WenQuanYi Micro Hei [Scale=0.9]WenQuanYi Micro Hei Mono song-WenQuanYi Micro Hei sfWenQuanYi Micro Hei "zh" = 0pt plus 1pt

ãĂŁLPython CookbookãĂŃçňňäÿL'çL'Ĺ

åŕŚåÿČ 3.0.0

çĘŁèČ¡

2018 åźť 03 æIJĹ 15 æŮě

Contents

1 Copyright **1**

2 åL'■èÍĂ **1**

2.1 éąźçŻőäÿžéąţ . . . 1
2.2 èřŚèĂĚçŽĎèřİ . . . 1
2.3 ä¡IJèĂĚçŽĎèřİ . . . 2
2.4 è£ŹæIJňäźęéĂĆåŘĹèřĄ . . . 2
2.5 è£ŹæIJňäźęäÿ■éĂĆåŘĹèřĄ . . . 3
2.6 åIJÍçž£çď'žä¿Ńäžčçă Ą . . . 3
2.7 ä¡£çŤÍçď'žä¿Ńäžčçă Ą . . . 3
2.8 èĄŤçşżæĹŚäžň . . . 3
2.9 èĞť'èřć . . . 4

3 çňňäÿĂçńäïijŽæŢřæ■őçżŞæđĎåŠŇçőŮæşŢ **4**

3.1 1.1 èğčåŐŃåžŔåĹŮètŃåĂijçżŹåď'ŽäÿłåŔŸéĞŔ . . . 5
3.2 1.2 èğčåŐŃåŔŕè£■äžčåŕźèšąètŃåĂijçżŹåď'ŽäÿłåŔŸéĞŔ . . . 6
3.3 1.3 ä£İçŢŹæIJĂåŘŐ N äÿłåĚČçt'ă . . . 9
3.4 1.4 æ§ěæL'¿æIJĂåď'ğæĹŰæIJĂåřŔçŽĎ N äÿłåĚČçt'ă . . . 11
3.5 1.5 åőđçŐřäÿĂäÿłäijŸåĚĹçžğéŸ§åĹŮ . . . 12
3.6 1.6 å■ŮåĚÿäÿ■çŽĎéŤőæŸăåřĎåď'ŽäÿłåĂij . . . 15
3.7 1.7 å■ŮåĚÿæŐŠåžŔ . . . 16
3.8 1.8 å■ŮåĚÿçŽĎè£ŘçőŮ . . . 17
3.9 1.9 æ§ěæL'¿äÿď'å■ŮåĚÿçŽĎçŻÿåŘŇçĆź . . . 19
3.10 1.10 åĹăéŹď'åžŔåĹŮçŻÿåŘŇåĚČçt'ăăźűä£İæŇĄéąžåžŔ . . . 20
3.11 1.11 åŚ¡åŘ■åĹĞçL'Ğ . . . 22
3.12 1.12 åžŔåĹŮäÿ■åĞžçŐřæňąæŢřæIJĂåď'ŽçŽĎåĚČçt'ă . . . 23
3.13 1.13 éĂŽè£Ğæ§ŘäÿłåĕşéŤőå■ŮæŐŠåžŔäÿĂäÿłå■ŮåĚÿåĹŮèąĺ . . . 25
3.14 1.14 æŐŠåžŔäÿ■æŤřæŇĄåŐ§çŤ§ærŤè¿ČçŽĎåŕźèšą . . . 27
3.15 1.15 éĂŽè£Ğæ§Řäÿłå■Ůæőţåřĕèőřå¡ŢåĹĘçžĎ . . . 28
3.16 1.16 è£Ğæžď'åžŔåĹŮåĚČçt'ă . . . 30
3.17 1.17 äžŐå■ŮåĚÿäÿ■æŔŘåŔŰå■ŘéŽĘ . . . 32
3.18 1.18 æŸăåřĎåŘ■çğřåĹřåžŔåĹŮåĚČçt'ă . . . 33
3.19 1.19 è¡ňæ■ćåźűåŘŇæŮűèőąçŐŮæŢřæ■ő . . . 35

4.14 2.14 åŘĹåźűæŃijæŐěå■Ůçňęäÿš 60
4.15 2.15 å■Ůçňęäÿšäÿ■æŔŠåĚěåŔŸéĞŔ 63
4.16 2.16 äžěæŇĞåőŽåĹŮåő¡æäijåijŔåŇŰå■Ůçňęäÿš 65
4.17 2.17 åIJĺå■Ůçňęäÿšäÿ■åď ĎçŘĘhtmlåŠŇxml 66
4.18 2.18 å■Ůçňęäÿšäžď çĽŇèğčæđŘ 68
4.19 2.19 åőđçŐřäÿĂäÿłçőĂå■ŢçŽĎéĂŠå¡ŠäÿŃéŹ■åĹĘæđŘåŹÍ 70
4.20 2.20 å■ŮèŁĆå■Ůçňęäÿšäÿ ŁçŽĎå■ŮçňęäÿšæŞ■ä¡IJ 78

5 çňňäÿL’çńäïijŽæŢřå■ŮæŮěæIJ§åŠŇæŮűéŮt’ 80
5.1 3.1 æŢřå■ŮçŽĎåŻŻèĹ■äžŤåĚě 81
5.2 3.2 æĽğèąŇçš¿çąőçŽĎæţőçĆźæŢřè£ŘçőŮ 82
5.3 3.3 æŢřå■ŮçŽĎæăijåijŔåŇŰè¿ŞåĞž 84
5.4 3.4 äžŇåĚńå■ĄåĚ■è£ŻåĹűæŢt’æŢř 86
5.5 3.5 å■ŮèŁĆåĹřåď ğæŢt’æŢřçŽĎæĽŞåŇĚäÿŐèğčåŇĚ 88
5.6 3.6 åď■æŢřçŽĎæŢřå■ęè£ŘçőŮ 89
5.7 3.7 æŮăçľůåď ğäÿŐNaN 91
5.8 3.8 åĹĘæŢřè£ŘçőŮ 93
5.9 3.9 åď ğåđŃæŢřçżĎè£ŘçőŮ 94
5.10 3.10 ç§l’éŸţäÿŐçž£æĂğäžčæŢřè£ŘçőŮ 97
5.11 3.11 éŽŔæIJžéĂĽ’æŃl’ 99
5.12 3.12 å§žæIJňçŽĎæŮěæIJ§äÿŐæŮűéŮt’è¡ňæ■ć 101
5.13 3.13 èőąçőŮæIJĂåŘŐäÿĂäÿłåŠĺäžŤçŽĎæŮěæIJ§ 103
5.14 3.14 èőąçőŮå¡ŞåĽ■æIJĹäż¡çŽĎæŮěæIJ§èŇČåŻt’ 105
5.15 3.15 å■Ůçňęäÿšè¡ňæ■ćäÿžæŮěæIJ§ 107
5.16 3.16 çżŞåŘĹæŮűåŇžçŽĎæŮěæIJ§æŞ■ä¡IJ 108

6 çňňåŻŻçńäïijŽè£■äžčåŹĺäÿŐçŤ§æĹŘåŹÍ 110
6.1 4.1 æĽŃåŁĺéĄ■åŐĘè£■äžčåŹÍ 110
6.2 4.2 äžčçŘĘè£■äžč 111
6.3 4.3 ä¡£çŤĺçŤ§æĹŘåŹĺåĹŻåżžæŰřçŽĎè£■äžčæĺąåijŔ 112
6.4 4.4 åőđçŐřè£■äžčåŹĺå■Ŕèőő 114
6.5 4.5 åŔ■åŘŚè£■äžč 116
6.6 4.6 åÿęæIJL’åď ŰéČĺçŁűæĂĄçŽĎçŤ§æĹŘåŹĺåĞ¡æŢř 117
6.7 4.7 è£■äžčåŹĺåĹĞçL’Ğ 119
6.8 4.8 èůşè£ĞåŔŕè£■äžčåŕźèśąçŽĎåijĂåğŃéČĺåĹĘ 120
6.9 4.9 æŐŠåĹŮçżĎåŘĹçŽĎè£■äžč 122
6.10 4.10 åžŔåĹŮäÿŁçt’ćåijŢåĂijè£■äžč 124
6.11 4.11 åŘŇæŮűè£■äžčåď Žäÿłåžŕåĺů 126
6.12 4.12 äÿ■åŘŃéŽĘåŘĹäÿŁåĚČçt’ăçŽĎè£■äžč 128
6.13 4.13 åĹŻåżžæŢřæ■őåď ĎçŘĘçőąéĄŞ 129
6.14 4.14 åśŢåijĂåţŇåěŮçŽĎåžŔåĹů 132
6.15 4.15 éąžåžŔè£■äžčåŘĹáźűåŘŐçŽĎæŐŠåžŔè£■äžčåŕźèśą 133
6.16 4.16 è£■äžčåŹĺäžčæŻ£whileæŮăéŹŘå¿łçŐŕ 134

7 çňňäžŤçńäïijŽæŰĞäžűäÿŐIO 136
7.1 5.1 èŕżåĘŹæŰĞæIJňæŢřæ■ő 136
7.2 5.2 æĽŞå■řè¿ŞåĞžèĞşæŰĞäžűäÿ■ 138
7.3 5.3 ä¡£çŤĺåĚűäžŰåĹĘéŽŤçňęæĹŰèąŇçżĹæ■ćçňęæĽŞå■ř 139
7.4 5.4 èŕżåĘŹå■ŮèŁĆæŢřæ■ő 140

7.5 5.5 æŰĞäžűäÿ■å■ŸåIJÍæĽ■èČ¡åĘŹåĚě 142
7.6 5.6 å■Ůçňęäÿšç ŽĎI/OæŞ■ä¡IJ 143
7.7 5.7 èŕžåĘŹåŐŃçijľæŰĞäžű 144
7.8 5.8 åŻžåőŽåď ğåřŔèőřå¡ŢçŽĎæŰĞäžűè£■äžč 145
7.9 5.9 èŕžåŔŰäžŇè£ŻåĹűæŢřæ■őåĹřåŔŕåŔŸçijŞåĘšåŇžäÿ■ 146
7.10 5.10 åĘĚå■ŸæŸăářĎçŽĎäžŇè£ŻåĹűæŰĞäžű 148
7.11 5.11 æŰĞäžűèůŕå¿ĎåŘ■çŽĎæŞ■ä¡IJ 150
7.12 5.12 æţŃèŕŢæŰĞäžűæŸŕåŘęå■ŸåIJÍ 151
7.13 5.13 èŐůåŔŰæŰĞäžűůåď źäÿ■çŽĎæŰĞäžűåĹŮèąÍ 152
7.14 5.14 å£¡çŢěæŰĞäžűåŘ■çijŰçăĄ 154
7.15 5.15 æĽŞå■ŕäÿ■åŘĹæşŢçŽĎæŰĞäžűåŘ■ 155
7.16 5.16 åćđåŁăæĹŰæŤźåŔŸåůšæĽŞåijĂæŰĞäžűçŽĎçijŰçăĄ 157
7.17 5.17 åřĘå■ŮèŁĆåĘŹåĚěæŰĞæIJňæŰĞäžű 160
7.18 5.18 åřĘæŰĞäžűæŔŔè£řçňęåŇĚèčĚæĹŘæŰĞäžűåŕźèšą 160
7.19 5.19 åĹŻåžžäÿťæŮűæŰĞäžűåŠŇæŰĞäžűåď ź 162
7.20 5.20 äÿŐäÿšèąŇçńŕåŔčçŽĎæŢřæ■őéĂŽä£ą 165
7.21 5.21 åžŔåĹŮåŇŰPythonåŕźèšą 165

8 çňňåĚ■çńăïijŽæŢřæ■őçijŰçăĄåŠŇåď ĎçŘĘ 169
8.1 6.1 èŕžåĘŹCSVæŢřæ■ő 169
8.2 6.2 èŕžåĘŹJSONæŢřæ■ő 172
8.3 6.3 èğčæđŘçőĂå■ŢçŽĎXMLæŢřæ■ő 177
8.4 6.4 åćđéĞŔåijŔèğčæđŘåď ğåđŃXMLæŰĞäžű 180
8.5 6.5 åřĘå■ŮåĚÿè¡ňæ■ćäÿžXML 183
8.6 6.6 èğčæđŘåŠŇä£őæŤźXML 185
8.7 6.7 åĹľçŤĺåŚ¡åŘ■çľžéŮťèğčæđŘXMLæŰĞæąč 187
8.8 6.8 äÿŐåĚşçşżåđŃæŢřæ■őåžŞçŽĎäžď äžŠ 189
8.9 6.9 çijŰçăĄåŠŇèğčçăĄå■ĄåĚ■è£ŻåĹűæŢř 191
8.10 6.10 çijŰçăĄèğčçăĄBase64æŢřæ■ő 192
8.11 6.11 èŕžåĘŹäžŇè£ŻåĹűæŢřçżĎæŢřæ■ő 193
8.12 6.12 èŕžåŔŰåţŇåěŮåŠŇåŕŕåŔŸéŢ£äžŇè£ŻåĹűæŢřæ■ő 197
8.13 6.13 æŢřæ■őçŽĎçťŕåŁăäÿŐçż§èőąæŞ■ä¡IJ 207

9 çňňäÿČçńăïijŽåĞ¡æŢř 209
9.1 7.1 åŔŕæŐěåŔŮäżżæĎŔæŢřéĞŔåŔĆæŢřçŽĎåĞ¡æŢř 209
9.2 7.2 åŔŁæŐěåŔŮåĚşéŤőå■ŮåŔĆæŢřçŽĎåĞ¡æŢř 210
9.3 7.3 çżŹåĞ¡æŢřåŔĆæŢřåćđåŁăåĚČä£ąæĄŕ 212
9.4 7.4 è£ŤåŻđåď ŽäÿłåĂijçŽĎåĞ¡æŢř 212
9.5 7.5 åőŽäźĽæIJĽéżŸèőď åŔĆæŢřçŽĎåĞ¡æŢř 213
9.6 7.6 åőŽäźĽåŇ£åŘ■æĹŰåĘĚèĄŤåĞ¡æŢř 216
9.7 7.7 åŇ£åŘ■åĞ¡æŢřæ■ŢèŐůåŔŸéĞŔåĂij 217
9.8 7.8 åĞŔářŠåŔŕěřČçŤĺåŕźèšąçŽĎåŔĆæŢřäÿłæŢř 219
9.9 7.9 åřĘå■ŢæŰźæşŢçŽĎçśżè¡ňæ■ćäÿžåĞ¡æŢř 222
9.10 7.10 åÿęéćÍåď ŰçŁűæĂĄä£ąæĄŕçŽĎåŻđěřČåĞ¡æŢř 223
9.11 7.11 åĘĚèĄŤåŻđěřČåĞ¡æŢř 226
9.12 7.12 èő£éŮőéŮ■åŇĚäÿ■åőŽäźĽçŽĎåŔŸéĞŔ 228

10 çňňåĚńçńăïijŽçśżäÿŐåŕźèšą 231
10.1 8.1 æŤźåŔŸåŕźèšąçŽĎå■ŮçňęäÿšæŸ¿çď ž 231

10.2 8.2 èĞłåőŽäźĽå■Ůçňęäÿšç ŽĎæäijåijŔåŇŰ 233
10.3 8.3 èőľåŕźèśąæŤŕæŇĄäÿŁäÿŃæŰĞçőąçŘĘå■Ŕèőő 234
10.4 8.4 åĹŽåžžåď ğéĞŔåŕźèśąæŮűèŁĆçIJĄåĘĚå■ŸæŰźæşŢ 236
10.5 8.5 åIJĺçśžäÿ■åřĄèčĚåśđæĂğåŘ■ 237
10.6 8.6 åĹŽåžžåŔŕçőąçŘĘçŽĎåśđæĂğ 239
10.7 8.7 èřČçŤĺçĹűçśżæŰźæşŢ 243
10.8 8.8 å■Řçśžäÿ■æĽľåśŢproperty 247
10.9 8.9 åĹŽåžžæŰřçŽĎçśżæĹŰåőđä¿ŃåśđæĂğ 251
10.108.10 ä¡£çŤĺåžűè£§èőąçőŮåśđæĂğ 254
10.118.11 çőĂåŇŰæŢřæ■őçżŞæđĎçŽĎåĹİåğŃåŇŰ 257
10.128.12 åőŽäźĽæŐěåŔčæĹŰèĂĚæŁ¡èśąå§žçśż 261
10.138.13 åőđçŐřæŢřæ■őæĺąåđŃçŽĎçśżåđŃçžęæİ§ 263
10.148.14 åőđçŐřèĞłåőŽäźĽåőźåŹÍ 268
10.158.15 åśđæĂğçŽĎäžčçŘĘèő£éŮő 271
10.168.16 åIJĺçśžäÿ■åőŽäźĽåď ŽäÿłæđĎéĂääŹÍ 276
10.178.17 åĹŽåžžäÿ■èřČçŤĺinitæŰźæşŢçŽĎåőđä¿Ń 277
10.188.18 åĹľçŤĺMixinsæĽľåśŢçśżåŁ§èČ¡ 278
10.198.19 åőđçŐřçŁűæĂĄåŕźèśąæĹŰèĂĚçŁűæĂĄæIJž 281
10.208.20 éĂŽè£Ğå■ŮçňęäÿšèřČçŤĺåŕźèśąæŰźæşŢ 284
10.218.21 åőđçŐřèő£éŮőèĂĚæĺąåijŔ 286
10.228.22 äÿ■çŤĺéĂŠå¡ŠåőđçŐřèő£éŮőèĂĚæĺąåijŔ 289
10.238.23 å¿łçŐřåijŢçŤĺæŢřæ■őçżŞæđĎçŽĎåĘĚå■ŸçőąçŘĘ 293
10.248.24 èőľçśżæŤŕæŇĄæŕŤè¿ČæŞ■ä¡IJ 296
10.258.25 åĹŽåžžçijŞå■Ÿåőđä¿Ń 298

11 çňňäźİçńäïijŽåĚČçijŰçĺŃ 302
11.1 9.1 åIJĺåĞ¡æŢřäÿŁæůżåŁäåŇĚèčĚåŹÍ 303
11.2 9.2 åĹŽåžžèčĚéěřåŹĺæŮűä£İçŢŹåĞ¡æŢřåĚČä£ąæĄŕ 304
11.3 9.3 èğčéŹď äÿĂäÿłèčĚéěřåŹÍ 306
11.4 9.4 åőŽäźĽäÿĂäÿłåÿęåŔĆæŢřçŽĎèčĚéěřåŹÍ 308
11.5 9.5 åŔŕèĞłåőŽäźĽåśđæĂğçŽĎèčĚéěřåŹÍ 309
11.6 9.6 åÿęåŔŕéĂĽåŔĆæŢřçŽĎèčĚéěřåŹÍ 312
11.7 9.7 åĹľçŤĺèčĚéěřåŹĺåijžåĹűåĞ¡æŢřäŸŁçŽĎçśżåđŃæčĂæ§ě 314
11.8 9.8 åřĘèčĚéěřåŹĺåőŽäźĽäÿžçśżçŽĎäÿĂéČĺåĹĘ 317
11.9 9.9 åřĘèčĚéěřåŹĺåőŽäźĽäÿžçśż 319
11.109.10 äÿžçśżåŠŇéİŹæĂĄæŰźæşŢæŔŘä¿ŻèčĚéěřåŹÍ 322
11.119.11 èčĚéěřåŹĺäÿžèćńåŇĚèčĚåĞ¡æŢřåćđåŁäåŔĆæŢř 324
11.129.12 ä¡£çŤĺèčĚéěřåŹĺæĽľåĚĚçśżçŽĎåŁ§èČ¡ 327
11.139.13 ä¡£çŤĺåĚČçśżæŐğåĹűåőđä¿ŃçŽĎåĹŻåžž 328
11.149.14 æ■ŢèŐůçśżçŽĎåśđæĂğåőŽäźĽéąžåžŔ 331
11.159.15 åőŽäźĽæIJĽåŔŕéĂĽåŔĆæŢřçŽĎåĚČçśż 334
11.169.16 *argsåŠŇ**kwargsçŽĎåijžåĹűåŔĆæŢřç■¿åŘ■ 336
11.179.17 åIJĺçśžäÿŁåijžåĹűä¡£çŤĺçijŰçĺŃèğĎçžę 339
11.189.18 äžěçijŰçĺŃæŰźåijŔåőŽäźĽçśż 342
11.199.19 åIJĺåőŽäźĽçŽĎæŮűåĂŹåĹİåğŃåŇŰçśżçŽĎæĹŘåŚŸ 345
11.209.20 åĹľçŤĺåĞ¡æŢřæşlèğčåőđçŐřæŰźæşŢéĞ■è¡¡ 347
11.219.21 éĄ£åĚ■éĞ■åď■çŽĎåśđæĂğæŰźæşŢ 353
11.229.22 åőŽäźĽäÿŁäÿŃæŰĞçőąçŘĘąŹĺçŽĎçőĂå■ŢæŰźæşŢ 355

11.239.23 åIJÍåśĂéČÍåŔŸéĞŔå§§äÿ■æĽğèąŇäžčçăĄ 357
11.249.24 èğčæđŘäÿŐåĹĘæđŘPythonæžŘçăĄ 359
11.259.25 æŃĘèğčPythonå■ŮèŁĆçăĄ 363

12 çňňå■AçńäïijŽæĺąåİŮäÿŐåŇĚ **366**
12.1 10.1 æđĎåžžäÿĂäÿłæĺąåİŮçŽĎåśĆçžğåŇĚ 366
12.2 10.2 æŐğåĹűæĺąåİŮèćńåĚĺéČÍåŕijåĚěçŽĎåĘĚåőź 367
12.3 10.3 ä¡£çŤÍçŻÿåŕźèůŕå¿ĎåŘ■åŕijåĚěåŇĚäÿ■å■ŘæĺąåİŮ 368
12.4 10.4 åřĘæĺąåİŮåĹĘåĽšæĹŘåď˙ŽäÿłæŰĞäžű 369
12.5 10.5 åĹľçŤÍåŚ¡åŘ■çľžéŮť˙åŕijåĚěçŻőå¡ŢåĹĘæŢčçŽĎäžčçăĄ 371
12.6 10.6 éĞ■æŰřåŁäè¡¡æĺąåİŮ 373
12.7 10.7 è£ŘèąŇçŽőå¡ŢæĹŰåŐŃçijľæŰĞäžű 374
12.8 10.8 èŕżåŔŰä¡■äžŐåŇĚäÿ■çŽĎæŢřæ■őæŰĞäžű 375
12.9 10.9 åřĘæŰĞäžűåď˙źåŁäåĚěåĹřsys.path 376
12.1010.10 éĂŽè£Ğå■Ůçňęäÿšå Ř■åŕijåĚěæĺąåİŮ 377
12.1110.11 éĂŽè£ĞéŠľ˙å■Řè£IJçÍŃåŁäè¡¡æĺąåİŮ 378
12.1210.12 åŕijåĚěæĺąåİŮçŽĎåŘŇæŮűä£őæŤźæĺąåİŮ 393
12.1310.13 åőĽèčĚçğĄæIJĽçŽĎåŇĚ 396
12.1410.14 åĹŻåžžæŰřçŽĎPythonçŐŕåćČ 396
12.1510.15 åĹĘåŔŚåŇĚ 398

13 çňňå■AąäÿĂçńäïijŽç¡ŚçżIJäÿŐWebçijŰçÍŃ **399**
13.1 11.1 ä¡IJäÿžåőćæĹůçńŕäÿŐHTTPæIJ■åŁąäžď˙äžŠ 399
13.2 11.2 åĹŻåžžTCPæIJ■åŁąåŹÍ 404
13.3 11.3 åĹŻåžžUDPæIJ■åŁąåŹÍ 407
13.4 11.4 éĂŽè£ĞCIDRåIJřåİĂçŤ§æĹŘåŕźåžŤçŽĎIPåIJřåİĂéŻĘ 409
13.5 11.5 åĹŻåžžäÿĂäÿłçőĂå■ŢçŽĎRESTæŐěåŔč 411
13.6 11.6 éĂŽè£ĞXML-RPCåőđçŐřçőĂå■ŢçŽĎè£IJçÍŃèřČçŤÍ 415
13.7 11.7 åIJÍäÿ■åŘŇçŽĎPythonèğčéĞŁåŹÍäźŃéŮť˙äžď˙äžŠ 418
13.8 11.8 åőđçŐřè£IJçÍŃæŰźæşŢèřČçŤÍ 419
13.9 11.9 çőĂå■ŢçŽĎåőćæĹůçńŕèőď˙èŕĄ 423
13.1011.10 åIJÍç¡ŚçżIJæIJ■åŁąäÿ■åŁäåĚěSSL 425
13.1111.11 è£ŻçÍŃéŮť˙äijäéĂŠSocketæŰĞäžűæŔŔè£řçňę 431
13.1211.12 çŘĘèğčäžŃäžűéľ˙śåŁÍçŽĎIO 436
13.1311.13 åŔŚéĂĄäÿŐæŐěæŤűåď˙ğåđŃæŢřçżĎ 441

14 çňňå■AąäžŇçńäïijŽåźűåŔŚçijŰçÍŃ **443**
14.1 12.1 åŘŕåŁÍäÿŐåĄIJæ■ćçž£çÍŃ 444
14.2 12.2 åĹď˙æŰ■çž£çÍŃæŸŕåŘęåůšçżŔåŘŕåŁÍ 446
14.3 12.3 çž£çÍŃéŮť˙éĂŽä£ą 449
14.4 12.4 çżŹåĚşéŤőéČÍåĹĘåŁäéŤĄ 454
14.5 12.5 éŸšæ■ćæ■żéŤĄçŽĎåŁäéŤĄæIJžåĹű 456
14.6 12.6 ä£Ìå■Ÿçž£çÍŃçŽĎçŁűæĂĄä£ąæĄŕ 460
14.7 12.7 åĹŻåžžäÿĂäÿłçž£çÍŃæśă 461
14.8 12.8 çőĂå■ŢçŽĎåźűèąŇçijŰçÍŃ 465
14.9 12.9 PythonçŽĎåĚÍåśĂéŤĄéŮőécŸ 469
14.1012.10 åőŽäźĽäÿĂäÿłActoräžžåŁą 471
14.1112.11 åőđçŐřæűĹæĄŕåŔŚåÿČ/èőćéŸĚæĺąåđŃ 475
14.1212.12 ä¡£çŤÍçŤ§æĹŘåŹÍäžčæŻ£çž£çÍŃ 478

14.1312.13 åď Žäÿłçž£çÍŃéŸ§åĹŮè¡őèŕć 486
14.1412.14 åIJÍUnixçşżçż§äÿŁéİćåŘŕåŁĺåőĹæŁd’è£ŻçÍŃ 489

15 çňňå■ĄäÿL’çńäïijŽèĎŽæIJňçijŰçÍŃäÿŐçşżçż§çőąçŘĘ 492
15.1 13.1 éĂŽè£ĞéĞ■åőŽåŘŚ/çőąéĄŞ/æŰĞäžűæŐěåŔŮè¿ŞåĚě 493
15.2 13.2 çżĹæ■ćçÍŃåžŔåźűçżŹåĞžéŤŹèŕŕä£ąæĄŕ 494
15.3 13.3 èğčæđŘåŚ¡äžď èąŇéĂĽéąź 494
15.4 13.4 è£Řèą Ňæ Ůűåijźå Ğžår Eçă Aè¿ŞåĚěæŔŘçď ž 497
15.5 13.5 èŐůåŔŰçżĹçńŕçŽĎåď ğářŔ 498
15.6 13.6 æĽğèąŇåď ŰéČĺåŚ¡äžď åźűèŐůåŔŰåőČçŽĎè¿ŞåĞž 499
15.7 13.7 åď■åĹűæĹŰèĂĚçğżåŁĺæŰĞäžűåŠŇçŻőå¡Ţ 501
15.8 13.8 åĹŻåżžåŠŇèğčåŐŃå¡Šæąčæ ŰĞäžű 503
15.9 13.9 éĂŽè£ĞæŰĞäžűåŘ■æ§ěæĽ¿æŰĞäžű 503
15.1013.10 èŕżåŔŰéĚ■ç¡őæŰĞäžű 505
15.1113.11 çżŹçőĂå■ŢèĎŽæIJňáćđåŁăæŮěå£ŮåŁ§èČ¡ 508
15.1213.12 çżŹåĞ¡æŢřåžŞáćđåŁăæŮěå£ŮåŁ§èČ¡ 511
15.1313.13 åőđçŐřäÿĂäÿłèőąæŮűåŹÍ 512
15.1413.14 éŹŘåĹűåĘĚå■ŸåŠŇCPUçŽĎä¡£çŤĺéĞŔ 514
15.1513.15 åŘŕåŁĺäÿĂäÿłWEBæţŔèğĹåŹÍ 515

16 çňňå■ĄåŻŻçńäïijŽæţŃèŕŢãĂĄèřČèŕŢåŠŇåijĆåÿÿ 516
16.1 14.1 æţŃèŕŢstdoutè¿ŞåĞž 516
16.2 14.2 åIJĺå■ŢåĚČæţŃèŕŢäÿ■çżŹåŕźèśąæĽŞèąěäÿĄ 518
16.3 14.3 åIJĺå■ŢåĚČæţŃèŕŢäÿ■æţŃèŕŢåijĆåÿÿæČĚåĘţ 521
16.4 14.4 åřĘæţŃèŕŢè¿ŞåĞžçŤĺæŮěå£Ůè őŕå¡ŢåĹřæŰĞäžűäÿ■ 523
16.5 14.5 å£¡çŢěæĹŰæIJ§æIJŽæţŃèŕŢåď śèť ě 524
16.6 14.6 åď ĎçŘĘåď ŽäÿłåijĆåÿÿ 525
16.7 14.7 æ■ŢèŐůæĽĂæIJĽåijĆåÿÿ 527
16.8 14.8 åĹŻåżžèĞłåőŽäźĽåijĆåÿÿ 529
16.9 14.9 æ■ŢèŐůåijĆåÿÿåŘŐæŁŻåĞžåŔęåď ŰçŽĎåijĆåÿÿ 531
16.1014.10 éĞ■æŰřæŁŻåĞžèćńæ■ŢèŐůçŽĎåijĆåÿÿ 533
16.1114.11 è¿ŞåĞžè■ęåŚŁä£ąæĄŕ 534
16.1214.12 èřČèŕŢå§žæIJňçŽĎçÍŃåžŔåť ľ æžČéŤŹèŕŕ 535
16.1314.13 çżŹä¡ăçŽĎçÍŃåžŔåĄŽæĂğèČ¡æţŃèŕŢ 538
16.1414.14 åŁăéĂ§çÍŃåžŔè£ŘèąŇ 541

17 çňňå■ĄäžŤçńäïijŽCèŕ■èÍĂæĽľåśŢ 545
17.1 15.1 ä¡£çŤĺctypesèő£éŮőCäžčçăĄ 547
17.2 15.2 çőĂå■ŢçŽĎCæĽľåśŢæĺąåİŮ 553
17.3 15.3 çijŰåĘŹæĽľåśŢåĞ¡æŢřæŞ■ä¡IJæŢřçžĎ 557
17.4 15.4 åIJÍCæĽľåśŢæĺąåİŮäÿ■æŞ■ä¡IJéŽŘå¡ćæŇĞéŠĹ 559
17.5 15.5 äžŐæĽľåśŢæĺąåİŮäÿ■åőŽäźĽåŠŇåŕijåĞžCçŽĎAPI 562
17.6 15.6 äžŐCèŕ■èÍĂäÿ■èřČçŤĺPythonäžčçăĄ 566
17.7 15.7 äžŐCæĽľåśŢäÿ■éĞŁæŤ¿åĚĺåśĂéŤĄ 571
17.8 15.8 CåŠŇPythonäÿ■çŽĎçž£çÍŃæůůçŤĺ 572
17.9 15.9 çŤĺWSIGåŇĚèčĚCäžčçăĄ 573
17.1015.10 çŤĺCythonåŇĚèčĚCäžčçăĄ 578
17.1115.11 çŤĺCythonåĘŹéńŸæĂğèČ¡çŽĎæŢřçžĎæŞ■ä¡IJ 585
17.1215.12 åřĘåĞ¡æŢřæŇĞéŠĹè¡ňæ■ćäÿžåŔŕèřČçŤĺåŕźèśą 589

17.1315.13 äijăéĂŠNULLçżŞář¿çŽĎå■ŮçňęäÿšçżŹCåĞ¡æŢřážŞ 590
17.1415.14 äijăéĂŠUnicodeå■ŮçňęäÿšçżŹCåĞ¡æŢřážŞ 594
17.1515.15 Cå■Ůçňęäÿšè¡ňæ■ćäÿžPythonå■Ůçňęäÿš 599
17.1615.16 äÿ■çąőåőŽçijŰçăĄæäijåijŔçŽĎCå■Ůçňęäÿš 600
17.1715.17 äijăéĂŠæŰĞäżűåŘ■çżŹCæĽľåśŢ 603
17.1815.18 äijăéĂŠåůšæĽŞåijĂçŽĎæŰĞäżűçżŹCæĽľåśŢ 604
17.1915.19 äżŐCèŕ■èÍĂäÿ■èřżåŔŰçśżæŰĞäżűåřźèśą 605
17.2015.20 åď ĎçŘĘCèŕ■èÍĂäÿ■çŽĎåŔřè£■äžčåřźèśą 608
17.2115.21 èřŁæŰ■åĹĘæőţéŤŹèřŕ 609

18 éŹĎå¡ŢA **610**
18.1 åIJÍçž£èţĎæžŘ 610
18.2 Pythonå■ęäźăäźęçś■ 610
18.3 énŸçžğäźęçś■ 611

19 åĚşäžŐèŕŚèĂĚ **611**

20 Roadmap **611**

Contents:

1 Copyright

äźęåŘ■ïijŽ ãĂŁPython CookbookãĂŃ3rd Edition
ä¡IJèĂĚïijŽ David Beazley, Brian K. Jones
èŕŚèĂĚïijŽ çĘŁèČ¡
çĽĹæIJňïijŽ çňň3çĽĹ
åĞžçĽĹçď¿ïijŽ OâĂŹReilly Media, Inc.
åĞžçĽĹæŮěæIJ§ïijŽ 2013åźť5æIJĹ08æŮě
Copyright Âl' 2013 David Beazley and Brian Jones. All rights reserved.

æŻť åď ŽåŔŚåÿČä£ąæĄŕèřůåŔĆèĂČ
http://oreilly.com/catalog/errata.csp?isbn=9781449340377

2 åĽ■èÍĂ

2.1 éąźçŻőäÿżéąţ

https://github.com/yidao620c/python3-cookbook

2.2 èŕŚèĂĚçŽĎèŕİ

äžžçŤ§èŃęç§■ïijŇæĹŚçŤÍ Pythonïij Ą

èŕŚèĂĚäÿĂçŻťåİŽæŇĄä¡£çŤÍ Python 3ïijŇåŻääÿžåőČäżčèąĺäžĘ Python çŽĎæIJłæİěãĂĆèŹ¡çĎůåŘŚåŘŐåĚijåőźæŸŕåőČçŽĎçąňäijd'ïijŇä¡ĘæŸŕè£ŹäÿłåśĂéİćè£§æŮl'äijŽæŤźåŔŸçŽ èĂŇäÿŤ Python 3 çŽĎæIJłæİěéIJĂèęĄæŕŔäÿłäžžçŽĎåÿőåĽl'åŠŇæŤŕæŇĄãĂĆ çŻőåĽ■åÿĆéİćäÿŁçŽĎæŢŹçÍŃäźęçś■ïijŇç¡Śäÿ ŁçŽĎæĽ'ŃåĘŇåd'ğéČĺåĹĘå§žæIJňéČ¡æŸŕ 2.x çşżåĹŮçŽĎïijŇäÿŞéŮĺå§žäžŐ 3.x çşżåĹŮçŽĎäźęçś■åřŚçŽĎåŔŕæĂIJãĂĆ

æIJĂè£ŚçIJŃåĹřäÿĂæIJňãĂŁPython CookbookãĂŃ3rd Edi-tionïijŇåőŇåĚĺå§žäžŐ Python 3ïijŇåĘŹçŽĎäź§å¿Ĺäÿ■éŤŹãĂĆ äÿžäžĘ Python 3 çŽĎæŹőåŔŁïijŇæĹŚäź§äÿ■èĞłéĞŔåŁŻïijŇæČşåĄŽçĆźäżĂäźĹäžŃæČĚãĂĆäžŐæŸŕäźŐïijŇåřśæIJL'äžĘç£ è£Źäÿ■æŸŕäÿĂéąźè¡żæİ¿çŽĎåůěä¡IJïijŇå■ť'æŸŕäÿĂäżűåĂijå¿ŮåĄŽçŽĎåůěä¡IJïijŽäÿ■äżĚæŰźä¿£äžĘåĹńä

èŕŚèĂĚäijŽåİŽæŇĄåŕźèĞłåůśæŕŔäÿĂåŔěçŽĎç£żèŕŚèť§èť'čïijŇåŁŻæśĆéńŸèť'ĺéĞŔãĂĆä¡ĘåŔŮèČ¡åŁŽ åęĆæđIJèŕŚæŰĞäÿ■æIJL'äžĂäźĹéŤŹæijŔçŽĎåIJřæŰźèŕůåd'ğåőűèğĄèŕËïijŇäź§æňćè£Őåd'ğåőűéŽŔæŮűæŇ yidao620@gmail.com

2.3 ä¡IJèĂĚçŽĎèŕİ

èĞłäžŐ 2008 åźť'äžěæİěïijŇPython 3 æĺłçl'žåĞžäÿŰåźűæĚćæĚćè£ŻåŇŰãĂĆPython 3 çŽĎætĄèąŇäÿĂçŻť'ècńèőd'äÿžéIJĂèęĄå¿ĹéŢ£äÿĂæőţæŮűéŮť'ãĂĆ äžŇåődäÿŁïijŇåĹřæĹŚåĘŹè£ŹæIJňäźęçŽĎ 2013 åźť'ïijŇçżİåd'ğéČĺåĹĘçŽĎ Python çÍŃåžŔåŚŸäż■çĎůåIJÍçŤ§äžğçŐŕåćČäÿ■ä¡£çŤÍçŽĎæŸŕçL'ĹæIJň 2 çşżåĹŮïijŇ æIJĂäÿżèęĄæŸŕåŻääÿž Python 3 äÿ■åŘŚåŘŐåĚijåőźãĂĆæŕńæŮăçŰŚéŮőïijŇåŕźäžŐåůěä¡IJåIJÍéĄŮçŢŹäżčç ä¡ĘæŸŕæŤ¿çIJijæIJłæİěïijŇä¡ăåŕśäijŽåŔŚçŐř Python 3 çżŹä¡ăåÿęæİěäÿ■äÿĂæăůçŽĎæČŁåŰIJãĂĆ

æ■čåęĆ Python 3 äžčèąĺæIJłæİěäÿĂæăůïijŇæŰřçŽĎãĂŁPython Cook-bookãĂŃçL'ĹæIJňçŻÿæŕŤè¿ČäźŃåL'■çŽĎçL'ĹæIJňæIJL'äžĘäÿĂäÿłåĚĺæŰřçŽĎæŤźåŔŸãĂĆ éęŰåĚĹïijŇäź§æŸŕæIJĂéĞ■èęĄçŽĎïijŇè£ŹæĎŘåŠşçİĂæIJňäźęæŸŕäÿĂæIJňéİđåÿÿåL'■æš£çŽĎåŔĆèĂČäź Python 3.3 çL'ĹæIJňäÿŃéİćçijŰåĘŹåŠŇætŃèŕŢçŽĎïijŇ åźűæšąæIJL'èĂČèŹŠäźŃåL'■èĂĄçL'ĹæIJňçŽĎåĚijå ä¡ĘæŸŕæĹŠäžňæIJĂçżĹçŽĎçŻőçŽĎæŸŕåĘŹäÿĂæIJňáőŇåĚĺå§žäžŐçŐřäžčåůěåĚůåŠŇèŕ■èĺĂçŽĎäźęçś■ãĂ æĹŠäžňåÿŇæIJŻæIJňäźęèČ¡åd'§æŇĞåŕijäžžäžňä¡£çŤÍ Python 3 çijŰåĘŹæŰřçŽĎäžčçăĄæĹŰèĂĚå■ĞçžğäźŃåL'■çŽĎéĄŮçŢŹäžčçăĄãĂĆ

æŕńæŮăçŰŚéŮőïijŇçijŰåĘŹäÿĂæIJňè£ŹæăůçŽĎäźęçżŹçijŰè¿Śåůěä¡IJåÿęæİěäÿĂáőŽçŽĎæŇŚæĹŸãĂ Python çğŸçś■çŽĎèŕİïijŇäijŽåIJĺèŕÿåęĆ ActiveStateâĂŹs Python recipes æĹŰèĂĚ Stack Overflow çŽĎç¡ŚçńŹäÿŁæŘIJåĹřæŢřäžěå■Čè őąçŽĎæIJL'çŤĺçŽĎçğŸçś■ïijŇä¡ĘæŸŕåĚűäÿ■çżİåd'ğéČĺåĹĘéC è£ŹäžŽçğŸçś■éŹd'äžĘæŸŕå§žäžŐ Python 2 çijŰåĘŹäźŃåd'ŰïijŇåŕèČ¡è£ŸæIJL'å¿Låd'Žèğčå Eşæ Ű źæąĹåIJ ïijĹæŕŤåęĆ 2.3 åŠŇ 2.4 çL'ĹæIJňïijL'ãĂĆ åŘęåd'ŰïijŇåőČäžňè£ŸäijŽçżŔåÿÿä¡£çŤĺäÿĂäžŻè£ĞæŮűçŽĎæĹA

Python 3.3 éĞŇéİćåŐżäžĘãĂĆåŕżæĽ¿åőŇåĚĺå§žäžŐ Python 3 çŽĎçğŸçś■çIJ§çŽĎéŽ¿äÿŁåŁăéŽ¿åŢŁãĂĆ

è£ŹæIJñäźęçŽĎæĽĂæIJL'äÿžécŸéČ¡æŸŕå§žäžŐåůšçżŔå■ŸåIJÍçŽĎäżčçăĄåŠŇæŁĂæIJfŕijŇèĂŇäÿ■æ
Python 3 çĽźæIJL'çŽĎçğŸçś■ãĂĆ åIJĺåŐ§æIJL'äžčçăĄå§žçąĂäÿŁïijŇæĹŚäżňåőŇåĚĺä¡£çŢĺæIJĂæŰřçŽĎ
Python æŁĂæIJŕåŐżæŢźéĂãăĂĆ æĽĂäžěïijŇäżżä¡ŢæČşä¡£çŢĺæIJĂæŰřæŁĂæIJŕçijŰåĘŹäżčçăĄçŽĎçÍŇåž

åIJÍéĂĽ'æŃl'èęĄåŇĚåŘńåŞłäžŽçğŸçś■æŰźéİćïijŇå¿ĹæŸŐæŸ¿äÿ■åŔŕèČ¡çijŰåĘŹäÿĂæIJñäźęåŻŁæŃ
Python écĘå§§æĽĂæIJL'çŽĎäÿIJèě£ãĂĆ åŻăæ■d'ïijŇæĹŚäżňäijŸåĚĹéĂĽ'æŃl'äžĘ Python
èŕ■èÍĂæăÿå£ČéČÍåĹĘïijŇäżèåŔŁéĆčäžŻæIJL'çİĂåź£æşŻåžŢçŢĺéćĘå§§çŽĎéŰőécŸãĂĆ
åŔęåd'ŰïijŇåĚűäÿ■æIJL'å¿Ĺåd'ŽçğŸçś■çŢĺæİěåśŢçd'ž Python 3 çŽĎæŰřçL'źæĂğïijŇ
è£Źåŕźäžőå¿Ĺåd'Žäžžæİěèŕť'æŸŕæŕŢè¿ČéŹŇçŢ§çŽĎïijŇåŞłæĂŢæŸŕä¡£çŢĺ
Python èĂĄçL'ĹæIJñçŽĎçżŔélŇäÿřárŇçŽĎçÍŇåžŔåŚŸãĂĆ
è£ŹäžŻçd'žä¿ŃçÍŇåžŔäź§äijŽåĄŔåŘŚäžŐåśŢçd'žäÿĂäžŻæIJL'çİĂåź£æşŻåžŢçŢĺçŽĎçijŰçÍŇæŁĂæIJŕ
ïijĹå■şçijŰçÍŇæĺąåijŔïijL'ïijŇ èĂŇäÿ■æŸŕäžĚäžĚåőŽä¡■åIJÍäÿĂäžŻåĚůä¡ŞçŽĎéŰőécŸäÿŁãĂĆår¡çőąäź§æ
Python èŕ■èÍĂæăÿå£ČåŠŇæăĞåĞĘåžŞãĂĆ

2.4 è£ŹælJñäźęéĂĆåŘĹèřĄ

è£ŹæIJñäźęçŽĎçŻőæăĞèŕżèĂĚæŸŕéĆčäžŻæČşæůśåĚěçŘĘèğč Python
èŕ■èÍĂæIJžåĹűåŠŇçŐřäżčçijŰçÍŇéčŐæăijçŽĎæIJL'çżŔélŇçŽĎ Python çÍŇåžŔåŚŸãĂĆ
æIJñäźęåd'ğéČÍåĹĘåĘĚåőźéŻĘäÿ■äžŐåIJÍæăĞåĞĘåžŞïijŇæąĘæđűåŠŇåžŢçŢĺçÍŇåžŔäÿ■åź£æşŻä¡£çŢĺçŽĎe
æIJñäźęæĽĂæIJL'çd'žä¿ŃåİĞåĄĞèő¿èŕżèĂĚåĚůæIJL'äÿĂåőŽçŽĎçijŰçÍŇèČŇæŹŕåźűäÿŤåŔŕäżěèŕżæĞĆçŻ
ïijĹæŕŤåęĆå§žæIJñçŽĎèőąçőŮæIJžçğŚå■ęç§ěèŕĘïijŇæŢŕæ■őçżŞædĎç§ěèŕĘïijŇçőŮæşŢåd'■æİĆåžęïijŇçşżç
èŕ■èÍĂçijŰçÍŇç■L'ïijL'ãĂĆ åŔęåd'ŰïijŇæŕŔäÿłçd'žä¿ŃéČ¡åŔłæŸŕäÿĂäÿłåĚěéŮÍæŇĞåŕijïijŇåęĆædIJèŕżèĂĚ
æĹŚäżňåĄĞåőŽèŕżèĂĚåŔŕäżěå¿ĹçĘ§çżČçŽĎä¡£çŤĺæŘIJçt'ćåijŢæŞŐäżěåŘŁç§ěéĄŞæĂŐæăůæ§ěèŕćåIJÍçž£
Python æŰĞæąčãĂĆ

æIJL'äÿĂäžŻæŻt'åŁäéńŸçžğçŽĎçğŸçś■ïijŇåęĆædIJèĂŘå£ČéŸĚèŕżïijŇårĘæIJL'åŁl'äžŐçŘĘèğč
Python åžŢåśĆçŽĎåůěä¡IJåŐ§çŘĘãĂĆ äžŐäÿ■ä¡äårĘå■ęåĹřäÿĂäžŻæŰřçŽĎæŁĂåůğåŠŇæŁĂæIJfŕijŇåźűåž

2.5 è£ŹælJñäźęäÿ■éĂĆåŘĹèřĄ

è£ŹæIJñäźęäÿ■éĂĆåŘĹ Python çŽĎåĹİå■ęèĂĚãĂĆäžŃåőđäÿŁïijŇæIJñäźęåĄĞåőŽèŕżèĂĚåĚůæIJL'
Python æŢŹçÍŇæĹŮåĚééŮÍäźęçś■äÿ■æL'ĂæŢŹæŐĹçŽĎå§žçąĂç§ěèŕĘãĂĆ
æIJñäźęäź§äÿ■æŸŕéČčçğ■å£ńéĂ§åŔĆèĂČæL'ŃåĘŇ ïijĹä¿ŃåęĆå£ńéĂ§æ§ěèŕćæ§ŘäÿłæĺąåİŮäÿŇçŽĎæ§Řä
æIJñäźęæŮĺåIJÍèĄŽçĎęåĞääÿłæIJĂéĞ■èęĄçŽĎäÿżécŸïijŇæijŤçd'žåĞăçğ■åŔŕèČ¡çŽĎèğčåĘşæŰźæąĹïijŇ
æŘŘä¿ŻäÿĂäÿłèůşæİ£åijŢåŕijèŕżèĂĚè£ŻåĚěäÿĂäžŻæŻt'éńŸçžğçŽĎåĘĚåőźïijĹè£ŹäžŻåŔŕäżěåIJÍç¡Śäÿ ŁæĹ

2.6 åIJÍçž£çd'žä¿Ńäżčçă Ą

æIJñäźęåĞääźŐæL'ĂæIJL'æžŘäżčçăĄåİĞåŔŕäżěåIJÍ http://github.com/dabeaz/python-cookbook äÿŁéİćæL'¿åĹřãĂĆ ä¡IJèĂĚæńćè£ŐåŘĎä¡■èŕżèĂĚä£őæ■č bugïijŇæŤźè£Żäżčçă ĄåŠŇèŕĎèőžãĂĆ

2.7 ä¡£çŤÍçď'žä¿ŃäžčçăĄ

æIJňäźęåřśæŸŕåÿőåŁľ'ä¡ăåőŇæĹŘä¡ăçŽĎåůěä¡IJçŽĎãĂĆ
äÿĂèĹňæİěèőšïijŇåŔłèęĄæŸŕæIJňäźęäÿŁéİćçŽĎçď'žä¿ŃäžčçăĄïijŇä¡ăéČ¡åŔŕäžěéŽŔæŮűæŃ£è£ĞåŐžåIJĺä
éŹď'éİđä¡ăä¡£çŤÍäžĘåď'ğéĞŔçŽĎäžčçăĄïijŇåŘęåĹŹäÿ■éIJĂèęĄåŘŚæĹŚäžňçŤşèřůèőÿåŔŕãĂĆ
ä¿ŃåęĆïijŇä¡£çŤÍåĞăäÿłäžčçăĄçL'Ğæőţå0žåőŇæĹŘäÿĂäÿłçÍŃåžŔäÿ■éIJĂèęĄèőÿåŔŕïijŇèt'l'å■ŰæĹŰèĂĚ
åijŢçŤÍæIJňäźęåŠŇçď'žä¿ŃäžčçăĄåŐžç¡ŚäÿŁåŻđç■ŤäÿĂäÿłéŮőéćŸäÿ■éIJĂèęĄèőÿåŔŕïijŇä¡ĘæŸŕåŘĹåźűå

æĹŚäžňäÿ■äijŽèęĄæśĆä¡ăæůżåŁăäžčçăĄçŽĎåĞžåď'ĎïijŇä¡ĘæŸŕåęĆæđIJä¡ăè£ŹäźĹåĄŽäžĘïijŇæĹŚäž
åijŢçŤÍéĂŽåÿÿåŇĚåŘńæăĞéćŸïijŇä¡IJèĂĚïijŇåĞžçL'Ĺçď¿ïijŇISBNãĂĆ ä¿ŃåęĆïijŽ*Python Cookbook*, 3rd edition, by David Beazley and Brian K. Jones (OâĂŹReilly). Copyright 2013 David Beazley and Brian Jones, 978-1-449-34037-7.

åęĆæđIJä¡ăèğL'å¿Ůä¡ăářźçď'žä¿ŃäžčçăĄçŽĎä¡£çŤÍèűĚåĞžäžĘåŘĹçŘEä¡£çŤÍæĹŰèĂĚäÿŁè£řåĹŮåĞžç
èřůéŽŔæŮűèĄŢçşżæĹŚäžňïijŇæĹŚäžňçŽĎéĆőçőśæŸŕ permissions@oreilly.comãĂĆ

2.8 èĄŢçşżæĹŚäžň

èřůářĘåĚşäžŐæIJňäźęçŽĎèŕĎèőžåŠŇéŮőéćŸåŔŚéĂĄçżŹåĞžçL'Ĺçď¿ïijŽ

OâĂŹReilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)

æĹŚäžňäÿžæIJňäźęåžžçńŃäžĘäÿĂäÿłç¡ŚéąţïijŇ åĚűäÿ■åŇĚåŘńåŃŸèŕŕèąĺïijŇçď'žä¿ŃåŠŇäÿĂäžŻåĚűäž
åŘŕäžěéĂŽè£ĞéŞ¿æŐě http://oreil.ly/python_cookbook_3e è6£éŮőãĂĆ

åĚşäžŐæIJňäźęçŽĎåžžèőőåŠŇæŁĂæIJŕæĂğéŮőéćŸïijŇèřůåŘŚéĂĄéĆőäžűèĞşïijŽ bookquestions@oreilly.com

åĚşäžŐæĹŚäžňçŽĎäźęçś■ïijŇèőÍèőžäijŽïijŇæŰřéŮżçŽĎæŻť'åď'Žä£ąæĄŕïijŇ èřůèő£éŮőæĹŚäžňçŽĎç¡ŚçńŹïijŽ http://www.oreilly.com

åIJÍ Facebook äÿŁæL'¿åĹřæĹŚäžňïijŽhttp://facebook.com/oreilly

åIJÍ Twitter äÿŁåĚşæşÍæĹŚäžňïijŽhttp://twitter.com/oreillymedia

åIJÍ YouTube äÿŁèğĆçIJŃæĹŚäžňïijŽhttp://www.youtube.com/oreillymedia

2.9 èĞťèřć

æĹŚäžňèąůå£ČæĎ§èřćæIJňäźęçŽĎæŁĂæIJŕæăąåőąäžžåŚŸ Jake Vanderplasïij ŇRobert Kern åŠŇ Andrea Crotti éİđåÿÿæIJL’çŤĺçŽĎèřĎèőžåŠŇåžžèőőïijŇ è£ŸæIJL’ Python çd’¿åŇžçŽĎåÿőåŁl’åŠŇéijŞåŁśãĂĆæĹŚäžňåŘŇæăůæĎ§èřćäÿŁäÿĂäÿłçL’ĹæIJňçŽĎçijŰè¿Ś Alex MartelliïijŇAnna Ravenscroft åŠŇ David AscherãĂĆ åř¡çőąè£Źäÿłç L’ĹæIJňæŸŕæŰřåĹŻä¡IJçŽĎïijŇä¡ĘæŸŕåL’■äÿĂäÿłçL’ĹæIJňäÿžæIJňäźęæŔŘä¿ŻäžĘäÿĂäÿłæŇ æIJĂåŘŐäź§æŸŕæIJĂéĞ■èęĄçŽĎïijŇæĹŚäžňèęĄæĎ§èřćæL’ĂæIJL’æŮl’æIJ§écĎèğĹçL’ĹæIJňçŽĎèŕżèĂĚïij

3 çňňäÿĂçńăïijŽæŢřæ■őçżŞæđĎåŠŇçőŮæşŢ

Python æŔŘä¿ŻäžĘåd’ğéĞŔçŽĎåĘĚç¡őæŢřæ■őçżŞæđĎïijŇåŇĚæŃňåĹŮèąĺïijŇéŻĘåŘĹäžěåŘŁå■ŮåĚ ä¡ĘæŸŕïijŇæĹŚäžňäź§äijŽçżŔåÿÿçćřåĹřåĹřěŕÿåęĆæ§ěèŕćïijŇæŐŠåžŔåŠŇè£Ğæżd’ç■L’ç■L’è£ŹäžŻæŹőéĄ■ åŽăæ■d’ïijŇè£ŹäÿĂçńăçŽĎçŻőçŽĎářśæŸŕèőĺèőžè£ŹäžŻæŕŤè¿Čåÿÿè ğĄçŽĎéŮőécŸåŠŇçőŮæşŢãĂĆ åŔęåd’ŰïijŇæĹŚäžňäź§äijŽçżŹåĞžåIJĺéŻĘåŘĹæĺąåİŮ `collections` å¡Şäÿ■æŞ■ä¡IJè£ŹäžŻæŢřæ■őçżŞæđĎçŽĎæŰźæşŢãĂĆ

3.1 1.1 èğčåŐŃåžŔåĹŮèţŃåĂijçżŹåd’ŽäÿłåŔŸéĞŔ

éŮőécŸ

çŐřåIJĺæIJL’äÿĂäÿłåŇĚåŘń N äÿłåĚČçt’ăçŽĎåĚČçżĎæĹŰèĂĚæŸŕåžŔåĹŮïijŇæĂŐæăůårĘåőČéĞŇéİć N äÿłåŔŸéĞŔïij§

èğčåĘşæŰźæąĹ

äžża¡ŢçŽĎåžŔåĹŮïijĹæĹŰèĂĚæŸŕåŔŕè£■äžčårźèšąïijL’åŔŕäžěéĂŽè£ĞäÿĂäÿłçőĂå■ŢçŽĎèţŃåĂijèŕ■å åŤŕäÿĂçŽĎåL’■æŘŘårśæŸŕåŔŸéĞŔçŽĎæŢřéĞŔå£Ëéążèů§åžŔåĹŮåĚČçt’ăçŽĎæŢřéĞŔæŸŕäÿĂæăůçŽĎãĂ

äžčçăĄçd’ža¿ŃïijŽ

```
>>> p = (4, 5)
>>> x, y = p
>>> x
4
>>> y
5
>>>
>>> data = [ 'ACME', 50, 91.1, (2012, 12, 21) ]
>>> name, shares, price, date = data
>>> name
'ACME'
>>> date
(2012, 12, 21)
>>> name, shares, price, (year, mon, day) = data
```

```
>>> name
'ACME'
>>> year
2012
>>> mon
12
>>> day
21
>>>
```

åęĆæđIJåŔŸéĞŔäÿłæŢřåŠŇåžŔåĹŮåĚČçť ăçŽĎäÿłæŢřäÿ■åŇźéĚ■ïijŇäijŽäžğçŤ§äÿĂäÿłåijĆåÿÿãĂĆ

äżčçăĄçď žä¿ŃïijŽ

```
>>> p = (4, 5)
>>> x, y, z = p
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ValueError: need more than 2 values to unpack
>>>
```

èőÍèőž

åőđéŹĚäÿŁïijŇè£Źçğ■èğčåŐŃètŃåĂijåŔŕäžěçŤĺåIJÍäżżä¡ŢåŔŕè£■äžčåŕźèśąäÿŁéİćïijŇèĂŇäÿ■äżĚäżĚæåŇĚæŃňå■ŮçňęäÿšïijŇæŰĞäżűåŕźèśąïijŇè£■äžčåŹĺåŠŇçŤ§æĹŘåŹĺãĂĆ

äżčçăĄçď žä¿ŃïijŽ

```
>>> s = 'Hello'
>>> a, b, c, d, e = s
>>> a
'H'
>>> b
'e'
>>> e
'o'
>>>
```

æIJL'æŮűåĂŹïijŇä¡ăåŔŕèČ¡åŔłæČşèğčåŐŃäÿĂéČĺåĹĘïijŇäÿćåijČåĚűäżŰçŽĎåĂijãĂĆåŕźäžŐè£Źçğ■a
Python åźűæšąæIJL'æŘŘä¿ŻçL'źæőŁçŽĎèí■æşŢãĂĆ ä¡ĘæŸŕä¡ăåŔŕäžěä¡£çŤĺäżżæĎŔåŘŸéĞŔåŘ■åŐżå■ää¡

äżčçăĄçď žä¿ŃïijŽ

```
>>> data = [ 'ACME', 50, 91.1, (2012, 12, 21) ]
>>> _, shares, price, _ = data
>>> shares
50
>>> price
91.1
>>>
```

ä¡åå£Ëéążä£ÌèŕĄä¡ăéĂL’çŤĺçŽĎéĆčäžŻå■ăä¡■åŔŸéĞŔåŘ■åIJĺåËűäžŰåIJřæŰźæšąèćńä¡£çŤĺåĹřãĂĆ

3.2 1.2 èğčåŐŃåŘŕè£■äžčåŕźèśąęţŃåĂijçżŹåď̌ŽäÿłåŔŸéĞŔ

éŮőécŸ

åęĆæđIJäÿĂäÿłåŔŕè£■äžčåŕźèśąçŽĎåËČçţ̌ääÿłæŢřèűËè£ĞåŔŸéĞŔäÿłæŢřæŮűïijÑäijŽæŁŻåĞžäÿĂäÿł
`ValueError` ãĂĆ éĆčäźĹæĂŐæåůæL’■èČ¡äžŐè£ŹäÿłåŔŕè£■äžčåŕźèśąäÿ■èğčåŐŃåĞž N
äÿłåËČçţ̌ääĞžæÌěïij§

èğčåĘşæŰźæąĹ

Python çŽĎæŸ§åŘůèąĺè¿¿åijŔåŔŕäžěçŤĺæÌěèğčåĘşè£ŹäÿłéŮőécŸãĂĆærŤåęĆïijÑä¡ääIJĺå■ęäźääÿĂéŮĺ
ä¡ăæČşçż§èőąäÿŃåőűåž■ä¡IJäÿŽçŽĎåźşåİĞæĹŘçżl’ïijÑä¡ĘæŸŕæŐŠéŹď̌æŐL’çňňäÿĂäÿłåŠÑæIJĂåŘŐäÿĂä
ä¡ĘåęĆæđIJæIJL’ 24 äÿłåŚćïij§è£ŹæŮűåĂŹæŸ§åŘůèąĺè¿¿åijŔåřśæť¿äÿŁçŤĺåIJžäžĘïijŽ

```
def drop_first_last(grades):
    first, *middle, last = grades
    return avg(middle)
```

åŘęåď̌ŰäÿĂçğ■æČĚåĘţïijÑåĄĞèő¿ä¡ăçŐřåIJĺæIJL’äÿĂäžŻçŤĺæĹůçŽĎèőřå¡ŢåĹŮèąĺïijÑærŔæİąèőřå¡Ţ
ä¡ääŔŕäžěåČŔäÿŃéİćè£ŹæăůåĹĘèğčè£ŹäžŻèőřå¡Ţïij Ž

```
>>> record = ('Dave', 'dave@example.com', '773-555-1212', '847-555-
↪1212')
>>> name, email, *phone_numbers = record
>>> name
'Dave'
>>> email
'dave@example.com'
>>> phone_numbers
['773-555-1212', '847-555-1212']
>>>
```

åĂijå¿ŮæşĺæĎŔçŽĎæŸŕäÿŁéİćèğčåŐŃåĞžçŽĎ `phone_numbers`
åŔŸéĞŔæřÿè£IJéČ¡æŸŕåĹŮèąĺçśžåđŃïijÑäÿ■çőąèğčåŐŃçŽĎçŢţèŕİåŔůçăĄæŢřéĞŔæŸŕåď̌ŽåřŠïijĹåŃĚæŃ
0 äÿłïijL’ãĂĆ æL’Ăäžěïij Ñäżżä¡Ţä¡£çŤĺåĹř `phone_numbers`
åŔŸéĞŔçŽĎäžčçăĄåřśäÿ■éIJĂèęĄåĄŽåď̌Žä¡ŹçŽĎçśžåđŃæčĂæ§ěåŐżçąőèőď̌åőČæŸŕåŘęæŸŕåĹŮèąĺçśžåđ

æŸ§åŘůèąĺè¿¿åijŔäźş§èČ¡çŤĺåIJĺåĹŮèąĺçŽĎåijĂåğŃéČĺåĹĘãĂĆærŤåęĆïijÑä¡ăæIJL’äÿĂäÿłåĚñåŔŷåL’■
8 äÿłæIJĹéŤĂåŤőæŢřæ■őçŽĎåžŔåĹŮïijÑ ä¡ĘæŸŕä¡ăæČşçIJÑäÿŃæIJĂè£ŠäÿĂäÿłæIJĹæŢřæ■őåŠÑåL’■éİć
7 äÿłæIJĹçŽĎåźşåİĞåĂijçŽĎåŕźærŤãĂĆä¡ääŔŕäžěè£ŹæăůåĄŽïijŽ

```
*trailing_qtrs, current_qtr = sales_record
trailing_avg = sum(trailing_qtrs) / len(trailing_qtrs)
return avg_comparison(trailing_avg, current_qtr)
```

äÿŃéİćæŸŕåIJĺ Python èğčéĞŁåŹĺäÿ■æL’ğèąŇçŽĎçżŞædIJïijŽ

```
>>> *trailing, current = [10, 8, 7, 1, 9, 5, 10, 3]
>>> trailing
[10, 8, 7, 1, 9, 5, 10]
>>> current
3
```

èőÍèőž

æĽľåśŢçŽĎè£■äžčèğčåŐŃèŕ■æşŢæŸŕäÿŞéŮĺäÿžèğčåŐŃäÿ■çąőåőŽäÿłæŢřæĹŰäżżæĎŔäÿłæŢřåĚČçť
éĂŽåÿÿijÑè£ŹäžŽåŔŕè£■äžčåŕźèśąçŽĎåĚČçť ăçżŞæđĎæIJĽçąőåőŽçŽĎèğĎåĹŹïijĹæŕŤåęĆçňň
1 äÿłåĚČçť ăåŘŐéÍćéČ¡æŸŕçŢţèŕÍåŔůçăĄïijĽ ïijÑ æŸ§åŔůèąÍè¿¿åijŔèőľ åijĂåŔŚäžžåŚŸåŔŕäżěå¿ĹåőźæŸ§ç
èĂŇäÿ■æŸŕéĂŽè£ĞäÿĂäžŽæŕŤè¿Čåď■æÍĆçŽĎæĽŃæőţåŐżèŐůåŔŰè£ŹäžŽåĚşèĄŢçŽĎåĚČçť ăåĂijãĂĆ

åĂijå¿ŮæşÍæĎŘçŽĎæŸŕïijÑæŸ§åŔůèąÍè¿¿åijŔåIJÍè£■äžčåĚČçť ăäÿžåŔŕåŘŸéŢ£åĚČçżĎçŽĎåžŘåĹŮæ
æŕŤåęĆïijÑäÿŃéÍćæŸŕäÿĂäÿłåÿęæIJĽæăĞç■¿çŽĎåĚČçżĎåžŘåĹŮïijŽ

```
records = [
    ('foo', 1, 2),
    ('bar', 'hello'),
    ('foo', 3, 4),
]

def do_foo(x, y):
    print('foo', x, y)

def do_bar(s):
    print('bar', s)

for tag, *args in records:
    if tag == 'foo':
        do_foo(*args)
    elif tag == 'bar':
        do_bar(*args)
```

æŸ§åŔůèğčåŐŃèŕ■æşŢåIJÍå■ŮçňęäÿšæŞ■ä¡IJçŽĎæŮűåĂŹäź§äijŽå¿ĹæIJĽçŤĺïijÑæŕŤåęĆå■Ůçňęäÿšç

äžčçăĄçď žä¿ŃïijŽ

```
>>> line = 'nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/
↪false'
>>> uname, *fields, homedir, sh = line.split(':')
>>> uname
'nobody'
>>> homedir
'/var/empty'
>>> sh
'/usr/bin/false'
>>>
```

æIJL'æŮűåĂŹïijÑä¡ăæČşèğčåŐÑäÿĂäžŻåĚČçť'ååŘŐäÿćåijČåőČäžňïijÑä¡ăäÿ■èČ¡çőĂå■Ţåřśä¡£çŤÍ
* ïijÑ ä¡ĘæŸŕä¡ăåŔŕäžěä¡£çŤĺäÿĂäÿłæŹőéĂŽçŽĎåž§åijČåŘ■çğřïijÑæŕŤåęĆ _ æĹŰèĂĚ
ign ïijĹignoreïijL'ãĂĆ

äžčçăĄçď'žä¿ŃïijŽ

```
>>> record = ('ACME', 50, 123.45, (12, 18, 2012))
>>> name, *_, (*_, year) = record
>>> name
'ACME'
>>> year
2012
>>>
```

åIJĺå¿Ĺåď'ŽåĞ¡æŢřåijŔèŕ■èĺĂäÿ■ïijÑæŸ§åŔůèğčåŐÑèŕ■æşŢèů§åĹŮèąĺåď'ĎçŘĘæIJL'èőÿåď'ŽçŻÿäijijä
ä¡ăåŔŕäžěå¿ĹåőźæŸŞçŽĎåřĘåőČåĹĘåL'šæĹŘåL'■åŘŐäÿď'éČĺåĹĘïijŽ

```
>>> items = [1, 10, 7, 4, 5, 9]
>>> head, *tail = items
>>> head
1
>>> tail
[10, 7, 4, 5, 9]
>>>
```

åęĆæđIJä¡ăåď'§èĄłæŸŐçŽĎèŕİïijÑè£ŸèČ¡çŤĺè£Źçğ■åĹĘåL'šèŕ■æşŢåŐżåůğåęŹçŽĎåőđçŐŕéĂŠå¡ŠçőŮ

```
>>> def sum(items):
...     head, *tail = items
...     return head + sum(tail) if tail else head
...
>>> sum(items)
36
>>>
```

çĎűåŘŐïijÑçŤśäžŐèŕ■èĺĂåśĆéİćçŽĎéŹŘåĹűïijÑéĂŠå¡Šåźűäÿ■æŸŕ Python
æŞĚéŢ£çŽĎãĂĆ åŽäæ■d'ïijÑæIJĂåŘŐéĆčäÿłéĂŠå¡ŠæijŤçď'žäžĚäžĚæŸŕäÿłåě¡åěĞçŽĎæŐćçť'ćç¡ćäžĘïijÑå

3.3 1.3 ä£İçŢŹælJĂåŘŐ N äÿłåĚČçť'ă

éŮőécŸ

åIJĺè£■äžčæŞ■ä¡IJæĹŰèĂĚåĚűäžŰæŞ■ä¡IJçŽĎæŮűåĂŹïijÑæĂŐæăůåŔłä£İçŢŹæIJĂåŘŐæIJL'éŹŘåĞă

èğčåĘşæŰźæąĹ

ä£İçŢŹæIJL'éŹŘåŐĘåŔšèőřå¡Ţæ■čæŸŕ `collections.deque`
åď'ğæŸ¿èžńæL'ŃçŽĎæŮűåĂŹãĂĆæŕŤåęĆïijÑäÿŃéİćçŽĎäžčçăĄåIJĺåď'ŽèąŇäÿŁéİćåĄŽçőĂå■ŢçŽĎæŰĞæ
åźűè£ŤåŽđåŇźéĚ■æL'ĂåIJĺèąÑçŽĎæIJĂåŘŐNèąÑïijŽ

```
from collections import deque


def search(lines, pattern, history=5):
    previous_lines = deque(maxlen=history)
    for line in lines:
        if pattern in line:
            yield line, previous_lines
        previous_lines.append(line)

# Example use on a file
if __name__ == '__main__':
    with open(r'../../cookbook/somefile.txt') as f:
        for line, prevlines in search(f, 'python', 5):
            for pline in prevlines:
                print(pline, end='')
            print(line, end='')
            print('-' * 20)
```

èőÍèőž

æĹŚäžňåIJĺåĘŹæ§ěèřćåĚČçť ăçŽĎäžčçăĄæŮűïijŇéĂŽåÿÿäijŽä¡£çŤĺåŇĚåŘń `yield` èąĺè¿¿åijŔçŽĎçŤ§æĹŘåŹĺåĞ¡æŢřïijŇäź§åřśæŸŕæĹŚäžňäÿŁéÍćçď žä¿ŇäžčçăĄäÿ■çŽĎéĆčæăůãĂĆ è£ŹæăůåŔŕäžěåřĘæŘIJçť ćè£ĞçÍŇäžčçăĄåŠŇä¡£çŤĺæŘIJçť ćçżŞædIJäžčçăĄèğčèĂęãĂĆåęĆædIJä¡ăè£Ÿäÿ■a 4.3 èŁĆãĂĆ

ä¡£çŤÍ `deque(maxlen=N)` ædĎéĂăåĞ¡æŢřäijŽæŰřåžžäÿĂäÿłåŽžåőŽåď ğåřŔçŽĎéŸ§åĹŮãĂĆå¡ŞæŰ æIJĂèĂĄçŽĎåĚČçť ăäijŽèĞłåŁĺèćńçğžéŹď æŐĽ˜ãĂĆ

äžčçăĄçď žä¿ŃïijŽ

```
>>> q = deque(maxlen=3)
>>> q.append(1)
>>> q.append(2)
>>> q.append(3)
>>> q
deque([1, 2, 3], maxlen=3)
>>> q.append(4)
>>> q
deque([2, 3, 4], maxlen=3)
>>> q.append(5)
>>> q
deque([3, 4, 5], maxlen=3)
```

ař¡çőąä¡ăäź§åŔŕäžěæĽ˜ŃåŁĺåIJĺäÿĂäÿłåĹŮèąĺäÿŁåőđçŐřè£ŹäÿĂçŽĎæŞ■ä¡IJïijĹæŕŤåęĆåćđåŁăãĂĄåĹ

æŻť˜äÿĂèĹńçŽĎïijŇ `deque` çśžåŔŕäžěèćńçŤĺåIJĺäżžä¡Ţä¡ääŔłéIJĂèęĄäÿĂäÿłçőĂå■ŢéŸ§åĹŮæŢřæ■őç åęĆædIJä¡ääÿ■èő¿ç¡őæIJĂåď ğéŸ§åĹŮåď ğåřŔïijŇéĆčäźĹåřśäijŽå¿ŮåĹřäÿĂäÿłæŮăéŹŘåď ğåřŔéŸ§åĹŮïijŇ

äžčçăĄçď žä¿ŃïijŽ

```
>>> q = deque()
>>> q.append(1)
>>> q.append(2)
>>> q.append(3)
>>> q
deque([1, 2, 3])
>>> q.appendleft(4)
>>> q
deque([4, 1, 2, 3])
>>> q.pop()
3
>>> q
deque([4, 1, 2])
>>> q.popleft()
4
```

åIJÍéŸ§åĹŮäÿď'çńŕæŔŠåĚěæĹŰåĹăéŹď'åĚČçť'ăæŮűéŮť'åď'■æİĆåžęéČ¡æŸŕ O(1) ïijŇèĂŇåIJÍåĹŮèąÍçŽĎåijĂåď'ťæŔŠåĚěæĹŰåĹăéŹď'åĚČçť'ăçŽĎæŮűéŮť'åď'■æİĆåžęäÿž O(N) ãĂĆ

3.4 1.4 æ§ěæĽ¿æIJĂåď'ğæĹŰæIJĂåřŔçŽĎ N äÿłåĚČçť'ă

éŮőécŸ

æĂŐæăůäżŐäÿĂäÿłéŻĘåŘĹäÿ■èŐůå¿ŮæIJĂåď'ğæĹŰèĂĚæIJĂåřŔçŽĎ N äÿłåĚČçť'ăåĹŮèąÍïij§

èğčåĘşæŰźæąĹ

heapq æÍąåİŮæIJL'äÿď'äÿłåĞ¡æŢřïijŽnlargest() åŠŇ nsmallest() åŘŕäžěåőŇç¿Őèğčåĕşè£ŹäÿłéŮőécŸãĂĆ

```
import heapq
nums = [1, 8, 2, 23, 7, -4, 18, 23, 42, 37, 2]
print(heapq.nlargest(3, nums)) # Prints [42, 37, 23]
print(heapq.nsmallest(3, nums)) # Prints [-4, 1, 2]
```

äÿď'äÿłåĞ¡æŢřéČ¡èČ¡æŐěåŘŮäÿĂäÿłåĚşéŤőå■ŮåŘĆæŢřïijŇçŤÍäžŐæŻť'åď'■æİĆçŽĎæŢřæ■őçżŞædĎă

```
portfolio = [
    {'name': 'IBM', 'shares': 100, 'price': 91.1},
    {'name': 'AAPL', 'shares': 50, 'price': 543.22},
    {'name': 'FB', 'shares': 200, 'price': 21.09},
    {'name': 'HPQ', 'shares': 35, 'price': 31.75},
    {'name': 'YHOO', 'shares': 45, 'price': 16.35},
    {'name': 'ACME', 'shares': 75, 'price': 115.65}
]
```

```
cheap = heapq.nsmallest(3, portfolio, key=lambda s: s['price'])
expensive = heapq.nlargest(3, portfolio, key=lambda s: s['price'])
```

èřŚèĂĚæşÍïijŽäÿŁéÍćäžčçăĄåIJĺářźæřŔäÿłåĚČçť'äè£ŻèąŇářźæřŤçŽĎæŮűåĂŹïijŇäijŽäžě price çŽĎåĂijè£ŻèąŇærŤè¿ČãĂĆ

èőÍèőž

åęĆæđIJä¡ăæČşåIJĺäÿĂäÿłéŻĘåŘĹäÿ■æ§ěæL'¿æIJĂářŔæĹŰæIJĂåď'ğçŽĎ N äÿłåĚČçť'ăïijŇåźűäÿŤ N åřŔäžŐéŻĘåŘĹåĚČçť'äæŢřéĞŔïijŇéĆčäźĹè£ŹäžŻåĞ¡æŢřæŔŘä¿ŻäžĘå¿Ĺåě¡çŽĎæ åŻääÿžåIJĺážŢåśĆåőđçŐřéĞŇéÍćïijŇéęŰåĚĹäijŽåĚĹářĘéŻĘåŘĹæŢřæ■ős£ŻèąŇääĘæŐŠäžŔåŘŐæŤ¿åĚěäÿ

```
>>> nums = [1, 8, 2, 23, 7, -4, 18, 23, 42, 37, 2]
>>> import heapq
>>> heap = list(nums)
>>> heapq.heapify(heap)
>>> heap
[-4, 2, 1, 23, 7, 2, 18, 23, 42, 37, 8]
>>>
```

ääĘæŢřæ■őçżŞæđĎæIJĂéĞ■èęĄçŽĎçL'źå¿ĄæŸý heap[0] æřÿè£IJæŸýæIJĂářŔçŽĎåĚČçť'ăãĆĆåźűäÿŤåL'l'ä¡ŹçŽĎåĚČçť'ääŔýäžěå¿Ĺåőźæ ŸŞçŽĎéĂŽè£ĞèřČçŤÍ heapq.heappop() æŰźæşŢå¿ŮåĹřïijŇ èřěæŰźæşŢäijŽåĚĹářĘçňňäÿĂäÿłåĚČçť'ääijźåĞžæÍěïijŇçĎűåŘŐ O(log N)ïijŇN æŸýåäĘåđ'ğářŔïijL'ãĂĆ æřŤåęĆïijŇåęĆæđIJæČşèęĄæ§ěæL'¿æIJĂářŔçŽĎ 3 äÿłåĚČçť'ăïijŇä¡ääŔýäžěè£ŹæäůåĄŽïijŽ

```
>>> heapq.heappop(heap)
-4
>>> heapq.heappop(heap)
1
>>> heapq.heappop(heap)
2
```

å¡ŞèęĄæ§ěæL'¿çŽĎåĚČçť'ääÿłæŢřçŻÿåźźæřŤè¿ČářŔçŽĎæŮűåĂŹïijŇåĞ¡æŢř nlargest() åŠŇ nsmallest() æŸýå¿ĹåŘĹéĂĆçŽĎãĂĆ åęĆæđIJä¡ääžĚäžĚæČşæ§ěæL'¿åŤýäŸĂçŽĎæIJĂářŔæĹŰæIJĂåď'ğïijĹN=1ïijL'çŽĎåĚČçť'ăçŽĎèřÍïijŇéĆčäźĹ min() åŠŇ max() åĞ¡æŢřäijŽæŻť'å£ńäžŽãĂĆ çśžäijijçŽĎïijŇåęĆæđIJ N çŽĎåď'ğářŔåŠŇéŻĘåŘĹåď'ğářŔæŐèè£ŠçŽĎæŮűåĂŹïijŇéĂŽåÿÿåĚĹæŐŠäžŔè£ŹäÿłéŻĘåŘĹçĎűåŘŐåĘ■ä¡ ïijĹ sorted(items)[:N] æĹŰèĂĚæŸý sorted(items)[-N:] ïijL'ãĂĆ éIJĂèęĄåIJĺæ■čçąőåIJžåŘĹä¡£çŤĺåĞ¡æŢř nlargest() åŠŇ nsmallest() æL'■èČ¡åŔŠæŇěåőČäžňçŽĎäijŸå£ ïijĹåęĆæđIJ N å£ńæŐèè£ŠéŻĘåŘĹåď'ğářŔäžĘïijŇéĆčäźĹä¡£çŤĺæŐŠäžŔæŞ■ä¡IJäijŽæŻť'åě¡äžŻïijL'ãĂĆ

år¡çőąä¡ăæšąæIJL'å£ĚèęĄäÿĂåőŽä¡£çŤĺè£ŹéĞŇçŽĎæŰźæşŢïijŇä¡ĘæŸýåäĘæŢřæ■őçżŞæđĎçŽĎåőđçĆ å§žæIJñäÿŁåŔłèęĄæŸýæŢřæ■őçżŞæđĎåŠŇçőŮæşŢäźęçś■éĞŇéÍćéČ¡äijŽæIJL'æŔŘåŔŁåĹřãĂĆ heapq æĺąåİŮçŽĎåőŸæŰźæŰĞæąčéĞŇéÍćäź§èŕęçżĘçŽĎäżŃçż■äžĘääĘæŢřæ■őçżŞæđĎåžŢåśĆçŽĎåőđçĆ

3.5 1.5 åőđçŐřäÿĂäÿłäijŸåĚĹçžğéŸ§åĹů

éůőécŸ

æĂŐæăůåőđçŐřäÿĂäÿłæŇL'äijŸåĚĹçžğæŐŠåžŔçŽĎéŸ§åĹŮïij§åźűäÿŤåIJĺè£Źäÿłéŷ§åĹŮäÿŁéÍćæŕŔæňą pop æŞ■ä¡IJæĂżæŸŕè£ŤåŽđäijŸåĚĹçžğæIJĂéńŸçŽĎéĆčäÿłåĚČçt

èğčåĘşæŰźæąĹ

äÿŃéÍćçŽĎçśżåĹl'çŤÍ heapq æĺąåİŮåőđçŐřäžĘäÿĂäÿłçőĂå■ŢçŽĎäijŸåĚĹçžğéŸ§åĹŮïijŽ

```
import heapq

class PriorityQueue:
    def __init__(self):
        self._queue = []
        self._index = 0

    def push(self, item, priority):
        heapq.heappush(self._queue, (-priority, self._index, item))
        self._index += 1

    def pop(self):
        return heapq.heappop(self._queue)[-1]
```

äÿŃéÍćæŸŕåőČçŽĎä¡£çŤÍæŰźåijŔïijŽ

```
>>> class Item:
...     def __init__(self, name):
...         self.name = name
...     def __repr__(self):
...         return 'Item({!r})'.format(self.name)
...
>>> q = PriorityQueue()
>>> q.push(Item('foo'), 1)
>>> q.push(Item('bar'), 5)
>>> q.push(Item('spam'), 4)
>>> q.push(Item('grok'), 1)
>>> q.pop()
Item('bar')
>>> q.pop()
Item('spam')
>>> q.pop()
Item('foo')
>>> q.pop()
Item('grok')
>>>
```

äžŤçżĘèğĆåŕ§åŔŕäžěåŔŚçŐřïijŇçňňäÿĂäÿł pop() æŞ■ä¡IJè£ŤåŽđäijŸåĚĹçžğæIJĂéńŸçŽĎåĚČçt'ãã

åŔęåď ŰæşÍæĎŔåĹřåęĆæđIJäÿď äÿłæIJL’çİĂçŻÿåŘŇäijŸåĚĹçžğçŽĎåĚČçť ăïijĹ `foo` åŠŇ `grok` ïijL’ïijŇpop æŞ■ä¡IJæŇL’çĚğåőČäžňèćńæŔŠåĚèåĹřéŸ§åĹŮçŽĎéąžåžŔè£ŤåŻđçŽĎãĂĆ

èőÍèőž

è£ŹäÿĂåřŔèŁĆæĹŚäžňäÿżèęĄåĚşæşÍ `heapq` æĺąåİŮçŽĎä¡£çŤĺãĂĆ åĞ¡æŢř `heapq.heappush()` åŠŇ `heapq.heappop()` åĹĘåĹńåIJĺéŸ§åĹŮ `_queue` äÿŁæŔŠåĚèåŠŇåĹäéŹď’çňňäÿĂäÿłåĚČçť ăïijŇ åźűäÿŤéŸ§åĹŮ `_queue` ä£ÌèŕĄçňňäÿĂäÿłåĚČçť ăæŃěæIJL’æIJĂéńŸäijŸåĚĹçžğïijĹ 1.4 èŁĆåůšçżŔèőÍèőžè£ĞèŁŹäÿłéŮőécŸïijL’ãĂĆ `heappop()` åĞ¡æŢřæĂżæŸŕè£ŤåŻđâĂİæIJĂåřŔçŽĎâĂİçŽĎåĚČçť ăïijŇè£ŹåřśæŸŕä£ÌèŕĄéŸ§åĹŮpopæŞ■ä¡IJè£ŤåŻđæ åŔęåď ŰïijŇçŤśäžŐ push åŠŇ pop æŞ■ä¡IJæŮűéŮť åď■æİĆåžęäÿž O(log N)ïijŇåĚűäÿ■ N æŸŕåăĘçŽĎåď ğåřŔïijŇåŻăæ■d’åřśőŮæŸŕ N å¿Ĺåď ğçŽĎæŮűåĂŹåőČäžňè£ŘèąŇéĂ§åžęäź§ä¿İæŮğå¿Ĺå£ńãĂĆ

åIJĺäÿŁéİćäžčçăĄäÿ■ïijŇéŸ§åĹŮåŇĚåŘńäžĘäÿĂäÿł `(-priority, index, item)` çŽĎåĚČçżĎãĂĆ äijŸåĚĹçžğäÿžèť §æŢřçŽĎçŽőçŽĎæŸŕä¡£å¿ŮåĚČçť ăæŇL’çĚğäijŸåĚĹçžğäžŐéńŸè£ŹäÿłèůšæŹőéĂŽçŽĎæŇL’äijŸåĚĹçžğäžŐä¡ŐåĹřéńŸæŐŠåžŔçŽĎåăĘæŐŠåžŔæĄřåůğçŽÿåŔ■ãĂĆ

`index` åŔŸéĞŔçŽĎä¡IJçŤĺæŸŕä£ÌèŕĄåŘŇç■L’äijŸåĚĹçžğåĚČçť ăçŽĎæ■čçąőæŐŠåžŔãĂĆ éĂŽè£ĞäŁİå■ŸäÿĂäÿłäÿ■æŰ■áćđåŁăçŽĎ `index` äÿŃæăĞåŔŸéĞŔïijŇåŔŕäžěçąőä£İåĚČçť ăæŇL’çĚğåőČäź èĂŇäÿŤïijŇ `index` åŔŸéĞŔäź§åIJĺçŽÿåŘŇäijŸåĚĹçžğåĚČçť ăæŕŤè¿ČçŽĎæŮűåĂŹètůåĹřéĞ■èęĄä¡IJçŤĺã

äÿžäžĘéŸŘæŸŐè£ŹäžŻïijŇåĚĹåĄĞåőŽ `Item` åðđä¿ŃæŸŕäÿ■æŤŕæŇĄæŐŠåžŔçŽĎïijŽ

```
>>> a = Item('foo')
>>> b = Item('bar')
>>> a < b
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: unorderable types: Item() < Item()
>>>
```

åęĆæđIJä¡ăä¡£çŤĺåĚČçżĎ `(priority, item)` ïijŇåŔłèęĄäÿď äÿłåĚČçť ăçŽĎäijŸåĚĹçžğäÿ■åŘŇår ä¡ĘæŸŕåęĆæđIJäÿď äÿłåĚČçť ăäijŸåĚĹçžğäÿĂæăůçŽĎèŕİïijŇéĆčäźĹærŤè¿ČæŞ■ä¡IJårśäijŽèů§äźŇåL’■äÿĂ

```
>>> a = (1, Item('foo'))
>>> b = (5, Item('bar'))
>>> a < b
True
>>> c = (1, Item('grok'))
>>> a < c
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: unorderable types: Item() < Item()
>>>
```

éĂŽè£ĞåijŢåĚèåŔęåď ŰçŽĎ `index` åŔŸéĞŔçżĎæĹŘäÿL’åĚČçżĎ `(priority, index, item)` ïijŇårśèČ¡å¿Ĺåě¡çŽĎéĄ£åĚ■äÿŁéİćçŽĎéŤŹèŕŕïijŇ åŽääÿžäÿ■åŔŕèČ¡æIJL’äÿď äÿłåĚČçť ăæIJL’çŽÿåŘŇçŽĎ `index` åĂijãĂĆPython

åIJÍåĄŽåĚČçżĎæŕŤè¿ČæŮűåĂŹïijŇåęĆæđIJåĽ■éÍćçŽĎæŕŤè¿ČåůšçżŕåŕŕäžěçąőåőŽçżŞæđIJäžĘïijŇ
åŘŐéÍćçŽĎæŕŤè¿ČæŞ■ä¡IJåřśäÿ■äijŽåŔŠçŤ§äžĘïijŽ

```
>>> a = (1, 0, Item('foo'))
>>> b = (5, 1, Item('bar'))
>>> c = (1, 2, Item('grok'))
>>> a < b
True
>>> a < c
True
>>>
```

åęĆæđIJäj¡äæČşåIJÍåd'Žäÿłçž£çÍŃäÿ■ä¡£çŤÍåŘŇäÿĂäÿłéŸ§åĹŮïijŇéĆčäźĹä¡äéIJĂèęĄåćđåŁäéĂĆå¡ŞçŽ
åŔŕäžěæ§ěçIJŃ 12.3 åřŔèŁĆçŽĎä¿Ńå■ŘæijŤçd'žæŸŕæĂŐæăůåĄŽçŽĎãĂĆ

heapq æÍąåİŮçŽĎåőŸæŰźæŰĞæąčæIJL'æŻt'èŕęçżEçŽĎä¿Ńå■ŘçÍŃåžŔäžěåŘŁåŕźäžŐåäEçŘĘèőžåŘŁ

3.6 1.6 å■ŮåĚÿäÿ■çŽĎéŤőæŸăåřĎåd'ŽäÿłåĂij

éŮőécŸ

æĂŐæăůåőđçŐřäÿĂäÿłéŤőåŕźåžŤåd'ŽäÿłåĂijçŽĎå■ŮåĚÿïijĹäź§åŔń
multidictïijL'ïij§

èğčåEşæŰźæąĹ

äÿĂäÿłå■ŮåĚÿåřśæŸŕäÿĂäÿłéŤőåŕźåžŤäÿĂäÿłå■ŢåĂijçŽĎæŸăåřĎãĂĆåęĆæđIJäj¡äæČşèęĄäÿĂäÿłéŤőá
æŕŤåęĆåĹŮèąĺæĹŰèĂĚéŻĘåŘĹéĞŃéÍćãĂĆæŕŤåęĆïijŇäj¡ääŔŕäžěåČŔäÿŃéÍćè£ŹæăůæđĎéĂäè£ŹæăůçŽĎå

```
d = {
    'a' : [1, 2, 3],
    'b' : [4, 5]
}
e = {
    'a' : {1, 2, 3},
    'b' : {4, 5}
}
```

éĂL'æŃl'ä¡£çŤÍåĹŮèąĺè£ŸæŸŕéŻĘåŘĹåŘŰåEşäžŐä¡äçŽĎåőđéŹĚéIJĂæśĆãĂĆåęĆæđIJäj¡äæČşä£ÍæŇA
åęĆæđIJæČşåŐżæŐL'éĞ■åd'■åĚČçt'äářśä¡£çŤĺéŻĘåŘĹïijĹázűäÿŤäÿ■åĚşå£ČåĚČçt'äçŽĎéąžäžŔéŮőécŸïijL

ä¡ääŔŕäžěå¿ĹæŰźä¿£çŽĎä¡£çŤÍ collections æÍąåİŮäÿ■çŽĎ
defaultdict æÍěæđĎéĂäè£ŹæăůçŽĎå■ŮåĚÿãĂĆ defaultdict
çŽĎäÿĂäÿłçL'źå¿ĄæŸŕåőČäijŽèĞłåŁĺåĹÍåğŇåŇŰærŔäÿł key
åĹŽåijÅåğŇåŕźäžŤçŽĎåĂijïijŇæL'Ääžěä¡ääŔłéIJĂèęĄåĚşæşĺæůżåŁääĚČçt'äæŞ■ä¡IJäžĘãĂĆærŤåęĆïijŽ

```
from collections import defaultdict

d = defaultdict(list)
```

```
d['a'].append(1)
d['a'].append(2)
d['b'].append(4)

d = defaultdict(set)
d['a'].add(1)
d['a'].add(2)
d['b'].add(4)
```

éIJÄèęĄæşÍæĎŘçŽĎæŸřïijŇ `defaultdict` äijŽèĞłåŁÍäÿžåřĘèęĄèő£éŮőçŽĎéŤőïijĹåřśçőŮçŻőåL'■å åęĆæđIJä¡ääźűäÿ■éIJÄèęĄè£ŹæăůçŽĎçL'źæĂğïijŇä¡ääŘŕäżěåIJĺäÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿäÿŁä¡£çŤÍ `setdefault()` æŰźæşŢæİěäżčæŻ£ãĂĆæŕŤåęĆïijŽ

```
d = {} # A regular dictionary
d.setdefault('a', []).append(1)
d.setdefault('a', []).append(2)
d.setdefault('b', []).append(4)
```

ä¡ĘæŸŕå¿Ĺåď'ŽçÍŃåžŘåŚŸèğL'å¿Ů `setdefault()` çŤÍèţůæİěæIJL'çĆźåĹńæL'■ãĂĆåŻääÿžæŕŔæňąèř `[]` ïijL'ãĂĆ

èőÍèőž

äÿĂèĹňæİěèőšïijŇåĹŻåżžäÿĂäÿłåď'ŽåĂijæŸăářĎå■ŮåĚÿæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆä¡ĘæŸŕïijŇåęĆæđIJä ä¡ääŘŕèČ¡äijŽåČŘäÿŃéİćè£ŹæăůæİěåőđçŐřïijŽ

```
d = {}
for key, value in pairs:
    if key not in d:
        d[key] = []
    d[key].append(value)
```

åęĆæđIJä¡£çŤÍ `defaultdict` çŽĎèŕİäżčçăĄåřśæŻť'åŁăçőĂæť'ĄäžĘïijŽ

```
d = defaultdict(list)
for key, value in pairs:
    d[key].append(value)
```

è£ŹäÿĂåřŔèŁĆæL'Äèőĺèőžç ŽĎéŮőécŸèů§æŢřæ■őåď'ĎçŘĘäÿ■çŽĎèőřå¡Ţå¡ŠçśżéŮőécŸæIJL'åď'ğçŽĎ 1.15 åřŔèŁĆçŽĎä¿Ńå■ŘãĂĆ

3.7 1.7 å■ŮåĚÿæŐŠåžŔ

éŮőécŸ

ä¡ăæČşåĹŻåżžäÿĂäÿłå■ŮåĚÿïijŇåźűäÿŤåIJĺè£■äżčæĹŰåžŘåĹŮåŇŰè£ŹäÿłåŮåĚÿçŽĎæŮűåĂŹèČ¡åd

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡æŐğåĹűäÿĂäÿłå■ŮåĚÿäÿ■åĚČçť'äçŽĎéąžåžŔïijŇä¡ääŔŕäżěä¡£çŤÍ collections æÍąåİŮäÿ■çŽĎ OrderedDict çśžãĂĆ åIJÍè£■äžčæŞ■ä¡IJçŽĎæŮűåĂŹåőČäijŽä£İæŇĄåĚČçť'äèćńæŔŠåĚěæŮűçŽĎéąžåžŔïijŇçď'žä¿ŃåęĆäÿŃïijŽ

```
from collections import OrderedDict

d = OrderedDict()
d['foo'] = 1
d['bar'] = 2
d['spam'] = 3
d['grok'] = 4
# Outputs "foo 1", "bar 2", "spam 3", "grok 4"
for key in d:
    print(key, d[key])
```

å¡Şä¡äæČşèęĄæđĎåžžäÿĂäÿłářĘæİěéIJĂèęĄåžŔåĹŮåŇŰæĹŰçijŰçăĄæĹŘåĚűäžŰæäijåijŔçŽĎæŸäářI OrderedDict æŸŕéİđåÿÿæIJL'çŤÍçŽĎãĂĆ æŕŤåęĆïijŇä¡äæČşçš¿çąőæŐğåĹűäžě JSON çijŰçăĄåŘŐå■Ůæőţç ŽĎéąžåžŔïijŇä¡ääŔŕäżěåĚĹä¡£çŤÍ OrderedDict æİěæđĎåžžè£ŹæäůçŽĎæŢřæ■őïijŽ

```
>>> import json
>>> json.dumps(d)
'{"foo": 1, "bar": 2, "spam": 3, "grok": 4}'
>>>
```

èőÍèőž

OrderedDict åĘĚéČÍçžť'æŁď'çİĂäÿĂäÿłæäźæ■őéŤőæŔŠåĚěéąžåžŔæŐŠåžŔçŽĎåŔŇåŘŠéŞ¿èąĺãĂĆ åőČäijŽèćńæŤ¿åĹřéŞ¿èąĺçŽĎår¿éČÍãĂĆårźäžŐäÿĂäÿłåůšçżŔå■ŸåIJÍçŽĎéŤőçŽĎéĞ■åď'■ètŇåĂijäÿ■äijŽæŤ

éIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇäÿĂäÿł OrderedDict çŽĎåď'ğårŔæŸŕäÿĂäÿłæŹőéĂŽå■ŮåĚÿçŽĎäÿď'å æĽĂäżěåęĆædIJä¡äèęĄædĎåžžäÿĂäÿłéIJĂèęĄåď'ğéĞŔ OrderedDict åőđä¿ŃçŽĎæŢřæ■őçżŞædĎçŽĎæŮűåĂŹïijĹærŤåęĆèŕžåŔŰ 100,000 èąŇ CSV æŢřæ■őåĹřäÿĂäÿł OrderedDict åĹŮèąĺäÿ■åŐżïijL'ïijŇ éČčäźĹä¡ääřšå¿ŮäžŤçżĘæİČèąąäÿĂäÿŃæŸŕåŘęä¡£çŤÍ OrderedDict åÿęæİěçŽĎåě¡åď'ĎèęĄåď'ğè£ĞéčÍåď'ŰåĘĚå■ŸæűĹèĂŮçŽĎå¡šåŞ■ãĂĆ

3.8 1.8 å■ŮåĚÿçŽĎè£ŘçőŮ

éŮőécŸ

æĂŐæăůåIJÍæŢřæ■őå■ŮåĚÿäÿ■æĽğèąŇäÿĂäžŻèőąçőŮæŞ■ä¡IJïijĹærŤåęĆæśĆæIJĂåřŔåĂijãĂĄæIJĂå

èğčåĘşæŰźæąĹ

èĂČèŹŚäÿŃéİćçŽĎèĆąçěĺåŘ■åŠŇäžůæäijæŸăářĎå■ŮåĚÿïijŽ

```
prices = {
    'ACME': 45.23,
    'AAPL': 612.78,
    'IBM': 205.55,
    'HPQ': 37.20,
    'FB': 10.75
}
```

äÿžäžĘåŕźå■ŮåĚÿåĂijæĽğèąŇèőąçőŮæŞ■ä¡IJïijŇéĂŽåÿÿéIJĂèęĄä¡£çŤĺ `zip()` åĞ¡æŢřåĚĹářĘéŤőåŠŇåĂijåŘ■è¡ňè£ĞæİěãĂĆ æŕŤåęĆïijŇäÿŃéİćæŸŕæ§ěæĽ¿æIJĂåřŔåŠŇæIJĂåď ğèĆąçěĺ

```
min_price = min(zip(prices.values(), prices.keys()))
# min_price is (10.75, 'FB')
max_price = max(zip(prices.values(), prices.keys()))
# max_price is (612.78, 'AAPL')
```

çśżäijijçŽĎïijŇåŔŕäžěä¡£çŤĺ `zip()` åŠŇ `sorted()` åĞ¡æŢřæİěæŐŠåĹŮå■ŮåĚÿæŢřæ■őïijŽ

```
prices_sorted = sorted(zip(prices.values(), prices.keys()))
# prices_sorted is [(10.75, 'FB'), (37.2, 'HPQ'),
#                   (45.23, 'ACME'), (205.55, 'IBM'),
#                   (612.78, 'AAPL')]
```

æĽğèąŇè£ŹäžŻèőąçőŮçŽĎæŮűåĂŹïijŇéIJĂèęĄæşĺæĎŘçŽĎæŸŕ `zip()` åĞ¡æŢřåĹŻåżžçŽĎæŸŕäÿĂäÿłåŘłèČ¡èő£éŮőäÿĂæňąçŽĎè£■äžčåŹĺãĂĆ æŕŤåęĆïijŇäÿŃéİćçŽĎäžčçăĄåŕšäijŽäžğçŤ§éŤŹèŕŕïijŽ

```
prices_and_names = zip(prices.values(), prices.keys())
print(min(prices_and_names)) # OK
print(max(prices_and_names)) # ValueError: max() arg is an empty␣
↪sequence
```

èőĺèőž

åęĆæđIJä¡ăåIJĺäÿĂäÿłå■ŮåĚÿäÿŁæĽğèąŇæŹőéĂŽçŽĎæŢřå■ęè£ŘçőŮïijŇä¡ääijŽåŔŚçŐřåőČäžňäžĚäž

```
min(prices) # Returns 'AAPL'
max(prices) # Returns 'IBM'
```

è£ŹäÿłçżŞæđIJåźűäÿ■æŸŕä¡äæČşèęĄçŽĎïijŇåŻääÿžä¡äæČşèęĄåIJĺå■ŮåĚÿçŽĎåĂijéŻĘåŘĹäÿŁæĽğèą æĹŰèőÿä¡ääijŽářİèŕŢçİĂä¡£çŤĺå■ŮåĚÿçŽĎ `values()` æŰźæşŢæİèèğčåĘşè£ŹäÿłéŮőécŸïijŽ

```
min(prices.values()) # Returns 10.75
max(prices.values()) # Returns 612.78
```

äÿ■åźÿçŽĎæŸŕïijŇéĂŽåÿÿè£Źäÿłçżş æđIJåŘŇæăůäź§äÿ■æŸŕä¡ăæČşèęĄçŽĎãĂĆ
ä¡ăåŔŕèČ¡è£ŸæČşèęĄç§ěéĄŞåŕźåžŤçŽĎéŤőçŽĎä£ąæĄŕïijĹæŕŤåęĆéĆČçğ■èĆąçĚĺäžůæăijæŸŕæIJĂä¡ŐçŽĎï

ä¡ăåŔŕäżěåIJÍ `min()` åŠŇ `max()` åĞ¡æŢřäÿ■æŘŘä¿Ż `key` åĞ¡æŢřåŔĆæŢřæİěèŐůåŘŰæIJĂåřŔåĂijæĹŰæIJĂåď ğåĂijåŕźåžŤçŽĎéŤőçŽĎä£ąæĄŕãĂĆæŕŤåęĆïijŽ

```
min(prices, key=lambda k: prices[k]) # Returns 'FB'
max(prices, key=lambda k: prices[k]) # Returns 'AAPL'
```

ä¡ĘæŸŕïijŇåęĆæđIJè£ŸæČşèęĄå¿ŮåĹřæIJĂåřŔåĂijïijŇä¡ăåŔĹå¿ŰæL'ğèąŇäÿĂæňąæ§ěæL'¿æŞ■ä¡IJãĂ

```
min_value = prices[min(prices, key=lambda k: prices[k])]
```

åL'■éİćçŽĎ `zip()` åĞ¡æŢřæŰźæąĹéĂŽè£ĞåřĘå■ŮåĚÿâĂİåŔ■è¡ňâĂİäÿž (åĂijïijŇéŤő) åĚČçżĎåžŔåĹŰæİèèğčåĘşäžĘäÿŁè£řéŮőécŸãĂĆ
å¡ŞæŕŤè¿Čäÿď äÿłåĚČçżĎçŽĎæŮűåĂŹïijŇåĂijäijŽåĚĹè£ŻèąŇæŕŤè¿ČïijŇçĎűåŘŐæL'■æŸŕéŤőãĂĆ
è£ŹæăůçŽĎèŕİä¡ăåřśèČ¡éĂŽè£ĞäÿĂæİąçőĂå■ŢçŽĎèŕ■åŔèåřśèČ¡å¿Ĺè¡żæİ¿çŽĎåőđçŐřåIJĺå■ŮåĚÿäÿŁçŽĎ

éIJĂèęĄæşĺæĎŘçŽĎæŸŕåIJĺèőąçőŮæŞ■ä¡IJäÿ■ä¡£çŤĺåĹřäžĘ (åĂijïijŇéŤő) åŕźãĂĆå¡Şåď Žäÿłåőđä¡ŞæŃěæIJL'çŽÿåŘŇçŽĎåĂijçŽĎæŮűåĂŹïijŇéŤőäijŽåĘşåőŽè£ŤåŻđçżŞæđIJãĂĆ
æŕŤåęĆïijŇåIJĺæL'ğèąŇ `min()` åŠŇ `max()` æŞ■ä¡IJçŽĎæŮűåĂŹïijŇåęĆæđIJæĄřåůğæIJĂåřŔæĹŰæIJĂåď

```
>>> prices = { 'AAA' : 45.23, 'ZZZ': 45.23 }
>>> min(zip(prices.values(), prices.keys()))
(45.23, 'AAA')
>>> max(zip(prices.values(), prices.keys()))
(45.23, 'ZZZ')
>>>
```

3.9 1.9 æ§ěæL'¿äÿď å■ŮåĚÿçŽĎçŻÿåŘŇçĆź

éŮőécŸ

æĂŐæăůåIJĺäÿď äÿłå■ŮåĚÿäÿ■åŕżåŕżæL'¿çŻÿåŘŇçĆźïijĹæŕŤåęĆçŻÿåŘŇçŽĎéŤőãĂĄçŻÿåŘŇçŽĎåĂij

èğčåĘşæŰźæąĹ

èĂČèŹŚäÿŃéİćäÿď äÿłå■ŮåĚÿïijŽ

```
a = {
    'x' : 1,
    'y' : 2,
    'z' : 3
}

b = {
    'w' : 10,
    'x' : 11,
```

```
    'y' : 2
}
```

äÿžäžĘåŕżæĽ¿äÿďäÿłå■ŮåĚÿçŽĎçŻÿåŘŇçĆźïijŇåŔŕäžěçőĂå■ŢçŽĎåIJÍäÿďå■ŮåĚÿçŽĎ `keys()` æĹŰèĂĚ `items()` æŰźæşŢè£ŤåŻđçżŞæđIJäÿŁæĽğèąŇéŻĘåŘĹæŞ■ä¡IJãĂĆæŕŤåęĆïijŽ

```
# Find keys in common
a.keys() & b.keys() # { 'x', 'y' }
# Find keys in a that are not in b
a.keys() - b.keys() # { 'z' }
# Find (key,value) pairs in common
a.items() & b.items() # { ('y', 2) }
```

è£ŹäžŻæŞ■ä¡IJäź§åŔŕäžěçŢĺäžŐä£őæŢźæĹŰèĂĚè£ĞæżďåŮåĚÿåĚČçťăãĂĆ æŕŤåęĆïijŇåĄĞåęĆä¡ăæČşäžěçŐřæIJL'å■ŮåĚÿæđĎéĂääÿĂäÿłæŐŠéŹďåĞăäÿłæŇĞåőŽéŤőçŽĎæŰřå■ŮåĚÿäÿŃéÍćåĹl'çŢĺå■ŮåĚÿæŐĺåŕijæÍěåðđçŐřè£ŹæăůçŽĎéIJĂæśĆïijŽ

```
# Make a new dictionary with certain keys removed
c = {key:a[key] for key in a.keys() - {'z', 'w'}}
# c is {'x': 1, 'y': 2}
```

èőÍèőž

äÿĂäÿłå■ŮåĚÿåŕśæŸŕäÿĂäÿłéŤőéŻĘåŘĹäÿŐåĂijéŻĘåŘĹçŽĎæŸăåřĎåĚşçşżãĂĆ å■ŮåĚÿçŽĎ `keys()` æŰźæşŢè£ŤåŻđäÿĂäÿłåśŢçŐřéŤőéŻĘåŘĹçŽĎéŤőèğĘåŻ¿åŕźèśąãĂĆ éŤőèğĘåŻ¿çŽĎäÿĂäÿłå¿ĹåřŚèćńäžĘèğčçŽĎçĽźæĂğåŕśæŸŕåőČäžňäź§æŤŕæŇĄéŻĘåŘĹæŞ■ä¡IJïijŇæŕŤåęĆ æĽĂäžěïijŇåęĆæđIJä¡ăæČşåŕźéŻĘåŘĹçŽĎéŤőæĽğèąŇäÿĂäžŻæŹőéĂŽçŽĎéŻĘåŘĹæŞ■ä¡IJïijŇåŔŕäžěçŻťsetãĂĆ

å■ŮåĚÿçŽĎ `items()` æŰźæşŢè£ŤåŻđäÿĂäÿłåŇĚåŘń (éŤőïijŇåĂij) åŕźçŽĎåĚČçťăèğĘåŻ¿åŕźèśąãĂĆ è£ŹäÿłåŕźèśąåŘŇæăůäź§æŤŕæŇĄéŻĘåŘĹæŞ■ä¡IJïijŇåźűäÿŤåŔŕäžěèćńçŤ

år¡çőąå■ŮåĚÿçŽĎ `values()` æŰźæşŢäź§æŸŕçśżäijijïijŇä¡ĘæŸŕåőČåźűäÿ■æŤŕæŇĄè£ŹéĞŇäżŃçż■ç æ§Řçğ■çÍŃåžęäÿŁæŸŕåŻăäÿžåĂijèğĘåŻ¿äÿ■èČ¡ä£ÌèŕĄæĽĂæIJL'çŽĎåĂijäžŠäÿ■çŻÿåŘŇïijŇè£ŹæăůäijŽår äÿ■è£ĞïijŇåęĆæđIJä¡ăçąňèęĄåIJÍåĂijäÿŁéÍćæĽğèąŇè£ŹäžŻéŻĘåŘĹæŞ■ä¡IJçŽĎèŕÍïijŇä¡ääŔŕäžěåĚĹårĘåĂ setïijŇçĎűåŘŐåĘ■æĽğèąŇéŻĘåŘĹè£ŘçőŮårśèąŇäžĘãĂĆ

3.10 1.10 åĹăéŹď åžŔåĹŮçŻÿåŘŇåĚČçťăäźűä£ÌæŇĄéąžåžŘ

éŮőécŸ

æĂŐæăůåIJÍäÿĂäÿłåžŔåĹŮäÿŁéÍćä£ÌæŇĄåĚČçťăéąžåžŘçŽĎåŘŇæŮűæűĹéŹď éĞ■åď■çŽĎåĂijïij§

èğčåĘşæŰźæąĹ

åęĆæđIJåžŔåĹŮäÿŁçŽĎåĂijéČ¡æŸŕ `hashable` çśżåđŃïijŇéĆčäźĹåŔŕäžěå¿ĹçőĂå■ŢçŽĎåĹl'çŢĺéŻĘå

```
def dedupe(items):
    seen = set()
    for item in items:
        if item not in seen:
            yield item
            seen.add(item)
```

äÿŃéÍćæŸŕä¡£çŤÍäÿŁè£ŕåĞ¡æŢřçŽĎä¿Ńå■ŘïijŽ

```
>>> a = [1, 5, 2, 1, 9, 1, 5, 10]
>>> list(dedupe(a))
[1, 5, 2, 9, 10]
>>>
```

è£ŹäÿłæŰźæşŢäżĚäżĚåIJÍåžŔåĹŮäÿ■åĚČçť´ääÿž hashable çŽĎæŮűåĂŹæL'■çőąçŤÍãĂĆ åęĆæđIJä¡ăæČşæűĹéŹđ'åĚČçť´ääÿ■åŔŕåŞĹåÿŇïijĹæŕŤåęĆ dict çśżåđŃïijL'çŽĎåžŔåĹŮäÿ■éĞ■åđ'■åĚČçť´ăçŽĎèŕÍïijŇä¡ăéIJĂèęĄåřĘäÿŁè£ŕäžčçăĄçÍ■å¿őæŤźåŔŸäÿĂ

```
def dedupe(items, key=None):
    seen = set()
    for item in items:
        val = item if key is None else key(item)
        if val not in seen:
            yield item
            seen.add(val)
```

è£ŹéĞŇçŽĎkeyåŔĆæŢřæŇĞåőŽäžĘäÿĂäÿłåĞ¡æŢřïijŇåřĘåžŔåĹŮåĚČçť´ăè¡ňæ■ćæĹŘ hashable çśżåđŃãĂĆäÿŃéÍćæŸŕåőČçŽĎçŤÍæşŢçđ'žä¿ŃïijŽ

```
>>> a = [ {'x':1, 'y':2}, {'x':1, 'y':3}, {'x':1, 'y':2}, {'x':2, 'y
↪':4}]
>>> list(dedupe(a, key=lambda d: (d['x'],d['y'])))
[{'x': 1, 'y': 2}, {'x': 1, 'y': 3}, {'x': 2, 'y': 4}]
>>> list(dedupe(a, key=lambda d: d['x']))
[{'x': 1, 'y': 2}, {'x': 2, 'y': 4}]
>>>
```

åęĆæđIJä¡ăæČşå§žäžŐå■Ţäÿłå■Ůæőţã˙ĂĄåśđæĂğæĹŰèĂĚæ§ŘäÿłæŻť'åđ'ğçŽĎæŢřæ■őçżŞæđĎæİěæű

èőÍèőž

åęĆæđIJä¡ăäżĚäżĚåřśæŸŕæČşæűĹéŹđ'éĞ■åđ'■åĚČçť´ăïijŇéĂŽåÿÿåŔŕäžěçőĂå■ŢçŽĎæđĎéĂääÿĂäÿłé

```
>>> a
[1, 5, 2, 1, 9, 1, 5, 10]
>>> set(a)
{1, 2, 10, 5, 9}
>>>
```

çĎűèĂŇïijŇè£Źçğ■æŰźæşŢäÿ■èČ¡çżť'æŁđ'åĚČçť´äçŽĎéąžåžŔïijŇçŤ§æĹŘçŽĎçżŞæđIJäÿ■çŽĎåĚČçť

åIJÍæIJňèŁĆäÿ■æĹŚäżňä¡£çŤÍäžĘçŤ§æĹŘåŹÍåĞ¡æŢřèőľæĹŚäżňçŽĎåĞ¡æŢřæŻťåŁăéĂŽçŤÍïijŇäÿ■äż
æŕŤåęĆïijŇåęĆæđIJåęĆæđIJä¡ăæČşèŕżåŔŰäÿĂäÿłæŰĞäżűïijŇæűĹéŹďéĞ■åď■èąŇïijŇä¡ăåŔŕäżěå¿ĹåőźæŸ

```
with open(somefile,'r') as f:
for line in dedupe(f):
    ...
```

äÿŁè£řkeyåĞ¡æŢřåŔĆæŢřæĺąäż£äžĘ sorted() , min() åŠŇ max() ç■ĽåĘĚç¡őåĞ¡æŢřçŽĎçŻÿäijijåŁ§èČ¡ãĂĆ åŔŕäżěåŔĆèĂČ 1.8 åŠŇ 1.13 åřŔèŁĆäžĘèğčæŻťåďŽãĂĆ

3.11 1.11 åŚ¡åŘ■åĹĞçĽĞ

éŮőéćŸ

ä¡ăçŽĎçÍŃåžŔåůšçżŔåĞžçŐřäÿĂåďğåăĘåůšæŮăæşŢçŻťèğĘçŽĎçąňçijŰçăĄåĹĞçĽĞäÿŃæăĞïijŇçĎű

èğčåĘşæŰźæąĹ

åĄĞåőŽä¡ăæIJĽäÿĂæőţäżčçăĄèęĄäżŐäÿĂäÿłèőřå¡Ţå■Ůçňęäÿšäÿ■åĞăäÿłåŻžåőŽä¡■ç¡őæŔŘåŔŰåĞžç

```
######
↪012345678901234567890123456789012345678901234567890123456789 0'
record = '.....................100 .......513.25 ..........'
cost = int(record[20:23]) * float(record[31:37])
```

äÿŐåĚűéĆčæăůåĘŹïijŇäÿžäżĂäźĹäÿ■æČşè£ŹæăůåŚ¡åŘ■åĹĞçĽĞåŚćïijŽ

```
SHARES = slice(20, 23)
PRICE = slice(31, 37)
cost = int(record[SHARES]) * float(record[PRICE])
```

çňňäžŇçğ■çĽĹæIJňäÿ■ïijŇä¡ăéĄ£åĚ■äžĘåďğéĞŔæŮăæşŢçŘĘèğčçŽĎçąňçijŰçăĄäÿŃæăĞïijŇä¡£å¿Ů

èőĺèőž

äÿĂèĹňæİěèőšïijŇäżčçăĄäÿ■åęĆæđIJåĞžçŐřåďğéĞŔçŽĎçąňçijŰçăĄäÿŃæăĞåĂijäijŽä¡£å¿ŮåŔŕèŕżæ
æŕŤåęĆïijŇåęĆæđIJä¡ăåŻđè£ĞæİěçIJŃçIJŃäÿĂåźťåĽ■ä¡ăåĘŹçŽĎäżčçăĄïijŇä¡ăäijŽæŚÿçİĂèĎŚèćŃæČşéĆ
è£ŹéĞŇçŽĎèğčåĘşæŰźæąĹæŸŕäÿĂäÿłå¿ĹçőĂå■ŢçŽĎæŰźæşŢèőľä¡ăæŻťåŁăæÿĚæŹřçŽĎèąĺè¿¿äżčçăĄåĹ

åĘĚç¡őçŽĎ slice() åĞ¡æŢřåĹŻåżžäžĘäÿĂäÿłåĹĞçĽĞåŕźèśąïijŇåŔŕäżěèćńçŤĺåIJĺäżżä¡ŢåĹĞçĽĞåĚ

```
>>> items = [0, 1, 2, 3, 4, 5, 6]
>>> a = slice(2, 4)
>>> items[2:4]
[2, 3]
>>> items[a]
```

```
[2, 3]
>>> items[a] = [10,11]
>>> items
[0, 1, 10, 11, 4, 5, 6]
>>> del items[a]
>>> items
[0, 1, 4, 5, 6]
```

åęĆæđIJä¡ăæIJL'äÿĂäÿłåĹĞçL'ĞåŕźèśąaïijŇä¡ăåŔŕäžěåĹĘåĹńèřČçŤĺåőČçŽĎ `a.start`, `a.stop`, `a.step` åśđæĂğæİěèŐůåŔŰæŻt'åd'ŽçŽĎä£ąæĄŕãĂĆæŕŤåęĆïijŽ

```
>>> a = slice(5, 50, 2)
>>> a.start
5
>>> a.stop
50
>>> a.step
2
>>>
```

åŔęåd'ŰïijŇä¡ăè£ŸèČ¡éĂŽè£ĞèřČçŤĺåĹĞçL'ĞçŽĎ `indices(size)` æŰźæşŢåřĘåőČæŸăåřĎåĹřäÿĂäÿłçąőåőŽåd'ğåřŔçŽĎåžŔåĹŮäÿŁïijŇ è£ŹäÿłæŰźæşŢè£ŤåŻđäÿĂäÿłäÿL'åĚČçżĎ `(start, stop, step)` ïijŇæL'ĂæIJL'åĂijéČ¡äijŽèćńåŘĹéĂĆçŽĎçjl'åřŔäžěæżąèűşè¿źçŢŇéŹŘåĹűïijŇ äžŐèĂŇä¡£çŤĺçŽĎæŮűåĂŹéĄ£åĚ■åĞžçŐř `IndexError` åijĆåÿÿãĂĆæŕŤåęĆïijŽ

```
>>> s = 'HelloWorld'
>>> a.indices(len(s))
(5, 10, 2)
>>> for i in range(*a.indices(len(s))):
...     print(s[i])
...
W
r
d
>>>
```

3.12 1.12 åžŔåĹŮäÿ■åĞžçŐřæňąæŢřæIJĂåd'ŽçŽĎåĚČçt'ă

éŮőécŸ

æĂŐæăůæL'¿åĞžäÿĂäÿłåžŔåĹŮäÿ■åĞžçŐřæňąæŢřæIJĂåd'ŽçŽĎåĚČçt'ăåŚćïij§

èğčåĘşæŰźæąĹ

`collections.Counter` çśżåřśæŸŕäÿŞéŮĺäÿžè£ŹçśżéŮőécŸèĂŇèő¿èőąçŽĎïijŇ åőČçŤŽèĞşæIJL'äÿĂäÿłæIJL'çŤĺçŽĎ `most_common()` æŰźæşŢçŻt'æŐěçżŹäžĘä¡äç■ŤæąĹãĂĆ

äÿžäžĘæijŤçď žïijŇåĚĹåĄĜèő¿ä¡ăæIJL’äÿĂäÿłå■Ţèŕ■åĹŮèąĺåźűäÿŤæČşæL’¿åĞžåŞłäÿłå■Ţèŕ■åĞžçŐřé

```
words = [
    'look', 'into', 'my', 'eyes', 'look', 'into', 'my', 'eyes',
    'the', 'eyes', 'the', 'eyes', 'the', 'eyes', 'not', 'around',
↪'the',
    'eyes', "don't", 'look', 'around', 'the', 'eyes', 'look', 'into
↪',
    'my', 'eyes', "you're", 'under'
]
from collections import Counter
word_counts = Counter(words)
# åĞžçŐřéćŚçŐĞæIJĂéńŸçŽĎ3äÿłå■Ţèŕ■
top_three = word_counts.most_common(3)
print(top_three)
# Outputs [('eyes', 8), ('the', 5), ('look', 4)]
```

èőĺèőž

ä¡IJäÿžè¿ŞåĚëïijŇ `Counter` åŕźèśąåŔŕäżěæŐěåŔŮäżżæĎŔçŽĎçŤśåŔŕåŞĹåÿŇïijĹhashableïijL’åĚČ
åIJĺåžŢåśĆåőđçŐřäÿŁïijŇäÿĂäÿł `Counter` åŕźèśąåřśæŸŕäÿĂäÿłå■ŮåĚÿïijŇåřĘåĚČçť’ăæŸăåřĎåĹřåőČåĞžç

```
>>> word_counts['not']
1
>>> word_counts['eyes']
8
>>>
```

åęĆæđIJä¡ăæČşæL’ŃåŁĺåćđåŁăèőąæŢřïijŇåŔŕäżěçőĂå■ŢçŽĎçŤĺåŁăæşŢïijŽ

```
>>> morewords = ['why','are','you','not','looking','in','my','eyes']
>>> for word in morewords:
...     word_counts[word] += 1
...
>>> word_counts['eyes']
9
>>>
```

æĹŰèĂĚä¡ăåŔŕäżěä¡£çŤĺ `update()` æŰźæşŢïijŽ

```
>>> word_counts.update(morewords)
>>>
```

`Counter` åőđä¿ŃäÿĂäÿłéšIJäÿžäžžç§ěçŽĎçL’źæĂğæŸŕåőČäżňåŔŕäżěå¿ĹåőźæŸ§çŽĎèů§æŢřå■ęè£Řç

```
>>> a = Counter(words)
>>> b = Counter(morewords)
>>> a
Counter({'eyes': 8, 'the': 5, 'look': 4, 'into': 3, 'my': 3, 'around
↪': 2,
```

```
"you're": 1, "don't": 1, 'under': 1, 'not': 1})
>>> b
Counter({'eyes': 1, 'looking': 1, 'are': 1, 'in': 1, 'not': 1, 'you
 ↪': 1,
'my': 1, 'why': 1})
>>> # Combine counts
>>> c = a + b
>>> c
Counter({'eyes': 9, 'the': 5, 'look': 4, 'my': 4, 'into': 3, 'not':␣
 ↪2,
'around': 2, "you're": 1, "don't": 1, 'in': 1, 'why': 1,
'looking': 1, 'are': 1, 'under': 1, 'you': 1})
>>> # Subtract counts
>>> d = a - b
>>> d
Counter({'eyes': 7, 'the': 5, 'look': 4, 'into': 3, 'my': 2, 'around
 ↪': 2,
"you're": 1, "don't": 1, 'under': 1})
>>>
```

æŕńæŮăçŰŚéŮőïijŇ Counter åŕźèśąåIJĺåĞääźŐæĽĂæIJL'éIJĂèęĄåĹűèąĺæĹŰèĂĚèőąæŢřæŢřæ■őçŽÌ
åIJĺèğčåEşè£ŹçśżéŮőécŸçŽĎæŮűåĂŹä¡ăäžŤèŕěäijŸåĚĹéĂĽ'æŃľåőČïijŇèĂŇäÿ■æŸŕæĽ'ŃåŁĺçŽĎåĹľçŤĺå

3.13 1.13 éĂŽè£Ğæ§ŘäÿłåĚşéŤőå■ŮæŐŠåžŔäÿĂäÿłå■ŮåĚÿåĹŮèąĺ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłå■ŮåĚÿåĹŮèąĺïijŇä¡ăæČşæäźæ■őæ§ŘäÿłæĹŰæ§ŘåĞääÿłå■ŮåĚÿå■ŮæőţæİěæŐŠåžŔè£

èğčåEşæŰźæąĹ

éĂŽè£Ğä¡£çŤĺ operator æĺąåİŮçŽĎ itemgetter
åĞ¡æŢřïijŇåŔŕäżěéİđåÿÿåőźæŸŞçŽĎæŐŠåžŔè£ŹæäůçŽĎæŢřæ■őçżŞæđĎãĂĆ
åĄĞèő¿ä¡ăäžŐæŢřæ■őåžŞäÿ■æčĂçť'ćåĞžæİěç¡ŚçńŹäijŽåŚŸä£qæĄŕåĹŮèąĺïijŇåźűäÿŤäžěäÿŃåĹŮçŽĎæŢřa

```
rows = [
    {'fname': 'Brian', 'lname': 'Jones', 'uid': 1003},
    {'fname': 'David', 'lname': 'Beazley', 'uid': 1002},
    {'fname': 'John', 'lname': 'Cleese', 'uid': 1001},
    {'fname': 'Big', 'lname': 'Jones', 'uid': 1004}
]
```

æăźæ■őäżżæĎŔçŽĎå■ŮåĚÿå■ŮæőţæİěæŐŠåžŔè¿ŞåĚěçżŞæđIJèąŇæŸŕå¿Ĺåőźæ ŸŞåőđçŐřçŽĎïijŇäżč

```
from operator import itemgetter
rows_by_fname = sorted(rows, key=itemgetter('fname'))
rows_by_uid = sorted(rows, key=itemgetter('uid'))
```

```
print(rows_by_fname)
print(rows_by_uid)
```

äżčçăĄçŽĎè¿ŞåĞžåęĆäÿŃïijŽ

```
[{'fname': 'Big', 'uid': 1004, 'lname': 'Jones'},
{'fname': 'Brian', 'uid': 1003, 'lname': 'Jones'},
{'fname': 'David', 'uid': 1002, 'lname': 'Beazley'},
{'fname': 'John', 'uid': 1001, 'lname': 'Cleese'}]
[{'fname': 'John', 'uid': 1001, 'lname': 'Cleese'},
{'fname': 'David', 'uid': 1002, 'lname': 'Beazley'},
{'fname': 'Brian', 'uid': 1003, 'lname': 'Jones'},
{'fname': 'Big', 'uid': 1004, 'lname': 'Jones'}]
```

itemgetter() åĞ¡æŢřäź§æŤŕæŇĄåď Žäÿł keysïijŇæŕŤåęĆäÿŃéİćçŽĎäżčçăĄ

```
rows_by_lfname = sorted(rows, key=itemgetter('lname','fname'))
print(rows_by_lfname)
```

äijŽäžğçŤ§åęĆäÿŃçŽĎè¿ŞåĞžïijŽ

```
[{'fname': 'David', 'uid': 1002, 'lname': 'Beazley'},
{'fname': 'John', 'uid': 1001, 'lname': 'Cleese'},
{'fname': 'Big', 'uid': 1004, 'lname': 'Jones'},
{'fname': 'Brian', 'uid': 1003, 'lname': 'Jones'}]
```

èőĺèőž

åIJĺäÿŁéİćä¿Ńå■Řäÿ■ïijŇ rows èćńäijăéĂŠçżŹæŐěåŔŮäÿĂäÿłåĚşéŤőå■ŮåŔĆæŢřçŽĎ sorted() åĘĚç¡őåĞ¡æŢřãĂĆ è£ŹäÿłåŔĆæŢřæŸŕ callable çśżåđŃïijŇåźűäÿŤäżŐ rows äÿ■æŐěåŔŮäÿĂäÿłå■ŢäÿĂåĚČçť ăïijŇçĎűåŘŐè£ŤåŻđèćńçŤĺæİěæŐŠåžŔçŽĎåĂijãĂĆ itemgetter() åĞ¡æŢřåřśæŸŕèť §èť čåĹŻåżžè£Źäÿł callable åŕźèśąçŽĎãĂĆ

operator.itemgetter() åĞ¡æŢřæIJĽäÿĂäÿłèćń rows äÿ■çŽĎèőřå¡ŢçŤĺæİěæ§ěæĽ¿åĂijçŽĎçť ćåijŢåŔĆæŢřãĂĆåŔŕäżěæŸŕäÿĂäÿłå■ŮåĚÿéŤőåŘ■çğřïijŇ äÿĂäÿłæŢť å¡ćåĂijæĹŰèĂĚäżżä¡ŢèČ¡åď §äijăåĚěäÿĂäÿłåŕźèśąçŽĎ __getitem__() æŰźæşŢçŽĎåĂijãĂĆ åęĆæđIJä¡ăäijăåĚěåď Žäÿłçť ćåijŢåŔĆæŢřçżŹ itemgetter() ïijŇåőČçŤ§æĹŘçŽĎ callable åŕźèśąäijŽè£ŤåŻđäÿĂäÿłåŇĚåŘńæĽĂæIJĽåĚČçť ăåĂijçŽĎåĚČçżĎïijŇ åźűäÿŤ sorted() åĞ¡æŢřäijŽæăźæ■őè£ŹäÿłåĚČçżĎäÿ■åĚČçť ăéąžåžŔåŐżæŐŠåžŔãĂĆ ä¡Ęä¡ăæČşèęĄåŘŇæŮűåIJĺåĞăäÿłå■ŮæőţäÿŁéİćè£ŻèąŇæŐŠåžŔïijĹæŕŤåęĆéĂŽè£ĞåğŞåŠŇåŘ■æİěæŐŠåž

itemgetter() æIJĽæŮűåĂŹäź§åŔŕäżěçŤĺ lambda èąĺè¿¿åijŔäżčæŻ£ïijŇæŕŤåęĆïijŽ

```
rows_by_fname = sorted(rows, key=lambda r: r['fname'])
rows_by_lfname = sorted(rows, key=lambda r: (r['lname'],r['fname']))
```

è£Źçğ■æŰźæąĹäź§äÿ■éŤŹãĂĆä¡ĘæŸŕïijŇä¡£çŤĺ itemgetter() æŰźåijŔäijŽè£ŘèąŇçŽĎçĺ■å¿őå£ńçĆźãĂĆåŻăæ■ď ïijŇåęĆæđIJä¡ăåŕźæĂğèČ¡èęĄæśĆæŕŤè¿ČéńŸçŽĎèŕİåřś itemgetter() æŰźåijŔãĂĆ

æIJĂåŘŐïijŇäÿ■èęĄå£ŸäžĘè£ŹèŁĆäÿ■åśŢçd'žçŽĎæŁĂæIJŕäź§åŘŇæăůéĂĆçŤĺäžŐ `min()` åŠŇ `max()` ç■L'åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> min(rows, key=itemgetter('uid'))
{'fname': 'John', 'lname': 'Cleese', 'uid': 1001}
>>> max(rows, key=itemgetter('uid'))
{'fname': 'Big', 'lname': 'Jones', 'uid': 1004}
>>>
```

3.14 1.14 æŐŠåžŔäÿ■æŤŕæŇĄåŐ§çŢ§æŕŤè¿ČçŽĎåŕźèśą

éŮőécŸ

ä¡ăæČşæŐŠåžŔçśżåđŇçŻÿåŘŇçŽĎåŕźèśąïijŇä¡ĘæŸŕäżŰäžňäÿ■æŤŕæŇĄåŐ§çŢ§çŽĎæŕŤè¿ČæŞ■ä¡IJã

èğčåĘşæŰźæąĹ

åĘĚç¡őçŽĎ `sorted()` åĞ¡æŢřæIJL'äÿĂäÿłåĚşéŤőå■ŮåŔĆæŢř `key` ïijŇåŔŕäżěäijăåĚěäÿĂäÿł `callable` åŕźèśąçżŹåőČïijŇ è£ŹäÿłåĂijïijŇè£ŹäÿłåĂijäijŽèćń `sorted` çŤĺæİěæŐŠåžŔè£ŹäžŻåŕźèśąãĂĆ æŕŤåęĆïijŇåęĆæđIJä¡ăåIJĺåžŢçŤĺçĺŇåžŔéĞŇéİćæIJL'äÿĂäÿł `User` åőđä¿ŃåžŔåĹŮïijŇåźűäÿŤä¡ăåÿŇæIJŻéĂŽè£ĞäżŰäžňçŽĎ `user_id` åśđæĂğè£ŻèąŇæŐŠåžŔïijŇ ä¡ăåŔŕäżěæŔŘä¿ŻäÿĂäÿłäžě `User` åőđä¿Ńä¡IJäÿžè¿ŞåĚěåźűè¿ŞåĞžåŕźåžŤ `user_id` åĂijçŽĎ `callable` åŕźèśąãĂĆæŕŤåęĆïijŽ

```
class User:
    def __init__(self, user_id):
        self.user_id = user_id

    def __repr__(self):
        return 'User({})'.format(self.user_id)


def sort_notcompare():
    users = [User(23), User(3), User(99)]
    print(users)
    print(sorted(users, key=lambda u: u.user_id))
```

åŔęåd'ŰäÿĂçğ■æŰźåijŔæŸŕä¡£çŤĺ `operator.attrgetter()` æİěäžčæŻ£ lambda åĞ¡æŢřïijŽ

```
>>> from operator import attrgetter
>>> sorted(users, key=attrgetter('user_id'))
[User(3), User(23), User(99)]
>>>
```

èőĺèőž

éĂĽæŃľä¡£çŤĺ lambda åĞ¡æŢřæĹŰèĂĚæŸŕ attrgetter() åŔŕèČ¡åŔŰåĘşäžŐäÿłäžžåŰIJåě¡ãĂĆ ä¡ĘæŸŕïijŇ attrgetter() åĞ¡æŢřéĂŽåÿÿäijŽè£ŘèąŇçŽĎå£ńçĆźïijŇåźűäÿŤè£ŸèČ¡åŘŇæŮűåĚĄèőÿåď Žäÿłå■Ůæőţè£ŻèąŇæŕŤè¿ČãĂ è£Źäÿłèů§ operator.itemgetter() åĞ¡æŢřä¡IJçŤĺäžŐå■ŮåĚÿçśżåđŃå¿ĹçśżäijijïijĹåŔĆèĂČ1.13åřŔè ä¿ŃåęĆïijŇåęĆæđIJ User åőđä¿Ńè£ŸæIJĽäÿĂäÿł first_name åŠŇ last_name åśđæĂğïijŇéĆčäźĹåŔŕäżěåŘŚäÿŃéİćè£ŹæăůæŐŠåžŔïijŽ

```
by_name = sorted(users, key=attrgetter('last_name', 'first_name'))
```

åŘŇæăůéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇè£ŹäÿĂåřŔèŁĆçŤĺåĹřçŽĎæŁĂæIJŕåŘŇæăůéĂĆçŤĺäžŐåČŔ min() åŠŇ max() äźŃçśżçŽĎåĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> min(users, key=attrgetter('user_id'))
User(3)
>>> max(users, key=attrgetter('user_id'))
User(99)
>>>
```

3.15 1.15 éĂŽè£Ğæ§Řäÿłå■ŮæőţåřĘèőřå¡ŢåĹĘçżĎ

éŮőéćŸ

ä¡ăæIJĽäÿĂäÿłå■ŮåĚÿæĹŰèĂĚåőđä¿ŃçŽĎåžŔåĹŮïijŇçĎűåŘŐä¡ăæČşæăźæ■őæ§ŘäÿłçĽźåőŽçŽĎå■Ů date æİěåĹĘçżĎè£■äżčèő£éŮőãĂĆ

èğčåĘşæŰźæąĹ

itertools.groupby() åĞ¡æŢřåŕźäžŐè£ŹæăůçŽĎæŢřæ■őåĹĘçżĎæŞ■ä¡IJéİđåÿÿåőđçŤĺãĂĆ äÿžäžĘæijŤçď žïijŇåĄĞèő¿ä¡ăåůšçżŔæIJĽäžĘäÿŃåĹŮçŽĎå■ŮåĚÿåĹŮèąĺïijŽ

```
rows = [
    {'address': '5412 N CLARK', 'date': '07/01/2012'},
    {'address': '5148 N CLARK', 'date': '07/04/2012'},
    {'address': '5800 E 58TH', 'date': '07/02/2012'},
    {'address': '2122 N CLARK', 'date': '07/03/2012'},
    {'address': '5645 N RAVENSWOOD', 'date': '07/02/2012'},
    {'address': '1060 W ADDISON', 'date': '07/02/2012'},
    {'address': '4801 N BROADWAY', 'date': '07/01/2012'},
    {'address': '1039 W GRANVILLE', 'date': '07/04/2012'},
]
```

çŐřåIJĺåĄĞèő¿ä¡ăæČşåIJĺæŇĽ date åĹĘçżĎåŘŐçŽĎæŢřæ■őåİŮäÿŁè£ŻèąŇè£■äżčãĂĆäÿžäžĘè£Źæăůa date)æŐŠåžŔïijŇ çĎűåŘŐèřČçŤĺ itertools.groupby() åĞ¡æŢřïijŽ

```
from operator import itemgetter
from itertools import groupby

# Sort by the desired field first
rows.sort(key=itemgetter('date'))
# Iterate in groups
for date, items in groupby(rows, key=itemgetter('date')):
    print(date)
    for i in items:
        print('  ', i)
```

è£ŘèąŇçżŞæđIJïijŽ

```
07/01/2012
  {'date': '07/01/2012', 'address': '5412 N CLARK'}
  {'date': '07/01/2012', 'address': '4801 N BROADWAY'}
07/02/2012
  {'date': '07/02/2012', 'address': '5800 E 58TH'}
  {'date': '07/02/2012', 'address': '5645 N RAVENSWOOD'}
  {'date': '07/02/2012', 'address': '1060 W ADDISON'}
07/03/2012
  {'date': '07/03/2012', 'address': '2122 N CLARK'}
07/04/2012
  {'date': '07/04/2012', 'address': '5148 N CLARK'}
  {'date': '07/04/2012', 'address': '1039 W GRANVILLE'}
```

èőÍèőž

groupby() åĞ¡æŢřæL'ńæŔŕæŢt'äÿłåžŕåĹŮåźűäÿŤæ§ěæL'¿è£đçż■çŻÿåŘŇåĂijïijĹæĹŰèĂĚæăźæ■ő
key åĞ¡æŢřè£ŤåŻđåĂijçŻÿåŘŇïijL'çŽĎåĚČçt'ăåžŕåĹŮãĂĆ
åIJÍæŕŕæňąè£■äžčçŽĎæŮűåĂŹïijŇåőČäijŽè£ŤåŻđäÿĂäÿłåĂijåŠŇäÿĂäÿłè£■äžčåŹÍåŕźèśąïijŇ
è£ŹäÿłèŁ■äžčåŹÍåŕźèśąåŔŕäžěçŢ§æĹŘåĚČçt'ăåĂijåĚÍéČÍç■L'äžŐäÿŁéÍćéĆčäÿłåĂijçŽĎçżĎäÿ■æL'ĂæIJL'åŕ

äÿĂäÿłéÍđåÿÿéĞ■èęĄçŽĎåĞĘåđ'Ğæ■ěélđ'æŸŕèęĄæăźæ■őæŇĞåőŽçŽĎå■ŮæőţåřĕæŢřæ■őæŐŠåžŔãĂ
åŻăäÿž groupby() äžĚäżĚæčĂæ§ěè£đçż■çŽĎåĚČçt'ăïijŇåęĆæđIJäžŇåĚĹåźűæšąæIJL'æŐŠåžŔåőŇæĹŘç

åęĆæđIJä¡ăäżĚäżĚåŔłæŸŕæČşæăźæ■ő date å■ŮæőţåřĕæŢřæ■őåĹĘçżĎåĹřäÿĂäÿłåđ'ğçŽĎæŢřæ■őçżŞ
éĆčäźĹä¡ăæIJĂåě¡ä¡£çŤÍ defaultdict() æÍěæđĎåżžäÿĂäÿłåđ'ŽåĂijå■ŮåĚÿïijŇåĚşäžŐåđ'ŽåĂijå■ŮåĚÿ
1.6 åŕŘèŁĆæIJL'è£ĞèŕęçżĘçŽĎäżŇçż■ãĂĆæŕŤåęĆïijŽ

```
from collections import defaultdict
rows_by_date = defaultdict(list)
for row in rows:
    rows_by_date[row['date']].append(row)
```

è£ŹæăůçŽĎèŕÍä¡ăåŔŕäžěå¿Ĺè¡żæÍ¿çŽĎåŕśèČ¡åŕźæŕŔäÿłæŇĞåőŽæŮěæIJ§èő£éŮőåŕźåžŤçŽĎèőŕå¡Ţïij Ž

```
>>> for r in rows_by_date['07/01/2012']:
...     print(r)
...
```

```
{'date': '07/01/2012', 'address': '5412 N CLARK'}
{'date': '07/01/2012', 'address': '4801 N BROADWAY'}
>>>
```

åIJÍäÿŁéİćè£ŹäÿŁä¿Ńå■Řäÿ■ïijŇæĹŚäżňæšąæIJL'å£ĚèęĄåĚĹårĘèőřå¡ŢæŐŠåžŔãĂĆåŻăæ■d'ïijŇåęĆæđ
è£ŹçğæŰźåijŔäijŽæŕŤåĚĹæŐŠåžŘçĎűåŘŐåĘéĂŽè£Ğ `groupby()`
åĞ¡æŢřè£äżčçŽĎæŰźåijŔè£ŘèąŇå¿Ůå£ńäÿĂäžŽãĂĆ

3.16 1.16 è£Ğæžď'åžŔåĹŮåĚČçť'ă

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłæŢřæ■őåžŔåĹŮïijŇæČşåĹľçŤĺäÿĂäžŻèğĎåĹŹäżŐäÿ■æŔŘåŔŰåĞžéIJĂèęĄçŽĎåĂijæĺ

èğčåĘşæŰźæąĹ

æIJĂçőĂå■ŢçŽĎè£ĞæžďåžŔåĹŮåĚČçť'ăçŽĎæŰźæşŢářśæŸŕă¡£çŤĺåĹŮèąĺæŐĺåŕijãĂĆæŕŤåęĆïijŽ

```
>>> mylist = [1, 4, -5, 10, -7, 2, 3, -1]
>>> [n for n in mylist if n > 0]
[1, 4, 10, 2, 3]
>>> [n for n in mylist if n < 0]
[-5, -7, -1]
>>>
```

ä¡£çŤĺåĹŮèąĺæŐĺåŕijçŽĎäÿĂäÿłæ¡IJåIJÍçijžéŹůåřśæŸŕåęĆæđIJè¿ŞåĚěéİđåÿÿåď'gçŽĎæŮűåĂŹäijŽäžğç
åęĆæđIJä¡ääŕźåĘĚå■ŸæŕŤè¿ČæŢŔæĎ§ïijŇéĆčäźĹä¡ääŔŕäžěä¡£çŤĺçŤ§æĹŘåŹĺèąĺè¿¿åijŔè£■äžčäžğçŤ§è£Ğ

```
>>> pos = (n for n in mylist if n > 0)
>>> pos
<generator object <genexpr> at 0x1006a0eb0>
>>> for x in pos:
... print(x)
...
1
4
10
2
3
>>>
```

æIJL'æŮűåĂŹïijŇè£Ğæžď'èğĎåĹŹæŕŤè¿Čåď'■æİĆïijŇäÿ■èČ¡çőĂå■ŢçŽĎåIJĺåĹŮèąĺæŐĺåŕijæĹŰèĂĚç
æŕŤåęĆïijŇåĄĞèő¿è£Ğæžď'çŽĎæŰűåĂŹéIJĂèęĄåď'ĎçŘĘäÿĂäžŻåijĆåÿÿæĹŰèĂĚåĚűäżŰåď'■æİĆæČĚåĘ
çĎűåŘŐä¡£çŤĺåĘĚåžžçŽĎ `filter()` åĞ¡æŢřãĂĆçď'žä¿ŇåęĆäÿŇïijŽ

```
values = ['1', '2', '-3', '-', '4', 'N/A', '5']
def is_int(val):
```

```
    try:
        x = int(val)
        return True
    except ValueError:
        return False
ivals = list(filter(is_int, values))
print(ivals)
# Outputs ['1', '2', '-3', '4', '5']
```

filter() åĞ¡æŢřåĹŻåžžäžĘäÿĂäÿłè£■äžčåŹĺïijŇåŻäæ■ď'åęĆæđIJä¡ăæČşå¿ŮåĹřäÿĂäÿłåĹŮèąĺçŽĎè
list() åŐżè¡ňæ■ćãĂĆ

èőĺèőž

åĹŮèąĺæŐĺåŕijåŠŇçŤ§æĹŘåŹĺèąĺè¿¿åijŔéĂŽåÿÿæČĚåĘţäÿŃæŸŕè£ĞæżďæŢřæ■őæIJĂçőĂå■ŢçŽĎæŰ
åĚűåőđåőČäžňè£ŸèČ¡åIJĺè£ĞæżďçŽĎæŰűåĂŹè¡ňæ■ćæŢřæ■őãĂĆæŕŤåęĆïijŽ

```
>>> mylist = [1, 4, -5, 10, -7, 2, 3, -1]
>>> import math
>>> [math.sqrt(n) for n in mylist if n > 0]
[1.0, 2.0, 3.1622776601683795, 1.4142135623730951, 1.
↪7320508075688772]
>>>
```

è£ĞæżďæŞ■ä¡IJçŽĎäÿĂäÿłåŔŸçğ■åřśæŸŕåřĘäÿ■çňęåŘĹæİąäžűçŽĎåĂijçŤĺæŰřçŽĎåĂijäžčæŻ£ïijŇèĂ
æŕŤåęĆïijŇåIJĺäÿĂåĹŮæŢřæ■őäÿ■ä¡ăåŔŕèČ¡äÿ■äžĚæČşæL'¿åĹřæ■čæŢřïijŇèĂŇäÿŤè£ŸæČşåřĘäÿ■æŸŕæ■
éĂŽè£ĞåřĘè£ĞæżďæİąäžűæŤ¿åĹřæİąäžűèąĺè¿¿åijŔäÿ■åŐżïijŇåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎèğčåĘşè£ŹäÿłéŮőéćŸ

```
>>> clip_neg = [n if n > 0 else 0 for n in mylist]
>>> clip_neg
[1, 4, 0, 10, 0, 2, 3, 0]
>>> clip_pos = [n if n < 0 else 0 for n in mylist]
>>> clip_pos
[0, 0, -5, 0, -7, 0, 0, -1]
>>>
```

åŔęåď'ŰäÿĂäÿłåĂijå¿ŮåĚşæşĺçŽĎè£Ğæżď'åůěåĚůåřśæŸŕ itertools.
compress() ïijŇ åőČäžěäÿĂäÿł iterable årźèśąåŠŇäÿĂäÿłçŻÿårźåžŤçŽĎ
Boolean éĂL'æŃľåŹĺåžŔåĹŮä¡IJäÿžè¿ŞåĚěåŔĆæŢřãĂĆ çĎűåŘŐè¿ŞåĞž
iterable årźèśąäÿ■årźåžŤéĂL'æŃľåŹĺäÿž True çŽĎåĚĆçť'ăãĂĆ
å¡Şä¡ăéIJĂèęĄçŤĺåŔęåď'ŰäÿĂäÿłçŻÿåĚşèĄŤçŽĎåžŔåĹŮæİèè£Ğæżď'æ§Ŕäÿłåž
æŕŤåęĆïijŇåĄĞåęĆçŐřåIJĺä¡ăæIJL'äÿŃéİćäÿď'åĹŮæŢřæ■őïijŽ

```
addresses = [
    '5412 N CLARK',
    '5148 N CLARK',
    '5800 E 58TH',
    '2122 N CLARK',
    '5645 N RAVENSWOOD',
```

```
    '1060 W ADDISON',
    '4801 N BROADWAY',
    '1039 W GRANVILLE',
]
counts = [ 0, 3, 10, 4, 1, 7, 6, 1]
```

çŐřåIJÍä¡ăæČşåřĘéĆčäžŻåŕźåžŤ `count` åĂijåď ğäžŐ5çŽĎåIJřåİĂåĚÍéČÍè¿ŞåĞžïijŇéĆčäźĹä¡ăåŔŕäżèè£

```
>>> from itertools import compress
>>> more5 = [n > 5 for n in counts]
>>> more5
[False, False, True, False, False, True, True, False]
>>> list(compress(addresses, more5))
['5800 E 58TH', '1060 W ADDISON', '4801 N BROADWAY']
>>>
```

è£ŹéĞŇçŽĎåĚşéŤőçĆźåIJÍäžŐåĚĹåĹŻåżžäÿĂäÿł Boolean åžŔåĹŮïijŇæŇĞçď žåŞłäžŻåĚČçť ăçňęåŘĹæİąäżűãĂĆ çĎűåŘŐ `compress()` åĞ¡æŢřæăźæ■ős£ŹäÿłåžŔåĹŮåŐžéĂĽ'æŃľè¿ŞåĞžåŕźåžŤä¡■ç¡őäÿž `True` çŽĎåĚČçť ăăĂĆ

åŠŇ `filter()` åĞ¡æŢřçśżäijijïijŇ `compress()` äźş§æŸŕrè£ŤåŻđçŽĎäÿĂäÿłè£■äżčåŹÍãĂĆåŻăæ■d'ïij éĆčäźĹä¡ăéIJĂèęĄä¡£çŤÍ `list()` æİěåřĘçżŞæđIJè¡ñæ■ćäÿžåĹŮèąÍçśżåđŃãĂĆ

3.17 1.17 äżŐå■ŮåĚÿäÿ■æŔŘåŔŰå■ŘéŻĘ

éŮőécŸ

ä¡ăæČşæđĎéĂăäÿĂäÿłå■ŮåĚÿïijŇåőČæŸŕåŘęåď ŰäÿĂäÿłå■ŮåĚÿçŽĎå■ŘéŻĘãĂĆ

èğčåĘşæŰźæąĹ

æIJĂçőĂå■ŢçŽĎæŰźåijŔæŸŕä¡£çŤĺå■ŮåĚÿæŐĺåŕijãĂĆærŤåęĆïijŽ

```
prices = {
    'ACME': 45.23,
    'AAPL': 612.78,
    'IBM': 205.55,
    'HPQ': 37.20,
    'FB': 10.75
}
# Make a dictionary of all prices over 200
p1 = {key: value for key, value in prices.items() if value > 200}
# Make a dictionary of tech stocks
tech_names = {'AAPL', 'IBM', 'HPQ', 'MSFT'}
p2 = {key: value for key, value in prices.items() if key in tech_
↪names}
```

èőĺèőž

åď ğåď ŽæŢřæČĚåĘţäÿŃå■ŮåĚÿæŐĺåŕijèČ¡åĄŽåĹřçŽĎïijŇéĂŽè£ĞåĹŻåżžäÿĂäÿłåĚČçżĎåžŔåĹŮçĎű dict() åĞ¡æŢřäź§èČ¡åőđçŐřãĂĆæŕŤåęĆïijŽ

```
p1 = dict((key, value) for key, value in prices.items() if value >␣
↪200)
```

ä¡ĘæŸŕïijŇå■ŮåĚÿæŐĺåŕijæŰźåijŔèąĺæĎŔæŻť æÿĚæŹřïijŇåźűäÿŤåőđéŹĚäÿŁäź§äijŽè£ŘèąŇçŽĎæŻť ïijĹåIJĺè£Źäÿłä¿Ńå■Řäÿ■ïijŇåőđéŹĚæţŃèŕŢåĞääźŐæŕŤ dcit() åĞ¡æŢřæŰźåijŔå£ńæŢť æŢť äÿĂåĂ■ïijĽãĂĆ

æIJĽæŮűåĂŹåőŇæĹŘåŘŇäÿĂäżűäžŃäijŽæIJĽåď Žçğ■æŰźåijŔãĂĆæŕŤåęĆïijŇçňňäžŇäÿłä¿Ńå■ŘçĺŃ

```
# Make a dictionary of tech stocks
tech_names = { 'AAPL', 'IBM', 'HPQ', 'MSFT' }
p2 = { key:prices[key] for key in prices.keys() & tech_names }
```

ä¡ĘæŸŕïijŇè£ŘèąŇæŮűéŮť æţŃèŕŢçżŞæđIJæŸ¿çď žè£Źçğ■æŰźæąĹåď ğæęĆæŕŤçňňäÿĂçğ■æŰźæąĹæ 1.6 åĂ■ãĂĆ åęĆæđIJåŕźçĺŃåžŔè£ŘèąŇæĂğèČ¡èęĄæśĆæŕŤè¿ČéńŸçŽĎèŕİïijŇéIJĂèęĄèŁśçĆźæŮűéŮť åŐżå åĚşäžŐæŻť åď ŽèőąæŮűåŠŇæĂğèČ¡æţŃèŕŢïijŇåŔŕäżěåŔĆèĂČ 14.13 åřŔèŁĆãĂĆ

3.18 1.18 æŸäåřĎåŘ■çğřåĹřåžŔåĹŮåĚČçť ă

éŮőéćŸ

ä¡ăæIJĽäÿĂæőţéĂŽè£ĞäÿŃæăĞèő£éŮőåĹŮèąĺæĹŰèĂĚåĚČçżĎäÿ■åĚČçť ăçŽĎäżčçăĄïijŇä¡ĘæŸŕè£Ź äžŐæŸŕä¡ăæČşéĂŽè£ĞåŘ■çğřæİěèő£éŮőåĚČçť ăãĂĆ

èğčåĘşæŰźæąĹ

collections.namedtuple() åĞ¡æŢřéĂŽè£Ğä¡£çŤĺäÿĂäÿłæŹőéĂŽçŽĎåĚČçżĎåŕźèśąæİěåÿőä¡ă è£ŹäÿłåĞ¡æŢřåőđéŹĚäÿŁæŸŕäÿĂäÿłè£ŤåŻđ Python äÿ■æăĞåĞĘåĚČçżĎçśżåđŃå■ŘçśżçŽĎäÿĂäÿłåůěåŐĆ ä¡ăéIJĂèęĄäijăéĂŠäÿĂäÿłçśżåđŃåŘ■åŠŇä¡ăéIJĂèęĄçŽĎå■ŮæőţçżŹåőČïijŇçĎűåŘŐåőČåŕśäijŽè£ŤåŻđäÿĂ äżčçăĄçď žä¿ŃïijŽ

```
>>> from collections import namedtuple
>>> Subscriber = namedtuple('Subscriber', ['addr', 'joined'])
>>> sub = Subscriber('jonesy@example.com', '2012-10-19')
>>> sub
Subscriber(addr='jonesy@example.com', joined='2012-10-19')
>>> sub.addr
'jonesy@example.com'
>>> sub.joined
'2012-10-19'
>>>
```

åř¡çőą `namedtuple` çŽĎåőđä¿ŃçIJŃèţůæİěåČŔäÿĂäÿłæŹőéĂŽçŽĎçśžåőđä¿ŃïijŇä¡ĘæŸŕåőČèů§åĚČ
ærŤåęĆïijŽ

```
>>> len(sub)
2
>>> addr, joined = sub
>>> addr
'jonesy@example.com'
>>> joined
'2012-10-19'
>>>
```

åŠ¡åŘ■åĚČçżĎçŽĎäÿĂäÿłäÿżèęĄçŤĺéĂŤæŸŕåřĘä¡ăçŽĎäżčçăĄäžŐäÿŃæăĞæŞ■ä¡IJäÿ■èğčèĎśåĞžæİě
åŻăæ■ď¨ïijŇåęĆædIJä¡ăäžŐæŢřæ■őåžŞèřČçŤĺäÿ■è£ŤåŻđäžĘäÿĂäÿłå¿Ĺåď¨ğçŽĎåĚČçżĎåĹÙèąĺïijŇéĂŽè£Ğ
å¡Şä¡ăåIJĺèąĺäÿ■æůżåŁääžĘæŰřçŽĎåĹÙçŽĎæÙűåĂŹä¡ăçŽĎäżčçăĄåŘŕèČ¡åŕšäijŽåĞžéŤŹäžĘãĆä¡ĘæŸŕå

äÿžäžĘèŕť¨æŸŐæÿĚæěŽïijŇäÿŃéİćæŸŕä¡£çŤĺæŹőéĂŽåĚČçżĎçŽĎäżčçăĄïijŽ

```
def compute_cost(records):
    total = 0.0
    for rec in records:
        total += rec[1] * rec[2]
    return total
```

äÿŃæăĞæŞ■ä¡IJéĂŽåÿÿäijŽèől¨äžčçăĄèąĺæĎŔäÿ■æÿĚæŹřijŇåźűäÿŤéİđåÿÿä¿ÌètŰèőřå¡ŢçŽĎçżŞæđĎ
äÿŃéİćæŸŕä¡£çŤĺåŠ¡åŘ■åĚČçżĎçŽĎçL¨ĹæIJñïijŽ

```
from collections import namedtuple

Stock = namedtuple('Stock', ['name', 'shares', 'price'])
def compute_cost(records):
    total = 0.0
    for rec in records:
        s = Stock(*rec)
        total += s.shares * s.price
    return total
```

èőÍèőž

åŠ¡åŘ■åĚČçżĎåŔęäÿĂäÿłçŤĺéĂŤåřśæŸŕä¡IJäÿžå■ŮåĚÿçŽĎæŻ£äžčijŇåŻăäÿžå■ŮåĚÿå■ŸåĆĺéIJĂèęĄ
åęĆædIJä¡ăéIJĂèęĄæđĎåżžäÿĂäÿłéİđåÿÿåď¨ğçŽĎåŇĚåŘńå■ŮåĚÿçŽĎæŢřæ■őçżŞæđĎïijŇéĆčäźĹä¡£çŤĺåŠ¡
ä¡ĘæŸŕéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇäÿ■åČŔå■ŮåĚÿéĆčæăůïijŇäÿĂäÿłåŠ¡åŘ■åĚČçżĎæŸŕäÿ■åŘŕæŻť¨æŤźç

```
>>> s = Stock('ACME', 100, 123.45)
>>> s
Stock(name='ACME', shares=100, price=123.45)
>>> s.shares = 75
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
```

```
AttributeError: can't set attribute
>>>
```

åęĆæđIJä¡ăçIJ§çŽĎéIJĂèęĄæŤźåŔŸåśđæĂğçŽĎåĂijïijŇéĆčäźĹåŔŕäżěä¡£çŤĺåŚ¡åŘ■åĚČçżĎåőđä¿ŃçŽ
_replace() æŰźæşŢïijŇ åőČäijŽåĹŻåżžäÿĂäÿłåĚĺæŰřçŽĎåŚ¡åŘ■åĚČçżĎåźűåřĘåŕźåžŤçŽĎå■ŮæőţçŤĺæ

```
>>> s = s._replace(shares=75)
>>> s
Stock(name='ACME', shares=75, price=123.45)
>>>
```

_replace() æŰźæşŢè£ŸæIJL'äÿĂäÿłå¿ĹæIJL'çŤĺçŽĎçL'źæĂğåřśæŸŕå¡Şä¡ăçŽĎåŚ¡åŘ■åĚČçżĎæŃě
åőČæŸŕäÿĂäÿłéİđåÿÿæŰźä¿£çŽĎåąńåĚĚæŢřæ■őçŽĎæŰźæşŢãĂĆ
ä¡ăåŔŕäżěåĚĹåĹŻåżžäÿĂäÿłåŇĚåŘńçijžçIJĄåĂijçŽĎåŐ§åđŃåĚČçżĎïijŇçĎűåŘŐä¡£çŤĺ
_replace() æŰźæşŢåĹŻåżžæŰřçŽĎåĂijèćńæŻť'æŰřè£ĞçŽĎåőđä¿ŃãĂĆæŕŤåęĆïijŽ

```
from collections import namedtuple

Stock = namedtuple('Stock', ['name', 'shares', 'price', 'date',
↪'time'])

# Create a prototype instance
stock_prototype = Stock('', 0, 0.0, None, None)

# Function to convert a dictionary to a Stock
def dict_to_stock(s):
    return stock_prototype._replace(**s)
```

äÿŃéİćæŸŕåőČçŽĎä¡£çŤĺæŰźæşŢïijŽ

```
>>> a = {'name': 'ACME', 'shares': 100, 'price': 123.45}
>>> dict_to_stock(a)
Stock(name='ACME', shares=100, price=123.45, date=None, time=None)
>>> b = {'name': 'ACME', 'shares': 100, 'price': 123.45, 'date':
↪'12/17/2012'}
>>> dict_to_stock(b)
Stock(name='ACME', shares=100, price=123.45, date='12/17/2012',␣
↪time=None)
>>>
```

æIJĂåŘŐèęĄèŕť'çŽĎæŸŕïijŇåęĆæđIJä¡ăçŽĎçŻőæăĞæŸŕåőŽäźL'äÿĂäÿłéIJĂèęĄæŻť'æŰřå¿Ĺåď'Žåőđä¿
è£ŹæŮűåĂŹä¡ăåžŤèŕěèĂČèŹŚåőŽäźL'äÿĂäÿłåŇĚåŘń __slots__
æŰźæşŢçŽĎçśżïijĹåŔĆèĂČ8.4åřŔèŁĆïijL'ãĂĆ

3.19 1.19 è¡ňæ■ćåźűåŘŇæŮűèőąçőŮæŢřæ■ő

éŮőécŸ

ä¡ăéIJĂèęĄåIJĺæŢřæ■őåžŔåĹŮäÿŁæL'ğèąŇèĄŽéŻĘåĞ¡æŢřïijĹæŕŤåęĆ `sum()`,`min()`,`max()` ïijL'ïijŇ ä¡ĘæŸŕéęŰåĚĹä¡ăéIJĂèęĄåĚĹè¡ňæ■ćæĹŰèĂĚè£Ğæżđ'æŢřæ■ő

èğčåĘşæŰźæąĹ

äÿĂäÿłéİđåÿÿäijŸéŻĚçŽĎæŰźåijŔåŐżçżŞåŘĹæŢřæ■ôèőąçőŮäÿŐè¡ňæ■ćåřśæŸŕä¡£çŤĺäÿĂäÿłçŤ§æĹŘæŕŤåęĆïijŇåęĆæđIJä¡ăæČşèőąçőŮåźşæŰźåŠŇïijŇåŘŕäžěåČŔäÿŃéİćè£ŹæăůåĄŽïijŽ

```
nums = [1, 2, 3, 4, 5]
s = sum(x * x for x in nums)
```

äÿŃéİćæŸŕæŻt'åđ'ŽçŽĎä¿Ńå■ŘïijŽ

```
# Determine if any .py files exist in a directory
import os
files = os.listdir('dirname')
if any(name.endswith('.py') for name in files):
    print('There be python!')
else:
    print('Sorry, no python.')
# Output a tuple as CSV
s = ('ACME', 50, 123.45)
print(','.join(str(x) for x in s))
# Data reduction across fields of a data structure
portfolio = [
    {'name':'GOOG', 'shares': 50},
    {'name':'YHOO', 'shares': 75},
    {'name':'AOL', 'shares': 20},
    {'name':'SCOX', 'shares': 65}
]
min_shares = min(s['shares'] for s in portfolio)
```

èőÍèőž

äÿŁéİćçŽĎçđ'žä¿ŃåŘŚä¡ăæijŤçđ'žäžĘå¡ŞçŤ§æĹŘåŹÍèąÍè¿¿åijŔä¡IJäÿžäÿĂäÿłå■ŢçŃňåŔĆæŢřäijăéĂŠçæŕŤåęĆïijŇäÿŃéİćè£ŹäžŻèŕ■åŘěæŸŕç■L'æŢĹçŽĎïijŽ

```
s = sum((x * x for x in nums)) #␣
↪æŸ¿çd'žçŽĎäijăéĂŠäÿĂäÿłçŤ§æĹŘåŹÍèąÍè¿¿åijŔårźèśą
s = sum(x * x for x in nums) #␣
↪æŻt'åŁääijŸéŻĚçŽĎåőđçŐřæŰźåijŔïijŇçIJĄçŢěäžĘæŃňåŔů
```

ä¡£çŤĺäÿĂäÿłçŤ§æĹŘåŹÍèąÍè¿¿åijŔä¡IJäÿžåŔĆæŢřäijŽæŕŤåĚĹåĹŻåżžäÿĂäÿłäÿť'æŮűåĹŮèąÍæŻť'åŁăéŕæŕŤåęĆïijŇåęĆæđIJä¡ăäÿ■ä¡£çŤĺçŤ§æĹŘåŹÍèąÍè¿¿åijŔçŽĎèŕÍïijŇä¡ăåŘŕèČ¡äijŽèĂČèŹŚä¡£çŤĺäÿŃéİćçŽĎå

```
nums = [1, 2, 3, 4, 5]
s = sum([x * x for x in nums])
```

è£Źçğ■æŰźåijŔåŘŇæăůåŘŕäžěè¿¿åĹřæČşèęĄçŽĎæŢĹæđIJïijŇä¡ĘæŸŕåőČäijŽåď'ŽäÿĂäÿłæ■ěéłď'ïijŇå
åŕźäžŐåřŘåđŃåĹŮèąĺåŘŕèČ¡æšąäżĂäźĹåĚşçşżïijŇä¡ĘæŸŕåęĆæđIJåĚČçť'ăæŢřéĞŔéİđåÿÿåď'ğçŽĎæŮűåĂŹi
åőČäijŽåĹŻåžžäÿĂäÿłåůĺåď'ğçŽĎäżĚäżĚèćńä¡£çŤĺäÿĂæňąåřśèćńäÿćåijČçŽĎäÿť'æŮűæŢřæ■őçżŞæđĎãĂĆè

åIJĺä¡£çŤĺäÿĂäžŽèĄŽéŻĘåĞ¡æŢřæŕŤåęĆ `min()` åŠŇ `max()` çŽĎæŮűåĂŹä¡ăåŔŕèČ¡æŽť'åŁăåĂ¿åŘŚäžŐä¡£çŤĺçŤ§æĹŘåŹĺçL'ĹæIJňïijŇ
åőČäžňæŐěåŔŮçŽĎäÿĂäÿł key åĚşéŤőå■ŮåŔĆæŢřæĹŰèőÿåŕźä¡ăå¿ĹæIJL'åÿőåŁl'ãĂĆ
æŕŤåęĆïijŇåIJĺäÿŁéİćçŽĎèŕĄåĹÿä¿Ńå■Řäÿ■ïijŇä¡ăåŔŕèČ¡äijŽèĂČèŹŠäÿŃéİćçŽĎåőđçŐřçL'ĹæIJňïijŽ

```
# Original: Returns 20
min_shares = min(s['shares'] for s in portfolio)
# Alternative: Returns {'name': 'AOL', 'shares': 20}
min_shares = min(portfolio, key=lambda s: s['shares'])
```

3.20 1.20 åŘĹåźűåď'Žäÿłå■ŮåĚÿæĹŰæŸăåřĎ

éŮőécŸ

çŐřåIJĺæIJL'åď'Žäÿłå■ŮåĚÿæĹŰèĂĚæŸăåřĎïijŇä¡ăæČşåřĘåőČäžňäžŐéĂżè¿ŚäÿŁåŘĹåźűäÿžäÿĂäÿłå■
æŕŤåęĆæ§ěæL'¿åĂijæĹŰèĂĚæčĂæ§ěæ§ŘäžŹéŤőæŸŕåŘęå■ŸåIJĺãĂĆ

èğčåĘşæŰźæąĹ

åĄĞåęĆä¡ăæIJL'åęĆäÿŃäÿď'äÿłå■ŮåĚÿ:

```
a = {'x': 1, 'z': 3 }
b = {'y': 2, 'z': 4 }
```

çŐřåIJĺåĄĞèő¿ä¡ăå£Ěéążå IJĺäÿď'äÿłå■ŮåĚÿäÿ■æL'ğèąŇæ§ěæL'¿æŞ■ä¡IJïijĹæŕŤåęĆåĚĹäžŐ
`a` äÿ■æL'¿ïijŇåęĆæđIJæL'¿äÿ■åĹřåĘ■åIJĺ `b` äÿ■æL'¿ïijL'ãĂĆ
äÿĂäÿłéİđåÿÿçőĂå■ŢçŽĎèğčåĘşæŰźæąĹåřśæŸŕä¡£çŤĺ `collections` æĺąåİŮäÿ■çŽĎ
`ChainMap` çśžãĂĆæŕŤåęĆïijŽ

```
from collections import ChainMap
c = ChainMap(a,b)
print(c['x']) # Outputs 1 (from a)
print(c['y']) # Outputs 2 (from b)
print(c['z']) # Outputs 3 (from a)
```

èőĺèőž

äÿĂäÿł `ChainMap` æŐěåŘŮåď'Žäÿłå■ŮåĚÿåźűåřĘåőČäžňåIJĺéĂżè¿ŚäÿŁåŘŸäÿžäÿĂäÿłå■ŮåĚÿãĂĆ
çĎűåŘŐïijŇè£ŹäžŻå■ŮåĚÿåźűäÿ■æŸŕçIJ§çŽĎåŘĹåźűåIJĺäÿĂèţůäžĘïijŇ `ChainMap`

çśżåŔłæŸŕåIJĺåĘĚéČĺåĹŻåżžäžĘäÿĂäÿłåőźçžşè£ŹäžŻå■ŮåĚÿçŽĎåĹŮèąĺ
åźűéĞ■æŰřåőŽäźL’äžĘäÿĂäžŻåÿÿèğAçŽĎå■ŮåĚÿæŞ■ä¡IJæİěéĄ■åŐĘè£ŹäÿłåĹŮèąĺãĂĆåď’ğéČĺåĹĘå■ŮåĚ

```
>>> len(c)
3
>>> list(c.keys())
['x', 'y', 'z']
>>> list(c.values())
[1, 2, 3]
>>>
```

åęĆædIJåĞžçŐřéĞ■åď’■éŤőïijŇéĆčäźĹçňňäÿĂæňąåĞžçŐřçŽĎæŸäářĎåĂijäijŽèćńè£ŤåŻđãĂĆ
åŻăæ■d’ïijŇä¿Ńå■ŘçĺŃåžŔäÿ■çŽĎ `c['z']` æĂżæŸŕäijŽè£ŤåŻđå■ŮåĚÿ `a`
äÿ■ářźåžŤçŽĎåĂijïijŇèĂŇäÿ■æŸŕ `b` äÿ■ářźåžŤçŽĎåĂijãĂĆ

ářźäžŐå■ŮåĚÿçŽĎæŻt’æŰřæĹŰåĹăéŹd’æŞ■ä¡IJæĂżæŸŕå¡śåŞ■çŽĎæŸŕåĹŮèąĺäÿ■çňňäÿĂäÿłå■ŮåĚÿă

```
>>> c['z'] = 10
>>> c['w'] = 40
>>> del c['x']
>>> a
{'w': 40, 'z': 10}
>>> del c['y']
Traceback (most recent call last):
...
KeyError: "Key not found in the first mapping: 'y'"
>>>
```

`ChainMap` åŕźäžŐçijŰçÍŃèŕ■èĺĂäÿ■çŽĎä¡IJçŤĺèŇČåŻt’åŔŸéĞŔïijĹærŤåęĆ
`globals` , `locals` ç■L’ïijL’æŸŕéİđåÿÿæIJL’çŤĺçŽĎãĂĆ
äžŃåőđäÿŁïijŇæIJL’äÿĂäžŻæŰźæşŢåŔŕäžëä¡£åőČåŔŸå¿ŮçőĂå■Ţïij Ž

```
>>> values = ChainMap()
>>> values['x'] = 1
>>> # Add a new mapping
>>> values = values.new_child()
>>> values['x'] = 2
>>> # Add a new mapping
>>> values = values.new_child()
>>> values['x'] = 3
>>> values
ChainMap({'x': 3}, {'x': 2}, {'x': 1})
>>> values['x']
3
>>> # Discard last mapping
>>> values = values.parents
>>> values['x']
2
>>> # Discard last mapping
>>> values = values.parents
>>> values['x']
```

```
1
>>> values
ChainMap({'x': 1})
>>>
```

ä¡IJäÿž `ChainMap` çŽĎæŻ£äżčïijŇä¡ăåŔŕèČ¡äijŽèĂČèŹŚä¡£çŤÍ `update()` æŰźæşŢåřĘäÿd'äÿłå■ŮåĚÿåŘĹåźűãĂĆæŕŤåęĆïijŽ

```
>>> a = {'x': 1, 'z': 3 }
>>> b = {'y': 2, 'z': 4 }
>>> merged = dict(b)
>>> merged.update(a)
>>> merged['x']
1
>>> merged['y']
2
>>> merged['z']
3
>>>
```

è£ŹæăůäźšèČ¡èąŇå¿ŮéĂŽïijŇä¡ĘæŸŕåőČéIJĂèęĄä¡ăåĹŻåżžäÿĂäÿłåőŇåĚíäÿ■åŘŇçŽĎå■ŮåĚÿåŕźèśąï
åŘŇæŮűïijŇåęĆæđIJåŐ§å■ŮåĚÿåĄŽäžĘæŻťæŰřïijŇè£Źçğ■æŤźåŔŸäÿ■äijŽåŔ■åžŤåĹřæŰřçŽĎåŘĹåźűå■Ů

```
>>> a['x'] = 13
>>> merged['x']
1
```

`ChainMap` ä¡£çŤĺåŐ§æİěçŽĎå■ŮåĚÿïijŇåőČèĞłåůśäÿ■åĹŻåżžæŰřçŽĎå■ŮåĚÿãĂĆæĽĂäžěåőČåźűäÿ

```
>>> a = {'x': 1, 'z': 3 }
>>> b = {'y': 2, 'z': 4 }
>>> merged = ChainMap(a, b)
>>> merged['x']
1
>>> a['x'] = 42
>>> merged['x'] # Notice change to merged dicts
42
>>>
```

4 çňňäžŇçńäïijŽå■ŮçňęäÿšåŠŇæŰĞælJň

åĞääźŐæĽĂæIJĽæIJĽçŤÍçŽĎçÍŃåžŔéČ¡äijŽæűĽåŔŁåĹřæ§ŘäžŽæŰĞæIJňåď'ĎçŘĘïijŇäÿ■çőąæŸŕèğ
è£ŹäÿĂçńäářĘéĞ■çĆźåĚşæşĺæŰĞæIJňçŽĎæŞ■ä¡IJåď'ĎçŘĘïijŇæŕŤåęĆæŔŘåŔŰå■ŮçňęäÿšïijŇæŘIJçť'ćïijŇ
åď'ğéČĺåĹęçŽĎéŮőécŸéČ¡èČ¡çőĂå■ŢçŽĎèřČçŤĺå■ŮçňęäÿšçŽĎåĘĚåžžæŰźæşŢåőŇæĹŘãĂĆ
ä¡ĘæŸŕïijŇäÿĂäžŽæŻť'äÿžåď'■æİĆçŽĎæŞ■ä¡IJåŔŕèČ¡éIJĂèęĄæ■čåĹŹèąĺè¿¿åijŔæĹŰèĂĚåijžåď'ğçŽĎèğčæ
åźűäÿŤåIJĺæŞ■ä¡IJUnicodeæŮűåĂŹçćřåĹřçŽĎäÿĂäžŽæčŸæĽŃçŽĎéŮőécŸåIJĺè£ŹéĞŇäź§äijŽèćńæŔŘåŔ

Contents:

4.1 2.1 ä¡£çŤĺåď Žäÿłç ŢŇåőŽçňęåĹĘåĽšå■Ůçňęäÿš

éŮőéćŸ

ä¡ăéIJĂèęĄåřĘäÿĂäÿłå■ŮçňęäÿšåĹĘåĽšäÿžåď Žäÿłå■ŮæőţïijŇä¡ĘæŸŕåĹĘéŽŤçňę(è£ŸæIJĽåŚĺåŻť çŽĎ

èğčåĘşæŰźæąĹ

string åŕźèśąçŽĎ split() æŰźæşŢåŔłéĂĆåžŤäžŐéİđåÿÿçőĂå■ŢçŽĎå■ŮçňęäÿšåĹĘåĽšæČĚå¡ćïij
åőČåźűäÿ■åĚĄèőÿæIJĽåď ŽäÿłåĹĘéŽŤçňęæĹŰèĂĚæŸŕåĹĘéŽŤçňęåŚĺåŻť äÿ■çąőåőŽçŽĎçľžæăijãĂĆ
å¡Şä¡ăéIJĂèęĄæŻť åŁăçĄţæť żçŽĎåĹĞåĽšå■ŮçňęäÿšçŽĎæŮűåĂŹïijŇæIJĂåě¡ä¡£çŤĺ re.
split() æŰźæşŢïijŽ

```
>>> line = 'asdf fjdk; afed, fjek,asdf, foo'
>>> import re
>>> re.split(r'[;,\s]\s*', line)
['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
```

èőĺèőž

åĞ¡æŢř re.split() æŸŕéİđåÿÿåőđçŤĺçŽĎïijŇåŻăäÿžåőČåĚĄèőÿä¡ăäÿžåĹĘéŽŤçňęæŇĞåőŽåď Žäÿł
æŕŤåęĆïijŇåIJĺäÿŁéİćçŽĎä¿Ńå■Řäÿ■ïijŇåĹĘéŽŤçňęåŔŕäżěæŸŕéĂŮåŔůïijŇåĹĘåŔůæĹŰèĂĚæŸŕçľžæăijïij
åŔłèęĄè£ŹäÿłæĺąåijŔèćńæĽ¿åĹřïijŇéĆčäźĹåŇźéĚ■çŽĎåĹĘéŽŤçňęäÿď è¿źçŽĎåőđä¡ŞéČ¡äijŽèćńå¡ŞæĹŘæ
è£ŤåŻđçżŞæđIJäÿžäÿĂäÿłå■ŮæőţåĹŮèąĺïijŇè£Źäÿłèů§ str.split()
è£ŤåŻđåĂijçśżåđŃæŸŕäÿĂæăůçŽĎãĂĆ

å¡Şä¡ăä¡£çŤĺ re.split() åĞ¡æŢřæŮűåĂŹïijŇéIJĂèęĄçĽźåĹńæşĺæĎŔçŽĎæŸŕæ■čåĹŹèąĺè¿¿åijŔäÿ
åęĆæđIJä¡£çŤĺäžĘæ■ŢèŐůåĹĘçżĎïijŇéĆčäźĹèćńåŇźéĚ■çŽĎæŰĞæIJňäź§åřĘåĞžçŐřåIJĺçżŞæđIJåĹŮèąĺäÿ

```
>>> fields = re.split(r'(;|,|\s)\s*', line)
>>> fields
['asdf', ' ', 'fjdk', ';', 'afed', ',', 'fjek', ',', 'asdf', ',',
↪'foo']
>>>
```

èŐůåŔŰåĹĘåĽšå■ŮçňęåIJĺæ§ŘäžŻæČĚåĘţäÿŃäź§æŸŕæIJĽçŤĺçŽĎãĂĆ
æŕŤåęĆïijŇä¡ăåŔŕèČ¡æČşä£İçŢŹåĹĘåĽšå■ŮçňęäÿšïijŇçŤĺæİěåIJĺåŘŐéİćéĞ■æŰřæđĎéĂăäÿĂäÿłæŰřçŽĎè

```
>>> values = fields[::2]
>>> delimiters = fields[1::2] + ['']
>>> values
['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
>>> delimiters
[' ', ';', ',', ',', ',', '']
>>> # Reform the line using the same delimiters
>>> ''.join(v+d for v,d in zip(values, delimiters))
'asdf fjdk;afed,fjek,asdf,foo'
>>>
```

åęĆæđIJä¡ăäÿ■æČşä£İçŢŹåĹĘåĽšå■ŮçňęäÿšåĹřçżŞæđIJåĹŮèąĺäÿ■åŐżïijŇä¡Ęäż■çĎűéIJĂèęĄä¡£çŤĺåĹ
çąőä£İä¡ăçŽĎåĹĘçżĎæŸŕéİđæ■ŢèŐůåĹĘçżĎïijŇå¡ćåęĆ (?:...) ãĂĆæŕŤåęĆïijŽ

```
>>> re.split(r'(?:,|;|\s)\s*', line)
['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
>>>
```

4.2 2.2 å■ŮçňęäÿšåijĂåďťæĹŰçżŞåř¿åŇźéĚ■

éŮőéćŸ

ä¡ăéIJĂèęĄéĂŽè£ĞæŇĞåőŽçŽĎæŰĞæIJňæĺąåijŔåŐżæčĂæ§ěå■ŮçňęäÿšçŽĎåijĂåďťæĹŰèĂĚçżŞåř¿
Schemeç■Ľç■Ľ ãĂĆ

èğčåĘşæŰźæąĹ

æčĂæ§ěå■ŮçňęäÿšåijĂåďťæĹŰçżŞåř¿çŽĎäÿĂäÿłçőĂå■ŢæŰźæşŢæŸŕä¡£çŤĺ str.startswith() æĹŰèĂĚæŸŕ str.endswith() æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> filename = 'spam.txt'
>>> filename.endswith('.txt')
True
>>> filename.startswith('file:')
False
>>> url = 'http://www.python.org'
>>> url.startswith('http:')
True
>>>
```

åęĆæđIJä¡ăæČşæčĂæ§ěåďŽçğ■åŇźéĚ■åŔŕèČ¡ïijŇåŔłéIJĂèęĄåřĘæĽĂæIJĽçŽĎåŇźéĚ■éąźæŤ¿åĚěåĹ
çĎűåŘŐäijăçżŹ startswith() æĹŰèĂĚ endswith() æŰźæşŢïijŽ

```
>>> import os
>>> filenames = os.listdir('.')
>>> filenames
[ 'Makefile', 'foo.c', 'bar.py', 'spam.c', 'spam.h' ]
>>> [name for name in filenames if name.endswith(('.c', '.h')) ]
['foo.c', 'spam.c', 'spam.h'
>>> any(name.endswith('.py') for name in filenames)
True
>>>
```

äÿŃéİćæŸŕåŔęäÿĂäÿłä¿Ńå■ŘïijŽ

```
from urllib.request import urlopen
```

```
def read_data(name):
    if name.startswith(('http:', 'https:', 'ftp:')):
        return urlopen(name).read()
    else:
        with open(name) as f:
            return f.read()
```

åěĞæĂłçŽĎæŸŕïijŇè£ŹäÿłæŰźæşŢäÿ■å£Ěéążèę Ąè¿ŞåĚěäÿĂäÿłåĚČçżĎä¡IJäÿžåŔĆæŢřãĂĆ åęĆæđIJä¡ăæĄřåůğæIJL'äÿĂäÿł `list` æĹŰèĂĚ `set` çśżåđŃçŽĎéĂL'æŃl'éąźïijŇ èęĄçąőä£Ïäijăé ĂŠåŔĆæŢřåL'■åĚĹèřČçŤĺ `tuple()` åřĘåĚűè¡ñæ■ćäÿžåĚČçżĎçśżåđŃãĂĆæŕŤåęĆïijŽ

```
>>> choices = ['http:', 'ftp:']
>>> url = 'http://www.python.org'
>>> url.startswith(choices)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: startswith first arg must be str or a tuple of str, not␣
 ↪list
>>> url.startswith(tuple(choices))
True
>>>
```

èőÍèőž

`startswith()` åŠŇ `endswith()` æŰźæşŢæŔŘä¿ŻäžĘäÿĂäÿłéÍđåÿÿæŰźä¿£çŽĎæŰźåijŔåŐżåĄŽå çśżäijijçŽĎæŞ■ä¡IJäź§åŔŕäžěä¡£çŤĺåĹĞçL'ĞæÍěåőđçŐřïijŇä¡ĘæŸŕäžčçăĄçIJŃèţůæÍěæšąæIJL'éĆčäźĹäijŸé

```
>>> filename = 'spam.txt'
>>> filename[-4:] == '.txt'
True
>>> url = 'http://www.python.org'
>>> url[:5] == 'http:' or url[:6] == 'https:' or url[:4] == 'ftp:'
True
>>>
```

ä¡ăåŔŕäžěèČ¡è£ŸæČşä¡£çŤĺæ■čåĹŹèąĺè¿¿åijŔåŐżåőđçŐřïijŇæŕŤåęĆïijŽ

```
>>> import re
>>> url = 'http://www.python.org'
>>> re.match('http:|https:|ftp:', url)
<_sre.SRE_Match object at 0x101253098>
>>>
```

è£Źçğ■æŰźåijŔäź§èąŇå¿ŮéĂŽïijŇä¡ĘæŸŕåŕźäžŐçőĂå■ŢçŽĎåŇźéĚ■åőđåIJĺæŸŕæIJL'çĆźåřŔæİŘåđ'ğ

æIJĂåŘŐæŔŘäÿĂäÿŃïijŇå¡ŞåŠŇåĚűäżŰæŞ■ä¡IJæŕŤåęĆæŹőéĂŽæŢřæ■őèĄŽåŘĹçŻÿçżŞåŘĹçŽĎæŮú `startswith()` åŠŇ `endswith()` æŰźæşŢæŸŕå¿Ĺäÿ■éŤŹçŽĎãĂĆ æŕŤåęĆïijŇäÿŃéÍćè£Źäÿłèŕ■åŘèæčĂæ§ěæ§ŘäÿłæŰĞäžűåđ'źäÿ■æŸŕåŘęå■ŸåIJĺæŇĞåőŽçŽĎæŰĞäžűçśżåđ

```
if any(name.endswith(('.c', '.h')) for name in listdir(dirname)):
...
```

4.3 2.3 çŤÍShelléĂŽéĚ■çňęåŇźéĚ■å■Ůçňęäÿš

éŮőéćŸ

ä¡ăæČşä¡£çŤÍ **Unix Shell** äÿ■åÿÿçŤÍçŽĎéĂŽéĚ■çňę(æŕŤåęĆ `*.py`,`Dat[0-9]*.csv` ç■Ľ)åŐżåŇźéĚ■æŰĞæIJňå■Ůçňęäÿš

èğčåĘşæŰźæąĹ

`fnmatch` æĺąåİŮæŔŘä¿ŻäžĘäÿď äÿłåĞ¡æŢřâĂŤâĂŤ `fnmatch()` åŠŇ `fnmatchcase()` ïijŇåŔŕäżěçŤÍæİěåőđçŐřè£ŹæăůçŽĎåŇźéĚ■ãĂĆçŤÍæşŢåęĆäÿŃïijŽ

```
>>> from fnmatch import fnmatch, fnmatchcase
>>> fnmatch('foo.txt', '*.txt')
True
>>> fnmatch('foo.txt', '?oo.txt')
True
>>> fnmatch('Dat45.csv', 'Dat[0-9]*')
True
>>> names = ['Dat1.csv', 'Dat2.csv', 'config.ini', 'foo.py']
>>> [name for name in names if fnmatch(name, 'Dat*.csv')]
['Dat1.csv', 'Dat2.csv']
>>>
```

`fnmatch()` åĞ¡æŢřä¡£çŤÍåžŢåśĆæŞ■ä¡IJçşżçż§çŽĎåď ğåřŔåĘŹæŢŔæĎ§èğĎåĹŹ(äÿ■åŘŇçŽĎçşżçż§

```
>>> # On OS X (Mac)
>>> fnmatch('foo.txt', '*.TXT')
False
>>> # On Windows
>>> fnmatch('foo.txt', '*.TXT')
True
>>>
```

åęĆæđIJä¡ăåŕźè£ŹäÿłåŇžåĹńå¿ĹåIJĺæĎŔïijŇåŔŕäżěä¡£çŤÍ `fnmatchcase()` æİěäżčæŻ£ãĂĆåőČåőŇåĚĺä¡£çŤÍä¡ăçŽĎæĺąåijŔåď ğåřŔåĘŹåŇźéĚ■ãĂĆæŕŤåęĆïijŽ

```
>>> fnmatchcase('foo.txt', '*.TXT')
False
>>>
```

è£Źäÿď äÿłåĞ¡æŢřéĂŽåÿÿäijŽèćńå£¡çŢěçŽĎäÿĂäÿłçĽźæĂğæŸŕåIJĺåď ĎçŘĘéİđæŰĞäżűåŘ■çŽĎå■Ůç
æŕŤåęĆïijŇåĄĞèő¿ä¡ăæIJĽäÿĂäÿłèąŮéĄŞåIJřåİĂçŽĎåĹŮèąĺæŢřæ■őïijŽ

```
addresses = [
    '5412 N CLARK ST',
    '1060 W ADDISON ST',
    '1039 W GRANVILLE AVE',
    '2122 N CLARK ST',
    '4802 N BROADWAY',
]
```

ä¡ååŔŕäžěåČŔè£ŹæăůåĘŹåĹŮèąĺæŐĺåŕijïijŽ

```
>>> from fnmatch import fnmatchcase
>>> [addr for addr in addresses if fnmatchcase(addr, '* ST')]
['5412 N CLARK ST', '1060 W ADDISON ST', '2122 N CLARK ST']
>>> [addr for addr in addresses if fnmatchcase(addr, '54[0-9][0-9]␣
↪*CLARK*')]
['5412 N CLARK ST']
>>>
```

èőÍèőž

fnmatch() åĞ¡æŢřåŇźéĚ■èČ¡åŁŻäżŇäžŐçőĂå■ŢçŽĎå■ŮçňęäÿšæŰźæşŢåŠŇåijžåď ğçŽĎæ■čåĹŹèą åęĆæđIJåIJĺæŢřæ■őåď ĎçŘĘæŞ■ä¡IJäÿ■åŔłéIJĂèęĄçőĂå■ŢçŽĎéĂŽéĚ■çňęářśèČ¡åőŇæĹŘçŽĎæŮűåĂŹïijŇ

åęĆæđIJä¡ăçŽĎäžčçăĄéIJĂèęĄåĄŽæŰĞäžűåŘ■çŽĎåŇźéĚ■ïijŇæIJĂåě¡ä¡£çŤÍ glob æĺąåİŮãĂĆåŔĆèĂČ5.13ářŔèŁĆãĂĆ

4.4 2.4 å■Ůçňęäÿšå ŇźéĚ■åŠŇæŘIJçť ć

éŮőécŸ

ä¡ăæČşåŇźéĚ■æĹŰèĂĚæŘIJçť ćçL’źåőŽæĺąåijŔçŽĎæŰĞæIJň

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşåŇźéĚ■çŽĎæŸŕå■Ůéİćĺ■ŮçňęäÿšïijŇéĆčäźĹä¡ăéĂŽåÿÿåŔłéIJĂèęĄèřČçŤĺå§žæIJňå■Ů æŕŤåęĆ str.find() , str.endswith() , str.startswith() æĹŰèĂĚçśžäijijçŽĎæŰźæşŢïijŽ

```
>>> text = 'yeah, but no, but yeah, but no, but yeah'
>>> # Exact match
>>> text == 'yeah'
False
>>> # Match at start or end
>>> text.startswith('yeah')
True
>>> text.endswith('no')
```

```
False
>>> # Search for the location of the first occurrence
>>> text.find('no')
10
>>>
```

åŕźäžŐåď■æİĆçŽĎåŇźéĚ■éIJĂèęĄä¡£çŤĺæ■čåĹŹèąĺè¿¿åijŔåŠŇ `re` æĺąåİŮãĂĆ äÿžäžĘèğčéĞŁæ■čåĹŹèąĺè¿¿åijŔçŽĎå§žæIJňåŐ§çŘĘïijŇåĄĞèő¿ä¡ăæČşåŇźéĚ■æŢřå■Ůæăijåij ŔçŽĎæŮěæ `11/27/2012` ïijŇä¡ääŔŕäžěè£ŹæăůåĄŽïijŽ

```
>>> text1 = '11/27/2012'
>>> text2 = 'Nov 27, 2012'
>>>
>>> import re
>>> # Simple matching: \d+ means match one or more digits
>>> if re.match(r'\d+/\d+/\d+', text1):
... print('yes')
... else:
... print('no')
...
yes
>>> if re.match(r'\d+/\d+/\d+', text2):
... print('yes')
... else:
... print('no')
...
no
>>>
```

åęĆæđIJä¡ăæČşä¡£çŤĺåŘŇäÿĂäÿłæĺąåijŔåŐżåĄŽåď'ŽæňąåŇźéĚ■ïijŇä¡ääžŤèŕěåĚĹåřĘæĺąåijŔå■Ůçňęä

```
>>> datepat = re.compile(r'\d+/\d+/\d+')
>>> if datepat.match(text1):
... print('yes')
... else:
... print('no')
...
yes
>>> if datepat.match(text2):
... print('yes')
... else:
... print('no')
...
no
>>>
```

`match()` æĂżæŸŕäžŐå■ŮçňęäÿšåijĂåğŇåŐżåŇźéĚ■ïijŇåęĆæđIJä¡ăæČşæ§ěæL'¿å■ŮçňęäÿšäżżæĎŔé ä¡£çŤĺ `findall()` æŰźæşŢåŐżäžčæŽ£ãĂĆæŕŤåęĆïijŽ

```
>>> text = 'Today is 11/27/2012. PyCon starts 3/13/2013.'
>>> datepat.findall(text)
['11/27/2012', '3/13/2013']
>>>
```

åIJÍåőŽäźL'æ■čåĹŹåijŔçŽĎæŮűåĂŹïijŇéĂŽåÿÿäijŽåĹľçŤÍæŃňåŔůåŐżæ■ŢèŐůåĹĘçżĎãĂĆæŕŤåęĆïij

```
>>> datepat = re.compile(r'(\d+)/(\d+)/(\d+)')
>>>
```

æ■ŢèŐůåĹĘçżĎåŔŕäżěä¡£å¿ŮåŘŐéİćçŽĎåd'ĎçŘĘæŻt'åŁäçőĂå■ŢïijŇåŻääÿžåŔŕäżěåĹĘåĹńářĘæŕŔäÿ

```
>>> m = datepat.match('11/27/2012')
>>> m
<_sre.SRE_Match object at 0x1005d2750>
>>> # Extract the contents of each group
>>> m.group(0)
'11/27/2012'
>>> m.group(1)
'11'
>>> m.group(2)
'27'
>>> m.group(3)
'2012'
>>> m.groups()
('11', '27', '2012')
>>> month, day, year = m.groups()
>>>
>>> # Find all matches (notice splitting into tuples)
>>> text
'Today is 11/27/2012. PyCon starts 3/13/2013.'
>>> datepat.findall(text)
[('11', '27', '2012'), ('3', '13', '2013')]
>>> for month, day, year in datepat.findall(text):
...     print('{}-{}-{}'.format(year, month, day))
...
2012-11-27
2013-3-13
>>>
```

`findall()` æŰźæşŢäijŽæŘIJçt'ćæŰĞæIJňåźűäžěåĹÙèąĺå¡ćåijŔè£ŤåŻđæL'ĂæIJL'çŽĎåŇźéĚ■ãĂĆ åęĆæđIJä¡ăæČşäžěè£■äžčæŰźåijŔè£ŤåŻđåŇźéĚ■ïijŇåŔŕäżěä¡£çŤÍ `finditer()` æŰźæşŢæİěäžčæŻ£ïijŇæŕŤåęĆïijŽ

```
>>> for m in datepat.finditer(text):
...     print(m.groups())
...
('11', '27', '2012')
('3', '13', '2013')
>>>
```

èőĺèőž

åĚşäžŐæ■čåĹŹèąĺè¿¿åijŔçŘĘèőžçŽĎæŢŹçĺŃåůšçżŔèűĚåĞžäžĘæIJňäźęçŽĎèŇČåŻťãĂĆ äÿ■è£ĞïijŇè£ŹäÿĂèŁĆéŸŘè£řäžĘä¡£çŤĺreæĺąåİŮè£ŻèąŇåŇźéĚ■åŠŇæŘIJçťćæŰĞæIJňçŽĎæIJĂå§žæIJňæ æăÿå£Čæ■ěéłďåřśæŸŕåĚĹä¡£çŤĺ `re.compile()` çijŰèŕŚæ■čåĹŹèąĺè¿¿åijŔå■ŮçňęäÿšïijŇ çĎűåŘŐä¡£çŤĺ `match()`, `findall()` æĹŰèĂĚ `finditer()` ç■ĽæŰźæşŢãĂĆ

å¡ŞåĘŹæ■čåĹŹåijŔå■ŮçňęäÿšçŽĎæŮűåĂŹïijŇçŻÿåŕźæŹőéĄ■çŽĎåĄŽæşŢæŸŕä¡£çŤĺåŐ§åğŃå■Ůçňęä `r'(\d+)/(\d+)/(\d+)'` ãĂĆ è£Źçğ■å■ŮçňęäÿšåřĘäÿ■åŐżèğčæđŘåŔ■æŰIJæİăïijŇè£ŹåIJĺæ■čåĹŹèąĺè åęĆæđIJäÿ■è£ŹæăůåĄŽçŽĎèŕİïijŇä¡ăå£Ěéążä¡£çŤĺäÿďäÿłåŔ■æŰIJæİăïijŇçśżäijij `'(\\d+)/(\\d+)/(\\d+)'` ãĂĆ

éIJĂèęĄæşĺæĎŔçŽĎæŸŕ `match()` æŰźæşŢäżĚäżĚæčĂæ§ěå■ŮçňęäÿšçŽĎåijĂåğŃéČĺåĹĘãĂĆåőČçŽ

```
>>> m = datepat.match('11/27/2012abcdef')
>>> m
<_sre.SRE_Match object at 0x1005d27e8>
>>> m.group()
'11/27/2012'
>>>
```

åęĆæđIJä¡ăæČşçš¿çąőåŇźéĚ■ïijŇçąőä£İä¡ăçŽĎæ■čåĹŹèąĺè¿¿åijŔäżě$çżŞåř¿ïijŇåřśåČŔè£ŹäźĹè£Źæăů

```
>>> datepat = re.compile(r'(\d+)/(\d+)/(\d+)$')
>>> datepat.match('11/27/2012abcdef')
>>> datepat.match('11/27/2012')
<_sre.SRE_Match object at 0x1005d2750>
>>>
```

æIJĂåŘŐïijŇåęĆæđIJä¡ăäżĚäżĚæŸŕåĄŽäÿĂæňąçőĂå■ŢçŽĎæŰĞæIJňåŇźéĚ■/æŘIJçťćæŞ■ä¡IJçŽĎèŕİ `re` æĺąåİŮçžğåĹńçŽĎåĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> re.findall(r'(\d+)/(\d+)/(\d+)', text)
[('11', '27', '2012'), ('3', '13', '2013')]
>>>
```

ä¡ĘæŸŕéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇåęĆæđIJä¡ăæĽŞçőŮåĄŽåďğéĞŔçŽĎåŇźéĚ■åŠŇæŘIJçťćæŞ■ä¡IJ æĺąåİŮçžğåĹńçŽĎåĞ¡æŢřäijŽåřĘæIJĂè£ŚçijŰèŕŚè£ĞçŽĎæĺąåijŔçijŞå■ŸèţůæİěïijŇåŻăæ■ďåźűäÿ■äijŽæűĹ ä¡ĘæŸŕåęĆæđIJä¡£çŤĺéćĎçijŰèŕŚæĺąåijŔçŽĎèŕİïijŇä¡ăåřĘäijŽåĞŔåřŚæ§ěæĽ¿åŠŇäÿĂäžŻéćİåďŰçŽĎåďĎ

4.5 2.5 å■ŮçňęäÿšæŘIJçťćåŠŇæŻ£æ■ć

éŮőéćŸ

ä¡ăæČşåIJĺå■Ůçňęäÿšäÿ■æŘIJçťćåŠŇåŇźéĚ■æŇĞåőŽçŽĎæŰĞæIJňæĺąåijŔ

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎå■ŮéİćæĺąåijŔïijŇçŻť'æŐëä¡£çŤĺ `str.replace()` æŰźæşŢå■şåŔŕïijŇæŕŤåęĆïijŽ

```
>>> text = 'yeah, but no, but yeah, but no, but yeah'
>>> text.replace('yeah', 'yep')
'yep, but no, but yep, but no, but yep'
>>>
```

åŕźäžŐåď'■æİĆçŽĎæĺąåijŔïijŇèŕůä¡£çŤĺ `re` æĺąåİŮäÿ■çŽĎ `sub()` åĞ¡æŢřãĂĆ äÿžäžĘèŕť'æŸŐè£ŹäÿłïijŇåĄĞèő¿ä¡ăæČşåřĘå¡ćåijŔäÿž `11/27/2012` çŽĎæŮěæIJ§å■ŮçňęäÿšæŤźæĹŘ `2012-11-27` ãĂĆçđ'žä¿ŃåęĆäÿŇïijŽ

```
>>> text = 'Today is 11/27/2012. PyCon starts 3/13/2013.'
>>> import re
>>> re.sub(r'(\d+)/(\d+)/(\d+)', r'\3-\1-\2', text)
'Today is 2012-11-27. PyCon starts 2013-3-13.'
>>>
```

`sub()` åĞ¡æŢřäÿ■çŽĎçňňäÿĂäÿłåŔĆæŢřæŸŕèćńåŇźéĚ■çŽĎæĺąåijŔïijŇçňňäžŇäÿłåŔĆæŢřæŸŕæŻ£æ
`\3` æŇĞåŘŠåĽ■éİćæĺąåijŔçŽĎæ■ŢèŐůçżĎåŔůãĂĆ

åęĆæđIJä¡ăæĽŞçőŮçŤĺçŻÿåŘŇçŽĎæĺąåijŔåĄŽåď'Žæňąæ Ż£æ■ćïijŇèĂČèŹŠåĚĹçijŰèŕŠåőČæİěæŔŘå■

```
>>> import re
>>> datepat = re.compile(r'(\d+)/(\d+)/(\d+)')
>>> datepat.sub(r'\3-\1-\2', text)
'Today is 2012-11-27. PyCon starts 2013-3-13.'
>>>
```

åŕźäžŐæŻť'åŁååď'■æİĆçŽĎæŻ£æ■ćïijŇåŔŕäžěäijăéĂŠäÿĂäÿłæŻ£æ■ćåŻđèřČåĞ¡æŢřæİěäžčæŻ£ïijŇæŕ

```
>>> from calendar import month_abbr
>>> def change_date(m):
... mon_name = month_abbr[int(m.group(1))]
... return '{} {} {}'.format(m.group(2), mon_name, m.group(3))
...
>>> datepat.sub(change_date, text)
'Today is 27 Nov 2012. PyCon starts 13 Mar 2013.'
>>>
```

äÿĂäÿłæŻ£æ■ćåŻđèřČåĞ¡æŢřçŽĎåŔĆæŢřæŸŕäÿĂäÿł `match` åŕźèśąïijŇäźş§åŕśæŸŕ `match()` æĹŰèĂĚ `find()` è£ŤåŻđçŽĎåŕźèśąãĂĆ ä¡£çŤĺ `group()` æŰźæşŢæİěæŔŘåŔŰçĽźåőŽçŽĎåŇźéĚ■éČĺåĹĘãĂĆåŻđèřČåĞ¡æŢřæIJĂåŘŐè£ŤåŻđæŻ£æ■ćå■Ůçňęäÿšã

åęĆæđIJéŹď'äžĘæŻ£æ■ćåŘŐçŽĎçżŞæđIJåď'ŰïijŇä¡ăè£ŸæČşç§ěéĄŞæIJĽåď'ŽåřŚæŻ£æ■ćåŔŚçŢ§äžĘ
`re.subn()` æİěäžčæŻ£ãĂĆæŕŤåęĆïijŽ

```
>>> newtext, n = datepat.subn(r'\3-\1-\2', text)
>>> newtext
```

```
'Today is 2012-11-27. PyCon starts 2013-3-13.'
>>> n
2
>>>
```

èőÍèőž

åĚşäžŐæ■čåĹŹèąÍè¿¿åijŔæŘIJçť ćåŠŇæŻ£æ■ćïijŇäÿŁéİćæijŤçď žçŽĎ sub() æŰźæşŢå§žæIJňåůšçżŔæűţçŻŰäžĘæĽ ĂæIJL ãĂĆ åĚűåőđæIJĂéŽ¿çŽĎéČĺåĹĘåřśæŸŕçijŰåĘŹæ■čåĹŹèąÍè¿

4.6 2.6 å■Ůçňęäÿšå£¡çŢěåď ğåřŔåĘŹçŽĎæŘIJçť ćæŻ£æ■ć

éŮőécŸ

ä¡ăéIJĂèęĄäžěå£¡çŢěåď ğåřŔåĘŹçŽĎæŰźåijŔæŘIJçť ćäÿŐæŻ£æ■ćæŰĞæIJňå■Ůçňęäÿš

èğčåĘşæŰźæąĹ

äÿžäžĘåIJÍæŰĞæIJňæŞ■ä¡IJæŮűå£¡çŢěåď ğåřŔåĘŹïijŇä¡ăéIJĂèęĄåIJÍä¡£çŤÍ re æĺąåİŮçŽĎæŮűåĂŹçżŹè£ŹäžŻæŞ■ä¡IJæŔŘä¿Ż re.IGNORECASE æăĞå£ŮåŔĆæŢřăĂĆæŕŤåęĆïijŽ

```
>>> text = 'UPPER PYTHON, lower python, Mixed Python'
>>> re.findall('python', text, flags=re.IGNORECASE)
['PYTHON', 'python', 'Python']
>>> re.sub('python', 'snake', text, flags=re.IGNORECASE)
'UPPER snake, lower snake, Mixed snake'
>>>
```

æIJĂåŘŐçŽĎéĆčäÿłä¿Ńå■ŘæŔ■çď žäžĘäÿĂäÿłåřŔçijžéŹůïijŇæŻ£æ■ćå■Ůçňęäÿšåźűäÿ■äijŽèĞłåŁĺèů§äÿžäžĘä£őåď ■è£ŹäÿłïijŇä¡ăåŔŕèČ¡éIJĂèęĄäÿĂäÿłè¿ĚåŁľåĞ¡æŢřïijŇåřšåČŔäÿŃéİćçŽĎè£Źæăůïijž

```
def matchcase(word):
    def replace(m):
        text = m.group()
        if text.isupper():
            return word.upper()
        elif text.islower():
            return word.lower()
        elif text[0].isupper():
            return word.capitalize()
        else:
            return word
    return replace
```

äÿŃéİćæŸŕä¡£çŤĺäÿŁè£ŕåĞ¡æŢřçŽĎæŰźæşŢïijž

```
>>> re.sub('python', matchcase('snake'), text, flags=re.IGNORECASE)
'UPPER SNAKE, lower snake, Mixed Snake'
>>>
```

èґŚèĂĚæşÍïijŽ `matchcase('snake')` è£ŤåŻđäžĘäÿĂäÿłåŻđèřČåĞ¡æŢř(åŔĆæŢřå£ĚéążæŸŕ `match` åŕźèśą)ïijŇåL’■éİćäÿĂèŁĆæŔŘåĹřè£ĞïijŇ `sub()` åĞ¡æŢřéŹď’äžĘæŐěåŔŮæŻ£æ■ćå■Ůçňęäÿšåď’ŰïijŇè£ŸèČ¡æŐěåŔŮäÿĂäÿłåŻđèřČåĞ¡æŢřãĂĆ

èőÍèőž

åŕźäžŐäÿĂèĹňçŽĎå£¡çŢ̈åď’ğåřŔåĘŹçŽĎåŇźéĚ■æŞ■ä¡IJïijŇçőĂå■ŢçŽĎäijåéĂŠäÿĂäÿł `re.IGNORECASE` æăĞå£ŮåŔĆæŢřåřśåůšçżŔèűşåď’§äžĘãĂĆ ä¡ĘæŸŕéIJĂèęĄæşÍæĎŔçŽĎæŸŕïijŇè£ŹäÿłåŕźäžŐæ§ŘäžŻéIJĂèęĄåď’ğåřŔåĘŹè¡ñæ■ćçŽĎUnicodeåŇźéĚ■å åŔĆèĂČ2.10åřŔèŁĆäžĘèğčæŻt’åď’ŽçżĘèŁĆãĂĆ

4.7 2.7 æIJĂç§■åŇźéĚ■æĺąåijŔ

éŮőécŸ

ä¡ăæ■čåIJĺèŕŢçİĂçŤĺæ■čåĹŹèąĺè¿¿åijŔåŇźéĚ■æ§Řäÿłæ̋ŰĞæIJñæĺąåijŔïijŇä¡ĘæŸŕåőČæL’¿åĹřçŽĎæŸ èĂŇä¡ăæČşä£őæŤźåőČåŔŸæĹŘæ§ěæL’¿æIJĂç§■çŽĎåŔŕèČ¡åŇźéĚ■ãĂĆ

èğčåĘşæŰźæąĹ

è£ŹäÿłéŮőécŸäÿĂèĹňåĞžçŐřåIJĺéIJĂèęĄåŇźéĚ■äÿĂåŕźåĹĘéŽŤçňęäźŃéŮt’çŽĎæŰĞæIJñçŽĎæŮűåĂ äÿžäžĘèŕt’æŸŐæÿĚæěŽïijŇèĂČèŹŚåęĆäÿŃçŽĎä¿Ńå■ŘïijŽ

```
>>> str_pat = re.compile(r'\"(.*)\"')
>>> text1 = 'Computer says "no."'
>>> str_pat.findall(text1)
['no.']
>>> text2 = 'Computer says "no." Phone says "yes."'
>>> str_pat.findall(text2)
['no." Phone says "yes.']
>>>
```

åIJĺè£Źäÿłä¿Ńå■Řäÿ■ïijŇæĺąåijŔ `r'\"(.*)\"'` çŽĎæĎŔåŻ¿æŸŕåŇźéĚ■èćńåŔŇåijŢåŔůåŇĚåŘńçŽĹ ä¡ĘæŸŕåIJĺæ■čåĹŹèąĺè¿¿åijŔäÿ■*æŞ■ä¡IJçňęæŸŕèt’łåľłçŽĎïijŇåŻăæ■d’åŇźéĚ■æŞ■ä¡IJäijŽæ§ěæL’¿æIJĂéŢ äžŐæŸŕåIJĺçňňäžŇäÿłä¿Ńå■Řäÿ■æŘIJçt’ć `text2` çŽĎæŮűåĂŹè£ŤåŻđçżŞæđIJåźűäÿ■æŸŕæĹŚäžňæČşèęĄç

äÿžäžĘä£őæ■čè£ŹäÿłéŮőécŸïijŇåŔŕäžěåIJĺæĺąåijŔäÿ■çŽĎ*æŞ■ä¡IJçňęåŘŐéİćåŁääÿŁ?ä£őéěřçňęïijŇå

```
>>> str_pat = re.compile(r'\"(.*?)\"')
>>> str_pat.findall(text2)
['no.', 'yes.']
>>>
```

è£Źæăůåřśä¡£å¿ŮåŇźéĚ■åŔŸæĹŘéİđèť łåľ łæĺąåijŔïijŇäżŐèĂŇå¿ŮåĹřæIJĂç§■çŽĎåŇźéĚ■ïijŇäź§åřśa

èőĺèőž

è£ŹäÿĂèŁĆåśŢçď žäžĘåIJĺåĘŹåŇĚåŘńçĆź(.)å■ŮçňęçŽĎæ■čåĹŹèąĺè¿¿åijŔçŽĎæŮűåĂŹéĄĞåĹřçŽĎäÿ
åIJĺäÿĂäÿłæĺąåijŔå■Ůçňęäÿšäÿ■ïijŇçĆź(.)åŇźéĚ■éŹď äžĘæ■ćèąŇåď ŰçŽĎäżżä¡Ţå■ŮçňęãĂĆ
çĎűèĂŇïijŇåęĆæđIJä¡ăåřĘçĆź(.)åŔůæŤ¿åIJĺåijĂåğŃäÿŐçżŞæİ§çňę(æŕŤåęĆåijŢåŔů)äźŃéŮť çŽĎæŮűåĂŹï
è£ŹæăůéĂŽåÿÿäijŽåŕijèĞť å¿Ĺåď Žäÿ■éŮť çŽĎèćńåijĂåğŃäÿŐçżŞæİ§çňęåŇĚåŘńçŽĎæŰĞæIJňèćńå£¡çŢěæ
éĂŽè£ĞåIJĺ * æĹŰèĂĚ + è£ŹæăůçŽĎæŞ■ä¡IJçňęåŘŐéİćæůżåŁăäÿĂäÿł ?
åŔŕäżěåijžåĹűåŇźéĚ■çőŮæşŢæŤźæĹŘåŕżæĽ¿æIJĂç§■çŽĎåŔŕèČ¡åŇźéĚ■ãĂĆ

4.8 2.8 åď ŽèąŇåŇźéĚ■æĺąåijŔ

éŮőéćŸ

ä¡ăæ■čåIJĺèŕŢçİĂä¡£çŤĺæ■čåĹŹèąĺè¿¿åijŔåŐżåŇźéĚ■äÿĂåď ğåİŮçŽĎæŰĞæIJňïijŇèĂŇä¡ăéIJĂèęĄèůĺ

èğčåĘşæŰźæąĹ

è£ŹäÿłéŮőéćŸå¿ĹåĚÿåđŃçŽĎåĞžçŐřåIJĺå¡Şä¡ăçŤĺçĆź(.)åŐżåŇźéĚ■äżżæĎŔå■ŮçňęçŽĎæŮűåĂŹïijŇ
æŕŤåęĆïijŇåĄĞèő¿ä¡ăæČşèŕŢçİĂåŐżåŇźéĚ■Cèŕ■èĺĂåĹĘåĽšçŽĎæşĺéĞŁïijŽ

```
>>> comment = re.compile(r'/\*(.*?)\*/')
>>> text1 = '/* this is a comment */'
>>> text2 = '''/* this is a
... multiline comment */
... '''
>>>
>>> comment.findall(text1)
[' this is a comment ']
>>> comment.findall(text2)
[]
>>>
```

äÿžäžĘä£őæ■čè£ŹäÿłéŮőéćŸïijŇä¡ăåŔŕäżěä£őæŤźæĺąåijŔå■ŮçňęäÿšïijŇåćđåŁăåŕźæ■ćèąŇçŽĎæŤŕæŇ

```
>>> comment = re.compile(r'/\*((?:.|\n)*?)\*/')
>>> comment.findall(text2)
[' this is a\n multiline comment ']
>>>
```

åIJĺè£ŹäÿłæĺąåijŔäÿ■ïijŇ (?:.|\n) æŇĞåőŽäžĘäÿĂäÿłéİđæ■ŢèŐůçżĎ
(äź§åřśæŸŕåőČåőŽäźĽäžĘäÿĂäÿłäżĚäżĚçŤĺæİěåĄŽåŇźéĚ■ïijŇèĂŇäÿ■èČ¡éĂŽè£Ğå■ŢçŃňæ■ŢèŐůæĹŰèĂ

èőÍèőž

re.compile() åĞ¡æŢřæŐěåŔŮäÿĂäÿłæăĞå£ŮåŔĆæŢřåŔń re.DOTALL ïijŇåIJÍè£ŹéĞŇéÍđåÿÿæIJL'çŤÍãĂĆ åőČåŔŕäžěèől'æ■čåĹŹèąÍè¿¿åijŔäÿ■çŽĎçĆź(.)åŇźéĚ■åŇĚæŃňæ■ćèąŇ

```
>>> comment = re.compile(r'/\*(.*?)\*/', re.DOTALL)
>>> comment.findall(text2)
[' this is a\n multiline comment ']
```

åŕźäžŐçőĂå■ŢçŽĎæČĚåĘţä¡£çŤÍ re.DOTALL æäĞèőřåŔĆæŢřåůěä¡IJçŽĎå¿Ĺåě¡ïijŇ ä¡ĘæŸŕåęĆæđIJæĺąåijŔéÍđåÿÿåd'■æÍĆæĹŰèĂĚæŸŕäÿžäžĘæđĎéĂäå■Ůçňęäÿšäžd'çL'ŇèĂŇåřĘåd'Žäÿłæĺąåi è£ŹæŮűåĂŹä¡£çŤÍè£ŹäÿłæäĞèőřåŔĆæŢřåřšåŔŕèČ¡åĞžçŐřäÿĂäžŻéŮőécŸãĂĆ åęĆæđIJèől'ä¡ăéĂL'æŃl'çŽĎèŕÍïijŇæIJĂåě¡è£ŸæŸŕåőŽäźL'èĞłåůšçŽĎæ■čåĹŹèąÍè¿¿åijŔæĺąåijŔïijŇè£Źæăů

4.9 2.9 åřĘUnicodeæŰĞæIJňæäĞåĞĘåŇŰ

éŮőécŸ

ä¡ăæ■čåIJÍåd'ĎçŘĘUnicodeå■ŮçňęäÿšïijŇéIJĂèęAçąőä£ÍæL'ĂæIJL'å■ŮçňęäÿšåIJÍåžŢåśĆæIJL'çŻÿåŘŇ

èğčåĘşæŰźæąĹ

åIJÍUnicodeäÿ■ïijŇæ§ŘäžŻå■Ůçňęè Č¡åd'§çŤÍåd'Žäÿłå ŘĹæşŢçŽĎçijŰçăĄèąÍçd'žãĂĆäÿžäžĘèŕt'æŸŐïij

```
>>> s1 = 'Spicy Jalape\u00f1o'
>>> s2 = 'Spicy Jalapen\u0303o'
>>> s1
'Spicy JalapeÃ±o'
>>> s2
'Spicy JalapeÃ±o'
>>> s1 == s2
False
>>> len(s1)
14
>>> len(s2)
15
>>>
```

è£ŹéĞŇçŽĎæŰĞæIJňâĂİSpicy JalapeÃ±oâĂİä¡£çŤÍäžĘäÿd'çğ■å¡ćåijŔæİěèąÍçd'žãĂĆ çňňäÿĂçğ■ä¡£çŤÍæŢt'ä¡Şå■Ůçňęâ ĂİÃ±âĂİ(U+00F1)ïijŇçňňäžŇçğ■ä¡£çŤÍæŃL'äÿĄå■Ůæŕ■âĂİnâĂİåŘŐéÍć

åIJÍéIJĂèęĄæŕŤè¿Čå■ŮçňęäÿšçŽĎçÍŃåžŔäÿ■ä¡£çŤÍå■ŮçňęçŽĎåd'Žçğ■èąÍçd'žäijŽäžğçŤ§éŮőécŸãĂĆ äÿžäžĘä£őæ■čè£ŹäÿłéŮőécŸïijŇä¡ăåŔŕäžěä¡£çŤÍunicodedataæÍąåİŮåĚĹårĘæŰĞæIJňæäĞåĞĘåŇŰïijŽ

```
>>> import unicodedata
>>> t1 = unicodedata.normalize('NFC', s1)
>>> t2 = unicodedata.normalize('NFC', s2)
>>> t1 == t2
```

```
True
>>> print(ascii(t1))
'Spicy Jalape\xf1o'
>>> t3 = unicodedata.normalize('NFD', s1)
>>> t4 = unicodedata.normalize('NFD', s2)
>>> t3 == t4
True
>>> print(ascii(t3))
'Spicy Jalapen\u0303o'
>>>
```

normalize() çňňäÿĂäÿłåŔĆæŢřæŇĞåőŽå■ŮçňęäÿšæăĞåĞĘåŇŰçŽĎæŰźåijŔãĂĆ NFCèąĺçd'žå■ŮçňęåžŤèŕěæŸŕæŢt'ä¡ŞçżĎæĹŘ(æŕŤåęĆåŔŕèČ¡çŽĎèŕÍåřśä¡£çŤĺå■ŢäÿĂçijŰçăA)ïijŇèĂŇNFI

PythonåŘŇæăůæŤŕæŇĄæĽľ'åśŢçŽĎæăĞåĞĘåŇŰå¡ćåijŔNFKCåŠŇNFKDïijŇåőČäžňåIJÍåd'ĎçŘĘæ§Ř

```
>>> s = '\ufb01' # A single character
>>> s
'ïňĄ'
>>> unicodedata.normalize('NFD', s)
'ïňĄ'
# Notice how the combined letters are broken apart here
>>> unicodedata.normalize('NFKD', s)
'fi'
>>> unicodedata.normalize('NFKC', s)
'fi'
>>>
```

èőÍèőž

æăĞåĞĘåŇŰåŕźäžŐäżä¡ŢéIJĂèęĄäžěäÿĂèĞt'çŽĎæŰźåijŔåd'ĎçŘĘUnicodeæŰĞæIJňçŽĎçÍŇåžŔéČ¡å å¡Şåd'ĎçŘĘæÍèèĞłçŤÍæĹůè¿ŞåĚěçŽĎå■Ůçňęäÿšè ĂŇä¡ăå¿ĹéŽ¿åÖżæŐğåĹűçijŰçăĄçŽĎæŮűåĂŹård'åĚűåę

åIJÍæÿĚçŘĘåŠŇè£Ğæżd'æŰĞæIJňçŽĎæŮűåĂŹå■ŮçňęçŽĎæăĞåĞĘåŇŰäź§æŸŕå¿ĹéĞ■èęĄçŽĎãĂĆ æŕŤåęĆïijŇåĄĞèő¿ä¡ăæČşæÿĚéŹd'æŐL'äÿĂäžŻæŰĞæIJňäÿŁéİćçŽĎåŔŸé§şçňęçŽĎæŮűåĂŹ(åŔŕèČ¡æŸŕäÿ

```
>>> t1 = unicodedata.normalize('NFD', s1)
>>> ''.join(c for c in t1 if not unicodedata.combining(c))
'Spicy Jalapeno'
>>>
```

æIJĂåŘŐäÿĂäÿłä¿Ńå■ŘåśŢçd'žäžĘ unicodedata æĺąåİŮçŽĎåŘęäÿĂäÿłéĞ■èęĄæŰźéİćïijŇäź§årśæ combining() åĞ¡æŢřåŔŕäžěæţŃèŕŢäÿĂäÿłå■ŮçňęæŸŕåŘęäÿžåŠŇé§şå■Ůçňęã ĂĆ åIJÍè£ŹäÿłæĺąåİŮäÿ■è£ŸæIJL'åĚűäžŰåĞ¡æŢřçŤÍäžŐæ§ěæL'¿å■ŮçňęçśžåĹńïijŇæţŃèŕŢæŸŕåŘęäÿžæŢřå■Ůå

UnicodeæŸ¿çĎűæŸŕäÿĂäÿłå¿Ĺåd'ğçŽĎäÿżéćŸãĂĆåęĆæđIJæČşæŻt'æůśåĚěçŽĎäžĘèğčåĚşäžŐæăĞåĞ èŕůçIJŃèĂČ UnicodeåőŸç¡Śäÿ■åĚşäžŐè£ŹéČĺåĹĘçŽĎèŕt'æŸŐ Ned BatchelderåIJÍ äżŰçŽĎç¡ŚçńŹ äÿŁåŕźPythonçŽĎUnicodeåd'ĎçŘĘéŮőécŸäź§æIJL'äÿĂäÿłå¿Ĺåě¡çŽĎäżŃçż■ãĂĆ

4.10 2.10 åIJÍæ■čåĹŹåijŔäÿ■ä¡£çŤÍUnicode

éŮőécŸ

ä¡ăæ■čåIJÍä¡£çŤÍæ■čåĹŹèąĺè¿¿åijŔåd'ĎçŘĘæŰĞæIJñïijÑä¡ĘæŸŕåĚşæşÍçŽĎæŸŕUnicodeå■Ůçňęåd'Ďç

èğčåĘşæŰźæąĹ

éžŸèőd'æČĚåĘţäÿŃ re æÍąåİŮåůšçżŔåŕźäÿĂäžŻUnicodeå■Ůçňęçśżæ IJL'äžĘå§žæIJñçŽĎæŤŕæŇĄãĂĆ æŕŤåęĆïijŇ \\d åůšçżŔåŇźéĚ■äżżæĎŔçŽĎunicodeæŢřå■Ůå■Ůçňęäžĕïijž

```
>>> import re
>>> num = re.compile('\d+')
>>> # ASCII digits
>>> num.match('123')
<_sre.SRE_Match object at 0x1007d9ed0>
>>> # Arabic digits
>>> num.match('\u0661\u0662\u0663')
<_sre.SRE_Match object at 0x101234030>
>>>
```

åęĆæđIJä¡ăæČşåIJÍæĺąåijŔäÿ■åŇĚåŘŕæŇĞåőŽçŽĎUnicodeå■ŮçňęïijŇä¡ääŔŕäżěä¡£çŤÍUnicodeå■Ůçň \uFFF æĹŰèĂĚ \UFFFFFFF)ãĂĆ æŕŤåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłåŇźéĚ■åĞääÿłäÿ■åŘŇéŸ£æŃL'äijŕçijŰç

```
>>> arabic = re.compile('[\u0600-\u06ff\u0750-\u077f\u08a0-\u08ff]+
↪')
>>>
```

å¡ŞæL'ğèąŇåŇźéĚ■åŠŇæŘIJçt'ćæŞ■ä¡IJçŽĎæŮűåĂŹïijŇæIJĂåě¡æŸŕåĚĹæăĞåĞĘåŇŰåźűäÿŤæÿĚçŘĚ ä¡ĘæŸŕåŘŇæăůäź§åžŤèŕěæşÍæĎŔäÿĂäžŻçL'źæőŁæČĚåĘţïijŇærŤåęĆåIJÍå£¡çŢěåd'ğářŔåĘŹåŇźéĚ■åŠŇåd'

```
>>> pat = re.compile('stra\u00dfe', re.IGNORECASE)
>>> s = 'straÃ§e'
>>> pat.match(s) # Matches
<_sre.SRE_Match object at 0x10069d370>
>>> pat.match(s.upper()) # Doesn't match
>>> s.upper() # Case folds
'STRASSE'
>>>
```

èőÍèőž

æůůåŘĹä¡£çŤÍUnicodeåŠŇæ■čåĹŹèąĺè¿¿åijŔéĂŽåÿÿäijŽèőľ'ä¡ăæŁŞçŃĆãĂĆ åęĆæđIJä¡ăçIJ§çŽĎæL'ŞçőŮè£ŹæăůåĄŽçŽĎèŕĺïijŇæIJĂåě¡èĂČèŹŚäÿŃåőL'èčĚçňňäÿL'æŰźæ■čåĹŹåijŔåžŞ åőČäžňäijŽäÿžUnicodeçŽĎåd'ğářŔåĘŹè¡ňæ■ćåŠŇåĚűäżŰåd'ğéĞŔæIJL'èűčçL'źæĂğæŔŘä¿ŻåĚĺéİćçŽĎæŤŕ

4.11 2.11 åĹǎéŹďå■Ůçňęäÿšäÿ■äÿ■éIJÄèęĄçŽĎå■Ůçňę

éŮőécŸ

ä¡ăæČşåŐżæŐĽ'æŰĞæIJňå■Ůçňęäÿšåij Ăåď't'ïijŇçżŞář¿æĹŰèĂĚäÿ■éŮt'äÿ■æČşèęĄçŽĎå■Ůçňęïij Ňæŕ

èğčåĘşæŰźæąĹ

strip() æŰźæşŢèČ¡çŤĺäžŐåĹǎéŹď'åijĂåğŃæĹŰçżŞář¿çŽĎå■Ůçňęã ĂĆ lstrip() åŠŇ rstrip() åĹĘąĹńäżŐåůęåŠŇäżŐåŔşæL'ğèąŇåĹǎéŹď'æŞ■ä¡IJãĂĆ éżŸèŏď'æČĚåĘţäÿŃïijŇè£ŹäžŻæŰźæşŢäijŽåŐżéŹď'çl'žçŹ¡å■Ůçňęïij Ňä¡ĘæŸŕä¡ääź§åŔŕäżěæŇĞåőŽåĚűäżŰ

```
>>> # Whitespace stripping
>>> s = ' hello world \n'
>>> s.strip()
'hello world'
>>> s.lstrip()
'hello world \n'
>>> s.rstrip()
' hello world'
>>>
>>> # Character stripping
>>> t = '-----hello====='
>>> t.lstrip('-')
'hello====='
>>> t.strip('-=')
'hello'
>>>
```

èőÍèőž

è£ŹäžŻ strip() æŰźæşŢåIJÍèŕżåŔŰåŠŇæÿĚçŘĘæŢřæ■őäžěåď'ĞåŘŐçż■åď'ĎçŘĘçŽĎæŮűåĂŹæŸŕç æŕŤåęĆïijŇä¡ääŔŕäżěçŤÍåőČäžňæÌěåŐżæŐĽ'çl'žæăijïijŇåijŢåŔůåŠŇåőŇæĹŘåĚűäżŰäżżåŁąãĂĆ

ä¡ĘæŸŕéIJÄèęĄæşÍæĎŔçŽĎæŸŕåŐżéŹď'æŞ■ä¡IJäÿ■äijŽåŕźå■ŮçňęäÿšçŽĎäÿ■éŮt'çŽĎæŰĞæIJňäžğçŤ

```
>>> s = ' hello      world \n'
>>> s = s.strip()
>>> s
'hello      world'
>>>
```

åęĆæđIJä¡ăæČşåď'ĎçŘĘäÿ■éŮt'çŽĎçl'žæăijïijŇéĆčäźĹä¡ăéIJÄèęĄæśĆåŁl'åĚűäżŰæŁĂæIJŕãĂĆæŕŤåę replace() æŰźæşŢæĹŰèĂĚæŸŕçŤÍæ■čåĹŹèąÍè¿¿åijŔæŽ£æ■ćãĂĆçď'žä¿ŃåęĆäÿŃïijŽ

```
>>> s.replace(' ', '')
'helloworld'
>>> import re
```

```
>>> re.sub('\s+', ' ', s)
'hello world'
>>>
```

éĂŽåÿÿæČĚåĘţäÿŃä¡ăæČşåřĘå■Ůçňęäÿš strip æŞ■ä¡IJåŠŇåĚűäżŰè£■äżčæŞ■ä¡IJçŻÿçżŞåŘĹïijŇæŕ
åęĆæđIJæŸŕè£ŹæăůçŽĎèŕİïijŇéĆčäźĹçŤ§æĹŘåŹĺèąĺè¿¿åijŔåřśåŔŕäżěåď ğæŸ¿èžńæĽŃäžĘãĂĆæŕŤåęĆïijŽ

```
with open(filename) as f:
    lines = (line.strip() for line in f)
    for line in lines:
        print(line)
```

åIJĺè£ŹéĞŇïijŇèąĺè¿¿åijŔ lines = (line.strip() for line in f) æĽğèąŇæŢřæ■őè¡ňæ■ćæŞ■ä¡IJãĂĆ è£Źçğ■æŰźåijŔéİđåÿÿéńŸæŢĹïijŇåŻăäÿžåőČäÿ■éIJĂèęĄéćĎåĚĹèŕżåĺ
åőČäżĚäżĚåŔłæŸŕåĹŻåżžäÿĂäÿłçŤ§æĹŘåŹĺïijŇåźűäÿŤæŕŔæňąè£ŤåŻđèąŇäźŃåĽ■äijŽåĚĹæĽğèąŇ
strip æŞ■ä¡IJãĂĆ

åŕźäžŐæŻť éńŸéŸűçŽĎstripïijŇä¡ăåŔŕèČ¡éIJĂèęĄä¡£çŤĺ translate() æŰźæşŢãĂĆèŕůåŔĆéŸĚäÿŃäÿĂèŁĆäžĘèğčæŻť åď ŽåĚşäžŐå■ŮçňęäÿšæÿĚçŘĘçŽĎåĘĚåőźãĂĆ

4.12 2.12 åőąæ§ěæÿĚçŘĘæŰĞæIJňå■Ůçňęäÿš

éŮőéćŸ

äÿĂäžŻæŮăèĄŁçŽĎåźijçĺŽéżŚåőćåIJĺä¡ăçŽĎç¡ŚçńŹéąţéİćèąĺå■Ţäÿ■è¿ŞåĚěæŰĞæIJňâĂİpÃ¡tÄěÃűÃś

èğčåĘşæŰźæąĹ

æŰĞæIJňæÿĚçŘĘéŮőéćŸäijŽæűĽåŔŁåĹřåŇĚæŃňæŰĞæIJňèğčæđŘäÿŐæŢřæ■őåď ĎçŘĘç■ĽäÿĂçşżå
åIJĺéİđåÿÿçőĂå■ŢçŽĎæČĚå¡ćäÿŃïijŇä¡ăåŔŕèČ¡äijŽéĂĽæŃľä¡£çŤĺå■ŮçňęäÿšåĞ¡æŢř(æŕŤåęĆ
str.upper() åŠŇ str.lower())åřĘæŰĞæIJňè¡ňäÿžæăĞåĞĘæăijåijŔãĂĆ ä¡£çŤĺ
str.replace() æĹŰèĂĚ re.sub() çŽĎçőĂå■ŢæŻ£æ■ćæŞ■ä¡IJèČ¡åĹăéŹď æĹŰèĂĚæŤźåŔŸæŇĞåőŹ
ä¡ăåŘŇæăůè£ŸåŔŕäżěä¡£çŤĺ2.9åřŔèŁĆçŽĎ unicodedata.normalize() åĞ¡æŢřåřĘunicodeæŰĞæIJňæăĞåĞĘåŇŰãĂĆ

çĎűåŘŐïijŇæIJĽæŮűåĂŹä¡ăåŔŕèČ¡è£ŸæČşåIJĺæÿĚçŘĘæŞ■ä¡IJäÿŁæŻť è£ŻäÿĂæ■ěãĂĆæŕŤåęĆïijŇä¡
äÿžäžĘè£ŹæăůåĄŽïijŇä¡ăåŔŕäżěä¡£çŤĺçżŔåÿÿäijŽèćńå£¡èğĘçŽĎ str.translate() æŰźæşŢãĂĆ äÿžäžĘæijŤçď žïijŇåĄĞèő¿ä¡ăçŐřåIJĺæIJĽäÿŃéİćè£ŹäÿłåĞŇäźśçŽĎå■ŮçňęäÿšïijŽ

```
>>> s = 'pÃ¡tÄěÃűÃś\fis\tawesome\r\n'
>>> s
'pÃ¡tÄěÃűÃś\x0cis\tawesome\r\n'
>>>
```

çňňäÿĂæ■ěæŸŕæÿĚçŘĘçľžçŹ¡å■ŮçňęãĂĆäÿžäžĘè£ŹæăůåĄŽïijŇåĚĹåĹŻåżžäÿĂäÿłåřŔçŽĎè¡ňæ■ćèąĺ
translate() æŰźæşŢïijŽ

```
>>> remap = {
...     ord('\t') : ' ',
...     ord('\f') : ' ',
...     ord('\r') : None # Deleted
... }
>>> a = s.translate(remap)
>>> a
'pÃ¡tÄëÃűÃś is awesome\n'
>>>
```

æ■čåęĆä¡äçIJŃçŽĎéĆčæăůïijŇçľžçŹ¡å■Ůçňę \t åŠŇ \f åůšçżŔèćńéĞ■æŰřæŸăářĎåĹřäÿĂäÿłçľžæăijãĂĆåŻđè¡ęå■ŮçňęrçŻť'æŐěèćńåĹăéŹď'ãĂĆ

ä¡ăåŔŕäžěäžěè£Źäÿłèąĺæăijäÿžå§žçąĂè£ŻäÿĂæ■ęæđĎåżžæŻť'åď'ğçŽĎèąĺæăijãĂĆæŕŤåęĆïijŇèőľ'æĹŚä

```
>>> import unicodedata
>>> import sys
>>> cmb_chrs = dict.fromkeys(c for c in range(sys.maxunicode)
...                          if unicodedata.combining(chr(c)))
...
>>> b = unicodedata.normalize('NFD', a)
>>> b
'pÃ¡tÄëÃűÃś is awesome\n'
>>> b.translate(cmb_chrs)
'python is awesome\n'
>>>
```

äÿŁéİćä¿Ńå■Řäÿ■ïijŇéĂŽè£Ğä¡£çŤĺ dict.fromkeys() æŰźæşŢæđĎéĂăäÿĂäÿłå■ŮåĚÿïijŇæŕŔäÿłUnicodeåŠŇé§şçňęä¡IJäÿžéŤőïijŇåŕźåžŤçŽĎåĂijåĚĺéČĺäÿž None ãĂĆ

çĎűåŘŐäj£çŤĺ unicodedata.normalize() åřĘåŐ§ågŃè¿ŞåĚěæăĞåĞĘåŇŰäÿžåĹĘèğčå¡ćåijŔå■Ů çĎűåŘŐåĘ■èřČçŤĺ translate åĞ¡æŢřåĹăéŹď'æĽ'ĂæIJL'éĞ■é§şçňęãĂĆ åŘŇæăůçŽĎæŁĂæIJŕäź§åŔŕäžěèćńçŤĺæİěåĹăéŹď'åĚűäžŰçśžåđŃçŽĎå■Ůçňę(æŕŤåęĆæŐğåĹűå■Ůçňęç■Ľ')ã

ä¡IJäÿžåŘęäÿĂäÿłä¿Ńå■ŘïijŇè£ŹéĞŇædĎéĂăäÿĂäÿłåřĘæĽ'ĂæIJL'UnicodeæŢřå■Ůå■ŮçňęæŸăářĎåĹi

```
>>> digitmap = { c: ord('0') + unicodedata.digit(chr(c))
...           for c in range(sys.maxunicode)
...           if unicodedata.category(chr(c)) == 'Nd' }
...
>>> len(digitmap)
460
>>> # Arabic digits
>>> x = '\u0661\u0662\u0663'
>>> x.translate(digitmap)
'123'
>>>
```

åŘęäÿĂçğ■æÿĚçŘĘæŰĞæIJňçŽĎæŁĂæIJŕæűL'åŔŁåĹřI/Oèğčçă Ąäÿŏçij Ůçă ĄåĞ¡æŢřãĂĆè£ŹéĞŇçŽĎ çĎűåŘŐåĘ■çżŞåŘĹ encode() æĹŰèĂĚ decode() æŞ■ä¡IJæİěæÿĚéŹď'æĹŰä£őæŤźåőČãĂĆæŕŤåęĆïijŽ

```
>>> a
'pÃ¡tÄëÃűÃś is awesome\n'
>>> b = unicodedata.normalize('NFD', a)
>>> b.encode('ascii', 'ignore').decode('ascii')
'python is awesome\n'
>>>
```

è£ŹéĞŇçŽĎæăĞåĞĘåŇŰæŞ■ä¡IJåřĘåŐ§æİěçŽĎæŰĞæIJňåĹĘèğčäÿžå■ŢçŃňçŽĎåŠŇé§şçňęãĂĆæŐě
å¡ŞçĎűïijŇè£Źçğ■æŰźæşŢäžĔäžĚåŔłåIJĺæIJĂåŘŐçŽĎçŻőæăĞåřśæŸŕèŐůåŘŰåĹřæŰĞæIJňåŕźåžŤACSIIèą

èőĺèőž

æŰĞæIJňå■Ůçňęæÿ̆ĔçŘĔäÿĂäÿłæIJĂäÿżèęĄçŽĎéŮőécŸåžŤèŕěæŸŕè£ŘèąŇçŽĎæĂğèČ¡ãĂĆäÿĂèĹňæ
åŕźäžŐçőĂå■ŢçŽĎæŻ£æ■ćæŞ■ä¡IJïijŇ `str.replace()` æŰźæşŢéĂŽåÿÿæŸŕæIJĂå£ńçŽĎïijŇçŤŽèĞşåIJ
æŕŤåęĆïijŇäÿžäžĘæÿĔçŘĔçl'žçŹ¡å■ŮçňęïijŇä¡ăåŘŕäžěè£ŹæăůåĄŽïijŽ

```
def clean_spaces(s):
    s = s.replace('\r', '')
    s = s.replace('\t', ' ')
    s = s.replace('\f', ' ')
    return s
```

åęĆæđIJä¡ăåŐżæţŃèŕŢçŽĎèŕÍïijŇä¡ăåřśäijŽåŘŞçŐřè£Źçğ■æŰźåijŔäijŽæŕŤä¡£çŤÍ
`translate()` æĹŰèĂĚæ■čåĹŹèąÍè¿¿åijŔèęĄå£ńå¿Ĺåď'ŽãĂĆ

åŘęäÿĂæŰźéÍćïijŇåęĆæđIJä¡ăéIJĂèęĄæL'ğèąŇäżżä¡Ţåď'■æİĆå■Ůçňęåŕźå■Ůçňę çŽĎéĞ■æŰřæŸăåřĎæ
`tanslate()` æŰźæşŢäijŽéÍđåÿÿçŽĎå£ńãĂĆ

äžŐåď'ğçŽĎæŰźéÍćæÍěèőšïijŇåŕźäžŐä¡ăçŽĎåžŤçŤÍçÍŃåžŘæÍěèŕť'æĂğèČ¡æŸŕä¡ääÿ■å¿Ůäÿ■åŐżèĞłåů
äÿ■åźÿçŽĎæŸŕïijŇæĹŚäżñäÿ■åŘŕèČ¡çżŹä¡ăåżžèőőäÿĂäÿłçL'źåőŽçŽĎæŁĂæIJŕïijŇä¡£åőČèČ¡åď'§éĂĆåžŤa
åŻäæ■ď'åőđéŹĚæČĚåĘţäÿ■éIJĂèęĄä¡ăèĞłåůšåŐżåŕÍèŕŢäÿ■åŘŇçŽĎæŰźæşŢåźűèŕĎäijřåőČãĂĆ

år¡çőąè£ŹäÿĂèŁĆéŻĘäÿ■èőĺèőžçŽĎæŸŕæŰĞæIJňïijŇä¡ĘæŸŕçśżäijijçŽĎæŁĂæIJŕäź§åŘŕäžěéĂĆçŤĺäž

4.13 2.13 å■Ůçňęäÿšåŕźé¡Ř

éŮőécŸ

ä¡ăæČşéĂŽè£Ğæ§Řçğ■åŕźé¡ŘæŰźåijŔæÍěæäijåijŔåŇŰå■Ůçňęäÿš

èğčåĘşæŰźæąĹ

åŕźäžŐå§žæIJňçŽĎå■Ůçňęäÿšåŕźé¡ŘæŞ■ä¡IJïijŇåŘŕäžěä¡£çŤĺå■ŮçňęäÿšçŽĎ `ljust()`
, `rjust()` åŠŇ `center()` æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> text = 'Hello World'
>>> text.ljust(20)
```

```
'Hello World          '
>>> text.rjust(20)
'          Hello World'
>>> text.center(20)
'    Hello World     '
>>>
```

æĽĂæIJL'è£ŹäžŻæŰźæşŢéČ¡èČ¡æŐěåŔŮäÿĂäÿłåŕŕéĂĽ'çŽĎåąńåĚĚå■Ůçňęã ĂĆæŕŤåęĆïijŽ

```
>>> text.rjust(20,'=')
'==========Hello World'
>>> text.center(20,'*')
'****Hello World*****'
>>>
```

åĞ¡æŢř format() åŘŇæăůåŔŕäžěçŤĺæİěå¿Ĺåőźæ ŸŞçŽĎåŕźé¡Řå■Ůçňęäÿšã ĂĆ ä¡ăèęĄåĄŽçŽĎåřśæŸŕä¡£çŤĺ <,> æĹŰèĂĚ ^ å■ŮçňęåŘŐéİćçť ğèů§äÿĂäÿłæŇĞåőŽçŽĎåő¡åžęã ĂĆæŕŤåęĆ

```
>>> format(text, '>20')
'          Hello World'
>>> format(text, '<20')
'Hello World          '
>>> format(text, '^20')
'    Hello World     '
>>>
```

åęĆæđIJä¡ăæČşæŇĞåőŽäÿĂäÿłéİđçl'žæăijçŽĎåąńåĚĚå■ŮçňęïijŇårĘåőČåĘŹåĹřåŕźé¡Řå■ŮçňęçŽĎåL'■

```
>>> format(text, '=>20s')
'==========Hello World'
>>> format(text, '*^20s')
'****Hello World*****'
>>>
```

å¡Şæăijåij ŔåŇŰåd'Žäÿłå Ăijç ŽĎæŮűåĂŹïijŇè£ŹäžŻæăijåijŔäžčçăĄäź§åŕŕäžěèćńçŤĺåIJĺ format() æŰźæşŢäÿ■ãĂĆæŕŤåęĆïijŽ

```
>>> '{:>10s} {:>10s}'.format('Hello', 'World')
'     Hello      World'
>>>
```

format() åĞ¡æŢřçŽĎäÿĂäÿłåě¡åd'ĎæŸŕåőČäÿ■äžĚéĂĆçŤĺäžŐå■Ůçňęäÿšã ĂĆåőČåŕŕäžěçŤĺæİěæăij æŕŤåęĆïijŇä¡ääŔŕäžěçŤĺåőČæİěæăijåijŔåŇŰæŢřå■ŮïijŽ

```
>>> x = 1.2345
>>> format(x, '>10')
'    1.2345'
>>> format(x, '^10.2f')
'   1.23   '
>>>
```

èőÍèőž

åIJÍèĂĄçŽĎäžčçăĄäÿ■ïijŇä¡ăçżŔåÿÿäijŽçIJŃåĹřèćńçŤÍæİěæäijåijŔåŇŰæŰĞæIJňçŽĎ % æŞ■ä¡IJçňęãĂĆæŕŤåęĆïijŽ

```
>>> '%-20s' % text
'Hello World          '
>>> '%20s' % text
'          Hello World'
>>>
```

ä¡ĘæŸŕïijŇåIJÍæŰřçL'ĹæIJňäžčçăĄäÿ■ïijŇä¡ăåžŤèŕěäijŸåĚĹéĂL'æŃl' format() åĞ¡æŢřæĹŰèĂĚæŰźæşŢãĂĆ format() èęĄæŕŤ % æŞ■ä¡IJçňęçŽĎåŁ§èČ¡æŻt'äÿžåijžåď'ğãĂĆ åźűäÿŤ format() äź§æŕŤä¡£çŤÍ ljust() , rjust() æĹŰ center() æŰźæşŢæŻt'éĂŽçŤĺïijŇ åŻääÿžåőČåŔŕäžěçŤÍæİěæäijåijŔåŇŰäżżæĎŔåŕźèśąïijŇèĂŇäÿ■äžĚäżĚæŸŕå■Ůçňęäÿšã ĂĆ

åęĆæđIJæČşèęĄåőŇåĚÍäžĘèğč format() åĞ¡æŢřçŽĎæIJL'çŤÍçL'źæĂğïijŇ èŕůåŔĆèĂČ åIJÍçž£PythonæŰĞæąč

4.14 2.14 åŘĹåźűæŃijæŐěå■Ůçňęäÿš

éŮőécŸ

ä¡ăæČşåřĘåĞääÿłåřŔçŽĎå■Ůçňęäÿšå ŘĹåźűäÿžäÿĂäÿłåď'ğçŽĎå■Ůçňęäÿš

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşèęĄåŘĹåźűçŽĎå■Ůçňęäÿšæ ŸŕåIJÍäÿĂäÿłåžŔåĹŮæĹŰèĂĚ iterable äÿ■ïijŇéĆčäźĹæIJĂå£nçŽĎæŰźåijŔåřśæŸŕä¡£çŤÍ join() æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> parts = ['Is', 'Chicago', 'Not', 'Chicago?']
>>> ' '.join(parts)
'Is Chicago Not Chicago?'
>>> ','.join(parts)
'Is,Chicago,Not,Chicago?'
>>> ''.join(parts)
'IsChicagoNotChicago?'
>>>
```

åĹİçIJŃètůæİěïijŇè£Źçğ■èŕ■æşŢçIJŃäÿŁåŐżäijŽæŕŤè¿ČæĂłïijŇä¡ĘæŸŕ join() èćńæŇĞåőŽäÿžå■Ůçňęäÿšç ŽĎäÿĂäÿłæŰźæşŢãĂĆ è£Źæăůå ĄŽçŽĎéČĺåĹĘåŐ§åŻăæŸŕä¡ăæČşåŐżè£đæŐěçŽĎåŕźèśąåŔŕèČ¡æİěèĞłåŘĎçğ■äÿ■åŘŇçŽĎæŢŕæ■ åęĆæđIJåIJÍæL'ĂæIJL'è£ŹäžŻåŕźèśąäÿŁéČ¡åőŽäźL'äÿĂäÿł join() æŰźæşŢæŸŐæŸ¿æŸŕåĘŮä¡ŹçŽĎãĂĆ åŻăæ■d'ä¡ăåŘłéIJĂèęĄæŇĞåőŽä¡ăæČşèęĄçŽĎåĹĘåL'šå■Ůçňęäÿšåź join() æŰźæşŢåŐżåřĘæŰĞæIJňçL'ĞæőţçżĎå ŘĹètůæİěãĂĆ

åęĆæđIJä¡ăäżĚäżĚåŘłæŸŕåŘĹåźűåřŚæŢřåĞääÿłå■ŮçňęäÿšïijŇä¡£çŤĺåŁăåŔů(+)éĂŽåÿÿåůšçżŔèűşåd'§ă

```
>>> a = 'Is Chicago'
>>> b = 'Not Chicago?'
>>> a + ' ' + b
'Is Chicago Not Chicago?'
>>>
```

åŁăåŔů(+)æŞ■ä¡IJçňęåIJĺä¡IJäÿžäÿĂäžŻåď■æİĆå■ŮçňęäÿšæăijåijŔåŇŰçŽĎæŻ£äżčæŰźæąĹçŽĎæŮűå

```
>>> print('{} {}'.format(a,b))
Is Chicago Not Chicago?
>>> print(a + ' ' + b)
Is Chicago Not Chicago?
>>>
```

åęĆæđIJä¡ăæČşåIJĺæžŘçăĄäÿ■åřĘäÿď äÿłå■Ůéİćå■ŮçňęäÿšåŘĹåźűèţůæİěïijŇä¡ăåŔłéIJĂèęĄçőĂå■ŢçŽ

```
>>> a = 'Hello' 'World'
>>> a
'HelloWorld'
>>>
```

èőĺèőž

å■ŮçňęäÿšåŘĹåźűåŔŕèČ¡çIJŃäÿŁåŐżåźűäÿ■éIJĂèęĄçŤĺäÿĂæŢť èŁĆæİěèőĺèőžãĂĆ
ä¡ĘæŸŕäÿ■åžŤèŕěåřŔçIJŃè£ŹäÿłéŮőéćŸïijŇçĺŃåžŔåŚŸéĂŽåÿÿåIJĺå■ŮçňęäÿšæăijåijŔåŇŰçŽĎæŮűåĂŹåŻ

æIJĂéĞ■èęĄçŽĎéIJĂèęĄåijŢèţůæşĺæĎŔçŽĎæŸŕïijŇå¡ŞæĹŚäżňä¡£çŤĺåŁăåŔů(+)æŞ■ä¡IJçňęåŐżè£đæ
åŻăäÿžåŁăåŔůè£đæŐěäijŽåijŢèţůåĘĚå■Ÿåď■åĹűäżěåŔŁåđČåIJ¿åŻđæŤűæŞ■ä¡IJãĂĆ
çĽźåĹńçŽĎïijŇä¡ăæřÿè£IJéČ¡äÿ■åžŤåČŔäÿŃéİćè£ŹæăůåĘŹå■Ůçňęäÿšè£đæŐěäżčçăĄïijŽ

```
s = ''
for p in parts:
    s += p
```

è£Źçğ■åĘŹæşŢäijŽæŕŤä¡£çŤĺ join() æŰźæşŢè£ŘèąŇçŽĎèęĄæĚćäÿĂäžŻïijŇåŻăäÿžæŕŔäÿĂæňąæĽ
ä¡ăæIJĂåě¡æŸŕåĚĹæŤűéŻĘæĽĂæIJĽçŽĎå■ŮçňęäÿšçĽĞæőţçĎűåŘŐåĘ■åřĘåőČäżňè£đæŐěèţůæİěãĂĆ

äÿĂäÿłçŻÿåŕźæŕŤè¿ČèĄłæŸŐçŽĎæŁĂåůğæŸŕåĹľçŤĺçŤ§æĹŘåŹĺèąĺè¿¿åijŔ(åŔĆèĂČ1.19åřŔèŁĆ)è¡ňa

```
>>> data = ['ACME', 50, 91.1]
>>> ','.join(str(d) for d in data)
'ACME,50,91.1'
>>>
```

åŘŇæăůè£Ÿå¿ŮæşĺæĎŔäÿ■å£ĚèęĄçŽĎå■Ůçňęäÿšè£đæŐěæŞ■ä¡IJãĂĆæIJĽæŮűåĂŹçĺŃåžŔåŚŸåIJĺæ

```
print(a + ':' + b + ':' + c) # Ugly
print(':'.join([a, b, c])) # Still ugly
print(a, b, c, sep=':') # Better
```

å¡Şæůůå ŘĹä¡£çŤÍI/OæŞ■ä¡IJåŠŇå■Ůçňęäÿšè£đæŐěæŞ■ä¡IJçŽĎæŮűůĂŹïijŇæIJL’æŮűůĂŹéIJÄèęĄäżŤ
ærŤåęĆïijŇèĂČèŹŚäÿŃéİćçŽĎäÿď’çńŕäžčçăĄçL’Ğæőţïij Ž

```
# Version 1 (string concatenation)
f.write(chunk1 + chunk2)

# Version 2 (separate I/O operations)
f.write(chunk1)
f.write(chunk2)
```

åęĆæđIJäÿď’äÿłå■Ůçňęäÿšå¿ĹåřŔïijŇéĆčäźĹçňňäÿĂäÿłçL’ĹæIJňæĂğèČ¡äijŽæŻť’åě¡äžŻïijŇåŻääÿžI/Oç
åŔęåď’Űäÿ ĂæŰźéİćïijŇåęĆæđIJäÿď’äÿłå■Ůçňęäÿšå¿Ĺåď’ğïijŇéĆčäźĹçňňäžŇäÿłçL’ĹæIJňåŔŕèČ¡äijŽæŻť’åŁä
åŻääÿžåőČéĄ£åĚ■äžĘåĹŻåžžäÿĂäÿłå¿Ĺåď’ğçŽĎäÿť’æŮűçżŞæđIJåźűäÿŤèęĄåď’■åĹűåď’ğéĞŔçŽĎåĘĚå■Ÿå
è£ŸæŸŕéĆčåŔěèŕÏïijŇæIJL’æŮűåĂŹæŸŕéIJÄèęĄæäźæ■őä¡äçŽĎåžŤçŤÍçÍŃåžŔçL’źçĆźæİěåĘşåőŽåžŤèŕěä¡£

æIJĂåŘŐèřĹäÿĂäÿŃïijŇåęĆæđIJä¡ăåĞĘåď’ĞçijŮåĘŹæđĎåžžåď’ğéĞŔåřŔå■Ůçňęäÿšç ŽĎè¿ŞåĞžäžčçăA
ä¡ăæIJĂåě¡èĂČèŹŚäÿŃä¡£çŤÍçŤ§æĹŘåŹÍåĞ¡æŢřïijŇåĹl’çŤÍyieldèŕ■åŔěäžğçŤ§è¿ŞåĞžçL’Ğæőţã ĆæŕŤåęĆ

```
def sample():
    yield 'Is'
    yield 'Chicago'
    yield 'Not'
    yield 'Chicago?'
```

è£ŹçğæŰźæşŢäÿĂäÿłæIJL’èűčçŽĎæŰźéİćæŸŕåőČåźűæšąæIJL’åŕźè¿ŞåĞžçL’Ğæőţå Ĺřåž Ţèę ĄæĂŐæăů
ä¿ŃåęĆïijŇä¡ăåŔŕäžěçőĂå■ŢçŽĎä¡£çŤÍ `join()` æŰźæşŢårĘè£ŹäžŻçL’Ğæőţå ŘĹåźűèţůæİěïijŽ

```
text = ''.join(sample())
```

æĹŰèĂĚä¡ääź§åŔŕäžěårĘå■Ůçňęäÿšç L’Ğæőţé Ğ■åőŽåŘŠåĹřI/Oïij Ž

```
for part in sample():
    f.write(part)
```

åĘ■æĹŰèĂĚä¡äè£ŸåŔŕäžěåĘŹåĞžäÿĂäžŻçżŞåŘĹI/OæŞ■ä¡IJçŽĎæůůåŘĹæŰźæąĹïijŽ

```
def combine(source, maxsize):
    parts = []
    size = 0
    for part in source:
        parts.append(part)
        size += len(part)
        if size > maxsize:
            yield ''.join(parts)
            parts = []
            size = 0
        yield ''.join(parts)

# çżŞåŘĹæŰĞäżűæŞ■ä¡IJ
with open('filename', 'w') as f:
    for part in combine(sample(), 32768):
        f.write(part)
```

è£ŹéĞŇçŽĎåĚşéŤőçĆźåIJÍäžŐåŐ§ågŃçŽĎçŤ§æĹŘåŹÍåĞ¡æŢřåźűäÿ■éIJĂèęAç§ěéĄŞä¡£çŤÍçżĘèŁĆïij

4.15 2.15 å■Ůçňęäÿšäÿ■æŔŠåĚěåŔŸéĞŔ

éŮőécŸ

ä¡ăæČşåĹŻåžžäÿĂäÿłåĘĚåţŇåŔŸéĞŔçŽĎå■ŮçňęäÿšïijŇåŔŸéĞŔèćńåőČçŽĎåĂijæĽĂèąÍçd'žçŽĎå■Ů

èğčåĘşæŰźæąĹ

PythonåźűæšąæIJL'åŕźåIJÍå■Ůçňęäÿšäÿ■çőĂå■ŢæŻ£æ■ćåŔŸéĞŔåĂijæŔŘä¿ŻçŻť'æŐěçŽĎæŤŕæŇĄãĂ
ä¡ĘæŸŕéĂŽè£Ğä¡£çŤÍå■Ůçňęäÿšç ŽĎ `format()` æŰźæşŢæİěèğčåĘşè£ŹäÿłéŮőécŸãĂĆærŤåęĆïijŽ

```
>>> s = '{name} has {n} messages.'
>>> s.format(name='Guido', n=37)
'Guido has 37 messages.'
>>>
```

æĹŰèĂĚïijŇåęĆæđIJèęĄèćńæŻ£æ■çŽĎåŔŸéĞŔèČ¡åIJÍåŔŸéĞŔå§§äÿ■æL'¿åĹřïijŇ
éĆčäźĹä¡ääŘŕäžěçżŞåŘĹä¡£çŤÍ `format_map()` åŠŇ `vars()`
ãĂĆåřśåČŘäÿŃéÍćè£ŹæăůïijŽ

```
>>> name = 'Guido'
>>> n = 37
>>> s.format_map(vars())
'Guido has 37 messages.'
>>>
```

`vars()` è£ŸæIJL'äÿĂäÿłæIJL'æĎŔæĂİçŽĎçL'źæĂğåřśæŸŕåőČäź§éĂĆçŤÍäžŐåŕźèśąåőđä¿ŃãĂĆærŤå

```
>>> class Info:
...     def __init__(self, name, n):
...         self.name = name
...         self.n = n
...
>>> a = Info('Guido',37)
>>> s.format_map(vars(a))
'Guido has 37 messages.'
>>>
```

`format` åŠŇ `format_map()` çŽĎäÿĂäÿłçijžéŹůåřśæŸŕåőČäžňåźűäÿ■èČ¡å¿Ĺåě¡çŽĎåd'ĎçŘĘåŔŸéĞ

```
>>> s.format(name='Guido')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'n'
>>>
```

äÿĂçğ■éĄ£åĚ■è£Źçğ■éŤŹèŕŕçŽĎæŰźæşŢæŸŕåŔęåď ŰåőŽäźĽäÿĂäÿłåŘńæIJĽ `__missing__()` æŰźæşŢçŽĎå■ŮåĚÿåŕźèśąïijŇåŕśåČŔäÿŃéİćè£ŹæăůïijŽ

```
class safesub(dict):
"""éŸšæ■ćkeyæĽ'¿äÿ■åĹř"""
def __missing__(self, key):
    return '{' + key + '}'
```

çŐřåIJĺä¡ăåŔŕäżěåĹľçŤĺè£ŹäÿłçśżåŇĚèčĚè¿ŞåĚěåŘŐäijăéĂŠçżŹ `format_map()` ïijŽ

```
>>> del n # Make sure n is undefined
>>> s.format_map(safesub(vars()))
'Guido has {n} messages.'
>>>
```

åęĆæđIJä¡ăåŔŚçŐřèĞłåůśåIJĺäżčçăĄäÿ■éćŚçźĄçŽĎæĽğèąŇè£ŹäžŻæ■ěéłď ïijŇä¡ăåŔŕäżěåřĘåŔŸéĞŔa

```
import sys

def sub(text):
    return text.format_map(safesub(sys._getframe(1).f_locals))
```

çŐřåIJĺä¡ăåŔŕäżěåČŔäÿŃéİćè£ŹæăůåĘŹäžĘïijŽ

```
>>> name = 'Guido'
>>> n = 37
>>> print(sub('Hello {name}'))
Hello Guido
>>> print(sub('You have {n} messages.'))
You have 37 messages.
>>> print(sub('Your favorite color is {color}'))
Your favorite color is {color}
>>>
```

èőĺèőž

åď Žåźť äżěæİěçŤśäžŐPythonçijžäźŔåŕźåŔŸéĞŔæŻ£æ■ćçŽĎåĘĚç¡őæŤŕæŇĄèĂŇåŕijèĞť äžĘåŘĎçğ■äÿ
ä¡IJäÿžæIJňèŁĆäÿ■åśŢçď žçŽĎäÿĂäÿłåŔŕèČ¡çŽĎèğčåĘşæŰźæąĹïijŇä¡ăåŔŕäżěæIJĽæŮűåĂŹäijŽçIJŃåĹřåČ

```
>>> name = 'Guido'
>>> n = 37
>>> '%(name) has %(n) messages.' % vars()
'Guido has 37 messages.'
>>>
```

ä¡ăåŔŕèČ¡è£ŸäijŽçIJŃåĹřå■Ůçňęäÿšæĺąæİ£çŽĎä¡£çŤĺïijŽ

```
>>> import string
>>> s = string.Template('$name has $n messages.')
>>> s.substitute(vars())
'Guido has 37 messages.'
>>>
```

çĎűèĂŇïijŇ format() åŠŇ format_map() çŻÿæŕŤè¿ČäÿŁéİćè£ŹäžŻæŰźæąĹèĂŇåůšæŻť åŁäåĚĹ
ä¡£çŤĺ format() æŰźæşŢè£ŸæIJĽäÿĂäÿłåě¡åď ĎåŕśæŸŕä¡ăåŔŕäżěèŐůå¿Ůåŕźå■ŮçňęäÿšæäijåijŔåŇŰçŽĎ
èĂŇè£ŹäžŻçĽźæĂğæŸŕä¡£çŤĺåČŔæĺąæİ£å■ŮçňęäÿšäźŃçśżçŽĎæŰźæąĹäÿ■åŔŕèČ¡èŐůå¿ŮçŽĎãĂĆ

æIJňæIJžè£ŸéČĺåĹĘäżŃçż■äžĘäÿĂäžŻéńŸçžğçĽźæĂğãĂĆæŸäåřĎæĹŰèĂĚå■ŮåĚÿçśżäÿ■éšIJäÿžäžž
__missing__() æŰźæşŢåŔŕäżěèőľä¡ăåőŽäźĽåęĆä¡Ţåď ĎçŘĘçijžåď śçŽĎåĂijãĂĆ åIJĺ
SafeSub çśżäÿ■ïijŇè£ŹäÿłæŰźæşŢèćńåőŽäźĽäÿžåŕźçijžåď śçŽĎåĂijè£ŤåŻđäÿĂäÿłå■ăä¡■çňęãĂĆ
ä¡ăåŔŕäżěåŔŚçŐřçijžåď śçŽĎåĂijäijŽåĞžçŐřåIJĺçżŞæđIJå■Ůçňęäÿšäÿ■(åIJĺèřČèŕŢçŽĎæŮűåĂŹåŔŕèČ¡å¿Ĺa
KeyError åijĆåÿÿãĂĆ

sub() åĞ¡æŢřä¡£çŤĺ sys._getframe(1) è£ŤåŻđèřČçŤĺèĂĚçŽĎæăĹåÿğãĂĆåŔŕäżěäżŐäÿ■èő£éŮ
f_locals æİěèŐůå¿ŮåśĂéČĺåŔŸéĞŔãĂĆ æŕńæŮăçŰŚéŮőçżİåď ğéČĺåĹĘæČĚåĘţäÿŃåIJĺäżčçăĄäÿ■åŐżç
ä¡ĘæŸŕïijŇåŕźäžŐåČŔå■ŮçňęäÿšæŻ£æ■ćåůěåĚůåĞ¡æŢřèĂŇèĺĂåőČæŸŕéİđåÿÿæIJĽçŤĺçŽĎãĂĆ
åŔęåď ŰïijŇåĂijå¿ŮæşĺæĎŔçŽĎæŸŕ f_locals æŸŕäÿĂäÿłåď■åĹűèřČçŤĺåĞ¡æŢřçŽĎæIJňåIJřåŔŸéĞŔçŽ
åř¡çőąä¡ăåŔŕäżěæŤźåŔŸ f_locals çŽĎåĘĚåőźïijŇä¡ĘæŸŕè£Źäÿłä£őæŤźåŕźäžŐåŘŐéİćçŽĎåŔŸéĞŔèő£é
æĽĂäżěïijŇèŹ¡èŕť èő£éŮőäÿĂäÿłæăĹåÿğçIJŃäÿŁåŐżå¿ĹéĆłæĄűïijŇä¡ĘæŸŕåŕźåőČçŽĎäżżä¡ŢæŞ■ä¡IJäÿ■ä

4.16 2.16 äżěæŇĞåőŽåĹŮåő¡æăijåijŔåŇŰå■Ůçňęäÿš

éŮőéćŸ

ä¡ăæIJĽäÿĂäžŻéŢ£å■ŮçňęäÿšïijŇæČşäżěæŇĞåőŽçŽĎåĹŮåő¡åřĘåőČäżňéĞ■æŰřæăijåijŔåŇŰãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺ textwrap æĺąåİŮæİěæăijåijŔåŇŰå■ŮçňęäÿšçŽĎè¿ŞåĞžãĂĆæŕŤåęĆïijŇåĄĞåęĆä¡ăæIJĽäÿŃå

```
s = "Look into my eyes, look into my eyes, the eyes, the eyes, \
the eyes, not around the eyes, don't look around the eyes, \
look into my eyes, you're under."
```

äÿŃéİćæijŤçď žä¡£çŤĺ textwrap æăijåijŔåŇŰå■ŮçňęäÿšçŽĎåď Žçğ■æŰźåijŔïijŽ

```
>>> import textwrap
>>> print(textwrap.fill(s, 70))
Look into my eyes, look into my eyes, the eyes, the eyes, the eyes,
not around the eyes, don't look around the eyes, look into my eyes,
you're under.

>>> print(textwrap.fill(s, 40))
Look into my eyes, look into my eyes,
the eyes, the eyes, the eyes, not around
```

```
the eyes, don't look around the eyes,
look into my eyes, you're under.

>>> print(textwrap.fill(s, 40, initial_indent='     '))
    Look into my eyes, look into my
eyes, the eyes, the eyes, the eyes, not
around the eyes, don't look around the
eyes, look into my eyes, you're under.

>>> print(textwrap.fill(s, 40, subsequent_indent='     '))
Look into my eyes, look into my eyes,
    the eyes, the eyes, the eyes, not
    around the eyes, don't look around
    the eyes, look into my eyes, you're
    under.
```

èőÍèőž

textwrap æÍąåİŮåŕźäžŐå■ŮçňęäÿšæĽŞå■řæŸŕéİđåÿÿæIJĽçŤÍçŽĎïijŇçĽźåĹńæŸŕå¡Şä¡ăåÿŇæIJŻè¿Ş
ä¡ăåŔŕäžěä¡£çŤÍ os.get_terminal_size() æŰźæşŢæİèèŐůåŔŰçżĹçńŕçŽĎåď ğářŔářžåŕÿãĂĆærŤåęĆ

```
>>> import os
>>> os.get_terminal_size().columns
80
>>>
```

fill() æŰźæşŢæŐěåŔŮäÿĂäžŻåĚűäžŰåŔŕéĂĽåŔĆæŢřæİěæŐğåĹűtabïijŇèŕ■åŔěçżŞář¿ç■Ľ ãĂĆ
åŔĆéŸĚ textwrap.TextWrapperæŰĞæąč èŐůåŔŰæŻť åď ŽåĘĚåőźãĂĆ

4.17 2.17 åIJĺå■Ůçňęäÿšäÿ■åď ĎçŘĘhtmlåŠŇxml

éŮőécŸ

ä¡ăæČşåřĘHTMLæĹŰèĂĚXMLåőđä¡ŞåęĆ &entity; æĹŰ &#code; æŻ£æ■ćäÿžåŕźåžŢçŽĎæŰĞæIJñãĂĆ åĘ■èĂĚïijŇä¡ăéIJĂèęĄè¡ňæ■ćæŰĞæIJñäÿ■çĽźåőŽçŽĎå■Ůçňę(æŕŤåę
>, æĹŰ &)ãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşæŻ£æ■ćæŰĞæIJňå■Ůçňęäÿšäÿ■çŽĎ âĂŸ<âĂŹ æĹŰèĂĚ âĂŸ>âĂŹ
ïijŇä¡£çŤÍ html.escape() åĞ¡æŢřåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåőŇæĹŘãĂĆærŤåęĆïijŽ

```
>>> s = 'Elements are written as "<tag>text</tag>".'
>>> import html
>>> print(s)
Elements are written as "<tag>text</tag>".
```

```
>>> print(html.escape(s))
Elements are written as &quot;&lt;tag&gt;text&lt;/tag&gt;&quot;.

>>> # Disable escaping of quotes
>>> print(html.escape(s, quote=False))
Elements are written as "&lt;tag&gt;text&lt;/tag&gt;".
>>>
```

åęĆæđIJä¡ăæ■čåIJĺåď ĎçŘĘçŽĎæŸŕASCIIæŰĞæIJňïijŇåźűäÿŤæČşåřĘéİđASCIIæŰĞæIJňåŕźåžŤçŽĎç
åŔŕäžěçżŹæ§ŘäžŻI/OåĞ¡æŢřäijăéĂŠåŔĆæŢř `errors='xmlcharrefreplace'`
æİěè¿¿åĹřè£ŹäÿłçŻőãĂĆæŕŤåęĆïijŽ

```
>>> s = 'Spicy JalapeÃśo'
>>> s.encode('ascii', errors='xmlcharrefreplace')
b'Spicy Jalape&#241;o'
>>>
```

äÿžäžĘæŻ£æ■ćæŰĞæIJňäÿ■çŽĎçijŰçăĄåőđä¡ŞïijŇä¡ăéIJĂèęĄä¡£çŤĺåŔęåď ŰäÿĂçğ■æŰźæşŢãĂĆ
åęĆæđIJä¡ăæ■čåIJĺåď ĎçŘĘHTMLæĹŰèĂĚXMLæŰĞæIJňïijŇèŕŢçİĂåĚĹä¡£çŤĺäÿĂäÿłåŘĹéĂĆçŽĎHTML
éĂŽåÿÿæČĚåĘţäÿŃïijŇè£ŹäžŻåůěåĚůäijŽèĞłåŁĺæŻ£æ■ćè£ŹäžŻçijŰçăĄåĂijïijŇä¡ăæŮăéIJĂæŃĚå£ČãĂĆ

æIJL'æŮűåĂŹïijŇåęĆæđIJä¡ăæŐěæŤűåĹřäžĘäÿĂäžŻåŘńæIJL'çijŰçăĄåĂijçŽĎåŐ§ågŃæŰĞæIJňïijŇél
éĂŽåÿÿä¡ăåŘłéIJĂèęĄä¡£çŤÍHTMLæĹŰèĂĚXMLèğčæđŘåŹÍçŽĎäÿĂäžŻçŻÿåĚşåůěåĚůåĞ¡æŢř/æŰźæşŢå■

```
>>> s = 'Spicy &quot;Jalape&#241;o&quot.'
>>> from html.parser import HTMLParser
>>> p = HTMLParser()
>>> p.unescape(s)
'Spicy "JalapeÃśo".'
>>>
>>> t = 'The prompt is &gt;&gt;&gt;'
>>> from xml.sax.saxutils import unescape
>>> unescape(t)
'The prompt is >>>'
>>>
```

èőÍèőž

åIJÍçŤ§æĹŘHTMLæĹŰèĂĚXMLæŰĞæIJňçŽĎæŮűåĂŹïijŇåęĆæđIJæ■čçąőçŽĎè¡ñæ■ćçL'źæőŁæăĞèć
çL'źåĹńæŸŕå¡Şä¡ăä¡£çŤÍ `print()` åĞ¡æŢřæĹŰèĂĚåĚűäžŰå■ŮçňęäÿšæăijåijŘåŇŰæİěäžğçŤ§è¿ŞåĞžçŽĎa
ä¡£çŤĺåČŔ `html.escape()` çŽĎåůěåĚůåĞ¡æŢřåŕäžěå¿Ĺåőźæ ŸŞçŽĎèğčåĘşè£ŹçśźéŰőécŸãĂĆ

åęĆæđIJä¡ăæČşäžěåĚűäžŰæŰźåijŔåď ĎçŘĘæŰĞæIJňïijŇè£ŸæIJL'äÿĂäžŻåĚűäžŰçŽĎåůěåĚůåĞ¡æŢřa
`xml.sax.saxutils.unescape()` åŔŕäžěåÿőåŁľä¡ăãĂĆ
çĎűěĂŇïijŇä¡ăåžŤèŕěåĚĹèŕČçăŤæÿĚæěŽæĂŐæăůä¡£çŤĺäÿĂäÿłåŘĹéĂĆçŽĎèğčæđŘåŹĺãĂĆ
æŕŤåęĆïijŇåęĆæđIJä¡ăåIJĺåď ĎçŘĘHTMLæĹŰXMLæŰĞæIJňïijŇ
ä¡£çŤĺæ§ŘäÿłèğčæđŘæĺąåİŮæŕŤåęĆ `html.parse` æĹŰ `xml.etree.ElementTree`
åůšçżŔåÿőä¡ăèĞłåŁĺåď ĎçŘĘäžĘçŻÿåĚşçŽĎæŻ£æ■ćçżĘèŁĆãĂĆ

4.18 2.18 å■Ůçňęäÿšäžď'çĽŇèğčæđŘ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłå■ŮçňęäÿšïijŇæČşäžŐåůęèĞşåŘşåřĘåĚűèğčæđŘäÿžäÿĂäÿłäžď'çL'ŇæţĄãĂĆ

èğčåĘşæŰźæąĹ

åĄĞåęĆä¡ăæIJL'äÿŃéÍćè£ŹæăůäÿĂäÿłæŰĞæIJňå■ŮçňęäÿšïijŽ

```
text = 'foo = 23 + 42 * 10'
```

äÿžäžĘäžď'çĽŇåŇŰå■ŮçňęäÿšïijŇä¡ääÿ■äžĚéIJÄèęĄåŇźéĚ■æĺąåijŔïijŇè£Ÿå¿ŮæŇĞåőŽæĺąåijŔçŽĎçæŕŤåęĆïijŇä¡ääŘŕèČ¡æČşåřĘå■ŮçňęäÿšåČŔäÿŃéÍćè£ŹæăůèĄñæ■ćäÿžåžŔåĹŮåŕźïijŽ

```
tokens = [('NAME', 'foo'), ('EQ','='), ('NUM', '23'), ('PLUS','+'),
          ('NUM', '42'), ('TIMES', '*'), ('NUM', '10')]
```

äÿžäžĘæL'ğèąŇè£ŹæăůçŽĎåĹĞåĹĘïijŇçňňäÿĂæ■ěåřśæŸŕåČŔäÿŃéÍćè£ŹæăůåĹl'çŤĺåŠ¡åŘ■æ■ŢèŐůçż

```
import re
NAME = r'(?P<NAME>[a-zA-Z_][a-zA-Z_0-9]*)'
NUM = r'(?P<NUM>\d+)'
PLUS = r'(?P<PLUS>\+)'
TIMES = r'(?P<TIMES>\*)'
EQ = r'(?P<EQ>=)'
WS = r'(?P<WS>\s+)'

master_pat = re.compile('|'.join([NAME, NUM, PLUS, TIMES, EQ, WS]))
```

åIJÍäÿŁéÍćçŽĎæĺąåijŔäÿ■ïijŇ `?P<TOKENNAME>` çŤĺäžŐçżŹäÿĂäÿłæĺąåijŔåŚ¡åŘ■ïijŇä¿ŻåŘŐéÍćä¡£çŤ

äÿŃäÿĂæ■ěïijŇäÿžäžĘäžď'çL'ŇåŇŰïijŇä¡£çŤĺæĺąåijŔåŕźèśąå¿ĹåřŚèćńäžžç§ěéĄŞçŽĎ `scanner()` æŰźæşŢãĂĆ è£ŹäÿłæŰźæşŢäijŽåĹŻåžžäÿĂäÿł `scanner` åŕźèśąïijŇ åIJÍè£ŹäÿłåŕźèśąäÿŁäÿ■æŰ■çŽĎèřČçŤÍ `match()` æŰźæşŢäijŽäÿĂæ■ěæ■ěçŽĎæL'ńæŔŘçŻőæăĞæŰŮĞæIJňïijŇærŔæ■ěäÿĂäÿłåŇźéĚ■ãĂĆ äÿŃéÍćæŸŕæijŤçď'žäÿĂäÿł `scanner` åŕźèśąåęĆä¡Ţåůěä¡IJçŽĎäžď'äžŠåijŔä¿Ńå■ŘïijŽ

```
>>> scanner = master_pat.scanner('foo = 42')
>>> scanner.match()
<_sre.SRE_Match object at 0x100677738>
>>> _.lastgroup, _.group()
('NAME', 'foo')
>>> scanner.match()
<_sre.SRE_Match object at 0x100677738>
>>> _.lastgroup, _.group()
('WS', ' ')
>>> scanner.match()
<_sre.SRE_Match object at 0x100677738>
```

```
>>> _.lastgroup, _.group()
('EQ', '=')
>>> scanner.match()
<_sre.SRE_Match object at 0x100677738>
>>> _.lastgroup, _.group()
('WS', ' ')
>>> scanner.match()
<_sre.SRE_Match object at 0x100677738>
>>> _.lastgroup, _.group()
('NUM', '42')
>>> scanner.match()
>>>
```

åőđéŹĚä¡£çŤÍè£Źçğ■æŁĂæIJŕçŽĎæŮűåĂŹïijŇåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåČŔäÿŃéİćè£ŹæăůåřĘäÿŁè£ŕäž

```
def generate_tokens(pat, text):
    Token = namedtuple('Token', ['type', 'value'])
    scanner = pat.scanner(text)
    for m in iter(scanner.match, None):
        yield Token(m.lastgroup, m.group())

# Example use
for tok in generate_tokens(master_pat, 'foo = 42'):
    print(tok)
# Produces output
# Token(type='NAME', value='foo')
# Token(type='WS', value=' ')
# Token(type='EQ', value='=')
# Token(type='WS', value=' ')
# Token(type='NUM', value='42')
```

åęĆæđIJä¡ăæČşè£ĞæżđʼäżđʼçĽŇæţĄïijŇä¡ăåŔŕäžěåőŽäźĽæŻtʼåđʼŽçŽĎçŤ§æĹŘåŹĺåĞ¡æŢřæĹŰèĂĚä¡æŕŤåęĆïijŇäÿŃéİćæijŤçđʼžæĂŐæăůè£ĞæżđʼæĽĂæIJĽçŽĎçlʼžçŹ¡äżđʼçĽŇïijŽ

```
tokens = (tok for tok in generate_tokens(master_pat, text)
          if tok.type != 'WS')
for tok in tokens:
    print(tok)
```

èőÍèőž

éĂŽåÿÿæİěèőšäžđʼçĽŇåŇŰæŸŕå¿ĹåđʼŽéńŸçžğæŰĞæIJňèğčæđŘäÿŐåđʼĎçŘĘçŽĎçňňäÿĂæ■ěãĂĆ äÿžäžĘä¡£çŤÍäÿŁéİćçŽĎæĽńæŔŘæŰźæşŢïijŇä¡ăéIJĂèęĄèőřä¡Ŕè£ŹéĞŇäÿĂäžŻéĞ■èęĄçŽĎåĞäçĆźãĂĆ çňňäÿĂçĆźåřśæŸŕä¡ăå£Ěéążçąõèőđʼä¡ăä¡£çŤÍæ■čåĹŹèąÍè¿¿åijŔæŇĞåőŽäžĘæĽĂæIJĽè¿ŞåĚěäÿ■åŔŕèČ¡åĞ åęĆæđIJæIJĽäżżä¡Ţäÿ■åŔŕåŇźéĚ■çŽĎæŰĞæIJňåĞžçŐřäžĘïijŇæĽńæŔŘåřśäijŽçŻtʼæŐěåĄIJæ■ćãĂĆè£Źä

äžđʼçĽŇçŽĎéąžåžŔäź§æŸŕæIJĽå¡śåŞ■çŽĎãĂĆ `re` æĺąåİŮäijŽæŇĽʼçĚğæŇĞåőŽåě¡çŽĎéąžåžŔåŐżåĄ åŻăæ■đʼïijŇåęĆæđIJäÿĂäÿłæĺąåijŔæĄřåě¡æŸŕåŔęäÿĂäÿłæŻtʼéŢ£æĺąåijŔçŽĎå■Řå■ŮçňęäÿšïijŇéĆčäźĹä¡ăé

```
LT = r'(?P<LT><)'
LE = r'(?P<LE><=)'
EQ = r'(?P<EQ>=)'

master_pat = re.compile('|'.join([LE, LT, EQ])) # Correct
# master_pat = re.compile('|'.join([LT, LE, EQ])) # Incorrect
```

çňňäžŇäýłæĺąåijŕæŸŕéŤŹçŽĎïijŇåŻääýžåőČäijŽåřĘæŰĞæIJň<=åŇźéĚ■äýžäžd’çL’ŇLTçt’ğèů§çİĂEQi

æIJĂåŘŐïijŇä¡ăéIJĂèęĄçŢŹæĎŔäýŃå■Řå■Ůçňęäýšå¡ćåijŔçŽĎæĺąåijŔãĂĆæŕŤåęĆïijŇåĄĞèő¿ä¡ăæIJI

```
PRINT = r'(?P<PRINT>print)'
NAME = r'(?P<NAME>[a-zA-Z_][a-zA-Z_0-9]*)'

master_pat = re.compile('|'.join([PRINT, NAME]))

for tok in generate_tokens(master_pat, 'printer'):
    print(tok)

# Outputs :
# Token(type='PRINT', value='print')
# Token(type='NAME', value='er')
```

åĚşäžŐæŻt’énŸéŸűçŽĎäžd’çL’ŇåŇŰæŁĂæIJŕïijŇä¡ăåŔŕèČ¡éIJĂèęĄæ§ěçIJŃ PyParsing æĹÚèĂĚ PLY åŇĚãĂĆ äýĂäýłèřČçŤÍPLYçŽĎä¿Ńå■ŘåIJĺäýŃäýĂèŁĆäijŽæIJL’æijŤçd’žãĂĆ

4.19 2.19 åőđçŐřäýĂäýłçőĂå■ŢçŽĎéĂŠå¡ŠäýŃéŹ■åĹĘæđŘåŹÍ

éŮőécŸ

ä¡ăæČşæăźæ■őäýĂçżĎèŕ■æşŢèğĎåĹŹèğčæđŘæŰĞæIJňåźűæL’ğèąŇåŚ¡äžd’ïijŇæĹŰèĂĚæđĎéĂăäýĂäj
åęĆæđIJèŕ■æşŢéİđåÿÿçőĂå■ŢïijŇä¡ăåŔŕäžěèĞłåůśåĘŹè£ŹäýłèğčæđŘåŹĺïijŇèĂŇäÿ■æŸŕä¡£çŤĺäýĂäžŻæąĘ

èğčåĘşæŰźæąĹ

åIJĺè£ŹäýłéŮőécŸäÿ■ïijŇæĹŚäžňéŻĘäÿ■èőĺèőžæăźæ■őçL’źæőŁèŕ■æşŢåŐżèğčæđŘæŰĞæIJňçŽĎéŮőé
äÿžäžĘè£Źæăůå ĄŽïijŇä¡ăéęŰåĚĹèęĄäžěBNFæĹŰèĂĚEBNFå¡ćåijŔæŇĞåőŽäÿĂäýłæăĞåĞĘèŕ■æşŢãĂĆ
æŕŤåęĆïijŇäÿĂäýłçőĂå■ŢæŢŕå■ęèąĺè¿¿åijŔèŕ■æşŢåŔŕèČ¡åČŔäýŃéİćè£Źæăůïijž

```
expr ::= expr + term
    |    expr - term
    |    term

term ::= term * factor
    |    term / factor
    |    factor
```

```
factor ::= ( expr )
    |   NUM
```

æĹŰèĂĚïijÑäžěEBNFå¡ćåijŔïijŽ

```
expr ::= term { (+|-) term }*

term ::= factor { (*|/) factor }*

factor ::= ( expr )
    |   NUM
```

åIJÍEBNFäÿ■ïijÑèćńåŇĚåŘńåIJÍ { . . . }* äÿ■çŽĎèğĎåĹŹæŸŕåŘŕéĂL'çŽĎãĂĆ*äžčèąĺ0æňąæĹŰåď'Žæ

çŐřåIJÍïijÑåęĆæđIJä¡ăåŕźBNFçŽĎåůěä¡IJæIJžåĹűè£Ÿäÿ■æŸŕå¿ĹæŸŐçŹ¡çŽĎèŕİïijÑåřśæŁŁåőČå¡ŞåĄ
äÿĂèĹňæİěëőšïijÑèğčæđŘçŽĎåŐ§çŘĘåřśæŸŕä¡ääĹl'çŤÍBNFåőŇæĹŘåď'Žäÿ łæŻ£æ■ćåŠŇæL'l'åśŢäžěåŇźéĚ
äÿžäžĘæijŤçď'žïijÑåĄĞèő¿ä¡ăæ■čåIJÍèğčæđŘå¡ćåęĆ 3 + 4 * 5 çŽĎèąĺè¿¿åijŔãĂĆ
è£Źäÿłèąĺè¿¿åijŔåĚĹèęĄéĂŽè£Ğä¡£çŤÍ2.18èŁĆäÿ■äžŃçż■çŽĎæŁĂæIJŕåĹĘèğčäÿžäÿĂçżĎäžď'çL'ŇæţĄãĂ
çżŞæđIJåŘŕèČ¡æŸŕåČŔäÿŇåĹŮè£ŹæăůçŽĎäžď'çL'ŇåžŘåĹŮïijŽ

```
NUM + NUM * NUM
```

åIJÍæ■d'å§žçąĂäÿŁïijÑ èğčæđŘåŁÍä¡IJäijŽèŕŢçİĂåŐžéĂŽè£ĞæŻ£æ■ćæŞ■ä¡IJåŇźéĚ■èŕ■æşŢåĹřè¿ŞåĚ

```
expr
expr ::= term { (+|-) term }*
expr ::= factor { (*|/) factor }* { (+|-) term }*
expr ::= NUM { (*|/) factor }* { (+|-) term }*
expr ::= NUM { (+|-) term }*
expr ::= NUM + term { (+|-) term }*
expr ::= NUM + factor { (*|/) factor }* { (+|-) term }*
expr ::= NUM + NUM { (*|/) factor}* { (+|-) term }*
expr ::= NUM + NUM * factor { (*|/) factor }* { (+|-) term }*
expr ::= NUM + NUM * NUM { (*|/) factor }* { (+|-) term }*
expr ::= NUM + NUM * NUM { (+|-) term }*
expr ::= NUM + NUM * NUM
```

äÿŇéİćæL'ĂæIJL'çŽĎèğčæđŘæ■ěéłď'åŘŕèČ¡éIJĂèęĄèŁśçĆźæŮűéŮt'åijĎæŸŐçŹ¡ïijÑä¡ĘæŸŕåőČäžňåĆ
çňňäÿĂäÿłè¿ŞåĚěäžď'çL'ŇæŸŕfNUMïijÑåŽăæ■d'æŻ£æ■ćéęŰåĚĹäijŽåŇźéĚ■éĆčäÿłéČĺåĹĘãĂĆ
äÿĂæŮęåŇźéĚ■æĹŘåŁ§ïijÑåřśäijŽè£ŽåĚěäÿŇäÿĂäÿłäžď'çL'Ň+ïijÑäžěæ■d'çśźæŐĺãĂĆ
å¡ŞåůšçżŕçąőåőŽäÿ■èČ¡åŇźéĚ■äÿŇäÿĂäÿłäžď'çL'ŇçŽĎæŮűåĂŹïijÑåŘşè¿źçŽĎéČĺåĹĘ(æŕŤåęĆ
{ (*/) factor }*)åřśäijŽèćńæÿĚçŘĘæŐL'ãĂĆ åIJĺäÿĂäÿłæĹŘåŁ§çŽĎèğčæđŘäÿ■ïijÑæŢt'äÿłåŘşè¿źé

æIJL'äžĘåL'■éİćçŽĎç§ěèŕĘèČŇæŹŕïijÑäÿŇéİćæĹŚäžňäÿ¿äÿĂäÿłçőĂå■Ţçď'žä¿ŇæİěåśŢçď'žåęĆä¡Ţæđĺ

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
"""
Topic: äÿŃéŹ■èğčæđŘåŹĺ
Desc :
"""
```

```
import re
import collections

# Token specification
NUM = r'(?P<NUM>\d+)'
PLUS = r'(?P<PLUS>\+)'
MINUS = r'(?P<MINUS>-)'
TIMES = r'(?P<TIMES>\*)'
DIVIDE = r'(?P<DIVIDE>/)'
LPAREN = r'(?P<LPAREN>\()'
RPAREN = r'(?P<RPAREN>\))'
WS = r'(?P<WS>\s+)'

master_pat = re.compile('|'.join([NUM, PLUS, MINUS, TIMES,
                                  DIVIDE, LPAREN, RPAREN, WS]))
# Tokenizer
Token = collections.namedtuple('Token', ['type', 'value'])


def generate_tokens(text):
    scanner = master_pat.scanner(text)
    for m in iter(scanner.match, None):
        tok = Token(m.lastgroup, m.group())
        if tok.type != 'WS':
            yield tok


# Parser
class ExpressionEvaluator:
    '''
    Implementation of a recursive descent parser. Each method
    implements a single grammar rule. Use the ._accept() method
    to test and accept the current lookahead token. Use the ._
↪expect()
    method to exactly match and discard the next token on on the␣
↪input
    (or raise a SyntaxError if it doesn't match).
    '''

    def parse(self, text):
        self.tokens = generate_tokens(text)
        self.tok = None  # Last symbol consumed
        self.nexttok = None  # Next symbol tokenized
        self._advance()  # Load first lookahead token
        return self.expr()

    def _advance(self):
        'Advance one token ahead'
        self.tok, self.nexttok = self.nexttok, next(self.tokens,␣
↪None)
```

```
    def _accept(self, toktype):
        'Test and consume the next token if it matches toktype'
        if self.nexttok and self.nexttok.type == toktype:
            self._advance()
            return True
        else:
            return False

    def _expect(self, toktype):
        'Consume next token if it matches toktype or raise␣
↪SyntaxError'
        if not self._accept(toktype):
            raise SyntaxError('Expected ' + toktype)

    # Grammar rules follow
    def expr(self):
        "expression ::= term { ('+'|'-') term }*"
        exprval = self.term()
        while self._accept('PLUS') or self._accept('MINUS'):
            op = self.tok.type
            right = self.term()
            if op == 'PLUS':
                exprval += right
            elif op == 'MINUS':
                exprval -= right
        return exprval

    def term(self):
        "term ::= factor { ('*'|'/') factor }*"
        termval = self.factor()
        while self._accept('TIMES') or self._accept('DIVIDE'):
            op = self.tok.type
            right = self.factor()
            if op == 'TIMES':
                termval *= right
            elif op == 'DIVIDE':
                termval /= right
        return termval

    def factor(self):
        "factor ::= NUM | ( expr )"
        if self._accept('NUM'):
            return int(self.tok.value)
        elif self._accept('LPAREN'):
            exprval = self.expr()
            self._expect('RPAREN')
            return exprval
        else:
            raise SyntaxError('Expected NUMBER or LPAREN')
```

```
def descent_parser():
    e = ExpressionEvaluator()
    print(e.parse('2'))
    print(e.parse('2 + 3'))
    print(e.parse('2 + 3 * 4'))
    print(e.parse('2 + (3 + 4) * 5'))
    # print(e.parse('2 + (3 + * 4)'))
    # Traceback (most recent call last):
    #   File "<stdin>", line 1, in <module>
    #   File "exprparse.py", line 40, in parse
    #   return self.expr()
    #   File "exprparse.py", line 67, in expr
    #   right = self.term()
    #   File "exprparse.py", line 77, in term
    #   termval = self.factor()
    #   File "exprparse.py", line 93, in factor
    #   exprval = self.expr()
    #   File "exprparse.py", line 67, in expr
    #   right = self.term()
    #   File "exprparse.py", line 77, in term
    #   termval = self.factor()
    #   File "exprparse.py", line 97, in factor
    #   raise SyntaxError("Expected NUMBER or LPAREN")
    #   SyntaxError: Expected NUMBER or LPAREN


if __name__ == '__main__':
    descent_parser()
```

èőÍèőž

æŰĞæIJňèğčæđŘæŸŕäÿĂäÿłå¿Ĺåď ğçŽĎäÿžéćŸïijŇ äÿĂèĹňäijŽå■ăçŤĺå■ęçŤ§å■ęäźăçijŰèŕŚèŕ¿çÍŃæŮ
åęĆæđIJä¡ăåIJÍæL’¿åŕżåĚşäžŐèŕ■æşŢïijŇèğčæđŘçőŮæşŢç■L’çŻÿåĚşçŽĎèČŇæŹŕç§ěèŕEçŽĎèŕÍïijŇä¡ăåžŤèŕ
å¿ĹæŸ¿çĎűïijŇåĚşäžŐè£ŹæŰźéÍćçŽĎåĘĚåőźåď łåď ŽïijŇäÿ■åŘŕèČ¡åIJÍè£ŹéĞŇåĚĺéČĺĺśŢåijĂãĂĆ

åř¡çőąåęĆæ■d’ïijŇçijŰåĘŹäÿĂäÿłéĂŠå¡ŠäÿŃéŹ■èğčæđŘåŹÍçŽĎæŢť ä¡ŞæĂÌèůŕæŸŕæŕŤè¿ČçőĂå■ŢçŽ
åijĂåğŃçŽĎæŮűůåĂŹïijŇä¡ăåĚĹèŐůå¿ŮæL’ĂæIJL’çŽĎèŕ■æşŢèğĎåĹŹïijŇçĎűåŘŐářĘåĚűè¡ňæ■ćäÿžäÿĂäÿ
åŽăæ■d’åęĆæđIJä¡ăçŽĎèŕ■æşŢçśżäijijè£Źæăůïijž

```
expr ::= term { ('+'|'-') term }*

term ::= factor { ('*'|'/') factor }*

factor ::= '(' expr ')'
    | NUM
```

ä¡ăåžŤèŕěéęŰåĚĹářĘåőČäżňè¡ňæ■ćæĹŘäÿĂçžĎåČŔäÿŃéÍćè£ŹæăůçŽĎæŰźæşŢïijž

```
class ExpressionEvaluator:
    ...
    def expr(self):
    ...
    def term(self):
    ...
    def factor(self):
    ...
```

æŕŘäÿłæŰźæşŢèęĄåőŇæĹŘçŽĎäżżåŁąå¿ĹçőĂå■Ţ - åőČå£ĚéążäżŐåůęèĞşåŘşéĄ■åŐĘèŕ■æşŢèğĎåĹŹäžŐæ§ŘçğæĎŘäźĽäÿŁèőšïijŇæŰźæşŢçŽĎçŻőçŽĎåřśæŸŕèęĄäźĹåď˙ĎçŘĘåőŇèŕ■æşŢèğĎåĹŹïijŇèęĄäźĹäÿžäžĘè£ŹæăůåĄŽïijŇéIJĂéĞĞçŤĺäÿŇéÍćçŽĎè£ŹäžŽåőđçŐřæŰźæşŢïijŽ

- åęĆæđIJèğĎåĹŹäÿ■çŽĎäÿŇäÿłçňęåŘůæŸŕåŘęåď˙ŰäÿĂäÿłèŕ■æşŢèğĎåĹŹçŽĎåŘ■å■Ů(æŕŤåęĆtermæ
  è£ŹåřśæŸŕèŕěçőŮæşŢäÿ■âĂÍäÿŇéŹ■âĂÍçŽĎçŤśæÍě -
  æŐğåĹűäÿŇéŹ■åĹřåŘęäÿĂäÿłèŕ■æşŢèğĎåĹŹäÿ■åŐżãĂĆ
  æIJL˙æŰűåĂŹèğĎåĹŹäijŽèřČçŤĺåůšçżŘæL˙ğèąŇçŽĎæŰźæşŢ(æŕŤåęĆïijŇåIJÍ
  `factor ::= '('expr ')'` äÿ■åŕźexprçŽĎèřČçŤĺ)ãĂĆ
  è£ŹåřśæŸŕçőŮæşŢäÿ■âĂÍéĂŠå¡ŠâĂÍçŽĎçŤśæÍěãĂĆ

- åęĆæđIJèğĎåĹŹäÿ■äÿŇäÿĂäÿłçňęåŘůæŸŕäÿłçL˙źæőŁçňęåŘů(æŕŤåęĆ()ïijŇä¡åå¿Ůæ§ěæL˙¿äÿŇäÿĂäÿ
  åęĆæđIJäÿ■åŇźéĚ■ïijŇåřśäžğçŤ§äÿĂäÿłèŕ■æşŢéŤŹèŕŕãĂĆè£ŹäÿĂèŁĆäÿ■çŽĎ
  `_expect()` æŰźæşŢåřśæŸŕçŤĺæÍěåĄŽè£ŹäÿĂæ■ěçŽĎãĂĆ

- åęĆæđIJèğĎåĹŹäÿ■äÿŇäÿĂäÿłçňęåŘůäÿžäÿĂäžŽåŘŕèČ¡çŽĎéĂL˙æŃľéąź(æŕŤåęĆ +
  æĹŰ -)ïijŇ ä¡åå£ĚéążåŕźæŕŘäÿĂçğ■åŘŕèČ¡æČĚåĘţæčĂæ§ěäÿŇäÿĂäÿłäžď˙çL˙ŇïijŇåŘłæIJL˙å¡ŞåőČåŇ
  è£ŹäźśæŸŕæIJňèŁĆçď˙žä¿Ńäÿ■ `_accept()` æŰźæşŢçŽĎçŻőçŽĎãĂĆ
  åőČçŻÿå¡ŞäžŐ_expect()æŰźæşŢçŽĎåijśåŇŰçL˙ĹæIJňïijŇåŽääÿžåęĆæđIJäÿĂäÿłåŇźéĚ■æL˙¿åĹřäžĘåę
  ä¡ĘæŸŕåęĆæđIJæšąæL˙¿åĹřïijŇåőČäÿ■äijŽäžğçŤ§éŤŹèŕŕèĂŇæŸŕåŽđæżŽ(åĚĄèőÿåŘŐçż■çŽĎæčĂæş

- åŕźäžŐæIJL˙éĞ■åď˙■éČĺåĹĘçŽĎèğĎåĹŹ(æŕŤåęĆåIJÍèğĎåĹŹèąÍè¿¿åijŔ `::= term { ('+'|'-') term }*` äÿ■)ïijŇ éĞ■åď˙■åŁÍä¡IJéĂŽè£ĞäÿĂäÿłwhileå¿łçŐŕæÍěåőđçŐřãĂĆ
  å¿łçŐŕäÿżä¡ŞäijŽæŤűéŽĘæĹŰåď˙ĎçŘĘæL˙ĂæIJL˙çŽĎéĞ■åď˙■åĚČçt˙ăçŽt˙åĹřæšąæIJL˙åĚűäžŰåĚČçt˙ă

- äÿĂæŮęæŢt˙äÿłèŕ■æşŢèğĎåĹŹåď˙ĎçŘĘåőŇæĹŘïijŇæŕŘäÿłæŰźæşŢäijŽè£ŤåŽđæ§Řçğ■çżŞæđIJçżŹè
  è£ŹåřśæŸŕåIJÍèğčæđŘè£ĞçÍŃäÿ■åĂijæŸŕæĂŐæăůçt˙ŕåŁăçŽĎåŐ§çŘĘãĂĆ
  æŕŤåęĆïijŇåIJÍèąÍè¿¿åijŔæśĆåĂijçÍŃåžŘäÿ■ïijŇè£ŤåŽđåĂijäžčèąÍèąÍè¿¿åijŔèğčæđŘåŘŐçŽĎéČĺåĹĘç
  æIJĂåŘŐæL˙ĂæIJL˙åĂijäijŽåIJÍæIJĂéąűåśĆçŽĎèŕ■æşŢèğĎåĹŹæŰźæşŢäÿ■åŘĹázűètůæÍěãĂĆ

åř¡çőąåŘŚä¡ăæijŤçď˙žçŽĎæŸŕäÿĂäÿłçőĂå■ŢçŽĎä¿Ńå■ŘïijŇéĂŠå¡ŠäÿŇéŹ■èğčæđŘåŹĺåŔŕäžěçŤÍæÍěå
æŕŤåęĆïijŇPythonèŕ■èÍĂæIJňèžńářśæŸŕéĂŽè£ĞäÿĂäÿłéĂŠå¡ŠäÿŇéŹ■èğčæđŘåŹĺåŐżèğčéĞŁçŽĎãĂĆ
åęĆæđIJä¡ăářźæ■ď˙æĎ$åĚt˙èűčïijŇä¡ăåŔŕäžěéĂŽè£ĞæŞěçIJŃPythonæžŘçăĄæŰĞäžűGrammar/GrammaræÍ
çIJŃåőŇä¡ăäijŽåŘŚçŐřïijŇéĂŽè£ĞæL˙ŃåŁÍæŰźåijŔåŐżåőđçŐřäÿĂäÿłèğčæđŘåŹĺåĚűåőđäijŽæIJL˙å¿Ĺåď˙Žç

åĚűäÿ■äÿĂäÿłåśĂéŹŘåřśæŸŕåőČäžňäÿ■èČ¡èćńçŤÍäžŐåŇĚåŘńäżżä¡ŢåůęéĂŠå¡ŠçŽĎèŕ■æşŢèğĎåĹŹäÿ

```
items ::= items ',' item
    | item
```

äÿžäžĘè£ŹæăůåĄŽïijŇä¡ăåŔŕèČ¡äijŽåČŘäÿŇéÍćè£ŹæăůäĄ£çŤÍ `items()` æŰźæşŢïijŽ

```
def items(self):
    itemsval = self.items()
    if itemsval and self._accept(','):
        itemsval.append(self.item())
    else:
        itemsval = [ self.item() ]
```

åŤŕäÿĂçŽĎéŮőéćŸæŸŕè£ŹäÿłæŰźæşŢæăźæIJňäÿ■èČ¡åůěä¡IJïijŇäžŃåőđäÿŁïijŇåőČäijŽäžğçŤ§äÿĂäÿ

åĚşäžŐèŕ■æşŢèğĎåĹŹæIJňèžńä¡ăåŔŕèČ¡äź§äijŽçćřåĹřäÿĂäžŻæčŸæL’ŃçŽĎéŮőéćŸãĂĆ æŕŤåęĆïijŇä¡ăåŔŕèČ¡æČşç§ěéĄŞäÿŃéİćè£ŹäÿłçőĂå■ŢæL’ijèŕ■æşŢæŸŕåŘęèąĺè£řå¿Ůå¡Şïijž

```
expr ::= factor { ('+'|'-'|'*'|'/') factor }*

factor ::= '(' expression ')'
    | NUM
```

è£Źäÿłèŕ■æşŢçIJŃäÿŁåŐżæšąåŢěéŮőéćŸïijŇä¡ĘæŸŕåőČå■t’äÿ■èČ¡åŕ§ègL’åĹřæăĞåĞĘåŻŻåĹŹè£ŘçőŮ æŕŤåęĆïijŇèąĺè¿¿åijŔ "3 + 4 * 5" äijŽå¿ŮåĹř35èĂŇäÿ■æŸŕæIJ§æIJŻçŽĎ23. åĹĘåijĂä¡£çŤĺâĂİexprâĂİåŠŇâĂİtermâĂİèğĎåĹŹåŔŕäžěèőľ åőČæ■čçąőçŽĎåůěä¡IJãĂĆ

åŕźäžŐåď■æİĆçŽĎèŕ■æşŢïijŇä¡ăæIJĂåě¡æŸŕéĂL’æŃľæ§ŘäÿłèğčæđŘåůěåĚůæŕŤåęĆPyParsingæĹŰèĂ äÿŃéİćæŸŕä¡£çŤÍPLYæİěéĞ■åĘŹèąĺè¿¿åijŔæśĆåĂijçÍŃåžŔçŽĎäžčçăĄïijž

```
from ply.lex import lex
from ply.yacc import yacc

# Token list
tokens = [ 'NUM', 'PLUS', 'MINUS', 'TIMES', 'DIVIDE', 'LPAREN',
↪'RPAREN' ]
# Ignored characters
t_ignore = ' \t\n'
# Token specifications (as regexs)
t_PLUS = r'\+'
t_MINUS = r'-'
t_TIMES = r'\*'
t_DIVIDE = r'/'
t_LPAREN = r'\('
t_RPAREN = r'\)'

# Token processing functions
def t_NUM(t):
    r'\d+'
    t.value = int(t.value)
    return t

# Error handler
def t_error(t):
    print('Bad character: {!r}'.format(t.value[0]))
    t.skip(1)
```

```
# Build the lexer
lexer = lex()

# Grammar rules and handler functions
def p_expr(p):
    '''
    expr : expr PLUS term
         | expr MINUS term
    '''
    if p[2] == '+':
        p[0] = p[1] + p[3]
    elif p[2] == '-':
        p[0] = p[1] - p[3]


def p_expr_term(p):
    '''
    expr : term
    '''
    p[0] = p[1]


def p_term(p):
    '''
    term : term TIMES factor
    | term DIVIDE factor
    '''
    if p[2] == '*':
        p[0] = p[1] * p[3]
    elif p[2] == '/':
        p[0] = p[1] / p[3]

def p_term_factor(p):
    '''
    term : factor
    '''
    p[0] = p[1]

def p_factor(p):
    '''
    factor : NUM
    '''
    p[0] = p[1]

def p_factor_group(p):
    '''
    factor : LPAREN expr RPAREN
    '''
    p[0] = p[2]
```

```
def p_error(p):
    print('Syntax error')

parser = yacc()
```

è£ŹäÿłçÍŃåžŔäÿ■ïijŇæĽĂæIJL’äžčçăĄéČ¡ä¡■äžŐäÿĂäÿłæŕŤè¿ČéńŸçŽĎåśĆæňąãĂĆä¡ååŔłéIJĂèęĄäÿ
èĂŇåőđéŹĚçŽĎè£ŘèąŇèğčæđŘåŹĺïijŇæŐěåŔŮäžď’çĽŇç■Ľ’ç■Ľ’åžŢåśĆåŁĺä¡IJåůšçżŔèćńåžŞåĞ¡æŢřåőđçĆ

äÿŃéİćæŸŕäÿĂäÿłæĂŐæăůä¡£çŤĺå¿ŮåĹřçŽĎèğčæđŘåŕźèśąçŽĎä¿Ńå■ŘïijŽ

```
>>> parser.parse('2')
2
>>> parser.parse('2+3')
5
>>> parser.parse('2+(3+4)*5')
37
>>>
```

åęĆæđIJä¡ăæČşåIJĺä¡ăçŽĎçijŰçĺŃè£ĞçĺŃäÿ■æİěçĆźæŇŚæĹŸåŠŇåĹžæ£ĂïijŇçijŰåĘŹèğčæđŘåŹĺåŠŇ
åĘ■æňąïijŇäÿĂæIJňçijŰèŕŚåŹĺçŽĎäźęçś■äijŽåŇĚåŘńå¿Ĺåď’ŽåžŢåśĆçŽĎçŘĘèőžç§ěèŕĘãĂĆäÿ■è£Ğå¿Ĺåď’
PythonèĞłåůśçŽĎastæĺąåİŮäź§åĂijå¿ŮåŐžçIJŃäÿĂäÿŃãĂĆ

4.20 2.20 å■ŮèŁĆå■ŮçňęäÿšäÿŀçŽĎå■ŮçňęäÿšæŞ■ä¡IJ

éŮőécŸ

ä¡ăæČşåIJĺå■ŮèŁĆå■ŮçňęäÿšäÿŀæĽğèąŇæŹőéĂŽçŽĎæŰĞæIJňæŞ■ä¡IJ(æŕŤåęĆçğżéŹď’ijjŇæŘIJçť’ćå

èğčåĘşæŰźæąĹ

å■ŮèŁĆå■ŮçňęäÿšåŘŇæăůäź§æŤŕæŇĄåď’ğéČĺåĹĘåŠŇæŰĞæIJňå■ŮçňęäÿšäÿĂæăůçŽĎåĘĚç¡őæŞ■ä¡

```
>>> data = b'Hello World'
>>> data[0:5]
b'Hello'
>>> data.startswith(b'Hello')
True
>>> data.split()
[b'Hello', b'World']
>>> data.replace(b'Hello', b'Hello Cruel')
b'Hello Cruel World'
>>>
```

è£ŹäžŻæŞ■ä¡IJåŘŇæăůäź§éĂĆçŤĺäžŐå■ŮèŁĆæŢřçżĎãĂĆæŕŤåęĆïijŽ

```
>>> data = bytearray(b'Hello World')
>>> data[0:5]
bytearray(b'Hello')
```

```
>>> data.startswith(b'Hello')
True
>>> data.split()
[bytearray(b'Hello'), bytearray(b'World')]
>>> data.replace(b'Hello', b'Hello Cruel')
bytearray(b'Hello Cruel World')
>>>
```

ä¡ăåŔŕäžěä¡£çŤĺæ■čåĹŹèąĺè¿¿åijŔåŇźéĚ■å■ŮèŁĆå■ŮçňęäÿšïijŇä¡ĘæŸŕæ■čåĹŹèąĺè¿¿åijŔæIJňèžńå£Ě

```
>>>
>>> data = b'FOO:BAR,SPAM'
>>> import re
>>> re.split('[:,]',data)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "/usr/local/lib/python3.3/re.py", line 191, in split
return _compile(pattern, flags).split(string, maxsplit)
TypeError: can't use a string pattern on a bytes-like object
>>> re.split(b'[:,]',data) # Notice: pattern as bytes
[b'FOO', b'BAR', b'SPAM']
>>>
```

èőÍèőž

åď ğåď ŽæŢřæČĚåĘţäÿŃïijŇåIJĺæŰĞæIJňå■Ůçňęäÿšäÿ ŁçŽĎæŞ■ä¡IJåİĞåŔŕçŤĺäžŐå■ŮèŁĆå■Ůçňęäÿš
çĎűčèĂŇïijŇè£ŹéĞŇäź§æIJL’äÿĂäžŻéIJĂèęĄæşĺæĎŔçŽĎäÿ■åŘŇçĆźãĂĆéęŰåĚĹïijŇå■ŮèŁĆå■Ůçňęäÿšç

```
>>> a = 'Hello World' # Text string
>>> a[0]
'H'
>>> a[1]
'e'
>>> b = b'Hello World' # Byte string
>>> b[0]
72
>>> b[1]
101
>>>
```

è£Źçğ■èŕ■äźL’äÿŁçŽĎåŇžåĹńäijŽåŕźäžŐåď ĎçŘĘéİćåŘŠå■ŮèŁĆçŽĎå■ŮçňęæŢřæ■őæIJL’å¡śåŞ■ãĂĆ

çňňäžŇçĆźïijŇå■ŮèŁĆå■Ůçňęäÿšäÿ■äijŽæŔŘä¿ŻäÿĂäÿłç¿ŐèğĆçŽĎå■Ůçňęäÿšèąĺçď žïijŇäź§äÿ■èČ¡å

```
>>> s = b'Hello World'
>>> print(s)
b'Hello World' # Observe b'...'
>>> print(s.decode('ascii'))
Hello World
>>>
```

çśżäijijçŽĎïijÑäź§äÿ■å■ŸåIJÍäżżä¡ŢéĂĆçŤÍäžŐå■ŮèŁĆå■ŮçňęäÿšçŽĎæäijåijŔåŇŰæŞ■ä¡IJïijŽ

```
>>> b'%10s %10d %10.2f' % (b'ACME', 100, 490.1)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for %: 'bytes' and 'tuple'
>>> b'{} {} {}'.format(b'ACME', 100, 490.1)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: 'bytes' object has no attribute 'format'
>>>
```

åęĆæđIJä¡ăæČşæäijåijŔåŇŰå■ŮèŁĆå■ŮçňęäÿšïijÑä¡ăå¿ŮåĚĹä¡£çŤÍæăĞåĞĘçŽĎæŰĞæIJňå■Ůçňęäÿš

```
>>> '{:10s} {:10d} {:10.2f}'.format('ACME', 100, 490.1).encode(
 ↪'ascii')
b'ACME       100     490.10'
>>>
```

æIJĂåŘŐéIJĂèęĄæşÍæĎŔçŽĎæŸŕïijÑä¡£çŤÍå■ŮèŁĆå■ŮçňęäÿšåŕŕèČ¡äijŽæŤźåŔŸäÿĂäžŻæŞ■ä¡IJçŽĹ
æŕŤåęĆïijÑåęĆæđIJä¡ăä¡£çŤÍäÿĂäÿłçijÚçăĄäÿžå■ŮèŁĆçŽĎæŰĞäżűåŘ■ïijÑèĂŇäÿ■æŸŕäÿĂäÿłæŹőéĂŽçŽ

```
>>> # Write a UTF-8 filename
>>> with open('jalape\xf1o.txt', 'w') as f:
...     f.write('spicy')
...
>>> # Get a directory listing
>>> import os
>>> os.listdir('.') # Text string (names are decoded)
['jalapeÃ±o.txt']
>>> os.listdir(b'.') # Byte string (names left as bytes)
[b'jalapen\xcc\x83o.txt']
>>>
```

æşÍæĎŔä¿Ńå■Řäÿ■çŽĎæIJĂåŘŐéČÍåĹĘçżŹçŻőå¡ŢåŘ■äijăéĂŠäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšæŸŕæĂŐæă
åIJÍçŻőå¡Ţäÿ■çŽĎæŰĞäżűåŘ■åŇĚåŘńåŐ§ågŃçŽĎUTF-8çijŰçăĄãĂĆ
åŔĆèĂČ5.15åřŔèŁĆèŐůåŔŰæŻť'åđ'ŽæŰĞäżűåŘ■çŻÿåĚşçŽĎåĘĚåőźãĂĆ

æIJĂåŘŐæŔŘäÿĂçĆźïijÑäÿĂäžŻçÍŃåžŔåŚŸäÿžäžĘæŔŘå■ĞçÍŃåžŔæĽ'ğèąŇçŽĎéĂ§åžęäijŽåĂ¿åŘŚäž
år¡çőąæŞ■ä¡IJå■ŮèŁĆå■ŮçňęäÿšçąőåőđäijŽæŕŤæŰĞæIJňæŻť'åŁăéńŸæŢĹ(åŽăäÿžåđ'ĎçŘĘæŰĞæIJňåŽžæI
è£ŹæăůåĄŽéĂŽåÿÿäijŽåŕijèĞť'éİđåÿÿæİĆäźśçŽĎäžčçăĄãĂĆä¡ääijŽçżŔåÿÿåŘŚçŐŕå■ŮèŁĆå■Ůçňęäÿšåźűäÿ
åźűäÿŤä¡ăè£Ÿå¿ŮæĽ'ŃåŁĺåđ'ĎçŘĘæĽ'ĂæIJĽçŽĎçijŰçăĄ/èğčçăĄæŞ■ä¡IJãĂĆ
åİęçŹ¡ëőšïijÑåęĆæđIJä¡ăåIJĺåđ'ĎçŘĘæŰĞæIJňçŽĎèŕÏïijÑåřśçŻť'æŐëåIJÍçÍŃåžŔäÿ■ä¡£çŤÍæŹőéĂŽçŽĎæŰĆ

5 çňňäÿĽ'çńäïijŽæŢřå■ŮæŮěæIJ§åŠŇæŮűéŮť

åIJÍPythonäÿ■æĽ'ğèąŇæŢť'æŢřåŠŇæţőçĆźæŢřçŽĎæŢřå■ęè£ŘçőŮæŮűå¿ĹçőĂå■ŢçŽĎãĂĆ
år¡çőąåęĆæ■đ'ïijÑåęĆæđIJä¡ăéIJĂèęĄæĽ'ğèąŇåĹĘæŢřãĂĄæŢřçżĎæĹŰèĂĚæŸŕæŮěæIJ§åŠŇæŮűéŮť'çŽĎ

æIJňçńäéŻĘäÿ■èőÍèőžçŽĎåřśæŸŕè£ŹäžŻäÿżéćŸãĂĆ

Contents:

5.1 3.1 æŢřå■ŮçŽĎåŻŻèĹ■äžŤåĚě

éŮőécŸ

ä¡ăæČşåŕźæţőçĆźæŢřæĽğèąŇæŇĞåőŽçš¿åžęçŽĎèĹ■åĚěè£ŘçőŮãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎèĹ■åĚěè£ŘçőŮïijŇä¡£çŤĺåĘĚç¡őçŽĎ `round(value, ndigits)` åĞ¡æŢřå■şåŔŕãĂĆæŕŤåęĆïijŽ

```
>>> round(1.23, 1)
1.2
>>> round(1.27, 1)
1.3
>>> round(-1.27, 1)
-1.3
>>> round(1.25361,3)
1.254
>>>
```

å¡ŞäÿĂäÿłåĂijåĹŽåě¡åIJĺäÿd'äÿłè¿źçŢŇçŽĎäÿ■éŮt'çŽĎæŮűåĂŹïijŇ `round` åĞ¡æŢřè£ŤåŻđçężåőČæIJĂè£ŚçŽĎåĄűæŢřãĂĆ äź§åŕśæŸŕèŕt'ïijŇåŕź1.5æĹŰèĂĚ2.5çŽĎèĹ■åĚěè£ŘçőŮéČ¡

äijăçżŹ `round()` åĞ¡æŢřçŽĎ `ndigits` åŔĆæŢřåŔŕäžěæŸŕèt'§æŢřïijŇè£Źçğ■æČĚåĘţäÿŃïijŇèĹ■åĚěè£ŘçőŮäijŽä¡IJçŤĺåIJĺå■Ąä¡■ãĂĄçŹ¿ä¡■ãĂĄå■Čä¡■ç■L'äÿŁéÍćãĂĆæŕŤåęĆïijŽ

```
>>> a = 1627731
>>> round(a, -1)
1627730
>>> round(a, -2)
1627700
>>> round(a, -3)
1628000
>>>
```

èőÍèőž

äÿ■èęĄåřĘèĹ■åĚěåŠŇæăijåijŔåŇŰè¿ŞåĞžæŘđæůûæůĘäžĘãĂĆ åęĆæđIJä¡ăçŽĎçŻőçŽĎåŔłæŸŕçőĂå■ŢçŽĎè¿ŞåĞžäÿĂåőŽåő¡åžęçŽĎæŢřïijŇä¡ăäÿ■éIJĂèęĄä¡£çŤĺ `round()` åĞ¡æŢřãĂĆ èĂŇäžĔäžĔåŔłéIJĂèęĄåIJĺæăijåijŔåŇŰçŽĎæŮűåĂŹæŇĞåőŽçš¿åžęå■şåŔŕãĂĆæŕŤ

```
>>> x = 1.23456
>>> format(x, '0.2f')
'1.23'
>>> format(x, '0.3f')
'1.235'
>>> 'value is {:0.3f}'.format(x)
'value is 1.235'
>>>
```

åŘŇæăůïijŇäÿ■èęĄèŕŢçİĂåŐżèĹ■åĚěæţőçĆźåĂijæİěâĂİä£őæ■čâĂİèąĺéİćäÿŁçIJŃèţůæİěæ■čçąőçŽĎéŮ

```
>>> a = 2.1
>>> b = 4.2
>>> c = a + b
>>> c
6.300000000000001
>>> c = round(c, 2) # "Fix" result (???)
>>> c
6.3
>>>
```

åŕźäžŐåď ğåď ŽæŢřä¡£çŤĺåĹřæţőçĆźçŽĎçÍŃåžŔïijŇæšąæIJL’å£ĚèęĄäź§äÿ■æŐÍè■Řè£ŹæăůåĄŽãĂĆ
åř¡çőąåIJÍèőąçőŮçŽĎæŮűůåĂŹäijŽæIJL’äÿĂçĆźçĆźåřŔçŽĎèŕřåůőïijŇä¡ĘæŸŕè£ŹäžŽåřŔçŽĎèŕřåůőæŸŕèČ¡è
åęĆæđIJäÿ■èČ¡åĚĄèőÿè£ŹæăůçŽĎåřŔèŕřåůő(æŕŤåęĆæűĽ’åŔŁåĹřéĞŚèđ■éćĘå§§)ïijŇéĆčäźĹåřšå¿ŮèĂČèŹ
`decimal` æÍąåİŮäžĘïijŇäÿŃäÿĂèŁĆæĹŚäžňäijŽèŕęçżĘèőÍèőžãĂĆ

5.2 3.2 æĽ’ğèąŇçš¿çąőçŽĎæţőçĆźæŢřè£ŘçőŮ

éŮőécŸ

ä¡ăéIJĂèęĄåŕźæţőçĆźæŢřæL’ğèąŇçš¿çąőçŽĎèőąçőŮæŞ■ä¡IJïijŇåźűäÿŤäÿ■åÿŃæIJŻæIJL’äżżä¡ŢåřŔèŕŕ

èğčåĘşæŰźæąĹ

æţőçĆźæŢřçŽĎäÿĂäÿłæŹőéĄ■éŮőécŸæŸŕåőČäžňåźűäÿ■èČ¡çš¿çąőçŽĎèąĺçď žå■Ąè£ŻåĹűæŢřãĂĆ
åźűäÿŤïijŇå■şä¡£æŸŕæIJĂçőĂå■ŢçŽĎæŢřå■ęè£ŘçőŮäź§äijŽäžğçŤ§åřŔçŽĎèŕřåůőïijŇæŕŤåęĆïijŽ

```
>>> a = 4.2
>>> b = 2.1
>>> a + b
6.300000000000001
>>> (a + b) == 6.3
False
>>>
```

è£ŹäžŻéŤŹèŕŕæŸŕçŤśåžŢåśĆCPUåŠŇIEEE 754æăĞåĞĘéĂŽè£ĞèĞłåůśçŽĎæţőçĆźå■Ţä¡■åŐżæĽ’ğèąŇ
çŤśäžŐPythonçŽĎæţőçĆźæŢřæ■őçśżåđŃä¡£çŤĺåžŢåśĆèąĺçď žå■ŸåĆĺæŢřæ■őïijŇåŽăæ■d’ä¡ăęšąåŁđæşŢåŐ

åęĆædIJä¡ăæČşæŻťˇåŁăçš¿çąő(åźűèČ¡åőźå£■äÿĂåőŽçŽĎæĂğèČ¡æ■§èĂŮ)ïijŇä¡ăåŔŕäžěä¡£çŤÍ decimal æÍąåİŮïijŽ

```
>>> from decimal import Decimal
>>> a = Decimal('4.2')
>>> b = Decimal('2.1')
>>> a + b
Decimal('6.3')
>>> print(a + b)
6.3
>>> (a + b) == Decimal('6.3')
True
```

åĹİçIJŃèţůæİěïijŇäÿŁéİćçŽĎäžčçăĄåě¡åČŔæIJL’çĆźåěĞæĂłïijŇæŕŤåęĆæĹŚäžňçŤĺå■ŮçňęäÿšæİěèąÍç
çĎűèĂŇïijŇ Decimal åŕźèśąäijŽåČŔæŹőéĂŽæţőçĆźæŢřäÿĂæăůçŽĎåůěä¡IJ(æŤŕæŇĄæL’ĂæIJL’çŽĎåÿÿç
åęĆædIJä¡ăæL’Şå■řåőČäžňæĹŰèĂĚåIJÍå■ŮçňęäÿšæăijåijŔåŇŰåĞ¡æŢřäÿ■ä¡£çŤĺåőČäžňïijŇçIJŃèţůæİěèů§a

decimal æÍąåİŮçŽĎäÿĂäÿłäÿżèęĄçL’źå¿ĄæŸŕåĚĄèőÿä¡ăæŐğåĹűèőąçőŮçŽĎæŕŔäÿĂæŰźéİćïijŇåŇ
äÿžäžEè£ŹæăůåĄŽïijŇä¡ăåĚĹå¿ŮåĹŻåžžäÿĂäÿłæIJňåIJřäÿŁäÿŃæŰĞåźűæŻťˇæŤźåőČçŽĎèő¿ç¡őïijŇæŕŤåęĆ

```
>>> from decimal import localcontext
>>> a = Decimal('1.3')
>>> b = Decimal('1.7')
>>> print(a / b)
0.76470588235294117647058823 53
>>> with localcontext() as ctx:
...     ctx.prec = 3
...     print(a / b)
...
0.765
>>> with localcontext() as ctx:
...     ctx.prec = 50
...     print(a / b)
...
0.76470588235294117647058823529411764705882352941176
>>>
```

èőÍèőž

decimal æÍąåİŮåőđçŐřäžĘIBMçŽĎâĂİéĂŽçŤÍåřŔæŢřè£ŘçőŮèğĎèŇČâĂİãĂĆäÿ■çŤÍèŕťˇïijŇæIJL’å¿

PythonæŰřæL’ŃäijŽåĂ¿åŘŠäžŐä¡£çŤÍ decimal æÍąåİŮæİěåď’ĎçŘĘæţőçĆźæŢřçŽĎçš¿çąőè£ŘçőŮãĂ
çĎűèĂŇïijŇåĚĹçŘĘèğčä¡ăçŽĎåžŤçŤÍçÍŃåžŔçŻőçŽĎæŸŕéİđåÿÿéĞ■èęĄçŽĎãĂĆ
åęĆædIJä¡ăæŸŕåIJÍåĄŽçğŚå■ęèőąçőŮæĹŰåůěçÍŃécĘå§§çŽĎèőąçőŮãĂĄçŤţèĎŚçżŸåŻ¿ïijŇæĹŰèĂĚæŸŕç
éĆčäźĹä¡£çŤÍæŹőéĂŽçŽĎæţőçĆźçśžåđŇæŸŕæŕŤè¿ČæŹőéĄ■çŽĎåĄŽæşŢãĂĆ
åĚűäÿ■äÿĂäÿłåŐ§åŻăæŸŕïijŇåIJÍçIJ§åőđäÿŰçŢŃäÿ■å¿ĹåŕŚäijŽèęĄæśĆçš¿çąőåĽřæŹőéĂŽæţőçĆźæŢřèČ¡æ
åŽăæ■ď’ïijŇèőąçőŮè£ĞçÍŃäÿ■çŽĎéĆčäźĹäÿĂçĆźçĆźçŽĎèŕŕåůőæŸŕèćńåĚĄèőÿçŽĎãĂĆ
çňňäžŇçĆźåřśæŸŕïijŇåŐ§çŤ§çŽĎæţőçĆźæŢřèőąçőŮèęĄå£ńçŽĎåď’Ž-
æIJL’æŰűåĂŹä¡ăåIJÍæL’ğèąŇåď’ğéĞŔè£ŘçőŮçŽĎæŰűåĂŹéĂ§åžęäź§æŸŕéİđåÿÿéĞ■èęĄçŽĎãĂĆ

å■şä¿£åęĆæ■ď'ïijŇä¡ăå■ť'äÿ■èČ¡åőŇåĚĺå£¡çŢěèŕŕåůőãĂĆæŢřå■ęåőűèŁśäžĘåď'ğéĞŔæŮűéŮť'åŐžçăŤçĺ
ä¡ăäź§å¿ŮæşÍæĎŔäÿŃåĞŔæşŢåĹăéŹď'äžěåŔŁåď'ğæŢřåŠŇåřŔæŢřçŽĎåŁăåĹĘè£ŘçőŮæĽ'ĂåÿęæİěçŽĎå¡šå

```
>>> nums = [1.23e+18, 1, -1.23e+18]
>>> sum(nums) # Notice how 1 disappears
0.0
>>>
```

äÿŁéİćçŽĎéŤŹèŕŕåŔŕäžěåĹl'çŤÍ `math.fsum()` æĽ'ĂæŔŘä¿ŻçŽĎæŻť'çš¿çąőèőąçőŮèČ¡åŁŻæİěèğčåĘ

```
>>> import math
>>> math.fsum(nums)
1.0
>>>
```

çĎűèĂŇïijŇåŕźäžŐåĚűäžŰçŽĎçőŮæşŢïijŇä¡ääžŤèŕěäżŤçżEçăŤçl'űåőČåźűçŘĘèğčåőČçŽĎèŕŕåůőäžğçŤ

æĂżçŽĎæİěèŕť'ïijŇ `decimal` æÍąåİŮäÿżèęĄçŤĺåIJÍæűĽ'åŔŁåĽřéĞŚèđ■çŽĎécĘå§§ãĂĆ
åIJÍè£ŹçśżçÍŃåžŘäÿ■ïijŇåŞłæĂŢæŸŕäÿĂçĆźåřŔåřŘçŽĎèŕŕåůőåIJÍèőąçőŮè£ĞçÍŃäÿ■èŢŞåżűéČ¡æŸŕäÿ■åĚA
åŽăæ■ď'ïijŇ `decimal` æÍąåİŮäÿžèğčåĘşè£ŹçśżéŮőécŸæŔŘä¿ŻäžĘæŰźæşŢãĂĆ
å¡ŞPythonåŠŇæŢřæ■őåžŞæĽ'Şäžď'éĄŞçŽĎæŮűåĂŹäź§éĂŽåÿÿäijŽéĄĞåĹř `Decimal`
åŕźèśąïijŇåźűäÿŤïijŇéĂŽåÿÿäź§æŸŕåIJÍåď'ĎçŘĘéĞŚèđ■æŢřæ■őçŽĎæŮűåĂŹãĂĆ

5.3 3.3 æŢřå■ŮçŽĎæăijåijŔåŇŰè¿ŞåĞž

éŮőécŸ

ä¡ăéIJĂèęĄåřĘæŢřå■ŮæăijåijŕåŇŰåŘŐè¿ŞåĞžïijŇåźűæŐğåĹűæŢřå■ŮçŽĎä¡■æŢřãĂĄåŕźé¡ŘãĂĄå■Č

èğčåĘşæŰźæąĹ

æăijåijŔåŇŰè¿ŞåĞžå■ŢäÿłæŢřå■ŮçŽĎæŮűåĂŹïijŇåŔŕäžěä¡£çŤĺåĘĚç¡őçŽĎ
`format()` åĞ¡æŢřïijŇærŤåęĆïijŽ

```
>>> x = 1234.56789

>>> # Two decimal places of accuracy
>>> format(x, '0.2f')
'1234.57'

>>> # Right justified in 10 chars, one-digit accuracy
>>> format(x, '>10.1f')
'    1234.6'

>>> # Left justified
>>> format(x, '<10.1f')
'1234.6    '
```

```
>>> # Centered
>>> format(x, '^10.1f')
'  1234.6  '

>>> # Inclusion of thousands separator
>>> format(x, ',')
'1,234.56789'
>>> format(x, '0,.1f')
'1,234.6'
>>>
```

åęĆæđIJä¡ăæČşä¡£çŤĺæŇĞæŢřèőřæşŢïijŇåřĘfæŤźæĹŘeæĹŰèĂĚE(åŔŰåĘşäžŐæŇĞæŢřè¿ŞåĞžçŽĎåd

```
>>> format(x, 'e')
'1.234568e+03'
>>> format(x, '0.2E')
'1.23E+03'
>>>
```

åŘŇæŮűæŇĞåőŽåő¡åžęåŠŇçš¿åžęçŽĎäÿĂèĹňå¡ćåijŔæŸŕ '[<>^]?width[,]?(.digits)?' ïijŇ åĚűäÿ■ width åŠŇ digits äÿžæŢťæŢřïijŇïij§äžčèąĺåŔŕéĂL'éČĺåĹĘãĂĆ åŘŇæăůçŽĎæäijåijŔäź§èćńçŤĺåIJĺå■ŮçňęäÿšçŽĎ format() æŰźæşŢäÿ■ãĂĆærŤåęĆïijŽ

```
>>> 'The value is {:0,.2f}'.format(x)
'The value is 1,234.57'
>>>
```

èőĺèőž

æŢřå■ŮæäijåijŕåŇŰè¿ŞåĞžéĂŽåÿÿæŸŕærŤè¿ČçőĂå■ŢçŽĎãĂĆäÿŁéĺćæijŤçd'žçŽĎæŁĂæIJŕåŘŇæŮű decimal æĺąåİŮäÿ■çŽĎ Decimal æŢřå■Ůåŕźèśąã ĂĆ

å¡ŞæŇĞåőŽæŢřå■ŮçŽĎä¡■æŢřåŘŐïijŇçżŞæđIJåĂijäijŽæăźæ■ő round() åĞ¡æŢřåŘŇæăůçŽĎèğĎåĹŹè£ŽèąŇåŽŻèĹ■äžŤåĚèåŘŐè£ŤåŽđãĂĆærŤåęĆïijŽ

```
>>> x
1234.56789
>>> format(x, '0.1f')
'1234.6'
>>> format(-x, '0.1f')
'-1234.6'
>>>
```

åŇĚåŘńå■Čä¡■çňęçŽĎæäijåijŔåŇŰèů§æIJňåIJřåŇŰæšąæIJL'åĚşçşžãĂĆ åęĆæđIJä¡äéIJĂèęĄæăźæ■őåIJřåŇžæİěæŸ¿çd'žå■Čä¡■çňęïijŇä¡äéIJĂèęĄèĞłåůśåŐżèřČæ§ěäÿŃ locale æĺąåİŮäÿ■çŽĎåĞ¡æŢřäžĘãĂĆ ä¡ääŘŇæăůäź§åŔŕäżěä¡£çŤĺå■ŮçňęäÿšçŽĎ translate() æŰźæşŢæİěäžd'æ■ćå■Čä¡■çňęãĂĆærŤåęĆïijŽ

```
>>> swap_separators = { ord('.'):',', ord(','):'.' }
>>> format(x, ',').translate(swap_separators)
'1.234,56789'
>>>
```

åIJÍå¿Ĺåď ŽPythonäžčçăĄäÿ■äijŽçIJŃåĹřä¡£çŤÍ%æİěæäijåijŔåŇŰæŢřå■ŮçŽĎïijŇæŕŤåęĆïijŽ

```
>>> '%0.2f' % x
'1234.57'
>>> '%10.1f' % x
'    1234.6'
>>> '%-10.1f' % x
'1234.6    '
>>>
```

è£Źçğ■æäijåijŔåŇŰæŰźæşŢäź§æŸŕåŔŕèąŇçŽĎïijŇäÿ■è£ĞæŕŤæŻťåŁäåĚĹè£ŻçŽĎ `format()` èęĄåůőäÿĂçĆźãĂĆ æŕŤåęĆïijŇåIJÍä¡£çŤÍ%æŞ■ä¡IJçňęæäijåijŔåŇŰæŢřå■ŮçŽĎæŮűåĂŹïijŇäÿ

5.4 3.4 äžŇåĚńå■ĄåĚ■è£ŻåĹűæŢťæŢř

éŮőécŸ

ä¡ăéIJĂèęĄè¡ňæ■ćæĹŰèĂĚè¿ŞåĞžä¡£çŤÍäžŇè£ŻåĹűïijŇåĚńè£ŻåĹűæĹŰå■ĄåĚ■è£ŻåĹűèąÍçď žçŽĎæŢ

èğčåĘşæŰźæąĹ

äÿžäžĘåřĘæŢťæŢřè¡ňæ■ćäÿžäžŇè£ŻåĹűãĂĄåĚńè£ŻåĹűæĹŰå■ĄåĚ■è£ŻåĹűçŽĎæŰĞæIJňäÿšïijŇ åŔŕäžěåĹĘåĹńä¡£çŤÍ `bin()` , `oct()` æĹŰ `hex()` åĞ¡æŢřïijŽ

```
>>> x = 1234
>>> bin(x)
'0b10011010010'
>>> oct(x)
'0o2322'
>>> hex(x)
'0x4d2'
>>>
```

åŔęåď ŰïijŇåęĆæđIJä¡ääÿ■æČşè¿ŞåĞž `0b` , `0o` æĹŰèĂĚ `0x` çŽĎåL’■çijĂçŽĎèŕÍïijŇåŔŕäžěä¡£çŤÍ `format()` åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> format(x, 'b')
'10011010010'
>>> format(x, 'o')
'2322'
>>> format(x, 'x')
'4d2'
>>>
```

æŢťæŢřæŸŕæIJL’çňęåŔůçŽĎïijŇæL’Ăäžěåę Ćæđ IJä¡ăåIJĺåď’ĎçŘĘèť§æŢřçŽĎèŕĺïijŇè¿ŞåĞžçżŞæđIJäij

```
>>> x = -1234
>>> format(x, 'b')
'-10011010010'
>>> format(x, 'x')
'-4d2'
>>>
```

åęĆæđIJä¡ăæČşäžğçŤ§äÿĂäÿłæŮăçňęåŔůåĂijïijŇä¡ăéIJĂèęĄåćđåŁääÿĂäÿłæŇĞçď’žæIJĂåď’ğä¡■éŢ£åž

```
>>> x = -1234
>>> format(2**32 + x, 'b')
'11111111111111111111101100101110'
>>> format(2**32 + x, 'x')
'fffffb2e'
>>>
```

äÿžäžĘäžěäÿ■åŘŇçŽĎè£ŻåĹűè¡ňæ■ćæŢťæŢřå■ŮçňęäÿšïijŇçőĂå■ŢçŽĎä¡£çŤĺåÿęæIJL’è£ŻåĹűçŽĎ `int()` åĞ¡æŢřå■şåŔŕïijŽ

```
>>> int('4d2', 16)
1234
>>> int('10011010010', 2)
1234
>>>
```

èőĺèőž

åď’ğåď’ŽæŢřæČĚåĘţäÿŃåď’ĎçŘĘäžŇè£ŻåĹűãĂĄåĚńè£ŻåĹűåŠŇå■ĄåĚ■è£ŻåĹűæŢťæŢřæŸŕå¿ĹçőĂå
åŘłèęĄèőřä¡Ŕè£ŹäžŻè¡ňæ■ćåśđäžŐæŢťæŢřåŠŇåĚűåŕźåžŢçŽĎæŰĞæIJňèąĺçď’žäźŃéŮťçŽĎè¡ňæ■ćå■şåŔŕã

æIJĂåŘŐïijŇä¡£çŤĺåĚńè£ŻåĹűçŽĎçÍŃåžŘåŚŸæIJL’äÿĂçĆźéIJĂèęĄæşÍæĎŔäÿŃãĂĆ
PythonæŇĞåőŽåĚńè£ŻåĹűæŢřçŽĎèŕ■æşŢèů§åĚűäžŰèŕ■èÍĂçÍ■æIJL’äÿ■åŘŇãĂĆæŕŤåęĆïijŇåęĆæđIJä¡ăåČ

```
>>> import os
>>> os.chmod('script.py', 0755)
    File "<stdin>", line 1
        os.chmod('script.py', 0755)
                              ^
SyntaxError: invalid token
>>>
```

éIJĂçąőä£ÍåĚńè£ŻåĹűæŢřçŽĎåL’■çijĂæŸŕ `0o` ïijŇårśåČŔäÿŃéÍćè£Źæăůïij Ž

```
>>> os.chmod('script.py', 0o755)
>>>
```

5.5 3.5 å■ÙèŁĆåĹřåď ğæŢťæŢřçŽĎæĽŞåŇĚäÿŐèğčåŇĚ

éŮőécŸ

ä¡ăæIJL’äÿĂäÿłå■ÙèŁĆå■ŮçňęäÿšåźűæČşåřĘåőČèğčåŐŃæĹŘäÿĂäÿłæŢť æŢřăĂĆæĹŰèĂĚïijŇä¡ăéIJĂ

èğčåĘşæŰźæąĹ

åĄĞèő¿ä¡ăçŽĎçÍŃåžŔéIJĂèęĄåď ĎçŘĘäÿĂäÿłæŃěæIJL’128ä¡■éŢ£çŽĎ16äÿłåĚČçt’ăçŽĎå■ÙèŁĆå■Ůç

```
data = b'\x00\x124V\x00x\x90\xab\x00\xcd\xef\x01\x00#\x004'
```

äÿžäžĘåřĘbytesèğčæđŘäÿžæŢť æŢřïijŇä¡£çŤÍ int.from_bytes() æŰźæşŢïijŇåźűåČŔäÿŃéÍćè£ŹæăůæŇĞåőŽå■ÙèŁĆéąžåžŔïijŽ

```
>>> len(data)
16
>>> int.from_bytes(data, 'little')
69120565665751139577663547927094891008
>>> int.from_bytes(data, 'big')
94522842520747284487117727783387188
>>>
```

äÿžäžĘåřĘäÿĂäÿłåď ğæŢť æŢřè¡ňæ■ćäÿžäÿĂäÿłå■ÙèŁĆå■ŮçňęäÿšïijŇä¡£çŤÍ int.to_bytes() æŰźæşŢïijŇåźűåČŔäÿŃéÍćè£ŹæăůæŇĞåőŽå■ÙèŁĆæŢřåŠŇå■ÙèŁĆéąžåžŔïijŽ

```
>>> x = 94522842520747284487117727783387188
>>> x.to_bytes(16, 'big')
b'\x00\x124V\x00x\x90\xab\x00\xcd\xef\x01\x00#\x004'
>>> x.to_bytes(16, 'little')
b'4\x00#\x00\x01\xef\xcd\x00\xab\x90x\x00V4\x12\x00'
>>>
```

èőÍèőž

åď ğæŢť æŢřåŠŇå■ÙèŁĆå■Ůçňęäÿšäź ŃéÚť çŽĎè¡ňæ■ćæŞ■ä¡IJåźűäÿ■åÿÿèğĄãĂĆ çĎűèĂŇïijŇåIJÍäÿĂäžŽåžŤçŤÍécĘå§§æIJL’æŮűåĂŹäź§äijŽåĞžçŐřïijŇæŕŤåęĆåŕĘçăĄå■ęæĹŰèĂĚç¡ŚçżIJã ä¿ŃåęĆïijŇIPv6ç¡ŚçżIJåIJřåİĂä¡£çŤÍäÿĂäÿł128ä¡■çŽĎæŢť æŢřèąÍçď žãĂĆ åęĆæđIJä¡ăèęĄäžŐäÿĂäÿłæŢřæ■őèőřå¡Ţäÿ■æŔŘåŔŰè£ŹæăůçŽĎåĂijçŽĎæŮűåĂŹïijŇä¡ăåřśäijŽéÍćåŕźè£Źa

ä¡IJäÿžäÿĂçğ■æŻ£äžčæŰźæąĹïijŇä¡ăåŔŕèČ¡æČşä¡£çŤÍ6.11åřŔèŁĆäÿ■æL’ĂäżŃçż■çŽĎ struct æÍąåİŮæİèèğčåŐŃå■ÙèŁĆãĂĆ è£ŹæăůäźšèąŇå¿ŮéĂŽïijŇäÿ■è£ĞåĹl’çŤÍ struct æÍąåİŮæİèèğčåŐŃåŕźäžŐæŢť æŢřçŽĎåď ğåřÆŸŕæIJL’éŹŘåĹűçŽĎãĂĆ åŻăæ■d’ïijŇä¡ăåŔŕèČ¡æČşèğčåŐŃåď Žäÿłå■ÙèŁĆäÿšåźűåřĘçżŞæđIJåŘĹåźűäÿžæIJĂçżĹçŽĎçżŞæđIJïijŇåřś

```
>>> data
b'\x00\x124V\x00x\x90\xab\x00\xcd\xef\x01\x00#\x004'
>>> import struct
```

```
>>> hi, lo = struct.unpack('>QQ', data)
>>> (hi << 64) + lo
94522842520747284487117727783387188
>>>
```

å■ÙèŁĆéąžåžŔèğĎåĹŹ(littleæĹŰbig)äżĚäżĚæŇĞåőŽäžĘæđĎåżžæŢťæŢřæŮűçŽĎå■ÙèŁĆçŽĎä¡Őä¡■æĹŚäžňäżŐäÿŃéİćçš¿å£ČæđĎéĂäçŽĎ16è£ŻåĹűæŢřçŽĎèąÍçď žäÿ■åŔřäžěå¿Ĺåőźæ ŸŞçŽĎçIJŃåĞžæİěïijŽ

```
>>> x = 0x01020304
>>> x.to_bytes(4, 'big')
b'\x01\x02\x03\x04'
>>> x.to_bytes(4, 'little')
b'\x04\x03\x02\x01'
>>>
```

åęĆæđIJä¡ăèŕŢçİĂåřĘäÿĂäÿłæŢťæŢřæL'ŞåŇĚäÿžå■ÙèŁĆå■ŮçňęäÿšïijŇéĆčäźĹåőČåřśäÿ■åŘĹéĂĆäžE
åęĆæđIJéIJĂèęAçŽĎèŕÏïijŇä¡ääŔřäžěä¡£çŤÍ `int.bit_length()`
æŰźæşŢæİěåĘşåőŽéIJĂèęAåď'ŽåřŚå■ÙèŁĆä¡■æİěå■ŸåĆÍè£ŹäÿłåĂijãĂĆ

```
>>> x = 523 ** 23
>>> x
335381300113661875107536852714019056160355655333978849017944067
>>> x.to_bytes(16, 'little')
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
OverflowError: int too big to convert
>>> x.bit_length()
208
>>> nbytes, rem = divmod(x.bit_length(), 8)
>>> if rem:
... nbytes += 1
...
>>>
>>> x.to_bytes(nbytes, 'little')
b'\x03X\xf1\x82iT\x96\xac\xc7c\x16\xf3\xb9\xcf...\xd0'
>>>
```

5.6 3.6 åď■æŢřçŽĎæŢřå■ęè£ŘçőŮ

éŮőécŸ

ä¡ääĘŹçŽĎæIJĂæŰřçŽĎç¡ŚçżIJèőď'èŕĄæŰźæąĹäžčçăĄéĄĞåĹřäžĘäÿĂäÿłéŽ¿écŸïijŇåźűäÿŤä¡ääŤřäÿ
åĘ■æĹŰèĂĚæŸřä¡ääżĚäżĚéIJĂèęAä¡£çŤĺåď'■æŢřæİěæL'ğèąŇäÿĂäžŻèőąçőŮæŞ■ä¡IJãĂĆ

èğčåĘşæŰźæąĹ

åď■æŢřåŔŕäżěçŤĺä¡£çŤĺåĞ¡æŢř `complex(real, imag)` æĹŰèĂĚæŸŕåÿęæIJĽåŘŐçijĂjçŽĎætőçĆźæŢřæİěæŇĞåőŽãĂĆæŕŤåęĆïijŽ

```
>>> a = complex(2, 4)
>>> b = 3 - 5j
>>> a
(2+4j)
>>> b
(3-5j)
>>>
```

åŕźåžŤçŽĎåőđéČĺãĂąèŹŽéČĺåŠŇåĚśè¡■åď■æŢřåŔŕäżěå¿Ĺåőźæ ŸŞçŽĎèŐůåŔŰãĂĆåřśåČŔäÿŃéİćè£Ź

```
>>> a.real
2.0
>>> a.imag
4.0
>>> a.conjugate()
(2-4j)
>>>
```

åŘęåď Űïij ŇæĽĂæIJĽåÿÿèğĄçŽĎæŢřå■ęè£ŘçőŮéČ¡åŔŕäżěåůěä¡IJïijŽ

```
>>> a + b
(5-1j)
>>> a * b
(26+2j)
>>> a / b
(-0.4117647058823529+0.6470588235294118j)
>>> abs(a)
4.47213595499958
>>>
```

åęĆæđIJèęĄæĽğèąŇåĚűäżŰçŽĎåď■æŢřåĞ¡æŢřæŕŤåęĆæ■čåijęãĂĄä¡Źåijęæ ĹŰåźşæŰźæăźïijŇä¡£çŤĺ `cmath` æĺąåİŮïijŽ

```
>>> import cmath
>>> cmath.sin(a)
(24.83130584894638-11.356612711218174j)
>>> cmath.cos(a)
(-11.36423470640106-24.814651485634187j)
>>> cmath.exp(a)
(-4.829809383269385-5.5920560936409816j)
>>>
```

èőÍèőž

Pythonäÿ■åď ğéČÍåĹĘäÿŐæŢřå■ęçŻÿåĚşçŽĎæÍąåİŮéČ¡èČ¡åď ĎçŘĘåď ■æŢřãĂĆ æŕŤåęĆåęĆæđIJä¡ăä¡£çŤÍ numpy ïijŇåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎæđĎéĂääÿĂäÿłåď ■æŢřæŢřçżĎåźűåIJÍè£Źäÿłæ

```
>>> import numpy as np
>>> a = np.array([2+3j, 4+5j, 6-7j, 8+9j])
>>> a
array([ 2.+3.j, 4.+5.j, 6.-7.j, 8.+9.j])
>>> a + 2
array([ 4.+3.j, 6.+5.j, 8.-7.j, 10.+9.j])
>>> np.sin(a)
array([ 9.15449915 -4.16890696j, -56.16227422 -48.50245524j,
        -153.20827755-526.47684926j, 4008.42651446-589.49948373j])
>>>
```

PythonçŽĎæăĞåĞĘæŢřå■ęåĞ¡æŢřçąőåőđæČĚåĘţäÿŃåźűäÿ■èČ¡äžğçŤ§åď ■æŢřåĂijïijŇåŻăæ■ď ä¡ăçŽ

```
>>> import math
>>> math.sqrt(-1)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ValueError: math domain error
>>>
```

åęĆæđIJä¡ăæČşçŤ§æĹŘäÿĂäÿłåď ■æŢřè£ŤåŻđçżŞæđIJïijŇä¡ăå£Ěéążæ Ÿ¿çď žçŽĎä¡£çŤÍ cmath æÍąåİŮïijŇæĹŮèĂĚåIJÍæ§Řäÿłæ ŤŕæŇĄåď ■æŢřçŽĎåžŞäÿ■åčřæ ŸŐåď ■æŢřçśżåđŃçŽĎä¡£çŤÍãĂĆæ

```
>>> import cmath
>>> cmath.sqrt(-1)
1j
>>>
```

5.7 3.7 æŮăçľůåď ğäÿŐNaN

éŮőécŸ

ä¡ăæČşåĹŻåžžæĹŰæţŃèŕŢæ■čæŮăçľůãĂĄèť §æŮăçľůæĹŰNaN(éİđæŢřå■Ů)çŽĎæţőçĆźæŢřãĂĆ

èğčåĘşæŰźæąĹ

PythonåźűæšąæIJL'çL'źæőŁçŽĎèŕ■æşŢæİěèąÍçď žè£ŹäžŻçL'źæőŁçŽĎæţőçĆźåĂijïijŇä¡ĘæŸŕåŔŕäžěä¡£ float() æİěåĹŻåžžåőČäžñãĂĆæŕŤåęĆïijŽ

```
>>> a = float('inf')
>>> b = float('-inf')
>>> c = float('nan')
>>> a
```

```
inf
>>> b
-inf
>>> c
nan
>>>
```

äÿžäžĘæţŃèŕŢè£ŹäžŻåĂijçŽĎå■ŸåIJĺïijŇä¡£çŤĺ `math.isinf()` åŠŇ `math.isnan()` åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> math.isinf(a)
True
>>> math.isnan(c)
True
>>>
```

èőĺèőž

æČşäžĘèğčæŻt'åd'Žè£ŹäžŻçL'źæőŁæţőçĆźåĂijçŽĎä£ąæĄŕïijŇåŔŕäžěåŔĆèĂČIEEE 754èğĎèŇČãĂĆ çĎűèĂŇïijŇäź§æIJL'äÿĂäžŻåIJřæŰźéIJĂèęĄä¡ăçL'źåĹńæşĺæĎŔïijŇçL'źåĹńæŸŕèů§æŕŤè¿

æŮăçl'ůåd'ğæŢřåIJĺæL'ğèąŇæŢřå■ęèőąçőŮçŽĎæŮűůåĂŹäijŽäijăæŠ■ïijŇæŕŤåęĆïijŽ

```
>>> a = float('inf')
>>> a + 45
inf
>>> a * 10
inf
>>> 10 / a
0.0
>>>
```

ä¡ĘæŸŕæIJL'äžŻæŞ■ä¡IJæŮűæIJłåőŽäźL'çŽĎåźűäijŽè£ŤåŻđäÿĂäÿłNaNçżŞæđIJãĂĆæŕŤåęĆïijŽ

```
>>> a = float('inf')
>>> a/a
nan
>>> b = float('-inf')
>>> a + b
nan
>>>
```

NaNåĂijäijŽåIJĺæL'ĂæIJL'æŞ■ä¡IJäÿ■äijăæŠ■ïijŇèĂŇäÿ■äijŽäžğçŤ§äijĆåÿÿãĂĆæŕŤåęĆïijŽ

```
>>> c = float('nan')
>>> c + 23
nan
>>> c / 2
nan
```

```
>>> c * 2
nan
>>> math.sqrt(c)
nan
>>>
```

NaNåĂijçŽĎäÿĂäÿłçL'źåĹńçŽĎåIJřæŰźæŮűåőČäžňäźŃéŮt'çŽĎæŕŤè¿ČæŞ■ä¡IJæĂżæŸŕè£ŤåŻđFalseâ

```
>>> c = float('nan')
>>> d = float('nan')
>>> c == d
False
>>> c is d
False
>>>
```

çŤśäžŐè£ŹäÿłåŐ§åŻăïijŇæţŃèŕŢäÿĂäÿłNaNåĂijå¿ŮåŤŕäÿĂåőL'åĚÍçŽĎæŰźæşŢåřśæŸŕä¡£çŤÍ `math.isnan()` ïijŇäź§åřśæŸŕäÿŁéİćæijŤçd'žçŽĎéĆčæăůãĂĆ

æIJL'æŮűåĂŹçÍŃåžŔåŚŸæČşæŤźåŔŸPythonézŸèőd'èąŇäÿžïijŇåIJÍè£ŤåŻđæŮăçl'ůåd'ğæĹŰNaNçżŞæ `fpectl` æÍąåİŮåŔŕäżěçŤÍæİěæŤźåŔŸè£Źçğ■èąŇäÿžïijŇä¡ĘæŸŕåőČåIJÍæăĞåĞĘçŽĎPythonæđĎåžžäÿ■åźűâ åźűäÿŤéŠĹåŕźçŽĎæŸŕäÿŞåőűçžğçÍŃåžŔåŚŸãĂĆåŔŕäżěåŔĆèĂČåIJÍçž£çŽĎPythonæŰĞæąčèŐůåŔŰæŻt'åd

5.8 3.8 åĹĘæŢřè£ŘçőŮ

éŮőécŸ

ä¡ăè£ŻåĚěæŮűéŮt'æIJžåŹÍïijŇçłĄçĎűåŔŚçŐřä¡ăæ■čåIJÍåĄŽåřŔå■ęåőűåž■ä¡IJäÿŽïijŇåźűæűL'åŔŁåĹřă æĹŰèĂĚä¡ăåŔŕèČ¡éIJĂèęĄåĘŹäžčçăĄåŐżèőąçőŮåIJÍä¡ăçŽĎæIJÍåůěåůěåŐĆäÿ■çŽĎæţŃéĞŔåĂijãĂĆ

èğčåĘşæŰźæąĹ

`fractions` æÍąåİŮåŔŕäżěèćńçŤÍæİěæL'ğèąŇåŇĚåŘńåĹĘæŢřçŽĎæŢřå■ęè£ŘçőŮãĂĆæŕŤåęĆïijŽ

```
>>> from fractions import Fraction
>>> a = Fraction(5, 4)
>>> b = Fraction(7, 16)
>>> print(a + b)
27/16
>>> print(a * b)
35/64

>>> # Getting numerator/denominator
>>> c = a * b
>>> c.numerator
35
>>> c.denominator
64
```

```
>>> # Converting to a float
>>> float(c)
0.546875

>>> # Limiting the denominator of a value
>>> print(c.limit_denominator(8))
4/7

>>> # Converting a float to a fraction
>>> x = 3.75
>>> y = Fraction(*x.as_integer_ratio())
>>> y
Fraction(15, 4)
>>>
```

èőÍèőž

åIJÍåď ğåď ŽæŢřçÍŃåžŔäÿ■äÿĂèĹňäÿ■äijŽåĞžçŐřåĹĘæŢřçŽĎèőąçőŮéŮőécŸïijŇä¡ĘæŸŕæIJL’æŮűåĂŹ
æŕŤåęĆïijŇåIJÍäÿĂäÿłåĚĄèőÿæŐěåŔŮåĹĘæŢřå¡ćåijŔçŽĎæţŃèŕŢå■Ţä¡■åźűäżěåĹĘæŢřå¡ćåijŔæL’ğèąŇè£Ř
çŻť’æŐěä¡£çŤĺåĹĘæŢřåŔŕäżěåĞŔåřŚæL’ŃåŁĺè¡ňæ■ćäÿžåřŔæŢřæĹŰæţőçĆźæŢřçŽĎåůěä¡IJãĂĆ

5.9 3.9 åď ğåđŃæŢřçżĎè£ŘçőŮ

éŮőécŸ

ä¡ăéIJĂèęĄåIJÍåď ğæŢřæ■őéŻĘ(æŕŤåęĆæŢřçżĎæĹŰç¡Śæăij)äÿŁéÍćæL’ğèąŇèőąçőŮãĂĆ

èğčåĘşæŰźæąĹ

æűL’åŔŁåĹřæŢřçżĎçŽĎéĞ■éĞŔçžğè£ŘçőŮæŞ■ä¡IJïijŇåŔŕäżěä¡£çŤÍ `NumPy` åžŞãĂĆ
`NumPy` çŽĎäÿĂäÿłäÿżèęĄçL’źå¿ĄæŸŕåőĆäijŽçżŹPythonæŔŘä¿ŻäÿĂäÿłæŢřçżĎåŕźèśąïijŇçŻÿæŕŤæăĞåĞĘ
äÿŃéÍćæŸŕäÿĂäÿłçőĂå■ŢçŽĎåřŔä¿Ńå■ŘïijŇåŘŚä¡ăåśŢçď žæăĞåĞĘåĹŮèąĺåŕźèśąåŠŇ
`NumPy` æŢřçżĎåŕźèśąäźŃéŮt’çŽĎåůőåĹníijŽ

```
>>> # Python lists
>>> x = [1, 2, 3, 4]
>>> y = [5, 6, 7, 8]
>>> x * 2
[1, 2, 3, 4, 1, 2, 3, 4]
>>> x + 10
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: can only concatenate list (not "int") to list
>>> x + y
[1, 2, 3, 4, 5, 6, 7, 8]
```

```
>>> # Numpy arrays
>>> import numpy as np
>>> ax = np.array([1, 2, 3, 4])
>>> ay = np.array([5, 6, 7, 8])
>>> ax * 2
array([2, 4, 6, 8])
>>> ax + 10
array([11, 12, 13, 14])
>>> ax + ay
array([ 6, 8, 10, 12])
>>> ax * ay
array([ 5, 12, 21, 32])
>>>
```

æ■čåęĆæĽ'ĂèğĄïijŇäÿď'çğ■æŰźæąĹäÿ■æŢřçżĎçŽĎå§žæIJňæŢřå■ęè£ŘçőŮçżŞæđIJåźűäÿ■çŻÿåŘŇãĂ
çĽ'źåĹńçŽĎïijŇ NumPy äÿ■çŽĎæăĞéĞŔè£ŘçőŮ(æŕŤåęĆ ax
* 2 æĹŰ ax + 10)äijŽä¡IJçŤĺåIJĺæŕŔäÿĂäÿłåĚČçť'ăäÿŁãĂĆ
åŔęåď'ŰïijŇå¡Şäÿď'äÿłæŞ■ä¡IJæŢřéČ¡æŸŕæŢřçżĎçŽĎæŮűåĂŹæĽ'ğèąŇåĚČçť'ăåŕźç■Ľ'ä¡■ç¡őèőąçőŮïijŇåźú

åŕźæŢť'äÿłæŢřçżĎäÿ■æĽ'ĂæIJĽ'åĚČçť'ăåŘŇæŮűæĽ'ğèąŇæŢřå■ęè£ŘçőŮåŔŕäżěä¡£å¿Ůä¡IJçŤĺåIJĺæŢť'ä
æŕŤåęĆïijŇåęĆæđIJä¡ăæČşèőąçőŮåď'ŽéąźåijŔçŽĎåĂijïijŇåŔŕäżěè£ŹæăůåĄŽïijŽ

```
>>> def f(x):
... return 3*x**2 - 2*x + 7
...
>>> f(ax)
array([ 8, 15, 28, 47])
>>>
```

NumPy è£ŸäÿžæŢřçżĎæŞ■ä¡IJæŔŘä¿ŻäžĘåď'ğéĞŔçŽĎéĂŽçŤĺåĞ¡æŢřïijŇè£ŹäžŻåĞ¡æŢřåŔŕäżěä¡IJäÿ
math æĺąåİŮäÿ■çśżäijijåĞ¡æŢřçŽĎæŻ£äżčãĂĆæŕŤåęĆïijŽ

```
>>> np.sqrt(ax)
array([ 1. , 1.41421356, 1.73205081, 2. ])
>>> np.cos(ax)
array([ 0.54030231, -0.41614684, -0.9899925 , -0.65364362])
>>>
```

ä¡£çŤĺè£ŹäžŻéĂŽçŤĺåĞ¡æŢřèęĄæŕŤå¿łçŐŕæŢřçżĎåźűä¡£çŤĺ
math æĺąåİŮäÿ■çŽĎåĞ¡æŢřæĽ'ğèąŇèőąçőŮèęĄå£ńçŽĎåď'ŽãĂĆ
åŻăæ■ď'ïijŇåŔłèęĄæIJĽ'åŔŕèČ¡çŽĎèŕİåř¡éĞŔéĂĽ'æŃľ' NumPy çŽĎæŢřçżĎæŰźæąĹãĂĆ

åžŢåśĆåőđçŐřäÿ■ïijŇ NumPy æŢřçżĎä¡£çŤĺäžĘCæĹŰèĂĚFortranèŕ■èĺĂçŽĎæIJžåĹűåĹĘéĚ■åĘĚå■Ÿã
äź§åřśæŸŕèŕť'ïijŇåőČäżňæŸŕäÿĂäÿłéİđåÿÿåď'ğçŽĎè£đçż■çŽĎåźűçŤśåŘŇçśżåđŃæŢřæ■őçżĎæĹŘçŽĎåĘĚå
æĽ'ĂäżěïijŇä¡ăåŔŕäżěæđĎéĂăäÿĂäÿłæŕŤæŹőéĂŽPythonåĹŮèąĺåď'ğçŽĎåď'ŽçŽĎæŢřçżĎãĂĆ
æŕŤåęĆïijŇåęĆæđIJä¡ăæČşæđĎéĂăäÿĂäÿł10,000*10,000çŽĎæţőçĆźæŢřäžŇçżť'ç¡ŚæăijïijŇå¿Ĺè¡żæİ¿ïijŽ

```
>>> grid = np.zeros(shape=(10000,10000), dtype=float)
>>> grid
    array([[ 0., 0., 0., ..., 0., 0., 0.],
```

```
    [ 0., 0., 0., ..., 0., 0., 0.],
    [ 0., 0., 0., ..., 0., 0., 0.],
    ...,
    [ 0., 0., 0., ..., 0., 0., 0.],
    [ 0., 0., 0., ..., 0., 0., 0.],
    [ 0., 0., 0., ..., 0., 0., 0.]])
>>>
```

æL'ĂæIJL'çŽĎæŹőéĂŽæŞ■ä¡IJè£ŸæŸŕäijŽåŘŇæŮűä¡IJçŤĺåIJĺæL'ĂæIJL'åĚČçť'ääýŁïijŽ

```
>>> grid += 10
>>> grid
array([[ 10., 10., 10., ..., 10., 10., 10.],
    [ 10., 10., 10., ..., 10., 10., 10.],
    [ 10., 10., 10., ..., 10., 10., 10.],
    ...,
    [ 10., 10., 10., ..., 10., 10., 10.],
    [ 10., 10., 10., ..., 10., 10., 10.],
    [ 10., 10., 10., ..., 10., 10., 10.]])
>>> np.sin(grid)
array([[-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111],
    [-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111],
    [-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111],
    ...,
    [-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111],
    [-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111],
    [-0.54402111, -0.54402111, -0.54402111, ..., -0.54402111,
        -0.54402111, -0.54402111]])
>>>
```

åĚşäžŐ `NumPy` æIJL'äÿĂçĆźéIJĂèęĄçL'źåĹńçŽĎäÿżæĎŕïijŇéĆčåřśæŸŕåőČæL'l'åśŢPythonåĹŮèąĺçŽĎ - çL'źåĹńæŸŕåŕźäžŐåď'Žçżť'æŢřçżĎãĂĆ äÿžäžĘèŕť'æŸŐæÿĚæěŽïijŇåĚĹædĎéĂääÿĂäÿłçőĂå■ŢçŽĎäžŇçżt

```
>>> a = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
>>> a
array([[ 1, 2, 3, 4],
[ 5, 6, 7, 8],
[ 9, 10, 11, 12]])

>>> # Select row 1
>>> a[1]
array([5, 6, 7, 8])

>>> # Select column 1
>>> a[:,1]
array([ 2, 6, 10])
```

```
>>> # Select a subregion and change it
>>> a[1:3, 1:3]
array([[ 6, 7],
       [10, 11]])
>>> a[1:3, 1:3] += 10
>>> a
array([[ 1, 2, 3, 4],
       [ 5, 16, 17, 8],
       [ 9, 20, 21, 12]])

>>> # Broadcast a row vector across an operation on all rows
>>> a + [100, 101, 102, 103]
array([[101, 103, 105, 107],
       [105, 117, 119, 111],
       [109, 121, 123, 115]])
>>> a
array([[ 1, 2, 3, 4],
       [ 5, 16, 17, 8],
       [ 9, 20, 21, 12]])

>>> # Conditional assignment on an array
>>> np.where(a < 10, a, 10)
array([[ 1, 2, 3, 4],
       [ 5, 10, 10, 8],
       [ 9, 10, 10, 10]])
>>>
```

èőÍèőž

NumPy æŸŕPythonécĘå§§äÿ■å¿Ĺåď Žçğ Śå■ęäÿŐåůěçÍŃåžŞçŽĎå§žçąĂïijŇåŘŇæŮűäź§æŸŕèćńåź£æşŽ å■şä¿£åęĆæ■ď ïijŇåIJĺåĹŽåijĂåğŃçŽĎæŮűåĂŹéĂŽè£ĞäÿĂäžŽçőĂå■ŢçŽĎä¿Ńå■ŘåŠŇçŐľ åĚůçÍŃåžŔäź§

éĂŽåÿÿæĹŚäžňåŕijåĚě NumPy æĺąåİŮçŽĎæŮűåĂŹäijŽä¡£çŤĺèŕ■åŔě `import numpy as np` ãĂĆ è£ŹæăůçŽĎèŕÍä¡ăåřśäÿ■çŤĺåĘ■ä¡ăçŽĎçÍŃåžŔéĞŇéİćäÿĂéĄ■éĄ■çŽĎæŢšåĚě numpy ïijŇåŘłéIJĂèęĄè¿ŞåĚě np åřśèąŇäžĘïijŇèŁĆçIJĄäžĘäÿ■åřŚæŮűéŮťãĂĆ

åęĆædIJæČşèŐůåŘŰæŻťåď ŽçŽĎä£ąæĄŕïijŇä¡ăå¡ŞçĎűå¿ŮåŐż NumPy åőŸç¡ŚéĂŻéĂŻäžĘïijŇç¡ŚåİĂæŸŕïijŽ http://www.numpy.org

5.10 3.10 ç§ľéŸţäÿŐçž£æĂğäżčæŢřè£ŘçőŮ

éŮőécŸ

ä¡ăéIJĂèęĄæL’ğèąŇç§l’éŸţåŠŇçž£æĂğäżčæŢřè£ŘçőŮïijŇæŕŤåęĆç§l’éŸţäźŸæşŢãĂĄåŕżæL’¿èąŇåĹŮå

èğčåĘşæŰźæąĹ

NumPy åžŞæIJL'äÿĂäÿłç§l'éŸţåŕźèśąåŔŕäżěçŤĺæİěèğčåĘşè£ŹäÿłéŮőécŸăĂĆ
ç§l'éŸţçśżäijijäžŐ3.9åřŔèŁĆäÿ■æŢřçżĎåŕźèśąïijŇä¡ĘæŸŕéĄţå¿łçž£æĂğäžčæŢřçŽĎèőąçőŮèğĎåĹŹãĂ

```
>>> import numpy as np
>>> m = np.matrix([[1,-2,3],[0,4,5],[7,8,-9]])
>>> m
matrix([[ 1, -2, 3],
        [ 0, 4, 5],
        [ 7, 8, -9]])

>>> # Return transpose
>>> m.T
matrix([[ 1, 0, 7],
        [-2, 4, 8],
        [ 3, 5, -9]])

>>> # Return inverse
>>> m.I
matrix([[ 0.33043478, -0.02608696, 0.09565217],
        [-0.15217391, 0.13043478, 0.02173913],
        [ 0.12173913, 0.09565217, -0.0173913 ]])

>>> # Create a vector and multiply
>>> v = np.matrix([[2],[3],[4]])
>>> v
matrix([[2],
        [3],
        [4]])
>>> m * v
matrix([[ 8],
        [32],
        [ 2]])
>>>
```

åŔŕäżěåIJÍ numpy.linalg å■ŘåŇĚäÿ■æL'¿åĹřæŻt'åď'ŽçŽĎæŞ■ä¡IJåĞ¡æŢřïijŇæŕŤåęĆïijŽ

```
>>> import numpy.linalg

>>> # Determinant
>>> numpy.linalg.det(m)
-229.99999999999983

>>> # Eigenvalues
>>> numpy.linalg.eigvals(m)
array([-13.11474312, 2.75956154, 6.35518158])

>>> # Solve for x in mx = v
>>> x = numpy.linalg.solve(m, v)
```

```
>>> x
matrix([[ 0.96521739],
        [ 0.17391304],
        [ 0.46086957]])
>>> m * x
matrix([[ 2.],
        [ 3.],
        [ 4.]])
>>> v
matrix([[2],
        [3],
        [4]])
>>>
```

èőÍèőž

å¿ÍæŸ¿çĎűçž£æĂğäžčæŢřæŸŕäÿłéÍđåÿÿåď ğçŽĎäÿžécŸïijŇåůšçżŔèűĚåĞžäžĘæIJňäźęèČ¡èőÍèőžçŽĎè
ä¡ĘæŸŕïijŇåęĆæđIJä¡ăéIJĂèęĄæŞ■ä¡IJæŢřçżĎåŠŇåŘŚéĞŔçŽĎèŕÍïijŇ NumPy
æŸŕäÿĂäÿłäÿ■éŤŹçŽĎåĚěåŘčçĆźãĂĆ åŔŕäžěèő£éŮő NumPy åőŸç¡Ś http://www.numpy.org
èŐůåŘŰæŻť åď Žä£ąæĄŕãĂĆ

5.11 3.11 éŽŔælJžéĂĽæŃľ

éŮőécŸ

ä¡ăæČşäżŐäÿĂäÿłåžŔåĹŮäÿ■éŽŔæIJžæŁ¡åŘŰèŃěåźšåĚČçť ăïijŇæĹŰèĂĚæČşçŢ§æĹŘåĞääÿłéŽŔæIJ

èğčåĘşæŰźæąĹ

random æÍąåİŮæIJL'åď ğéĞŔçŽĎåĞ¡æŢřçŤÍæİěäžğçŤ§éŽŔæIJžæŢřåŠŇéŽŔæIJžéĂL'æŃľåĚČçť ăãĂĆ
ærŤåęĆïijŇèęĄæČşäżŐäÿĂäÿłåžŔåĹŮäÿ■éŽŔæIJžçŽĎæŁ¡åŘŰäÿĂäÿłåĚČçť ăïijŇåŔŕäžěä¡£çŤÍ
random.choice() ïijŽ

```
>>> import random
>>> values = [1, 2, 3, 4, 5, 6]
>>> random.choice(values)
2
>>> random.choice(values)
3
>>> random.choice(values)
1
>>> random.choice(values)
4
>>> random.choice(values)
6
>>>
```

äÿžäžĘæŔŘåŔŰåĞžNäÿłäÿ■åŘŇåĚČçť ăçŽĎæăůæIJňçŤĺæİěåĄŽè£ŻäÿĂæ■ěçŽĎæŞ■ä¡IJïijŇåŔŕäżěä¡£ `random.sample()` ïijŽ

```
>>> random.sample(values, 2)
[6, 2]
>>> random.sample(values, 2)
[4, 3]
>>> random.sample(values, 3)
[4, 3, 1]
>>> random.sample(values, 3)
[5, 4, 1]
>>>
```

åęĆæđIJä¡ăäżĚäżĚåŔłæŸŕæČşæL'Şäźśåž ŔåĹŮäÿ■åĚČçť ăçŽĎéąžåžŔïijŇåŔŕäżěä¡£çŤÍ `random.shuffle()` ïijŽ

```
>>> random.shuffle(values)
>>> values
[2, 4, 6, 5, 3, 1]
>>> random.shuffle(values)
>>> values
[3, 5, 2, 1, 6, 4]
>>>
```

çŤ§æĹŘéŽŔæIJžæŢť'æŢřïijŇèŕůä¡£çŤÍ `random.randint()` ïijŽ

```
>>> random.randint(0,10)
2
>>> random.randint(0,10)
5
>>> random.randint(0,10)
0
>>> random.randint(0,10)
7
>>> random.randint(0,10)
10
>>> random.randint(0,10)
3
>>>
```

äÿžäžĘçŤ§æĹR0åĹř1èŇČåŻť'åĘĚåİĞåŇĂåĹĘåÿČçŽĎæţőçĆźæŢřïijŇä¡£çŤÍ `random.random()` ïijŽ

```
>>> random.random()
0.9406677561675867
>>> random.random()
0.133129581343897
>>> random.random()
0.4144991136919316
>>>
```

åęĆædIJèęĄèŐůåŘŰNä¡■éŽŔæIJžä¡■(äžŇè£ŻåĹű)çŽĎæŢt'æŢřïijŇä¡£çŤĺ `random.getrandbits()` ïijŽ

```
>>> random.getrandbits(200)
335837000776573622800628485064121869519521710558559406913275
>>>
```

èőĺèőž

`random` æĺąåİŮä¡£çŤĺ *Mersenne Twister* çőŮæşŢæİěèőąçőŮçŤ§æĹŘéŽŔæIJžæŢřãĂĆè£ŹæŸŕäÿĂäÿłçä¡ĘæŸŕä¡ăåŘŕäžěéĂŽè£Ğ `random.seed()` åĞ¡æŢřä£őæŤźåĹİåğŇåŇŰçğ■å■ŘãĂĆærŤåęĆïijŽ

```
random.seed() # Seed based on system time or os.urandom()
random.seed(12345) # Seed based on integer given
random.seed(b'bytedata') # Seed based on byte data
```

éŹď'äžĘäÿŁè£řäžŇçż■çŽĎåŁ§èČ¡ïijŇrandomæĺąåİŮè£ŸåŇĚåŘńå§žäžŐåİĞåŇĂåĹĘåÿČãĂĄéńŸæŰŕå
ærŤåęĆïijŇ `random.uniform()` èőąçőŮåİĞåŇĂåĹĘåÿČéŽŔæIJžæŢřïijŇ `random.gauss()` èőąçőŮæ■čæĂĄåĹĘåÿČéŽŔæIJžæŢřãĂĆ åŕźäžŐåĚűäžŰçŽĎåĹĘåÿČæČĚåĘţèŕůåŘĆèĂČåIJĺçž£æŰĞæąčãĂĆ

åIJĺ `random` æĺąåİŮäÿ■çŽĎåĞ¡æŢřäÿ■åžŤèŕěçŤĺåIJĺåŠŇårĘçăĄå■ęçŻÿåĚşçŽĎçĺŃåžŔäÿ■ãĂĆ åęĆædIJä¡ăçąőåődéIJĂèęĄçśżäijijçŽĎåŁ§èČ¡ïijŇåŘŕäžěä¡£çŤĺsslæĺąåİŮäÿ■çŻÿåžŤçŽĎåĞ¡æŢřãĂĆ ærŤåęĆïijŇ `ssl.RAND_bytes()` åŘŕäžěçŤĺæİěçŤ§æĹŘäÿĂäÿłåőĽ'åĚĺçŽĎéŽŔæIJžå■ŮèŁĆåžŔåĹŮãĂĆ

5.12 3.12 å§žælJńçŽĎæŮěælJ§äÿŐæŮűéŮť'è¡ňæ■ć

éŮőécŸ

ä¡ăéIJĂèęĄæL'ğèąŇçőĂå■ŢçŽĎæŮűéŮť'è¡ňæ■ćïijŇærŤåęĆåd'l'åĽrçğŠïijŇářŔæŮűåĹřåĹĘéŠ§ç■L'çŽĎ

èğčåĘşæŰźæąĹ

äÿžäžĘæL'ğèąŇäÿ■åŘŇæŮűéŮť'å■Ţä¡■çŽĎè¡ňæ■ćåŠŇèőąçőŮïijŇèŕůä¡£çŤĺ `datetime` æĺąåİŮãĂĆ ærŤåęĆïijŇäÿžäžĘèąÍçd'žäÿĂäÿłæŮűéŮť'æőţïijŇåŘŕäžěåĹŻåžžäÿĂäÿł `timedelta` åőđä¿ŃïijŇářšåČŔäÿŃéİćè£Źæăůïijž

```
>>> from datetime import timedelta
>>> a = timedelta(days=2, hours=6)
>>> b = timedelta(hours=4.5)
>>> c = a + b
>>> c.days
2
>>> c.seconds
37800
>>> c.seconds / 3600
10.5
```

```
>>> c.total_seconds() / 3600
58.5
>>>
```

åęĆædIJä¡ăæČşèąÍçd'žæŇĞåőŽçŽĎæŮěæIJ§åŠŇæŮűéŮt'ïijŇåĚĹåĹŻåžžäÿĂäÿł datetime åőđä¿ŃçĎűåŘŐä¡£çŤÍæăĞåĞĘçŽĎæŢřå■ęè£ŘçőŮæİěæŞ■ä¡IJåőČäžñãĂĆærŤåęĆïijŽ

```
>>> from datetime import datetime
>>> a = datetime(2012, 9, 23)
>>> print(a + timedelta(days=10))
2012-10-03 00:00:00
>>>
>>> b = datetime(2012, 12, 21)
>>> d = b - a
>>> d.days
89
>>> now = datetime.today()
>>> print(now)
2012-12-21 14:54:43.094063
>>> print(now + timedelta(minutes=10))
2012-12-21 15:04:43.094063
>>>
```

åIJÍèőąçőŮçŽĎæŮűåĂŹïijŇéIJĂèęĄæşÍæĎŘçŽĎæŸŕ datetime äijŽèĞłåŁÍåd'ĎçŘĘéŮřåźt'ãĂĆærŤåęĆïijŽ

```
>>> a = datetime(2012, 3, 1)
>>> b = datetime(2012, 2, 28)
>>> a - b
datetime.timedelta(2)
>>> (a - b).days
2
>>> c = datetime(2013, 3, 1)
>>> d = datetime(2013, 2, 28)
>>> (c - d).days
1
>>>
```

èőÍèőž

åŕźåd'ğåd'ŽæŢřå§žæIJňçŽĎæŮěæIJ§åŠŇæŮűéŮt'åd'ĎçŘĘéŮőécŸïijŇ datetime æÍąåİŮåůšçżŔèűşåd'§äžĘãĂĆ åęĆædIJä¡ăéIJĂèęĄæL'ğèąŇæŻt'åŁăåd'■æİĆçŽĎæŮěæIJ§æŞ■ä¡IJïijŇærŤåęĆ åŘŕäżěèĂČèŹŚä¡£çŤÍ dateutilæÍąåİŮ

èőÿåd'Žçśżäijijç ŽĎæŮűéŮt'èőąçőŮåŘŕäżěä¡£çŤÍ dateutil.relativedelta() åĞ¡æŢřäžčæŻ£ãĂĆ ä¡ĘæŸŕïijŇæIJL'äÿĂçĆźéIJĂèęĄæşÍæĎŘçŽĎåřśæŸŕïijŇåőČäijŽåIJÍåd'ĎçŘĘæIJĹäż¡(è

```
>>> a = datetime(2012, 9, 23)
>>> a + timedelta(months=1)
```

```
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: 'months' is an invalid keyword argument for this function
>>>
>>> from dateutil.relativedelta import relativedelta
>>> a + relativedelta(months=+1)
datetime.datetime(2012, 10, 23, 0, 0)
>>> a + relativedelta(months=+4)
datetime.datetime(2013, 1, 23, 0, 0)
>>>
>>> # Time between two dates
>>> b = datetime(2012, 12, 21)
>>> d = b - a
>>> d
datetime.timedelta(89)
>>> d = relativedelta(b, a)
>>> d
relativedelta(months=+2, days=+28)
>>> d.months
2
>>> d.days
28
>>>
```

5.13 3.13 èőąçőŮæIJĂåŘŐäÿĂäÿłåŚĺäžŤçŽĎæŮěæIJ§

éŮőécŸ

ä¡ăéIJĂèęĄæ§ěæL'¿æŸ§æIJ§äÿ■æ§ŘäÿĂåď'ľæIJĂåŘŐåĞžçŐřçŽĎæŮěæIJ§ïijŇæŕŤåęĆæŸ§æIJ§äžŤãĂ

èğčåĘşæŰźæąĹ

PythonçŽĎ `datetime` æĺąåİŮäÿ■æIJL'åůěåĚůåĞ¡æŢřåŠŇçśżåŔŕäžěåÿőåŁľä¡ăæL'ğèąŇè£ŹæăůçŽĎèőäÿŃéIćæŸŕåŕźçśżäijijè£ŹæăůçŽĎéŮőécŸçŽĎäÿĂäÿłéĂŽçŤĺèğčåĘşæŰźæąĹïijŽ

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
"""
Topic: æIJĂåŘŐçŽĎåŚĺäžŤ
Desc :
"""
from datetime import datetime, timedelta

weekdays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday',
            'Friday', 'Saturday', 'Sunday']
```

```
def get_previous_byday(dayname, start_date=None):
    if start_date is None:
        start_date = datetime.today()
    day_num = start_date.weekday()
    day_num_target = weekdays.index(dayname)
    days_ago = (7 + day_num - day_num_target) % 7
    if days_ago == 0:
        days_ago = 7
    target_date = start_date - timedelta(days=days_ago)
    return target_date
```

åIJÍäžď'äžŠåijŔèğčéĞŁåŹÍäÿ■ä¡£çŤÍåęĆäÿŃïijŽ

```
>>> datetime.today() # For reference
datetime.datetime(2012, 8, 28, 22, 4, 30, 263076)
>>> get_previous_byday('Monday')
datetime.datetime(2012, 8, 27, 22, 3, 57, 29045)
>>> get_previous_byday('Tuesday') # Previous week, not today
datetime.datetime(2012, 8, 21, 22, 4, 12, 629771)
>>> get_previous_byday('Friday')
datetime.datetime(2012, 8, 24, 22, 5, 9, 911393)
>>>
```

åŔŕéĂL'çŽĎ `start_date` åŔĆæŢřåŔŕäžěçŤśåŔęåď'ŰäÿĂäÿł `datetime` åőđä¿ŃæÌěæŔŘä¿ŻãĂĆæŕŤåęĆïijŽ

```
>>> get_previous_byday('Sunday', datetime(2012, 12, 21))
datetime.datetime(2012, 12, 16, 0, 0)
>>>
```

èőÍèőž

äÿŁéÍćçŽĎçőŮæşŢåŐ§çŘĘæŸŕè£Źæăůç ŽĎïijŽåĚĹářĘåijĂåğŃæŮěæIJ§åŠŇçŻőæăĞæŮěæIJ§æŸăářĎ
çĎűåŘŐéĂŽè£ĞæÍąè£ŘçőÙèőąçőŮåĞžçŻőæăĞæŮěæIJ§èęAçżŔè£Ğåď'ŽářŚåď'l'æL'■èČ¡åĹřè¿¿åijĂåğŃæŮ

åęĆædIJä¡ăèęĄåČŔè£ŹæăůæL'ğèąŇåď'ğéĞŔçŽĎæŮěæIJ§èőąçőŮçŽĎèŕÌïijŇä¡ăæIJĂåě¡åőL'èčĚçňňäÿL
`python-dateutil` æÌěäžčæŻ£ãĂĆ æŕŤåęĆïijŇäÿŃéÌćæŸŕæŸŕä¡£çŤÍ `dateutil` æÍąåÌŮäÿ■çŽĎ `relativedelta()` åĞ¡æŢřæL'ğèąŇåŘŇæăůçŽĎèőąçőŮïijŽ

```
>>> from datetime import datetime
>>> from dateutil.relativedelta import relativedelta
>>> from dateutil.rrule import *
>>> d = datetime.now()
>>> print(d)
2012-12-23 16:31:52.718111

>>> # Next Friday
>>> print(d + relativedelta(weekday=FR))
2012-12-28 16:31:52.718111
```

```
>>>

>>> # Last Friday
>>> print(d + relativedelta(weekday=FR(-1)))
2012-12-21 16:31:52.718111
>>>
```

5.14 3.14 èőąçőŮå¡ŞåĽ■æIJĹäż¡çŽĎæŮěæIJ§èŇČåŻť

éŮőécŸ

ä¡ăçŽĎäžčçăĄéIJĂèęĄåIJĺå¡ŞåĽ■æIJĹäż¡äÿ■å¿łçŐŕæŕŔäÿĂåď'ľïijŇæČşæĽ¿åĹřäÿĂäÿłèőąçőŮè£Źäÿłaŕ

èğčåĘşæŰźæąĹ

åIJĺè£Źæăůç ŽĎæŮěæIJ§äÿŁå¿łçŐŕåźűéIJĂèęĄäžŇåĚĹæđĎéĂääÿĂäÿłåŇĚåŘńæĽĂæIJĽæŮěæIJ§çŽĎ
ä¡ääŔŕäžčåĚĹèőąçőŮåĞžåijĂåğŇæŮěæIJ§åŠŇçżŞæİ§æŮěæIJ§ïijŇ
çĎűåŘŐåIJĺä¡ăæ■čè£ŻçŽĎæŮűåĂŹä¡£çŤÍ `datetime.timedelta`
ářźèśąéĂŠåćđè£ŹäÿłæŮěæIJ§åŔŸéĞŔå■şåŔŕãĂĆ

äÿŇéİćæŸŕäÿĂäÿłæŐěåŘŮäżżæĎŔ `datetime` ářźèśąåźűè£ŤåŻđäÿĂäÿłçŤśå¡ŞåĽ■æIJĹäż¡åijĂåğŇæŮ

```
from datetime import datetime, date, timedelta
import calendar

def get_month_range(start_date=None):
    if start_date is None:
        start_date = date.today().replace(day=1)
    _, days_in_month = calendar.monthrange(start_date.year, start_
↪date.month)
    end_date = start_date + timedelta(days=days_in_month)
    return (start_date, end_date)
```

æIJĽäžĘè£ŹäÿłářśåŔŕäżěå¿ĹåőźæŸŞçŽĎåIJĺè£ŤåŻđçŽĎæŮěæIJ§èŇČåŻť äÿŁéİćåĄŽå¿łçŐŕæŞ■ä¡IJäž

```
>>> a_day = timedelta(days=1)
>>> first_day, last_day = get_month_range()
>>> while first_day < last_day:
...     print(first_day)
...     first_day += a_day
...
2012-08-01
2012-08-02
2012-08-03
2012-08-04
2012-08-05
2012-08-06
```

```
2012-08-07
2012-08-08
2012-08-09
#... and so on...
```

èőÍèőž

äÿŁéİćçŽĎäžčçăĄåĚĹèőąçőŮåĞžäÿĂäÿłåŕźåžŤæIJĹäż¡çňňäÿĂåď'ľçŽĎæŮĕæIJ§ãĂĆ äÿĂäÿłå£ńéĂ§çŽĎæŰźæşŢåřśæŸŕä¡£çŤĺ `date` æĹŰ `datetime` åŕźèśąçŽĎ `replace()` æŰźæşŢçőĂå■ŢçŽĎåřĘ `days` åśđæĂğèő¿ç¡őæĹŘ1å■şåŔŕãĂĆ `replace()` æŰźæşŢäÿĂäÿłåě¡åď'ĎåřśæŸŕåőČäijŽåĹŻåžžåŠŇä¡ăåijĂåğŃäijăåĚěåŕźèśąçśžåđŃçŻÿåŘŇçŽĎåŕźèśąãĂĆ æĽ'Ăäżěïij Ňåę Ćæđ IJè¿ŞåĚěåŔĆæŢřæŸŕäÿĂäÿł `date` åőđä¿Ńïij ŇéĆčäźĹçżŞæđIJäź§æŸŕäÿĂäÿł `date` åőđä¿ŃãĂĆ åŘŇæăůçŽĎïij Ňåę ĆæđIJè¿ŞåĚěæŸŕäÿĂäÿł `datetime` åőđä¿Ńïij ŇéĆčäźĹä¡ăå¿ŮåĹřçŽĎåřśæŸŕäÿĂäÿł `datetime` åőđä¿ŃãĂĆ

çĎűåŘŐïij Ňä¡£çŤĺ `calendar.monthrange()` åĞ¡æŢřæİěæĽ'¿åĞžèřěæIJĹçŽĎæĂżåď'ľæŢřãĂĆ äżżä¡ŢæŮűåĂŹåŔłèęĄä¡ăæČşèŐůå¿ŮæŮĕåŐĘä£ąæĄŕïij ŇéĆčäźĹ `calendar` æĺąåİŮåŕśéİđåÿÿæIJĽçŤĺäžĘãĂĆ `monthrange()` åĞ¡æŢřäijŽè£ŤåŻđåŇĚåŘńæŸ§æIJ§åŠŇèřěæIJĹåď'ľæŢřçŽĎåĚČçżĎãĂĆ

äÿĂæŮęèřěæIJĹçŽĎåď'ľæŢřåůšç§ëäžĘïij ŇéĆčäźĹçżŞæİ§æŮĕæIJ§åŕśåŘŕäżěéĂŽè£ĞåIJĺåijĂåğŃæŮĕæ æIJĽäÿłéIJĂèęĄæşĺæĎŔçŽĎæŸŕçżŞæİ§æŮĕæIJ§åźűäÿ■åŇĚåŘńåIJĺè£ŹäÿłæŮĕæIJ§èŇČåŻť'åĘĚ(äžŃåőđäÿ è£ŹäÿłåŠŇPythonçŽĎ `slice` äÿŐ `range` æŞ■ä¡IJèąŇäÿžä£ÍæŇĄäÿĂèĞť'ïijŇåŘŇæăůäź§äÿ■åŇĚåŘńçżŞåï

äÿžäžĘåIJĺæŮĕæIJ§èŇČåŻť'äÿŁå¿łçŐŕïijŇèęĄä¡£çŤĺåĹřæăĞåĞĘçŽĎæŢřå■ęåŠŇæŕŤè¿ĆæŞ■ä¡IJãĂĆ æŕŤåęĆïijŇåŔŕäžěåĹľçŤĺ `timedelta` åőđä¿ŃæÍěéĂŠáćđæŮĕæIJ§ïijŇåřŔäžŐåŔů<çŤĺæÍěæčĂæ§ëäÿĂäÿła

çŘĘæČşæČĚåĘţäÿŃïij ŇåęĆæđIJèČ¡äÿžæŮĕæIJ§è£■äžčåĹŻåžžäÿĂäÿłåŘŇåĘĚç¡őçŽĎ `range()` åĞ¡æŢřäÿĂæăůçŽĎåĞ¡æŢřåŕśåě¡äžĘãĂĆ åźÿè£ŘçŽĎæŸŕïijŇåŔŕäžěä¡£çŤĺäÿĂäÿłçŤ§æĹŘåŹĺçŽĎä¿Ńå■ŘïijŽ

```
def date_range(start, stop, step):
    while start < stop:
        yield start
        start += step
```

äÿŃéİćæŸŕä¡£çŤĺè£ŹäÿłçŤ§æĹŘåŹĺçŽĎä¿Ńå■ŘïijŽ

```
>>> for d in date_range(datetime(2012, 9, 1), datetime(2012,10,1),
                        timedelta(hours=6)):
...     print(d)
...
2012-09-01 00:00:00
2012-09-01 06:00:00
2012-09-01 12:00:00
2012-09-01 18:00:00
2012-09-02 00:00:00
2012-09-02 06:00:00
...
>>>
```

è£Źçğ■åőđçŐřäźŃæL’Ăäžěè£ŹäźĹçőĂå■ŢïijŇè£Ÿå¿Ůå¡ŠåŁ§äžŐPythonäÿ■çŽĎæŮěæIJ§åŠŇæŮűéŮť

5.15 3.15 å■Ůçňęäÿšè¡ňæ■ćäÿžæŮěæIJ§

éŮőécŸ

ä¡ăçŽĎåžŤçŤĺçĺŃåžŔæŐěåŔŮå■ŮçňęäÿšæăijåijŔçŽĎè¿ŞåĚěïijŇä¡ĘæŸŕä¡ăæČşářĘåőČäžňè¡ňæ■ćäÿž datetime åŕźèśąäžěä¿£åIJĺäÿŁéİćæL’ğèąŇéİđå■ŮçňęäÿšæŞ■ä¡IJãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺPythonçŽĎæăĞåĞĘæĺąåİŮ datetime åŔŕäžěå¿ĹåőźæŸ§çŽĎèğčåĘşè£ŹäÿłéŮőécŸãĂĆæŕŤåę

```
>>> from datetime import datetime
>>> text = '2012-09-20'
>>> y = datetime.strptime(text, '%Y-%m-%d')
>>> z = datetime.now()
>>> diff = z - y
>>> diff
datetime.timedelta(3, 77824, 177393)
>>>
```

èőĺèőž

datetime.strptime() æŰźæşŢæŤŕæŇĄå¿Ĺåd’ŽçŽĎæăijåijŔåŇŰäžčçăĄïijŇ æŕŤåęĆ %Y äžčèąĺ4ä¡■æŢřåźt’äž¡ïijŇ %m äžčèąĺäÿd’ä¡■æŢřæIJĹäž¡ãĂĆ è£ŸæIJL’äÿĂçĆźåĂijå¿ŮæşĺæĎŔçŽĎæŸŕè£ŹäžŻæăijåijŔåŇŰå■ăä¡■çňęäź§åŔŕäžěåŔ■è£Ğæĺěä¡£çŤĺïijŇårE

æŕŤåęĆïijŇåĄĞèő¿ä¡ăçŽĎäžčçăĄäÿ■çŤ§æĹŘäžĘäÿĂäÿł datetime åŕźèśąïijŇ ä¡ăæČşářĘåőČæăijåijŔåŇŰäÿžæijĆäžőæŸŞèŕźå¡ćåijŔåŘŐæŤ¿åIJĺèĞłåŁĺçŤ§æĹŘçŽĎä£ąäžűæĹŰèĂĚæŁěå

```
>>> z
datetime.datetime(2012, 9, 23, 21, 37, 4, 177393)
>>> nice_z = datetime.strftime(z, '%A %B %d, %Y')
>>> nice_z
'Sunday September 23, 2012'
>>>
```

è£ŸæIJL’äÿĂçĆźéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇ strptime() çŽĎæĂğèČ¡èęĄæŕŤä¡ăæČşèśąäÿ■çŽĎåůőå¿Ĺåd’ŽïijŇ åŻăäÿžåőČæŸŕä¡£çŤĺçžíPythonåőđçŐřïijŇåźűäÿŤå£Ì åęĆæđIJä¡ăèęĄåIJĺäžčçăĄäÿ■éIJĂèęĄèğčæđŘåd’ğéĞŔçŽĎæŮěæIJ§åźűäÿŤåůšçżŔç§ěéĄŞäžĘæŮěæIJ§å■Ůç æŕŤåęĆïijŇåęĆæđIJä¡ăåůšçżŔç§ěéĄŞæL’Ăäžěæ ŮěæIJ§æăijåijŔæŸŕ YYYY-MM-DD ïijŇä¡ăåŔŕäžěåČŔäÿŇéİćè£ŹæăůåőđçŐřäÿĂäÿłèğčæđŘåĞ¡æŢřïijŽ

```
from datetime import datetime
def parse_ymd(s):
```

```
    year_s, mon_s, day_s = s.split('-')
    return datetime(int(year_s), int(mon_s), int(day_s))
```

åőđéŹĚæţŃèřŢäÿ■ïijŇè£Źäÿłåĝ¡æŢřæŕŤ `datetime.strptime()` å£ń7åĂ■åď'ŽãĂĆ
åęĆæđIJä¡ăèęĄåď'ĎçŘĘåď'ğéĞŔçŽĎæűL'åŔŁåĹřæŮĕæIJ§çŽĎæŢřæ■őçŽĎèŕİïijŇéĆčäźĹæIJĂåě¡èĂČèŹŚäÿ

5.16 3.16 çżŞåŘĹæŮűåŇžçŽĎæŮĕæIJ§æŞ■ä¡IJ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłåőL'æŐŠåIJÍ2012åźť'12æIJĹ21æŮĕæŮl'äÿŁ9:30çŽĎçŤţèŕİäijŽèőőïijŇåIJřçĆźåIJÍèŁİåŁă
èĂŇä¡ăçŽĎæIJŃåŔŃåIJÍå■řåžęçŽĎçŔ■åŁăç¡ŮåřŤïijŇéĆčäźĹäżŰåžŤèŕĕåIJÍå¡ŞåIJřæŮűéŮť'åĞăçĆźåŔĆåŁă

èğčåĘşæŰźæąĹ

åŕźåĞääźŐæL'ĂæIJL'æűL'åŔŁåĹřæŮűåŇžçŽĎéŮőécŸïijŇä¡äéČ¡åžŤèŕĕä¡£çŤÍ
`pytz` æÍąåİŮãĂĆè£ŹäÿłåŇĚæŔŘä¿ŻäžĘOlsonæŮűåŇžæŢřæ■őåžŞïijŇ
åőČæŸŕæŮűåŇžä£ąæĄŕçŽĎäžŃåőđäÿŁçŽĎæăĞåĞĚïijŇåIJÍå¿Ĺåď'Žèŕ■èÍĂåŠŇæŞ■ä¡IJçşżçż§éĞŇéÍćéČ¡åŔ

`pytz` æÍąåİŮäÿĂäÿłäÿżèęĄçŤÍéĂŤæŸŕårĘ `datetime`
åžŞåĹŻåżžçŽĎçőĂå■ŢæŮĕæIJ§årźèśąæIJñåIJřåŇŰãĂĆ ærŤåęĆïijŇäÿŃéÍćåęĆä¡Ţèą[çď'žäÿĂäÿłèŁİåŁääŞěa

```
>>> from datetime import datetime
>>> from pytz import timezone
>>> d = datetime(2012, 12, 21, 9, 30, 0)
>>> print(d)
2012-12-21 09:30:00
>>>

>>> # Localize the date for Chicago
>>> central = timezone('US/Central')
>>> loc_d = central.localize(d)
>>> print(loc_d)
2012-12-21 09:30:00-06:00
>>>
```

äÿĂæŮęæŮĕæIJ§èćńæIJñåIJřåŇŰäžĘïijŇ åőČårśåŔŕäżěè¡ňæ■ćäÿžåĚűäżŰæŮűåŇžçŽĎæŮűéŮť'äžĘãĂ
äÿžäžĘå¿ŮåĹřçŔ■åŁăç¡ŮåřŤårźåžŤçŽĎæŮűéŮť'ïijŇä¡ääŔŕäżěè£Źæăůå[ŽïijŽ

```
>>> # Convert to Bangalore time
>>> bang_d = loc_d.astimezone(timezone('Asia/Kolkata'))
>>> print(bang_d)
2012-12-21 21:00:00+05:30
>>>
```

åęĆæđIJä¡ăæL'ŞçőŮåIJÍæIJñåIJřåŇŰæŮĕæIJ§äÿŁæL'ğèąŇèőąçőŮïijŇä¡ăéIJĂèęĄçL'źåĹńæşÍæĎŔåď'Ŕä
ærŤåęĆïijŇåIJÍ2013åźť'ïijŇç¿ŐåŻ¡æăĞåĞĘåď'Ŕäżď'æŮűæŮűéŮť'åijĂågŇäžŐæIJñåIJřæŮűéŮť'3æIJĹ13æŮè
åęĆæđIJä¡ăæ■čåIJÍæL'ğèąŇæIJñåIJřèőąçőŮïijŇä¡ääijŽå¿ŮåĹřäÿĂäÿłéŤŹèŕřãĂĆærŤåęĆïijŽ

```
>>> d = datetime(2013, 3, 10, 1, 45)
>>> loc_d = central.localize(d)
>>> print(loc_d)
2013-03-10 01:45:00-06:00
>>> later = loc_d + timedelta(minutes=30)
>>> print(later)
2013-03-10 02:15:00-06:00 # WRONG! WRONG!
>>>
```

çżŞæđIJéŤŹèŕŕæŸŕåŻääÿžåőČåźűæšąæIJL'èĂČèŹŠåIJÍæIJňåIJřæŮűéŮt'äÿ■æIJL'äÿĂářŔæŮűçŽĎèůşèů äÿžäžĘä£őæ■čè£ŹäÿłéŤŹèŕŕïijŇåŔŕäżčä¡£çŤÍæŮűåŇžářźèšą normalize() æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> from datetime import timedelta
>>> later = central.normalize(loc_d + timedelta(minutes=30))
>>> print(later)
2013-03-10 03:15:00-05:00
>>>
```

èőÍèőž

äÿžäžĘäÿ■èől'ä¡äèćnè£ŹäžŻäÿIJäÿIJåijĎçŽĎæŹŢåd'ť'è¡ňåŘŚïijŇåd'ĎçŘĘæIJňåIJřåŇŰæŮčæIJ§çŽĎéĂ åźűçŤÍåőČæİčæL'ğèąŇæL'ĂæIJL'çŽĎäÿ■éŮt'å■ŸåĆÍåŠŇæŞ■ä¡IJãĂĆæŕŤåęĆïijŽ

```
>>> print(loc_d)
2013-03-10 01:45:00-06:00
>>> utc_d = loc_d.astimezone(pytz.utc)
>>> print(utc_d)
2013-03-10 07:45:00+00:00
>>>
```

äÿĂæŮęè¡ňæ■ćäÿžUTCïijÑä¡ăářšäÿ■çŤÍåŐżæŃĚå£Čèů§åd'Ŕäžd'æŮűçŻÿåĚşçŽĎéŮőécŸäžĘãĂĆ åŻăæ■d'ïijÑä¡ăåŔŕäżčèů§äźŃåL'■äÿĂæăůæŤ¿å£ČçŽĎæL'ğèąŇåÿÿèğĄçŽĎæŮčæIJ§èőąçőŮãĂĆ å¡Şä¡äæČşářĘè¿ŞåĞžåŔŸäÿžæIJňåIJřæŮűéŮt'çŽĎæŮűåĂŹïijÑä¡£çŤÍåŘĹéĂĆçŽĎæŮűåŇžåŐżè¡ňæ■ćäÿŃå

```
>>> later_utc = utc_d + timedelta(minutes=30)
>>> print(later_utc.astimezone(central))
2013-03-10 03:15:00-05:00
>>>
```

å¡ŞæűL'åŔŁåĹřæŮűåŇžæŞ■ä¡IJçŽĎæŮűåĂŹïijÑæIJL'äÿłéŮőécŸåřśæŸŕæĹŚäžňåęĆä¡Ţå¿ŮåĹřæŮűåŇ æŕŤåęĆïijŇåIJÍè£Źäÿŀäż¿Ńå■Řäÿ■ïijÑæĹŚäžňåęĆä¡Ţç§ěéĄŞâĂIJAsia/KolkataâĂİåřśæŸŕå■ŕåžęåŕźåžŤçŽĎæ äÿžäžĘæ§ěæL'¿ïijŇåŔŕäżčä¡£çŤÍISO 3166åŻ¡åőűäžčçăĄä¡IJäÿžåĚşéŤőå■ŮåŐżæ§ěéŸĚå■ŮåĚÿ pytz.country_timezones ãĂĆæŕŤåęĆïijŽ

```
>>> pytz.country_timezones['IN']
['Asia/Kolkata']
>>>
```

æşÍïijŽå¡Şä¡ãéŸĚèŕžåĹŕè£ŹéĞŇçŽĎæŮűåĂŹïijŇæIJL’åŔŕèČ¡ pytz
æÍąåİŮåůšçżŔäÿ■åĘ■åžžèőőä¡£çŤÍäžĘïijŇåŽääÿžPEP431æŔŘåĞžäžĘæŻť’åĚĹè£ŻçŽĎæŮűåŇžæŤŕæŇĄãĂ
ä¡ĘæŸŕè£ŹéĞŇèřĹåĹřçŽĎå¿Ĺåd’ŽéŮőécŸè£ŸæŸŕæIJL’åŔĆèĂČäžůåĂijçŽĎ(æŕŤåęĆä¡£çŤÍUTCæŮěæIJ§ç

6 çňňåŻŻçńäïijŽè£■äžčåŹÍäÿŐçŤ§æĹŘåŹÍ

è£■äžčæŸŕPythonæIJĂåijžåd’ğçŽĎåŁ§èČ¡äźŃäÿĂãĂĆåĹİçIJŃèţůæİěïijŇä¡ăåŔŕèČ¡äijŽçőĂå■ŢçŽĎèőc
çĎűèĂŇïijŇçżÍéİđäžĚäžĚåŕśæŸŕåęĆæ■d’ïijŇè£ŸæIJL’å¿Ĺåd’Žä¡ăåŔŕèČ¡äÿ■ç§ěéĄŞçŽĎïijŇ
æŕŤåęĆåĹŻåžžä¡ăèĞłåůśçŽĎè£■äžčåŹÍåŕźèśąïijŇåIJÍitertoolsæÍąåİŮäÿ■ä¡£çŤÍæIJL’çŤÍçŽĎè£■äžčæÍąåijŔïij
è£ŹäÿĂçńäçŽőçŽĎåŕśæŸŕåŘŠä¡ăåśŢçd’žèů§è£■äžčæIJL’åĚşçŽĎåŘĎçğ■åÿÿèğĄéŮőécŸãĂĆ

Contents:

6.1 4.1 æĽŃåŁÍéĄ■åŐĘè£■äžčåŹÍ

éŮőécŸ

ä¡ăæČşéĄ■åŐĘäÿĂäÿłåŔŕè£■äžčåŕźèśąäÿ■çŽĎæL’ĂæIJL’åĚČçt’ăïijŇä¡ĘæŸŕå■t’äÿ■æČşä¡£çŤÍforå¿łçČ

èğčåĘşæŰźæąĹ

äÿžäžĘæL’ŃåŁÍçŽĎéĄ■åŐĘåŔŕè£■äžčåŕźèśąïijŇä¡£çŤÍ next()
åĞ¡æŢřåźűåIJÍäžčçăĄäÿ■æ■ŢèŐů StopIteration åijĆåÿÿãĂĆ
æŕŤåęĆïijŇäÿŃéİćçŽĎä¿Ńå■ŘæL’ŃåŁÍèŕžåŔŰäÿĂäÿłæŰĞäžűäÿ■çŽĎæL’ĂæIJL’èąŇïijŽ

```
def manual_iter():
    with open('/etc/passwd') as f:
        try:
            while True:
                line = next(f)
                print(line, end='')
        except StopIteration:
            pass
```

éĂŽåÿÿæİěèőšïijŇ StopIteration çŤÍæİěæŇĞçd’žè£■äžčçŽĎçżŞår¿ãĂĆ
çĎűèĂŇïijŇåęĆæđIJä¡ăæL’ŃåŁÍä¡£çŤÍäÿŁéİćæijŢçd’žçŽĎ next()
åĞ¡æŢřçŽĎèŕİïijŇä¡ăè£ŸåŔŕäžééĂŽè£Gè£ŤåŻđäÿĂäÿłæŇĞåőŽåĂijæİěæăĞèőřçżŞår¿ïijŇæŕŤåęĆ
None ãĂĆ äÿŃéİćæŸŕçd’žä¿ŃïijŽ

```
with open('/etc/passwd') as f:
    while True:
        line = next(f, None)
        if line is None:
            break
        print(line, end='')
```

èőÍèőž

åď ğåď ŽæŢřæČĚåĘţäÿŃïijŇæĹŚäžňäijŽä¡£çŤÍ `for` å¿łçŐŕèŕ■åŘěçŤÍæİěéĄ■åŐĘäÿĂäÿłåŘŕè£■äžčåŕźè
ä¡ĘæŸŕïijŇåĄűåřŤäź§éIJĂèęĄåŕźè£■äžčåĄŽæŻť åŁăçš¿çąőçŽĎæŐğåĹűïijŇè£ŹæŮűåĂŹäžĘèğčåžŢåśĆè£■ä

äÿŃéİćçŽĎäžď äžŠçď žä¿ŃåŘŚæĹŚäžňæijŤçď žäžĘè£■äžčæIJ§éÚť æĽĂåŘŚçŤ§çŽĎå§žæIJňçżĘèŁĆïijŽ

```
>>> items = [1, 2, 3]
>>> # Get the iterator
>>> it = iter(items) # Invokes items.__iter__()
>>> # Run the iterator
>>> next(it) # Invokes it.__next__()
1
>>> next(it)
2
>>> next(it)
3
>>> next(it)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
StopIteration
>>>
```

æIJňçńăæŐěäÿŃæİěåĞååřŔèŁĆäijŽæŻť æůśåĚěçŽĎèőšèğčè£■äžčçŻÿåĚşæŁĂæIJŕïijŇåĽ■æŔŘæŸŕä¡ăă
æĽĂäżěçąőä£Ia¡ăåůšçżŔæŁŁè£ŹçńăçŽĎåĘĚåőźçĽćçĽćèőřåIJĺå£Čäÿ■ãĂĆ

6.2 4.2 äžčçŘĘè£■äžč

éŮőécŸ

ä¡ăæđĎåžžäžĘäÿĂäÿłèĞłåőŽäźĽåőźåŹĺåŕźèśąïijŇéĞŇéİcåŇĚåŘńæIJĽåĹŮèąĺãĂĄåĚČçżĎæĹŰåĚűäżŮ
ä¡ăæČşçŻť æŐěåIJĺä¡äçŽĎè£ŹäÿłæŰřåőźåŹĺåŕźèśąäÿŁæĽğèąŇè£■äžčæŞ■ä¡IJãĂĆ

èğčåĘşæŰźæąĹ

åőđéŹĚäÿŁä¡ăåŘłéIJĂèęĄåőŽäźĽäÿĂäÿł `__iter__()`
æŰźæşŢïijŇářĘè£■äžčæŞ■ä¡IJäžčçŘĘåĹřåőźåŹĺåĘĚéČĺçŽĎåŕźèśąäÿŁåŐżãĂĆæŕŤåęĆïijŽ

```
class Node:
    def __init__(self, value):
        self._value = value
        self._children = []

    def __repr__(self):
        return 'Node({!r})'.format(self._value)

    def add_child(self, node):
        self._children.append(node)
```

```
    def __iter__(self):
        return iter(self._children)

# Example
if __name__ == '__main__':
    root = Node(0)
    child1 = Node(1)
    child2 = Node(2)
    root.add_child(child1)
    root.add_child(child2)
    # Outputs Node(1), Node(2)
    for ch in root:
        print(ch)
```

åIJÍäÿŁéİćäžčçăĄäÿ■ïijŇ __iter__() æŰźæşŢåŔłæŸŕçőĂå■ŢçŽĎåřĘè£■äżčèŕůæśĆäijăéĂŠçżŹåĘĚ
_children åśđæĂğãĂĆ

èőÍèőž

PythonçŽĎè£■äżčåŹÍå■ŔèőőéIJĂèęĄ __iter__() æŰźæşŢè£ŤåŻđäÿĂäÿłåőđçŐřäžĘ
__next__() æŰźæşŢçŽĎè£■äżčåŹÍåŕźèśąãĂĆ åęĆæđIJä¡ăåŔłæŸŕè£■äżčéĄ■åŐĘåĚűäżŰåőźåŹÍçŽĎåĘĚå

è£ŹéĞŇçŽĎ iter() åĞ¡æŢřçŽĎä¡£çŤÍçőĂåŇŰäžĘäžčçăĄïijŇ
iter(s) åŔłæŸŕçőĂå■ŢçŽĎéĂŽè£ĞèřČçŤÍ s.__iter__()
æŰźæşŢæİèè£ŤåŻđåŕźåžŤçŽĎè£■äżčåŹÍåŕźèśąïijŇ åřśèů§ len(s) äijŽèřČçŤÍ s.
__len__() åŐ§çŘĘæŸŕäÿĂæăůçŽĎãĂĆ

6.3 4.3 ä¡£çŤÍçŤ§æĹŘåŹÍåĹŻåžžæŰřçŽĎè£■äżčæÍąåijŔ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłèĞłåőŽäźL'è£■äżčæÍąåijŔïijŇèů§æŹőéĂŽçŽĎåĘĚç¡őåĞ¡æŢřæŕŤåęĆ
range() , reversed() äÿ■äÿĂæăůãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşåőđçŐřäÿĂçğ■æŰřçŽĎè£■äżčæÍąåijŔïijŇä¡£çŤÍäÿĂäÿłçŤ§æĹŘåŹÍåĞ¡æŢřæİèåőŽäźL'å
äÿŃéİćæŸŕäÿĂäÿłçŤ§äžğæ§ŘäÿłèŇČåŻt'åĘĚæţőçĆźæŢřçŽĎçŤ§æĹŘåŹÍïijŽ

```
def frange(start, stop, increment):
    x = start
    while x < stop:
        yield x
        x += increment
```

äÿžäžĘä¡£çŤÍè£ŹäÿłåĞ¡æŢřïijŇ ä¡ăåŔřäžěçŤÍforå¿łçŐřè£■äžčåőČæĹŰèĂĚä¡£çŤÍåĚűäžŰæŐěåŔŮäÿĂä
sum() ,list() ç■Ľ)ãĂĆçď žä¿ŃåęĆäÿŃïijŽ

```
>>> for n in frange(0, 4, 0.5):
...     print(n)
...
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
>>> list(frange(0, 1, 0.125))
[0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875]
>>>
```

èőÍèőž

äÿĂäÿłåĞ¡æŢřäÿ■éIJĂèęĄæIJL'äÿĂäÿł yield èŕ■åŔěå■şåŔřårĘåĚűè¡ňæ■ćäÿžäÿĂäÿłçŤ§æĹŘåŹÍãĂĆ
èů§æŹőéĂŽåĞ¡æŢřäÿ■åŘŇçŽĎæŸřïijŇçŤ§æĹŘåŹÍåŔłèČ¡çŤÍäžŐè£■äžčæŞ■ä¡IJãĂĆ
äÿŃéÍćæŸŕäÿĂäÿłåőđéłŇïijŇåŘŚä¡ăåśŢçď žè£ŹæăůçŽĎåĞ¡æŢřåžŢåśĆåůěä¡IJæIJžåĹűïijŽ

```
>>> def countdown(n):
...     print('Starting to count from', n)
...     while n > 0:
...         yield n
...         n -= 1
...     print('Done!')
...

>>> # Create the generator, notice no output appears
>>> c = countdown(3)
>>> c
<generator object countdown at 0x1006a0af0>

>>> # Run to first yield and emit a value
>>> next(c)
Starting to count from 3
3

>>> # Run to the next yield
>>> next(c)
2

>>> # Run to next yield
>>> next(c)
1
```

```
>>> # Run to next yield (iteration stops)
>>> next(c)
Done!
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
StopIteration
>>>
```

äÿĂäÿłçŢ§æĹŘåŹĺåĞ¡æŢřäÿżèęĄçĽźå¿ĄæŸŕåőČåŔłäijŽåŻđåžŢåIJĺè£■äżčäÿ■ä¡£çŢĺåĹřçŽĎ
next æŞ■ä¡IJãĂĆ äÿĂæŮęçŢ§æĹŘåŹĺåĞ¡æŢřè£ŢåŻđéĂĂåĞžïijŇè£■äżčçżĹæ■ćãĂĆæĹŚäžňåIJĺè£■äżčäÿ■é

6.4 4.4 åőđçŐřè£■äżčåŹĺå■Ŕèőő

éŮőécŸ

ä¡ăæČşæđĎåžžäÿĂäÿłèČ¡æŢŕæŇĄè£■äżčæŞ■ä¡IJçŽĎèĞłåőŽäźĽ'åŕźèśąïijŇåźűåÿŇæIJŻæĽ'¿åĹřäÿĂäÿł

èğčåĘşæŰźæąĹ

çŻőåĽ'■äÿžæ■ćïijŇåIJĺäÿĂäÿłåŕźèśąäÿŁåőđçŐřè£■äżčæIJĂçőĂå■ŢçŽĎæŰźåijŔæŸŕä¡£çŢĺäÿĂäÿłçŢ§æ
åIJĺ4.2åřŔèŁĆäÿ■ïijŇä¡£çŢĺNodeçśżæİěèąĺçđ'žæăŚå¡ćæŢřæ■őçżŞæđĎãĂĆä¡ăåŔŕèČ¡æČşåőđçŐřäÿĂäÿłäżěa
äÿŃéİćæŸŕäžčçăĄçđ'žä¿ŃïijŽ

```
class Node:
    def __init__(self, value):
        self._value = value
        self._children = []

    def __repr__(self):
        return 'Node({!r})'.format(self._value)

    def add_child(self, node):
        self._children.append(node)

    def __iter__(self):
        return iter(self._children)

    def depth_first(self):
        yield self
        for c in self:
            yield from c.depth_first()

# Example
if __name__ == '__main__':
    root = Node(0)
    child1 = Node(1)
    child2 = Node(2)
```

```
    root.add_child(child1)
    root.add_child(child2)
    child1.add_child(Node(3))
    child1.add_child(Node(4))
    child2.add_child(Node(5))

    for ch in root.depth_first():
        print(ch)
    # Outputs Node(0), Node(1), Node(3), Node(4), Node(2), Node(5)
```

åIJÍè£Źæőţäžčçă Ąäÿ■ïijŇdepth_first() æŰźæşŢçőĂå■ŢçŻť'èğĆãĂĆ åőČéęŰåĚĹè£ŤåŻđèĞłåůśæIJňèžńåźűè£■äžčæŕŔäÿĂäÿłå■ŘèŁĆçĆźåźű éĂŽè£ĞèřČçŤĺå■ŘèŁĆçĆźçŽĎ depth_first() æŰźæşŢ(ä¡£çŤÍ yield from èŕ■åŘě)è£ŤåŻđåŕźåžŤåĚČçť'ããĂĆ

èőÍèőž

PythonçŽĎè£■äžčå■Ŕèőőèę Ąæś ĆäÿĂäÿł __iter__() æŰźæşŢè£ŤåŻđäÿĂäÿłçL'źæőŁçŽĎè£■äžčåŹĺåŕźèśąïijŇ è£Źäÿłè£■äžčåŹĺåŕźèśąåőđçŐřäžĘ __next__() æŰźæşŢåźűéĂŽè£Ğ StopIteration åijĆåÿÿæăĞèŕĘè£■äžčçŽĎåőŇæĹŘãĂĆ ä¡ĘæŸŕïijŇåőđçŐřè£ŹäžŽéĂŽåÿÿäijŽæŕŤè¿ČçźĄçŘŘãĂĆ äÿŃéİćæĹŚäžňæijŤçd'žäÿŃè£Źçğ■æŰźåijŔïijŇå depth_first() æŰźæşŢïijŽ

```
class Node2:
    def __init__(self, value):
        self._value = value
        self._children = []

    def __repr__(self):
        return 'Node({!r})'.format(self._value)

    def add_child(self, node):
        self._children.append(node)

    def __iter__(self):
        return iter(self._children)

    def depth_first(self):
        return DepthFirstIterator(self)


class DepthFirstIterator(object):
    '''
    Depth-first traversal
    '''

    def __init__(self, start_node):
        self._node = start_node
        self._children_iter = None
```

```
        self._child_iter = None

    def __iter__(self):
        return self

    def __next__(self):
        # Return myself if just started; create an iterator for␣
↪children
        if self._children_iter is None:
            self._children_iter = iter(self._node)
            return self._node
        # If processing a child, return its next item
        elif self._child_iter:
            try:
                nextchild = next(self._child_iter)
                return nextchild
            except StopIteration:
                self._child_iter = None
                return next(self)
        # Advance to the next child and start its iteration
        else:
            self._child_iter = next(self._children_iter).depth_
↪first()
            return next(self)
```

DepthFirstIterator çśżåŠÑäÿŁéÍćä¡£çŤÍçŤ§æĹŘåŹÍçŽĎçL'ĹæIJňåůěä¡IJåŐ§çŘĘçśżäijijïijÑ
ä¡ĘæŸŕåőČåĘŹèţůæÍěå¿ĹçźAçŘŘïijÑåŽääÿžè£■äžčåŹÍå£Ěéążå IJÍè£■äžčåd'ĎçŘĘè£ĞçÍÑäÿ■çżt'æŁd'åd'ğéČ
åÍęçŹ¡æÍěèőšïijÑæšąäžžæĎ£æĎŔåĘŹè£ŹäźĹæŹęæűl'çŽĎäžčçăĄãĂĆåřĘä¡äçŽĎè£■äžčåŹÍåőŽäźL'äÿžäÿĂäÿ

6.5 4.5 åŔ■åŘŚè£■äžč

éŮőécŸ

ä¡ăæČşåŔ■æŰźåŘŚè£■äžčäÿĂäÿłåžŔåĹŮ

èğčåĘşæŰźæąĹ

ä¡£çŤÍåĘĚç¡őçŽĎ `reversed()` åĞ¡æŢřïijÑæŕŤåęĆïijŽ

```
>>> a = [1, 2, 3, 4]
>>> for x in reversed(a):
...     print(x)
...
4
3
2
1
```

åŕ■åŘŝè£■äžčäžĚäžĚå¡Şåŕźèśąçž Ďåď ğåřŕåŕŕéčĎåĚĹçąőåőŽæĹŰèĂĚåŕźèśąåőđçŐřäžĘ
__reversed__() çŽĎçĽźæőŁæŰźæşŢæŮűæĽ■èČ¡çŤ§æŢĹãĂĆ
åęĆæđIJäÿď èĂĚéČ¡äÿ■çňęåŘĹïijŇéĆčä¡ăå£Ěéążå ĚĹåřĘåŕźèśąè¡ňæ■ćäÿžäÿĂäÿłåĹŮèąÍæĽ■èąŇïijŇæŕŤåęĆ

```
# Print a file backwards
f = open('somefile')
for line in reversed(list(f)):
    print(line, end='')
```

èęĄæşÍæĎŘçŽĎæŸŕåęĆæđIJåŘŕè£■äžčåŕźèśąåĚČçť ăå¿Ĺåď ŽçŽĎèŕÍïijŇåřĘåĚűéčĎåĚĹè¡ňæ■ćäÿžäÿĂ

èőÍèőž

å¿Ĺåď ŽçÍŃåžŘåŚŸåźűäÿ■ç§ěéĄŞåŘŕäžěéĂŽè£ĞåIJÍèĞłåőŽäźĽçśžäÿŁåőđçŐř
__reversed__() æŰźæşŢæÌěåőđçŐřåŕ■åŘŝè£■äžčãĂĆæŕŤåęĆïijŽ

```
class Countdown:
    def __init__(self, start):
        self.start = start

    # Forward iterator
    def __iter__(self):
        n = self.start
        while n > 0:
            yield n
            n -= 1

    # Reverse iterator
    def __reversed__(self):
        n = 1
        while n <= self.start:
            yield n
            n += 1

for rr in reversed(Countdown(30)):
    print(rr)
for rr in Countdown(30):
    print(rr)
```

åőŽäźĽäÿĂäÿłåŕ■åŘŝè£■äžčåŹĺåŕŕäžěä¡£å¿ŮäžčçăĄéÌđåÿÿçŽĎéńŸæŢĹïijŇ
åŻääÿžåőČäÿ■åĘ■éIJĂèęĄåřĘæŢřæ■őåąńåĚĚåĹřäÿĂäÿłåĹŮèąÍäÿ■çĎűåŘŐåĘ■åŐżåŕ■åŘŝè£■äžčè£ŹäÿłåĹ

6.6 4.6 åÿęæIJĽåď ŰéČÍçŁűæĂĄçŽĎçŤ§æĹŘåŹĺåĞ¡æŢř

éŮőéćŸ

ä¡ăæČşåőŽäźĽäÿĂäÿłçŤ§æĹŘåŹĺåĞ¡æŢřïijŇä¡ĘæŸŕåőČäijŽèřČçŤÍæ§Řäÿłä¡ăæČşæŽť éIJšçżŹçŤÍæĹůă

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăæČşèőľ'ä¡ăçŽĎçŤ§æĹŘåŹĺæŽť'éIJšåď'ŰéČĺçŁűæĂĄçżŹçŤĺæĹůïijŇ
åĹńå£ŸäžĘä¡ăåŔŕäżěçőĂå■ŢçŽĎåřĘåőČåőđçŐřäÿžäÿĂäÿłçśžïijŇçĎűåŘŐæŁŁçŤ§æĹŘåŹĺåĞ¡æŢřæŤ¿åĹř
`__iter__()` æŰźæşŢäÿ■è£ĞåŐżãĂĆæŕŤåęĆïijŽ

```
from collections import deque

class linehistory:
    def __init__(self, lines, histlen=3):
        self.lines = lines
        self.history = deque(maxlen=histlen)

    def __iter__(self):
        for lineno, line in enumerate(self.lines, 1):
            self.history.append((lineno, line))
            yield line

    def clear(self):
        self.history.clear()
```

äÿžäžĘä¡£çŤĺè£ŹäÿłçśžïijŇä¡ăåŔŕäżěåřĘåőČå¡ŞåĄŽæŸŕäÿĂäÿłæŹőéĂŽçŽĎçŤ§æĹŘåŹĺåĞ¡æŢřãĂĆ
çĎűěĂŇïijŇçŤśäžŐåŔŕäżěåĹŻåžžäÿĂäÿłåőđä¿ŃåŕźèśąïijŇäžŐæŸŕä¡ăåŔŕäżěèő£éŮőåĘĚéČĺåšđæĂğåĂijïijŇ
æŕŤåęĆ `history` åšđæĂğæĹŰèĂĚæŸŕ `clear()` æŰźæşŢãĂĆäžčçăĄçď'žä¿ŃåęĆäÿŃïijŽ

```
with open('somefile.txt') as f:
    lines = linehistory(f)
    for line in lines:
        if 'python' in line:
            for lineno, hline in lines.history:
                print('{}:{}'.format(lineno, hline), end='')
```

èőĺèőž

åĚşäžŐçŤ§æĹŘåŹĺïijŇå¿Ĺåőźæ˜ŞæŐL'è£ŻåĞ¡æŢřæŮăæL'Ăäÿ■èČ¡çŽĎéŹůéŸśãĂĆ
åęĆædIJçŤ§æĹŘåŹĺåĞ¡æŢřéIJĂèęĄèů§ä¡ăçŽĎçĺŃåžŘåĚűäžŰéČĺåĹĘæL'Şäžď'éĄŞçŽĎèŕĺ(æŕŤåęĆæŽť'éIJšă
åŘŕèČ¡äijŽåŕijèĞť'ä¡ăçŽĎäžčçăĄåijĆåÿÿçŽĎåď'■æĺĆãĂĆ åęĆædIJæŸŕè£ŹçĞ■æČĚåĘţçŽĎèŕĺïijŇåŘŕäżěèĂĆ
åIJĺ `__iter__()` æŰźæşŢäÿ■åőŽäźL'ä¡ăçŽĎçŤ§æĹŘåŹĺäÿ■äijŽæŤźåŔŸä¡ääżä¡ŢçŽĎçőŮæşŢéĂżè¿ŚãĂĆ
çŤśäžŐåőČæŸŕçśžçŽĎäÿĂéČĺåĹĘïijŇæL'ĂäžěåĚĄèőÿä¡ăåőŽäźL'åŘĎçĞ■åšđæĂğåŠŇæŰźæşŢæİěä¿ŻçŤĺæĹ

äÿĂäÿłéIJĂèęĄæşĺæĎŘçŽĎåřŘåIJřæŰźæŸŕïijŇåęĆædIJä¡ăåIJĺè£■äžčæŞ■ä¡IJæŮűäÿ■ä¡£çŤĺforå¿łçŐŕè
`iter()` åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> f = open('somefile.txt')
>>> lines = linehistory(f)
>>> next(lines)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: 'linehistory' object is not an iterator
```

```
>>> # Call iter() first, then start iterating
>>> it = iter(lines)
>>> next(it)
'hello world\n'
>>> next(it)
'this is a test\n'
>>>
```

6.7 4.7 è£■äżčåŹĺåĹĞçĽĞ

éŮőécŸ

ä¡ăæČşå¿ŮåĹřäÿĂäÿłçŤśè£■äżčåŹĺçŤ§æĹŘçŽĎåĹĞçĽĞåŕźèśąïijŇä¡ĘæŸŕæăĞåĞĘåĹĞçĽĞæŞ■ä¡IJåź

èğčåĘşæŰźæąĹ

åĞ¡æŢř `itertools.islice()` æ■čåě¡éĂĆçŤĺäžŐåIJĺè£■äżčåŹĺåŠŇçŤ§æĹŘåŹĺäÿŁåĄŽåĹĞçĽĞæ

```
>>> def count(n):
...     while True:
...         yield n
...         n += 1
...
>>> c = count(0)
>>> c[10:20]
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: 'generator' object is not subscriptable

>>> # Now using islice()
>>> import itertools
>>> for x in itertools.islice(c, 10, 20):
...     print(x)
...
10
11
12
13
14
15
16
17
18
19
>>>
```

èőÍèőž

è£■äžčåŹÍåŠŇçŤ§æĹŘåŹÍäÿ■èČ¡ä¡£çŤÍæăĞåĞĘçŽĎåĹĞçL'ĞæŞ■ä¡IJïijŇåŻăäÿžåőČäžňçŽĎéŢ£åžęäžŇ
åĞ¡æŢř `islice()` è£ŤåŻđäÿĂäÿłåŘŕäžěçŤ§æĹŘæŇĞåőŽåĚČçť'ăçŽĎè£■äžčåŹÍïijŇåőČéĂŽè£ĞéĄ■åŐĘå
çĎűåŘŐæL'■åijĂåğŃäÿĂäÿłäÿłçŽĎè£ŤåŻđåĚČçť'ăïijŇåźűçŻť'åĹŕåĹĞçL'ĞçżŞæİ§çť'ćåijŢä¡■ç¡őãĂĆ

è£ŹéĞŇèęĄçİĂéĞ■åijžèřČçŽĎäÿĂçĆźæŸŕ `islice()`
äijŽæűĹèĂŮæŐL'äijăåĚěçŽĎè£■äžčåŹÍäÿ■çŽĎæŢřæ■őãĂĆ å£Ěéążè ĂČèŹŚåĹřè£■äžčåŹÍæŸŕäÿ■åŘréĂĘçŽ
æL'Ăäžěåę ĆæđIJä¡äéIJĂèęĄäźŃåŘŐåĘ■æňąèő£éŮőè£Źäÿłè£■äžčåŹÍçŽĎèŕİïijŇéĆčä¡ääřšå¿ŮåĚĹářĘåőČéĞ

6.8 4.8 èůşè£ĞåŘŕè£■äžčářźèšąçŽĎåijĂåğŃéČĺåĹĘ

éŮőécŸ

ä¡ăæČşéĄ■åŐĘäÿĂäÿłåŘŕè£■äžčářźèšąïijŇä¡ĘæŸŕåőČåijĂåğŃçŽĎæ§ŘäžŽåĚČçť'ăä¡ääźűäÿ■æĎ§åĚt'è

èğčåĘşæŰźæąĹ

`itertools` æĺąåİŮäÿ■æIJL'äÿĂäžŽåĞ¡æŢřåŘŕäžěåőŇæĹŘè£ŹäÿłäżżåŁąãĂĆ
éęŰåĚĹäżŃçż■çŽĎæŸŕ `itertools.dropwhile()` åĞ¡æŢřãĂĆä¡£çŤÍæŮűïijŇä¡ăçżŹåőČäijåéĂŠäÿĂäÿ
åőČäijŽè£ŤåŻđäÿĂäÿłè£■äžčåŹÍářźèšąïijŇäÿćåijČåŐ§æIJL'åžŘåĹŮäÿ■çŻť'åĹřåĞ¡æŢřè£ŤåŻđFlaseäźŃåL'■ç

äÿžäžĘæijŤçđ'žïijŇåĄĞåőŽä¡ăåIJÍèŕżåŘŰäÿĂäÿłåijĂåğŃéČÍåĹĘæŸŕåĞăèąŇæşÍéĞŁçŽĎæžŘæŰĞäžűã

```
>>> with open('/etc/passwd') as f:
...     for line in f:
...         print(line, end='')
...
##
# User Database
#
# Note that this file is consulted directly only when the system is␣
↪running
# in single-user mode. At other times, this information is provided␣
↪by
# Open Directory.
...
##
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
...
>>>
```

åęĆæđIJä¡ăæČşèůşè£ĞåijĂåğŃéČÍåĹĘçŽĎæşÍéĞŁèąŇçŽĎèŕİïijŇåŘŕäžěè£ŹæăůåĄŽïijŽ

```
>>> from itertools import dropwhile
>>> with open('/etc/passwd') as f:
...     for line in dropwhile(lambda line: line.startswith('#'), f):
```

```
...         print(line, end='')
...
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
...
>>>
```

è£Źäÿłä¿Ńå■ŘæŸŕå§žäžŐæăźæ■őæ§ŘäÿłæţŃèŕŢåĞ¡æŢřèůşè£ĞåijĂåğŃçŽĎåĚČçť ăãĂĆ åęĆæđIJä¡ăåůšçżŔæŸŐçąőç§ěéĄŞäžĘèęĄèůşè£ĞçŽĎåĚČçť ăçŽĎäÿłæŢřçŽĎèŕÍïijŇéĆčäźĹåŔŕäżěä¡£çŤÍ `itertools.islice()` æİěäżčæŻ£ãĂĆæŕŤåęĆïijŽ

```
>>> from itertools import islice
>>> items = ['a', 'b', 'c', 1, 4, 10, 15]
>>> for x in islice(items, 3, None):
...     print(x)
...
1
4
10
15
>>>
```

åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇ `islice()` åĞ¡æŢřæIJĂåŘŐéĆčäÿł `None` åŔĆæŢřæŇĞåőŽäžĘä¡ăèęĄèŐůåŔŰäżŐçňň3äÿłåĹřæIJĂåŘŐçŽĎæĽĂæIJĽåĚČçť ăïijŇ åęĆæđIJ `None` åŠŇ3çŽĎä¡■ç¡őåŕźèřČïijŇæĎŔæĂİåřśæŸŕäżĚäżĚèŐůåŔŰåĽ■äÿĽäÿłåĚČçť ăæĄřæĄřçŻÿåŔ (è£Źäÿłèů§åĹĞçĽĞçŽĎçŻÿåŔ■æŞ■ä¡IJ `[3:]` åŠŇ `[:3]` åŐ§çŘĘæŸŕäÿĂæăůçŽĎ)ãĂĆ

èőĺèőž

åĞ¡æŢř `dropwhile()` åŠŇ `islice()` åĚűåőđåřśæŸŕäÿď äÿłåÿőåŁľ åĞ¡æŢřïijŇäÿžçŽĎåřśæŸŕéĄ£åĽ

```
with open('/etc/passwd') as f:
    # Skip over initial comments
    while True:
        line = next(f, '')
        if not line.startswith('#'):
            break

    # Process remaining lines
    while line:
        # Replace with useful processing
        print(line, end='')
        line = next(f, None)
```

èůşè£ĞäÿĂäÿłåŔŕè£■äżčåŕźèśąçŽĎåijĂåğŃéČĺåĹĘèů§éĂŽåÿÿçŽĎè£ĞæżďæŸŕäÿ■åŘŇçŽĎãĂĆ æŕŤåęĆïijŇäÿŁè£řäżčçăĄçŽĎçňňäÿĂäÿłéČĺåĹĘåŔŕèČ¡äijŽè£ŹæăůéĞ■åĘŹïijŽ

```
with open('/etc/passwd') as f:
    lines = (line for line in f if not line.startswith('#'))
```

```
    for line in lines:
        print(line, end='')
```

è£ŹæăůåĘŹçąőåőđåŘŕäžěèůşè£ĞåijĂåğŃéČĺåĹĘçŽĎæşĺéĞŁèąŇïijŇä¡ĘæŸŕåŘŇæăůäź§äijŽèůşè£ĞæŰ
æ■ćåŘěèŕİèőšïijŇæĹŚäžňçŽĎèğčåĘşæŰźæąĹæŸŕäžĚäžĚèůşè£ĞåijĂåğŃéČĺåĹĘæżąèűşæţŃèŕŢæİąäžűçŽĎè

æIJĂåŘŐéIJĂèęĄçİĂéĞ■åijžèřČçŽĎäÿĂçĆźæŸŕïijŇæIJňèŁĆçŽĎæŰźæąĹéĂĆçŤĺäžŐæĽ’ĂæIJL’åŘŕè£
æŕŤåęĆçŤ§æĹŘåŹĺïijŇæŰĞäžűåŘŁåĚűçśžäijijçŽĎåŕźèśąãĂĆ

6.9 4.9 æŐŠåĹŮçżĎåŘĹçŽĎè£■äžč

éŮőécŸ

ä¡ăæČşè£■äžčéĄ■åŐĘäÿĂäÿłéŻĘåŘĹäÿ■åĚČçť ăçŽĎæĽ’ĂæIJL’åŘŕèČ¡çŽĎæŐŠåĹŮæĹŰçżĎåŘĹ

èğčåĘşæŰźæąĹ

itertoolsæĺąåİŮæŔŘä¿ŻäžĘäÿL’äÿłåĞ¡æŢřæİěèğčåĘşè£ŹçśźéŮőécŸãĂĆ
åĚűäÿ■äÿĂäÿłæŸŕ `itertools.permutations()` ïijŇ
åőČæŐěåŘŮäÿĂäÿłéŻĘåŘĹåźűäžğçŢ§äÿĂäÿłåĚČçżĎåžŘåĹŮïijŇæŕŘäÿłåĚČçżĎçŢśéŻĘåŘĹäÿ■æĽ’ĂæIJL’
äź§åřśæŸŕèŕť éĂŽè£ĞæĽ’Şäźśé ŻĘåŘĹäÿ■åĚČçť ăæŐŠåĹŮéąžåžŘçŢ§æĹŘäÿĂäÿłåĚČçżĎïijŇæŕŤåęĆïijŽ

```
>>> items = ['a', 'b', 'c']
>>> from itertools import permutations
>>> for p in permutations(items):
...     print(p)
...
('a', 'b', 'c')
('a', 'c', 'b')
('b', 'a', 'c')
('b', 'c', 'a')
('c', 'a', 'b')
('c', 'b', 'a')
>>>
```

åęĆæđIJä¡ăæČşå¿ŮåĹřæŇĞåőŽéŢ£åžęçŽĎæĽ’ĂæIJL’æŐŠåĹŮïijŇä¡ääŘŕäžěäijăéĂŠäÿĂäÿłåŘŕéĂL’çŽ

```
>>> for p in permutations(items, 2):
...     print(p)
...
('a', 'b')
('a', 'c')
('b', 'a')
('b', 'c')
('c', 'a')
('c', 'b')
>>>
```

ä¡£çŤÍ `itertools.combinations()` åŔŕå¿ŮåĹřè¿ŞåĚěéŻĘåŘĹäÿ■åĚČçť ăçŽĎæĽĂæIJĽçŽĎçżĽ

```
>>> from itertools import combinations
>>> for c in combinations(items, 3):
...     print(c)
...
('a', 'b', 'c')

>>> for c in combinations(items, 2):
...     print(c)
...
('a', 'b')
('a', 'c')
('b', 'c')

>>> for c in combinations(items, 1):
...     print(c)
...
('a',)
('b',)
('c',)
>>>
```

åŕźäžŐ `combinations()` æÍěèőšïijŇåĚČçť ăçŽĎéąžåžŔåůšçżŔäÿ■éĞ■èęĄäžĘãĂĆ äź§åřśæŸŕèŕťˇïijŇçżĎåŘĹ `('a', 'b')` èů§ `('b', 'a')` åĚűåőđæŸŕäÿĂæăůçŽĎ(æIJĂçżĹåŔłäijŽè¿ŞåĞžåĚűäÿ■äÿĂäÿł)ãĂĆ

åIJÍèőąçőŮçżĎåŘĹçŽĎæŮűåĂŹïijŇäÿĂæŮęåĚČçť ăèćńéĂĽˇåŔŰåřśäijŽäżŐåĂŹéĂĽˇäÿ■åĽˇŤéŹđˇæŐĽˇ èĂŇåĞ¡æŢř `itertools.combinations_with_replacement()` åĚĄèőÿåŘŇäÿĂäÿłåĚČçť ăèćńéĂĽˇæŃĺˇåđˇŽæňąïijŇæŕŤåęĆïijŽ

```
>>> for c in combinations_with_replacement(items, 3):
...     print(c)
...
('a', 'a', 'a')
('a', 'a', 'b')
('a', 'a', 'c')
('a', 'b', 'b')
('a', 'b', 'c')
('a', 'c', 'c')
('b', 'b', 'b')
('b', 'b', 'c')
('b', 'c', 'c')
('c', 'c', 'c')
>>>
```

èőÍèőž

è£ŹäÿĂåřŔèŁĆæĹŚäżňåŘŚä¡ăåśŢçđˇžçŽĎäżĚäżĚæŸŕ `itertools` æÍąåİŮçŽĎäÿĂéČÍåĹĘåŁ§èČ¡ãĂĆ år¡çőąä¡ăäź§åŔŕäžèèĞłåůśæĽˇŃåŁÍåőđçŐřæŐŠåĹŮçżĎåŘĹçőŮæşŢïijŇä

å¡ŞæĹŚäžňçćřåĽřçIJŃäÿŁåŐżæIJL’äžŻåď■æİĆçŽĎè£■äžčéŮőécŸæŮűïijŇæIJĂåě¡åŔŕäžěåĚĹåŐżçIJŃçIJŃi
åęĆæđIJè£ŹäÿłéŮőécŸå¿ĹæŹőéĄ■ïijŇéĆčäźĹå¿ĹæIJL’åŔŕèČ¡äijŽåIJĺéĞŇéİćæL’¿åĹřèğčåĘşæŰźæąĹïijĄ

6.10 4.10 åžŔåĹŮäÿŁçť’ćåijŢåĂijè£■äžč

éŮőécŸ

ä¡ăæČşåIJĺè£■äžčäÿĂäÿłåžŔåĹŮçŽĎåŘŇæŮűèů§èÿłæ■čåIJĺèćńåď’ĎçŘĘçŽĎåĚČçť’ăçť’ćåijŢãĂĆ

èğčåĘşæŰźæąĹ

åĘĚç¡őçŽĎ `enumerate()` åĞ¡æŢřåŔŕäžěå¿Ĺåě¡çŽĎèğčåĘşè£ŹäÿłéŮőécŸïijŽ

```
>>> my_list = ['a', 'b', 'c']
>>> for idx, val in enumerate(my_list):
...     print(idx, val)
...
0 a
1 b
2 c
```

äÿžäžĘæŇL’äijăçż§èąŇåŔůè¿ŞåĞž(èąŇåŔůäžŐ1åijĂåğŃ)ïijŇä¡ääŔŕäžěäijăéĂŠäÿĂäÿłåijĂåğŃåŔĆæŢř

```
>>> my_list = ['a', 'b', 'c']
>>> for idx, val in enumerate(my_list, 1):
...     print(idx, val)
...
1 a
2 b
3 c
```

è£Źçğ■æČĚåĘţåIJĺä¡ăéĄ■åŐĘæŰĞäžűæŮűæČşåIJĺéŤŹèŕŕæűĹæĄŕäÿ■ä¡£çŤĺèąŇåŔůåőŽä¡■æŮűåĂŹéĺ

```
def parse_data(filename):
    with open(filename, 'rt') as f:
        for lineno, line in enumerate(f, 1):
            fields = line.split()
            try:
                count = int(fields[1])
                ...
            except ValueError as e:
                print('Line {}: Parse error: {}'.format(lineno, e))
```

`enumerate()` åŕźäžŐèů§èÿłæ§ŘäžŻåĂijåIJĺåĹŮèąĺäÿ■åĞžçŐřçŽĎä¡■ç¡őæŸŕå¿ĹæIJL’çŤĺçŽĎãĂĆ
æL’ĂäžëïijŇåęĆæđIJä¡ăæČşåřĘäÿĂäÿłæŰĞäžűäÿ■åĞžçŐřçŽĎå■Ţèŕ■æŸåřĎåĹřåőČåĞžçŐřçŽĎèąŇåŔůäÿ
`enumerate()` æİěåőŇæĹŘïijŽ

```
word_summary = defaultdict(list)

with open('myfile.txt', 'r') as f:
    lines = f.readlines()

for idx, line in enumerate(lines):
    # Create a list of words in current line
    words = [w.strip().lower() for w in line.split()]
    for word in words:
        word_summary[word].append(idx)
```

åęĆæđIJä¡ăåď'ĎçŘĘåőŇæŰĞäżűåŘŐæL'Şå■ř word_summary ïijŇäijŽåŔŚçŐřåőČæŸŕäÿĂäÿłå■ŮåĚÿ(åĞEçąőæİèèőšæŸŕäÿĂäÿł defaultdict)ïijŇ åŕźäžŐæŕŔäÿłå■Ţèŕ■æIJL'äÿĂäÿł key ïijŇæŕŔäÿł key åŕźåžŢçŽĎåĂijæŸŕäÿĂäÿłçŢśè£Źäÿłå■Ţèŕ■åĞžçŐřè£Ğäÿď'æňąïijŇéĆčäźĹè£ŹäÿłèąŇåŔůżĎæĹŘçŽĎåĹŮèąĺãĂĆ åęĆæđIJæ§Ŕäÿłå■Ţèŕ■åIJĺäÿĂèąŇäÿ■åĞžçŐřè£Ğäÿď'æňąïijŇéĆčäźĹè£ŹäÿłèąŇåŔůäź§äijŽåĞžçŐřäÿď'æňąï åŘŇæŮűäź§åŔŕäżěä¡IJäÿžæŰĞæIJńçŽĎäÿĂäÿłçőĂå■Ţçż§èőąãĂĆ

èőĺèőž

å¡Şä¡ăæČşéćİåď'ŰåőŽäźL'äÿĂäÿłèőąæŢřåŔŸéĞŔçŽĎæŮűåĂŹïijŇä¡£çŤĺ enumerate() åĞ¡æŢřäijŽæŻť'åŁăçőĂå■ŢãĂĆä¡ăåŔŕèČ¡äijŽåČŔäÿŃéİćè£ŹæăůåĘŹäżčçăĄïijŽ

```
lineno = 1
for line in f:
    # Process line
    ...
    lineno += 1
```

ä¡ĘæŸŕåęĆæđIJä¡£çŤĺ enumerate() åĞ¡æŢřæİěäžčæŻ£åřśæŸ¿å¿ŮæŻť'åŁääijŸéŻĚäžĘïijŽ

```
for lineno, line in enumerate(f):
    # Process line
    ...
```

enumerate() åĞ¡æŢřè£ŤåŻđçŽĎæŸŕäÿĂäÿł enumerate åŕźèśąåőđä¿ŃïijŇ åőČæŸŕäÿĂäÿłè£■äżčåŹĺïijŇè£ŤåŻđè£đçż■çŽĎåŇĚåŘńäÿĂäÿłèőąæŢřåŠŇäÿĂäÿłåĂijçŽĎåĚČçżĎïijŇ åĚČçżĎäÿ■çŽĎåĂijéĂŽè£ĞåIJĺäijăåĚěåžŔåĹŮäÿŁèřČçŤĺ next() è£ŤåŻđãĂĆ

è£ŸæIJL'äÿĂçĆźåŔŕèČ¡åźűäÿ■å¿ĹéĞ■èęĄïijŇä¡ĘæŸŕäź§åĂijå¿ŮæşĺæĎŔïijŇ æIJL'æŮűåĂŹå¡Şä¡ăåIJĺäÿĂäÿłåůšçżŔèğčåŐŇåŘŐçŽĎåĚČçżĎåžŔåĹŮäÿŁä¡£çŤĺ enumerate() åĞ¡æŢřæŮűå¿Ĺåőźæ ŸŞèřČåĚěéŹůéŸśãĂĆ ä¡ăå¿ŮåČŔäÿŃéİćæ■čçąőçŽĎæŰźåijŔè£ŹæăůåĘŹïijŽ

```
data = [ (1, 2), (3, 4), (5, 6), (7, 8) ]

# Correct!
for n, (x, y) in enumerate(data):
    ...
```

```
# Error!
for n, x, y in enumerate(data):
    ...
```

6.11 4.11 åŘŇæŮűè£■äžčåď'Žäÿłåž ŕåĹŮ

éŮőécŸ

ä¡ăæČşåŘŇæŮűè£■äžčåď'Žäÿłåž ŕåĹŮïijŇæŕŕæňąåĹĘåĹńäżŐäÿĂäÿłåž ŕåĹŮäÿ■åŔŰäÿĂäÿłåĚČçť'ăãĂ

èğčåĘşæŰźæąĹ

äÿžäžĘåŘŇæŮűè£■äžčåď'Žäÿłåž ŕåĹŮïijŇä¡£çŤÍ `zip()` åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> xpts = [1, 5, 4, 2, 10, 7]
>>> ypts = [101, 78, 37, 15, 62, 99]
>>> for x, y in zip(xpts, ypts):
...     print(x,y)
...
1 101
5 78
4 37
2 15
10 62
7 99
>>>
```

`zip(a, b)` äijŽçŤ§æĹŘäÿĂäÿłåŕŕè£ŤåŻđåĚČçżĎ `(x, y)` çŽĎè£■äžčåŹĺïijŇåĚűäÿ■xæİěèĞłaïijŇyæİěèĞłbãĂĆ äÿĂæŮęåĚűäÿ■æ§Řäÿłåž ŕåĹŮåĹřåžŢçżŞář¿ïijŇè£■äž
åŻăæ■ď'è£■äžčéŢ£åžęèů§åŔĆæŢřäÿ■æIJĂç§■åžŕåĹŮéŢ£åžęäÿĂèĞť'ãĂĆ

```
>>> a = [1, 2, 3]
>>> b = ['w', 'x', 'y', 'z']
>>> for i in zip(a,b):
...     print(i)
...
(1, 'w')
(2, 'x')
(3, 'y')
>>>
```

åęĆæđIJè£Źäÿłäÿ■æŸŕä¡ăæČşèęĄçŽĎæŢĹæđIJïijŇéĆčäźĹè£ŸåŔŕäžěä¡£çŤÍ `itertools.zip_longest()` åĞ¡æŢřæİěäžčæŻ£ãĂĆæŕŤåęĆïijŽ

```
>>> from itertools import zip_longest
>>> for i in zip_longest(a,b):
...     print(i)
```

```
...
(1, 'w')
(2, 'x')
(3, 'y')
(None, 'z')

>>> for i in zip_longest(a, b, fillvalue=0):
...     print(i)
...
(1, 'w')
(2, 'x')
(3, 'y')
(0, 'z')
>>>
```

èőÍèőž

å¡Şä¡äæČşæĹŘåŕźåď'ĎçŘĘæŢřæ■őçŽĎæŮűåĂŹ zip() åĞ¡æŢřæŸŕå¿ĹæIJL'çŤÍçŽĎãĂĆ æŕŤåęĆïijŇåĄĞèő¿ä¡ääď'ťåĹŮèąĺåŠŇäÿĂäÿłåĂijåĹŮèąĺïijŇåřśåČŔäÿŇéIć

```
headers = ['name', 'shares', 'price']
values = ['ACME', 100, 490.1]
```

ä¡£çŤÍzip()åŔŕäžèèől'ä¡ääřĘåőČäžňæL'ŞåŇĚåźűçŤ§æĹŘäÿĂäÿłå■ŮåĚÿïijŽ

```
s = dict(zip(headers,values))
```

æĹŰèĂĚä¡äää§åŔŕäžěåČŔäÿŇéIćè£Źæăůäžğç Ť§è¿ŞåĞžïijŽ

```
for name, val in zip(headers, values):
    print(name, '=', val)
```

èŹ¡çĎűäÿ■åÿÿèğĄïijŇä¡ĘæŸŕ zip() åŔŕäžěæŐěåŔŮåď'ŽäžŐäÿd'äÿłçŽĎåžŔåĹŮçŽĎåŔĆæŢřãĂĆ è£ŹæŮűåĂŹæL'ĂçŤ§æĹŘçŽĎçżŞædIJåĚČçżĎäÿ■åĚČçt'ääÿłæŢřèů§è¿ŞåĚěåžŔåĹŮäÿłæŢřäÿĂæăůãĂĆærŤ

```
>>> a = [1, 2, 3]
>>> b = [10, 11, 12]
>>> c = ['x','y','z']
>>> for i in zip(a, b, c):
...     print(i)
...
(1, 10, 'x')
(2, 11, 'y')
(3, 12, 'z')
>>>
```

æIJĂåŘŐåijžèřČäÿĂçĆźåřśæŸŕïijŇ zip() äijŽåĹŻåżžäÿĂäÿłè£■äžčåŹĺæÍěä¡IJäÿžçżŞædIJè£ŤåŻđãĂĆ åęĆædIJä¡ăéIJĂèęĄåřĘçżŞåŕźçŽĎåĂijå■ŸåĆĺåIJĺåĹŮèąĺäÿ■ïijŇèęĄä¡£çŤÍ list() åĞ¡æŢřãĂĆærŤåęĆïijŽ

```
>>> zip(a, b)
<zip object at 0x1007001b8>
>>> list(zip(a, b))
[(1, 10), (2, 11), (3, 12)]
>>>
```

6.12 4.12 äÿ■åŘŇéŻĘåŘĹäÿŁåĚČçťäçŽĎè£■äžč

éŮőécŸ

ä¡ăæČşåIJĺåď'ŽäÿłåŕźèśąæL'ğèąŇçŻÿåŘŇçŽĎæŞ■ä¡IJïijŇä¡ĘæŸŕè£ŹäžŻåŕźèśąåIJĺäÿ■åŘŇçŽĎåőźåŹĺä

èğčåĘşæŰźæąĹ

itertools.chain() æŰźæşŢåŔŕäžěçŤĺæİěçőĂåŇŰè£ŹäÿłäżżåŁąãĂĆ åőČæŐěåŘŮäÿĂäÿłåŔŕè£■äžčåŕźèśąåĹŮèąĺä¡IJäÿžè¿ŞåĚčïijŇåźűè£ŤåŻđäÿĂäÿłè£■äžčåŹĺïijŇæIJL'æŢĹçŽĎ äÿžäžĘæijŤçď'žæÿĚæěŽïijŇèĂČèŹŚäÿŃéİćè£Źäÿłä¿Ńå■ŘïijŽ

```
>>> from itertools import chain
>>> a = [1, 2, 3, 4]
>>> b = ['x', 'y', 'z']
>>> for x in chain(a, b):
... print(x)
...
1
2
3
4
x
y
z
>>>
```

ä¡£çŤÍ chain() çŽĎäÿĂäÿłåÿÿèğĄåIJžæŹŕæŸŕå¡Şä¡ăæČşåŕźäÿ■åŘŇçŽĎéŻĘåŘĹäÿ■æL'ĂæIJL'åĚČç

```
# Various working sets of items
active_items = set()
inactive_items = set()

# Iterate over all items
for item in chain(active_items, inactive_items):
    # Process item
```

è£Źçğ■èğčåĘşæŰźæąĹèęĄæŕŤåČŔäÿŃéİćè£Źæăůä¡£çŤĺäÿď'äÿłå■ŢçŃňçŽĎå¿łçŐŕæŻť'åŁääijŸéŻĚïijŇ

```
for item in active_items:
    # Process item
```

```
    ...

for item in inactive_items:
    # Process item
    ...
```

èőÍèőž

itertools.chain() æŐěåŔŮäÿĂäÿłæĹŰåď'Žäÿłåŕŕè£■äžčåŕźèśąæIJĂäÿžè¿ŞåĚěåŔĆæŢřãĂĆ çĎűåŘŐåĹŻåžžäÿĂäÿłè£■äžčåŹĺïijÑä¿İæňąè£đçż■çŽĎè£ŤåŻđæŕŔäÿłåŔŕè£■äžčåŕźèśąäÿ■çŽĎåĚČçt'ããĂĆ è£Źçğ■æŰźåijŔèęĄæŕŤåĚĹářĘåžŔåĹŮåŘĹåźűåĘ■è£■äžčèęĄéńŸæŢĹçŽĎåď'ŽãĂĆæŕŤåęĆïijŽ

```
# Inefficent
for x in a + b:
    ...

# Better
for x in chain(a, b):
    ...
```

çňňäÿĂçğ■æŰźæąĹäÿ■ïijŇ a + b æŞ■ä¡IJäijŽåĹŻåžžäÿĂäÿłåĚĺæŰřçŽĎåžŔåĹŮåźűèęĄæśĆaåŠŇbçŽ chian() äÿ■äijŽæIJL'è£ŹäÿĂæ■čïijŇæL'Ăäžěåę ĆæđIJè¿ŞåĚěåžŔåĹŮéİđåÿÿåď'ğçŽĎæŮűåĂŹäijŽå¿ĹçIJA åźűäÿŤå¡ŞåŔŕè£■äžčåŕźèśąçśžåđŃäÿ■äÿĂæăůçŽĎæŮűåĂŹ chain() åŘŇæăůåŔŕäžěå¿Ĺåě¡çŽĎåůěä¡IJãĂĆ

6.13 4.13 åĹŻåžžæŢřæ■őåď'ĎçŘĘçőąéĄŞ

éŮőécŸ

ä¡ăæČşäžěæŢřæ■őçőąéĄŞ(çśžäijijUnixçőąéĄŞ)çŽĎæŰźåijŔè£■äžčåď'ĎçŘĘæŢřæ■őãĂĆ ærŤåęĆïijŇä¡ăæIJL'äÿłåď'ğéĞŔçŽĎæŢřæ■őéIJĂèęĄåď'ĎçŘĘïijŇä¡ĘæŸŕäÿ■èČ¡åřĘåőČäžňäÿĂæňąæĂğæŤ¿

èğčåĘşæŰźæąĹ

çŤ§æĹŘåŹĺåĞ¡æŢřæŸŕäÿĂäÿłåőđçŐřçőąéĄŞæIJžåĹűçŽĎåě¡åŁđæşŢãĂĆ äÿžäžĘæijŤçď'žïijŇåĄĞåőŽä¡ăèęĄåď'ĎçŘĘäÿĂäÿłéİđåÿÿåď'ğçŽĎæŮěå£ŮæŰĞäžűçŻőå¡ŢïijŽ

```
foo/
    access-log-012007.gz
    access-log-022007.gz
    access-log-032007.gz
    ...
    access-log-012008
bar/
    access-log-092007.bz2
```

```
    ...
    access-log-022008
```

åĄĞèő¿æŕŔäÿłæŮěå£ŮæŰĞäżűåŇĚåŘńè£ŹæăůçŽĎæŢřæ■őïijŽ

```
124.115.6.12 - - [10/Jul/2012:00:18:50 -0500] "GET /robots.txt ..."␣
↪200 71
210.212.209.67 - - [10/Jul/2012:00:18:51 -0500] "GET /ply/ ..." 200␣
↪11875
210.212.209.67 - - [10/Jul/2012:00:18:51 -0500] "GET /favicon.ico ..
↪." 404 369
61.135.216.105 - - [10/Jul/2012:00:20:04 -0500] "GET /blog/atom.xml␣
↪..." 304 -
...
```

äÿžäžĘåď ĎçŘĘè£ŹäžŻæŰĞäżűïijŇä¡åŔŕäżěåőŽäźĽäÿĂäÿłçŤšåď ŽäÿłæĽğèąŇçĽźåőŽäżżåŁąçŃňçńŇ

```
import os
import fnmatch
import gzip
import bz2
import re

def gen_find(filepat, top):
    '''
    Find all filenames in a directory tree that match a shell␣
↪wildcard pattern
    '''
    for path, dirlist, filelist in os.walk(top):
        for name in fnmatch.filter(filelist, filepat):
            yield os.path.join(path,name)

def gen_opener(filenames):
    '''
    Open a sequence of filenames one at a time producing a file␣
↪object.
    The file is closed immediately when proceeding to the next␣
↪iteration.
    '''
    for filename in filenames:
        if filename.endswith('.gz'):
            f = gzip.open(filename, 'rt')
        elif filename.endswith('.bz2'):
            f = bz2.open(filename, 'rt')
        else:
            f = open(filename, 'rt')
        yield f
        f.close()

def gen_concatenate(iterators):
    '''
```

```
    Chain a sequence of iterators together into a single sequence.
    '''
    for it in iterators:
        yield from it

def gen_grep(pattern, lines):
    '''
    Look for a regex pattern in a sequence of lines
    '''
    pat = re.compile(pattern)
    for line in lines:
        if pat.search(line):
            yield line
```

çŐřåIJÍä¡ăåŔŕäžěå¿Ĺåőźæ˙ŸŞçŽĎåřĘè£ŹäžŻåĞ¡æŢřè£đèţůæİěåĹŻåžžäÿĂäÿłåd'ĎçŘĘçőąéĄŞãĂĆ
æŕŤåęĆïijÑäÿžäžĘæ§ěæL'¿åŇĚåŘńå■Ţèŕ■pythonçŽĎæL'ĂæIJL'æŮěå£Ůèą×ïijÑä¡ăåŔŕäžěè£Źæăůå˛ŽïijŽ

```
lognames = gen_find('access-log*', 'www')
files = gen_opener(lognames)
lines = gen_concatenate(files)
pylines = gen_grep('(?i)python', lines)
for line in pylines:
    print(line)
```

åęĆædIJåřĘæİěçŽĎæŮűåĂŹä¡ăæČşæL'l'åśŢçőąéĄŞïijÑä¡ăçŤŽèĞşåŔŕäžěåIJÍçŤ§æĹŘåŹÍèąÍè¿¿åijŔäÿ■
æŕŤåęĆïijÑäÿÑéÍćè£ŹäÿłçL'ĹæIJňèőąçőŮåĞžäijăè¿ŞçŽĎå■ÙèŁĆæŢřåźűèőąçőŮåĚűæĂżåŠÑãĂĆ

```
lognames = gen_find('access-log*', 'www')
files = gen_opener(lognames)
lines = gen_concatenate(files)
pylines = gen_grep('(?i)python', lines)
bytecolumn = (line.rsplit(None,1)[1] for line in pylines)
bytes = (int(x) for x in bytecolumn if x != '-')
print('Total', sum(bytes))
```

èőÍèőž

äžěçőąéĄŞæŰźåijŔåd'ĎçŘĘæŢřæ■őåŔŕäžěçŤÍæİěèğčåĘşåŘĎçśżåĚűäžŰéŮőéćŸïijÑåŇĚæŃňèğčædŘïi

äÿžäžĘçŘĘèğčäÿŁè£řäžčçăĄïijÑéĞ■çĆźæŸŕèęĄæŸŐçŹ¡ `yield`
èŕ■åŘëä¡IJäÿžæŢřæ■őçŽĎçŤ§äžğèĂĚèĂŇ `for` å¿łçŐŕèŕ■åŘëä¡IJäÿžæŢřæ■őçŽĎæűĹèť'źèĂĚãĂĆ
å¡Şè£ŹäžŻçŤ§æĹŘåŹÍèćńè£ðåIJÍäÿĂèţůåŘŐïijÑærŔäÿł `yield`
äijŽåřĘäÿĂäÿłå■ŢçŃňçŽĎæŢřæ■őåĚČçť'ăäijăéĂŠçżŹè£■äžčåd'ĎçŘĘçőąéĄŞçŽĎäÿÑäÿĂéŸűæőţãĂĆ
åIJÍä¿Ńå■ŘæIJĂåŘŐéČÍåĹĘïijÑ `sum()` åĞ¡æŢřæŸŕæIJĂçżĹçŽĎçÍÑåžŔél'šåŁÍèĂĚïijÑærŔæąäžŐçŤ§æĹŘ

è£Źçğ■æŰźåijŔäÿĂäÿłéIđåÿÿåě¡çŽĎçL'źçĆźæŸŕærŔäÿłçŤ§æĹŘåŹÍåĞ¡æŢřå¿ĹåřŘåźűäÿŤéČ¡æŸŕçŃňç
å¿Ĺåd'ŽæŮűåĂŹïijÑè£ŹäžŻåĞ¡æŢřåęĆædIJærŕŤè¿ČéĂŽçŤÍçŽĎèŕÍåŘŕäžěåIJÍåĚűäžŰåIJžæŹŕéĞ■åd'■ä¡£çŤÍ
åźűäÿŤæIJĂçżĹåřĘè£ŹäžŻçżĎäžűçżĎåŘĹèţůæİěçŽĎäžčçăĄçIJŃäÿŁåŐżéIđåÿÿçőĂå■ŢïijÑäź§å¿Ĺåőźæ˙ŸŞç

ä¡£çŤÍè£ŹçğæŰźåijŔçŽĎåĘĚå■ŸæŢĹçŐĞäź§äÿ■å¿Ůäÿ■æŔŘãĂĆäÿŁè£řäžčçăĄå■şä¿£æŸŕåIJĺäÿĂäÿ
äžŇåőđäÿŁïijŇçŤśäžŐä¡£çŤĺäžĘè£■äžčæŰźåijŔåď'ĎçŘĘïijŇäžčçăĄè£ŘèąŇè£ĞçÍŇäÿ■åŔłéIJĂèęĄå¿ĹåřŔå

åIJÍèřČçŤÍ `gen_concatenate()` åĞ¡æŢřçŽĎæŮűåĂŹä¡ääŔŕèČ¡äijŽæIJL'äžŽäÿ■åď'łæŸŐçŹ¡ãĂĆ
è£ŹäÿłåĞ¡æŢřçŽĎçŽőçŽĎæŸŕåřĘè¿ŞåĚěåžŘåĹŮæŃijæŐěæĹŘäÿĂäÿłå¿ĹéŢ£çŽĎèąŇåžŘåĹŮãĂĆ
`itertools.chain()` åĞ¡æŢřåŘŇæăůæIJL'çśżäijijçŽĎåŁ§èČ¡ïijŇä¡ĘæŸŕåőČéIJĂèęĄåřĘæL'ĂæIJL'åŔŕè
åIJÍäÿŁéİćè£ŹäÿłäżŃå■Řäÿ■ïijŇä¡ääŔŕèČ¡äijŽåĘŹçśżäijijè£ŹæăůçŽĎèŕ■åŔě

```
lines = itertools.chain(*files)
```

ïijŇ è£ŹåřĘåŕijèĞť
`gen_opener()` çŤ§æĹŘåŹĺèćńæŘŘåL'■åĚĺéČĺæűĹèť'źæŐL'ãĂĆ ä¡ĘçŤśäžŐ
`gen_opener()` çŤ§æĹŘåŹĺæŕŘæňąçŤ§æĹŘäÿĂäÿłæL'ŞåijĂè£ĞçŽĎæŰĞäżűïijŇ
ç■L'åĹřäÿŃäÿĂäÿłè£■äžčæ■ěéłď'æŮűæŰĞäżűåřśåĚşéŮ■äžĘïijŇåŽăæ■ď' `chain()`
åIJÍè£ŹéĞŇäÿ■èČ¡è£Źæăůä¡£çŤĺãĂĆ äÿŁéİćçŽĎæŰźæąĹåŔŕäžěéĄ£åĚ■è£Źçğ■æČĚåĘţãĂĆ

`gen_concatenate()` åĞ¡æŢřäÿ■åĞžçŐřè£Ğ `yield from` èŕ■åŔěïijŇåőČårĘ
`yield` æŞ■ä¡IJäžčçŘĘåĹřçĹűçŤ§æĹŘåŹĺäÿŁåŐžãĂĆ èŕ■åŔě `yield from`
`it` çőĂå■ŢçŽĎè£ŤåŽđçŤ§æĹŘåŹÍ `it` æL'ĂäžğçŤ§çŽĎæL'ĂæIJL'åĂijãĂĆ
åĚşäžŐè£ŹäÿłæĹŚäžňåIJÍ4.14åřŔèŁĆäijŽæIJL'æŻť'è£ŽäÿĂæ■ěçŽĎæŘŔè£řãĂĆ

æIJĂåŘŐè£ŸæIJL'äÿĂçĆźéIJĂèęĄæşÍæĎŘçŽĎæŸŕïijŇçőąéĄŞæŰźåijŔåźűäÿ■æŸŕäÿĞèČ¡çŽĎãĂĆ
æIJL'æŮűåĂŹä¡ăæČşçńÑå■şåď'ĎçŘĘæL'ĂæIJL'æŢřæ■őãĂĆ çĎűèĂŇïijŇå■şä¿£æŸŕè£Źçğ■æČĚåĘţïijŇä¡£ç

David Beazley åIJÍäżŰçŽĎ Generator Tricks for Systems Programmers
æŢŹçÍŃäÿ■åŕźäžŐè£Źçğ■æŁĂæIJŕæIJL'éİđåÿÿæůśåĚěçŽĎèőšèğčãĂĆåŔŕäžěåŘĆèĂČè£ŹäÿłæŢŹçÍŃèŐůåŔ

6.14 4.14 åśŢåijĂåţŇåěŮçŽĎåžŔåĹŮ

éŮőécŸ

ä¡ăæČşåřĘäÿĂäÿłåď'ŽåśĆåţŇåěŮçŽĎåžŔåĹŮåśŢåijĂæĹŘäÿĂäÿłå■ŢåśĆåĹŮèąÍ

èğčåĘşæŰźæąĹ

åŔŕäžěåĘŹäÿĂäÿłåŇĚåŘń `yield from` èŕ■åŘěçŽĎéĂŠå¡ŠçŤ§æĹŘåŹĺæİěè¡żæİ¿èğčåĘşè£ŹäÿłéŮőéć

```
from collections import Iterable

def flatten(items, ignore_types=(str, bytes)):
    for x in items:
        if isinstance(x, Iterable) and not isinstance(x, ignore_
↪types):
            yield from flatten(x)
        else:
            yield x

items = [1, 2, [3, 4, [5, 6], 7], 8]
# Produces 1 2 3 4 5 6 7 8
for x in flatten(items):
    print(x)
```

åIJÍäÿŁéİćäžčçăĄäÿ■ïijŇ `isinstance(x, Iterable)` æčĂæ§ěæ§ŘäÿłåĚČçť ăæŸŕåŘęæŸŕåŘŕè£■äžčçŽĎãĂĆ åęĆæđIJæŸŕçŽĎèŕİïijŇ `yield from` åŕśäijŽè£ŤåŽđæĽĂæIJL'å■Řä¿ŇçÍŇçŽĎåĂijãĂĆæIJĂçżĹè£ŤåŽđçżŞæđIJåŕśæŸŕäÿĂäÿłæšąæIJL'åţŇ

éćİåď'ŰçŽĎåŔĆæŢř `ignore_types` åŠŇæčĂæţŃèŕ■åŘě `isinstance(x, ignore_types)` çŤÍæİěåřĘå■Ůçňęäÿšå ŠŇå■ŮèŁĆæŐŠéŹď'åIJÍåŘŕè£■äžčåŕźèśąåď'ŰïijŇéŸšæ■ćåřĘåőČäz è£Źæăůç ŽĎèŕİå■ŮçňęäÿšæŢřçżĎåŕśèČ¡æIJĂçźĹè£ŤåŽđæĹŚäžňæĽĂæIJ§æIJŻçŽĎçżŞæđIJäžĘãĂĆæŕŤåęĆ

```
>>> items = ['Dave', 'Paula', ['Thomas', 'Lewis']]
>>> for x in flatten(items):
...     print(x)
...
Dave
Paula
Thomas
Lewis
>>>
```

èőÍèőž

èŕ■åŘě `yield from` åIJÍä¡ăæČşåIJÍçŤ§æĹŘåŹÍäÿ■èřČçŤÍåĚűäžŰçŤ§æĹŘåŹÍä¡IJäÿžå■Řä¿ŇçÍŇçŽĎa åęĆæđIJä¡ääÿ■ä¡£çŤÍåőČçŽĎèŕİïijŇéĆčäźĹåŕśå£Ěéąžå ĘŹéćİåď'ŰçŽĎ `for` å¿łçŐŕäžĘãĂĆæŕŤåęĆïijŽ

```
def flatten(items, ignore_types=(str, bytes)):
    for x in items:
        if isinstance(x, Iterable) and not isinstance(x, ignore_
↪types):
            for i in flatten(x):
                yield i
        else:
            yield x
```

åř¡çőąåŘłæŤźäžĘäÿĂçĆźçĆźïijŇä¡ĘæŸŕ `yield from` èŕ■åŘěçIJŇäÿŁåŐżæĎ§ègL'æŻť'åě¡ïijŇåźűäÿŤäź§ä¡£å¿ŮäžčçăĄæŻť'çőĂæť'ĄæÿĚçĹ¡ãĂĆ

äźŇåL'■æŘŘåĹřçŽĎårźäžŐå■Ůçňęäÿšå ŠŇå■ŮèŁĆçŽĎéćİåď'ŰæčĂæ§ěæŸŕäÿžäžĘéŸšæ■ćåřĘåőČäžňå åęĆæđIJè£ŸæIJL'åĚűäžŰä¡ääÿ■æČşåśŢåijĂçŽĎçśžåđŇïijŇä£őæŤźåŔĆæŢř `ignore_types` å■şåŔŕãĂĆ

æIJĂåŘŐèęĄæşĺæĎŘçŽĎäÿĂçĆźæŸŕïijŇ `yield from` åIJÍæűL'åŘŁåĹřå§žäžŐå■ŘçÍŇåŠŇçŤ§æĹŘåŹÍçŽĎåźűåŘŠçijŰçÍŇäÿ■æL'őæijŤçİĂæŻť'åŁáéĞ■èęĄçŽĎèğŠè åŔŕäžěåŔĆèĂČ12.12åřŘèŁĆæ§ěçIJŇåŘęåď'ŰäÿĂäÿłä¿Ňå■ŘãĂĆ

6.15 4.15 éąžäžŔè£■äžčåŘĹåźűåŘŐçŽĎæŐŠåžŔè£■äžčåŕźèśą

éŮőécŸ

ä¡ăæIJL'äÿĂçşžåĹŮæŐŠåžŔåžŔåĹŮïijŇæČşåřĘåőČäžňåŘĹåźűåŘŐå¿ŮåĹřäÿĂäÿłæŐŠåžŔåžŔåĹŮåźűå

èğčåĘşæŰźæąĹ

heapq.merge() åĞ¡æŢřåŔŕäżěåÿőä¡ăèğčåĘşè£ŹäÿłéŮőéćŸãĂĆæŕŤåęĆïijŽ

```
>>> import heapq
>>> a = [1, 4, 7, 10]
>>> b = [2, 5, 6, 11]
>>> for c in heapq.merge(a, b):
...     print(c)
...
1
2
4
5
6
7
10
11
```

èőĺèőž

heapq.merge åŔŕè£■äżčçĽźæĂğæĎŔåŚşçİĂåőČäÿ■äijŽçńŃéľňèŕżåŔŰæĽĂæIJĽåžŔåĹŮãĂĆ è£ŹåřśæĎŔåŚşçİĂä¡ăåŔŕäżěåIJĺéİđåÿÿéŢ£çŽĎåžŔåĹŮäÿ■ä¡£çŤĺåőČïijŇèĂŇäÿ■äijŽæIJĽåďłåďğçŽĎåijĂé æŕŤåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘæİěæijŤçďžåęĆä¡ŢåŘĹåźűäÿďäÿłæŐŠåžŔæŰĞäżűïijŽ

```
with open('sorted_file_1', 'rt') as file1, \
    open('sorted_file_2', 'rt') as file2, \
    open('merged_file', 'wt') as outf:

    for line in heapq.merge(file1, file2):
        outf.write(line)
```

æIJĽäÿĂçĆźèęĄåijžèřČçŽĎæŸŕ heapq.merge() éIJĂèęĄæĽĂæIJĽè¿ŞåĚěåžŔåĹŮå£ĚéążæŸŕæŐŠ çĽźåĹńçŽĎïijŇåőČåźűäÿ■äijŽéćĎåĚĹèŕżåŔŰæĽĂæIJĽæŢřæ■őåĹřåăĘæăĹäÿ■æĹŰèĂĚéćĎåĚĹæŐŠåžŔïij åőČäżĚäżĚæŸŕæčĂæ§ěæĽĂæIJĽåžŔåĹŮçŽĎåijĂåğŃéČĺåĹĘåźűè£ŤåŻđæIJĂåřŔçŽĎéĆčäÿłïijŇè£ŹäÿłèĹĆ

6.16 4.16 è£■äżčåŹĺäżčæŻ£whileæŮăéŹŘå¿łçŐŕ

éŮőéćŸ

ä¡ăåIJĺäżčçăĄäÿ■ä¡£çŤĺ while å¿łçŐŕæİěè£■äżčåďĎçŘĘæŢřæ■őïijŇåŻăäÿžåőČéIJĂèęĄèřČçŤĺæ§Řäÿ èČ¡äÿ■èČ¡çŤĺè£■äżčåŹĺæİěéĞ■åĘŹè£Źäÿłå¿łçŐŕåŚćïij§

èğčåĘşæŰźæąĹ

äÿĂäÿłåÿÿèğĄçŽĎIOæŞ■ä¡IJçĺŃåžŔåŔŕèČ¡äijŽæČşäÿŃéİćè£ŹæăůïijŽ

```
CHUNKSIZE = 8192

def reader(s):
    while True:
        data = s.recv(CHUNKSIZE)
        if data == b'':
            break
        process_data(data)
```

è£Źçğ■äžčçăĄéĂŽåÿÿåŔŕäžěä¡£çŤÍ `iter()` æİěäžčæŻ£ïijŇåęĆäÿŇæĽĂçď žïijŽ

```
def reader2(s):
    for chunk in iter(lambda: s.recv(CHUNKSIZE), b''):
        pass
        # process_data(data)
```

åęĆæđIJä¡ăæĂĂçŰŠåőČåĹŕåžŢèČ¡äÿ■èČ¡æ■čåÿÿåůěä¡IJïijŇåŔŕäžěèŕŢéłŇäÿŇäÿĂäÿłçőÅå■ŢçŽĎä¿Ňå

```
>>> import sys
>>> f = open('/etc/passwd')
>>> for chunk in iter(lambda: f.read(10), ''):
...     n = sys.stdout.write(chunk)
...
nobody:*:-2:-2:Unprivileged User:/var/empty:/usr/bin/false
root:*:0:0:System Administrator:/var/root:/bin/sh
daemon:*:1:1:System Services:/var/root:/usr/bin/false
_uucp:*:4:4:Unix to Unix Copy Protocol:/var/spool/uucp:/usr/sbin/
 ↪uucico
...
>>>
```

èőĺèőž

`iter` åĞ¡æŢřäÿĂäÿłéšIJäÿžäžžçş§ěçŽĎçĽźæĂğæŸŕåőČæŐěåŔŮäÿĂäÿłåŔŕéĂĽçŽĎ `callable` åŕźèśąåŠŇäÿĂäÿłæăĞèőř(çżŞář¿)åĂijä¡IJäÿžè¿ŞåĚěåŔĆæŢřãĂĆ å¡Şäžěè£Źçğ■æŰźåijŔä¡£çŤÍçŽĎæŮűåĂŹïijŇåőČäijŽåĹŻåžžäÿĂäÿłè£■äžčåŹĺïijŇ è£Źäÿłè£■äžčåŹĺäijŽäÿ■æŰ■èřČçŤÍ `callable` åŕźèśąçŻť'åĹřè£ŤåŽđåĂijåŠŇæăĞèőřåĂijçŻÿç■L'äÿžæ■ćãĂ

è£Źçğ■çL'źæőŁçŽĎæŰźæşŢåŕźäžŐäÿĂäžŻçL'źåőŽçŽĎäijŽèćńéĞ■åď'■èřČçŤÍçŽĎåĞ¡æŢřå¿ĹæIJL'æŢÍ äÿ¿ä¿ŃæİèèőšïijŇåęĆæđIJä¡ăæČşäžŐåěŮæŐěå■ŮæĹŰæŰĞäžűäÿ■äžěæŢřæ■őåİŮçŽĎæŰźåijŔèŕżåŔŰæŢřæ `read()` æĹŰ `recv()` ïijŇ åźűåIJĺåŘŐéİćçť'ğèů§äÿĂäÿłæŰĞäžűçżŞář¿æţŃèŕŢæİěåĘşåőŽæŸŕåŘęçżĹæ■ćãĂ `iter()` èřČçŤĺåŕšåŔŕäžěåřĘäÿd'èĂĚçżŞåŘĹèţůæİěäžĘãĂĆ åĚűäÿ■ `lambda` åĞ¡æŢřåŔĆæŢřæŸŕäÿžäžžĘåĹŻåžžäÿĂäÿłæŮăåŔĆçŽĎ `callable` åŕźèśąïijŇåźűäÿž `recv` æĹŰ `read()` æŰźæşŢæŘŘä¿ŻäžĘ `size` åŔĆæŢřãĂĆ

7 çňňäžŤçńäïijŽæŰĞäżűäÿŐlO

æĽĂæIJL’çÍŃåžŔéČ¡èęĄåď’ĎçŘĘè¿ŞåĚěåŠŇè¿ŞåĞžãĂĆ è£ŹäÿĂçńäářĘæűţçŻŰåď’ĎçŘĘäÿ■åŘŇçśžåđŃçŽĎæŰĞäżűïijŇåŇĚæŃňæŰĞæIJňåŠŇäžŇè£ŻåĹűæŰĞäżűï åŕźæŰĞäżűåŘ■åŠŇçŻőå¡ŢçŽĎæŞ■ä¡IJäź§äijŽæűĽ’åŔŁåĹřãĂĆ

Contents:

7.1 5.1 èŕżåĘŹæŰĞælJňæŢřæ■ő

éŮőécŸ

ä¡ăéIJĂèęĄèŕżåĘŹåŘĎçğ■äÿ■åŘŇçijŰçăĄçŽĎæŰĞæIJňæŢřæ■őïijŇærŤåęĆASCIIïijŇUTF-8æĹŰUTF-16çijŰçăĄç■L’ãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺåÿęæIJL’ `rt` æĺąåijŔçŽĎ `open()` åĞ¡æŢřèŕżåŔŰæŰĞæIJňæŰĞäżűãĂĆåęĆäÿŇæL’Ăçď’žïijŽ

```
# Read the entire file as a single string
with open('somefile.txt', 'rt') as f:
    data = f.read()

# Iterate over the lines of the file
with open('somefile.txt', 'rt') as f:
    for line in f:
        # process line
        ...
```

çśżäijijçŽĎïijŇäÿžäžĘåĘŹåĚěäÿĂäÿłæŰĞæIJňæŰĞäżűïijŇä¡£çŤĺåÿęæIJL’ `wt` æĺąåijŔçŽĎ `open()` åĞ¡æŢřïijŇ åęĆædIJäźŇåL’■æŰĞäżűåĘĚåőźå■ŸåIJĺåĹŹæÿĚéŹď’åźűèęĘçŻŰæŐL’ãĂĆ

```
# Write chunks of text data
with open('somefile.txt', 'wt') as f:
    f.write(text1)
    f.write(text2)
    ...

# Redirected print statement
with open('somefile.txt', 'wt') as f:
    print(line1, file=f)
    print(line2, file=f)
    ...
```

åęĆædIJæŸŕåIJĺåůšå■ŸåIJĺæŰĞäżűäÿ■æůżåŁäåĘĚåőźïijŇä¡£çŤĺæĺąåijŔäÿž `at` çŽĎ `open()` åĞ¡æŢřãĂĆ

æŰĞäžűçŽĎèřžåĘŹæŞ■ä¡IJéżŸèőďä¡£çŤĺçşżçż§çijŰçăĄïijŇåŔřäžěéĂŽè£ĞèřČçŤĺ sys.getdefaultencoding() æİěå¿ŮåĹřãĂĆ åIJĺåď ğåď ŽæŢřæIJžåŹĺäÿŁéİćéČ¡æŸŕutf-8çijŰçăĄãĂĆåęĆæđIJä¡ăåůšçżŔç§ěéĄŞä¡ăèęĄèřžåĘŹçŽĎæŰĞæIJňæŸŕåĚűäžŰçijŰçăĄæŰźåijŔïijŇéĆčäźĹåŔřäžěéĂŽè£ĞäijáéĂŠäÿĂäÿłåŔřéĂL'çŽĎ encoding åŔĆæŢřçżŹopen()åĞ¡æŢřãĂĆåęĆäÿŇæL'Ăçď žïijŽ

```
with open('somefile.txt', 'rt', encoding='latin-1') as f:
    ...
```

PythonæŤŕæŇĄéİđåÿÿåď ŽçŽĎæŰĞæIJňçijŰçăĄãĂĆåĞääÿłåÿÿèğĄçŽĎçijŰçăĄæŸŕascii, latin-1, utf-8åŠŇutf-16ãĂĆ åIJĺwebåžŤçŤĺçÍŇåžŔäÿ■éĂŽåÿÿéČ¡ä¡£çŤĺçŽĎæŸŕUTF-8ãĂĆ asciiåŕźåžŤäžŐU+0000åĹřU+007FèŇČåŻť åĘĚçŽĎ7ä¡■å■Ůçňęã ĂĆ latin-1æŸŕå■ŮèŁĆ0-255åĹřU+0000èĞşU+00FFèŇČåŻť åĘĚUnicodeå■Ůçňęç ŽĎçŻť æŐěæŸäåřĎãĂĆ å¡Şèŕźå ŔŰäÿĂäÿłæIJłç§ěçijŰçăĄçŽĎæŰĞæIJňæŮűä¡£çŤĺlatin-1çijŰçăĄæřÿè£IJäÿ■äijŽäžğçŤ§èğčçăĄéŤŹèŕŕãĂĆ ä¡£çŤĺlatin-1çijŰçăĄèŕźåŔŰäÿĂäÿłæŰĞäžűçŽĎæŮůåŹäź§èőÿäÿ■èČ¡äžğçŤ§åőŇåĚĺæ■čçąőçŽĎæŰĞæIJňèğčçăĄæŢřä¡ĘæŸŕåőČäź§èČ¡äżŐäÿ■æŔŘåŔŰåĞžèűşåď §åď ŽçŽĎæIJL'çŤĺæŢřæ■őãĂĆåŘŇæŮűïijŇåęĆæđIJä¡ääźŇåŘ

èőĺèőž

èŕźåĘŹæŰĞæIJňæŰĞäžűäÿĂèĹňæİèèőšæŸŕæŕŤè¿ČçőĂå■ŢçŽĎãĂĆä¡ĘæŸŕäź§åĞäçĆźæŸŕéIJĂèęĄæşéęŰåĚĹïijŇåIJĺä¿Ńå■ŘçĺŇåžŔäÿ■çŽĎwithèŕ■åŔężŹèćńä¡£çŤĺåĹřçŽĎæŰĞäžűåĹŻåžžäžĘäÿĂäÿłäÿŁäÿŇæä¡Ę with æŐğåĹűåİŮçżŞæİ§æŮűïijŇæŰĞäžűäijŽèĞłåŁĺåĚşéŮ■ãĂĆä¡ääź§åŔŕäžěäÿ■ä¡£çŤĺ with èŕ■åŔěïijŇä¡ĘæŸŕè£ŹæŮűåĂŹä¡ääřšå£Ěéąžèőřå¿ŮæĽŃåŁĺåĚşéŮ■æŰĞäžűïijŽ

```
f = open('somefile.txt', 'rt')
data = f.read()
f.close()
```

åŔęåď ŰäÿĂäÿłéŮőécŸæŸŕåĚşäžŐæ■ćèąŇçňęçŽĎèŕĘåĹńéŮőécŸïijŇåIJĺUnixåŠŇWindowsäÿ■æŸŕäÿ■ \n åŠŇ \r\n)ãĂĆ éżŸèőď æČĚåĘţäÿŇïijŇPythonäijŽäžěçż§äÿĂæĺąåijŔåď ĎçŘĘæ■ćèąŇçňęãĂĆ è£Źçğ■æĺąåijŔäÿŇïijŇåIJĺèŕźåŔŰæŰĞæIJňçŽĎæŮűåĂŹïijŇPythonåŔŕäžěèŕĘåĹńæL'ĂæIJL'çŽĎæŹőéĂŽæ■ \n å■Ůçňęã ĂĆ çśžäijijçŽĎïijŇåIJĺè¿ŞåĞžæŮűäijŽářĘæ■ćèąŇçňę \n è¡ňæ■ćäÿžçşżçż§éżŸèőď çŽĎæ■ćèąŇçňęãĂĆ åęĆæđIJä¡ääÿ■åÿŇæIJŽè£Źçğ■éżŸèőď çŽĎåď ĎçŘĘæŰźåijŔïijŇ open() åĞ¡æŢřäijååĚěåŔĆæŢř newline='' ïijŇářšåČŔäÿŃéİćè£Źæăůïij Ž

```
# Read with disabled newline translation
with open('somefile.txt', 'rt', newline='') as f:
    ...
```

äÿžäžĘèŕť æŸŐäÿď èĂĚäźŃéŮť çŽĎåůőåijĆïijŇäÿŃéİćæĹŠåIJĺUnixæIJžåŹĺäÿŁéİćèŕźåŔŰäÿĂäÿłWindows hello world!\r\n ïijŽ

```
>>> # Newline translation enabled (the default)
>>> f = open('hello.txt', 'rt')
>>> f.read()
'hello world!\n'

>>> # Newline translation disabled
```

```
>>> g = open('hello.txt', 'rt', newline='')
>>> g.read()
'hello world!\r\n'
>>>
```

æIJĂåŘŐäÿĂäÿłéŮőéćŸåřśæŸŕæŰĞæIJňæŰĞäżűäÿ■åŔŕèČ¡åĞžçŐřçŽĎçijŰçăĄéŤŹèŕŕãĂĆ
ä¡Ęä¡ăèŕżåŔŰæĹŰèĂĚåĘŹåĚěäÿĂäÿłæŰĞæIJňæŰĞäżűæŮűïijÑä¡ăåŔŕèČ¡äijŽéĄĞåĹřäÿĂäÿłçijŰçăĄæĹŰè

```
>>> f = open('sample.txt', 'rt', encoding='ascii')
>>> f.read()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "/usr/local/lib/python3.3/encodings/ascii.py", line 26, in␣
 ↪decode
        return codecs.ascii_decode(input, self.errors)[0]
UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position
12: ordinal not in range(128)
>>>
```

åęĆæđIJåĞžçŐřè£ŹäÿłéŤŹèŕŕïijÑéĂŽåÿÿèąÍçd'žä¡ăèŕżåŔŰæŰĞæIJňæŮűæŇĞåőŽçŽĎçijŰçăĄäÿ■æ■čç
ä¡ăæIJĂåě¡äžŤçżĘéŸĚèŕżèŕť'æŸŐåźűçąőèőd'ä¡ăçŽĎæŰĞäżűçijŰçăĄæŸŕæ■čçąőçŽĎ(æŕŤåęĆä¡£çŤÍUTF-
8èĂÑäÿ■æŸŕLatin-1çijŰçăĄæĹŰåĚűäżŰ)ãĂĆ åęĆæđIJçijŰçăĄéŤŹèŕŕè£ŸæŸŕå■ŸåIJÍçŽĎèŕİïijÑä¡ăåŔŕäžě
`open()` åĞ¡æŢřäijăéĂŠäÿĂäÿłåŔŕéĂL'çŽĎ `errors` åŔĆæŢřæİěåď'ĎçŘĘè£ŹäžŻéŤŹèŕŕãĂĆ
äÿÑéIćæŸŕäÿĂäžŽåď'ĎçŘĘåÿÿèğĄéŤŹèŕŕçŽĎæŰźæşŢïijŽ

```
>>> # Replace bad chars with Unicode U+fffd replacement char
>>> f = open('sample.txt', 'rt', encoding='ascii', errors='replace')
>>> f.read()
'Spicy Jalape?o!'
>>> # Ignore bad chars entirely
>>> g = open('sample.txt', 'rt', encoding='ascii', errors='ignore')
>>> g.read()
'Spicy Jalapeo!'
>>>
```

åęĆæđIJä¡ăçżŔåÿÿä¡£çŤÍ `errors` åŔĆæŢřæİěåď'ĎçŘĘçijŰçăĄéŤŹèŕŕïijÑåŔŕèČ¡äijŽèőľä¡ăçŽĎçŢ§æt
åŕźäžŐæŰĞæIJňåď'ĎçŘĘçŽĎéęŰèęĄåŐ§åĹŹæŸŕçąőä£Ïä¡ăæĂżæŸŕä¡£çŤÍçŽĎæŸŕæ■čçąőçijŰçăĄãĂĆå¡Şa
8)ãĂĆ

7.2 5.2 æĽŞå■řè¿ŞåĞžèĞşæŰĞäżűäÿ■

éŮőécŸ

ä¡ăæČşåřĘ `print()` åĞ¡æŢřçŽĎè¿ŞåĞžéĞ■åőŽåŘŚåĹřäÿĂäÿłæŰĞäżűäÿ■åŐżãĂĆ

èğčåĘşæŰźæąĹ

åIJÍ `print()` åĞ¡æŢřäÿ■æŇĞåőŽ `file` åĚşéŤőå■ŮåŔĆæŢřïijŇåČŔäÿŃéİćè£ŹæăůïijŽ

```
with open('d:/work/test.txt', 'wt') as f:
    print('Hello World!', file=f)
```

èőĺèőž

åĚşäžŐè¿ŞåĞžéĞ■åőŽåŘŠåĹřæŰĞäżűäÿ■åřśè£ŹäžŻäžĘãĂĆä¡ĘæŸŕæIJL'äÿĂçĆźèęĄæşĺæĎŔçŽĎåřśæåęĆæđIJæŰĞäżűæŸŕäžŇè£ŻåĹűæĺąåijŔçŽĎèŕİïijŇæL'Şå■řåřśäijŽåĞžéŤŹãĂĆ

7.3 5.3 ä¡£çŤĺåĚűäżŰåĹĘéŽŤçňęæĹŰèąŇçżĹæ■ćçňęæL'Şå■ř

éŮőécŸ

ä¡ăæČşä¡£çŤÍ `print()` åĞ¡æŢřè¿ŞåĞžæŢřæ■őïijŇä¡ĘæŸŕæČşæŤźåŘŸéżŸèőd'çŽĎåĹĘéŽŤçňęæĹŰè

èğčåĘşæŰźæąĹ

åŘŕäżěä¡£çŤĺåIJÍ `print()` åĞ¡æŢřäÿ■ä¡£çŤÍ `sep` åŠŇ `end` åĚşéŤőå■ŮåŔĆæŢřïijŇäżěä¡ăæČşèęĄçŽĎæŰźåijŔè¿ŞåĞžãĂĆærŤåęĆïijŽ

```
>>> print('ACME', 50, 91.5)
ACME 50 91.5
>>> print('ACME', 50, 91.5, sep=',')
ACME,50,91.5
>>> print('ACME', 50, 91.5, sep=',', end='!!\n')
ACME,50,91.5!!
>>>
```

ä¡£çŤÍ `end` åŘĆæŢřäź§åŘŕäżěåIJÍè¿ŞåĞžäÿ■çęĄæ■ćæ■ćèąŇãĂĆærŤåęĆïijŽ

```
>>> for i in range(5):
...     print(i)
...
0
1
2
3
4
>>> for i in range(5):
...     print(i, end=' ')
...
0 1 2 3 4 >>>
```

èőÍèőž

å¡Şä¡ăæČşä¡£çŤÍéÍđçľžæăijåĹĘéŽŤçňęæÍěè¿ŞåĞžæŢřæ■őçŽĎæŮűåĂŹïijŇçżŹ `print()` åĞ¡æŢřäijăéĂŠäÿĂäÿł `sep` åŔĆæŢřæŸŕæIJĂçőÅå■ŢçŽĎæŰźæąĹãĂĆ æIJľæŮűåĂŹä¡ăäijŽçIJŃåĹřäÿĂäžŽçÍŃåžŔåŚŸäijŽä¡£çŤÍ `str.join()` æÍěåőŇæĹŘåŘŇæăůçŽĎäžŃæČĚãĂĆærŤåęĆïijŽ

```
>>> print(','.join(('ACME','50','91.5')))
ACME,50,91.5
>>>
```

`str.join()` çŽĎéŮőécŸåIJÍäžŐåőČäżĚäżĚéĂĆçŤÍäžŐå■Ůçňęäÿšã ĂĆè£ŹæĎŔåŚşçİĂä¡ăéĂŽåÿÿéIJ

```
>>> row = ('ACME', 50, 91.5)
>>> print(','.join(row))
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: sequence item 1: expected str instance, int found
>>> print(','.join(str(x) for x in row))
ACME,50,91.5
>>>
```

ä¡ăå¡ŞçĎűåŔŕäžěäÿ■çŤÍéĆčäźĹéžżçČęïijŇåŔłéIJĂèęĄåČŔäÿŃéÍćè£ŹæăůåĘŹïijŽ

```
>>> print(*row, sep=',')
ACME,50,91.5
>>>
```

7.4 5.4 èŕžåĘŹå■ŮèŁĆæŢřæ■ő

éŮőécŸ

ä¡ăæČşèŕžåĘŹäžŃè£ŽåĹűæŰĞäžűïijŇærŤåęĆåŻ¿çĽĞïijŇåčŕé§şæŰĞäžűç■Ľç■Ľ ãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍæÍąåijŔäÿž `rb` æĹŰ `wb` çŽĎ `open()` åĞ¡æŢřæÍěèŕžåŔŰæĹŰåĘŹåĚěäžŃè£ŽåĹűæŢřæ■őãĂĆærŤ

```
# Read the entire file as a single byte string
with open('somefile.bin', 'rb') as f:
    data = f.read()

# Write binary data to a file
with open('somefile.bin', 'wb') as f:
    f.write(b'Hello World')
```

åIJÍèŕžåŔŰäžŃè£ŽåĹűæŢřæ■őæŮűïijŇéIJĂèęĄæŇĞæŸŐçŽĎæŸŕæĽĂæIJĽè£ŤåŻđçŽĎæŢřæ■őéČ¡æ çśżäijijçŽĎïijŇåIJÍåĘŹåĚěçŽĎæŮűåĂŹïijŇå£Ěéążä£ÍèŕĄåŔĆæŢřæŸŕäžěå■ŮèŁĆå¡ćåijŔåŕźåď ŰæŽť éIJšæŢ

èőÍèőž

åIJÍèŕžåŘŰäžÑè£ŻåĹűæŢřæ■őçŽĎæŮűåĂŹïijÑå■ŮèŁĆå■ŮçňęäÿšåŠÑæŰĞæIJňå■ŮçňęäÿšçŽĎèŕ■äźĽ
çĽźåĹńéIJĂèęĄæşÍæĎŔçŽĎæŸŕïijÑçť́ćåijŢåŠÑè£■äžčåŁÍä¡IJè£ŤåŽđçŽĎæŸŕå■ŮèŁĆçŽĎåĂijèĂÑäÿ■æŸ

```
>>> # Text string
>>> t = 'Hello World'
>>> t[0]
'H'
>>> for c in t:
...     print(c)
...
H
e
l
l
o
...
>>> # Byte string
>>> b = b'Hello World'
>>> b[0]
72
>>> for c in b:
...     print(c)
...
72
101
108
108
111
...
>>>
```

åęĆæđIJä¡ăæČşäżŐäžÑè£ŻåĹűæÍąåijŔçŽĎæŰĞäžűäÿ■èŕžåŘŰæĹŰåĘŹåĚěæŰĞæIJňæŢřæ■őïijÑå£Ěéą

```
with open('somefile.bin', 'rb') as f:
    data = f.read(16)
    text = data.decode('utf-8')

with open('somefile.bin', 'wb') as f:
    text = 'Hello World'
    f.write(text.encode('utf-8'))
```

äžÑè£ŻåĹűI/Oè£ŸæIJL'äÿĂäÿłéšIJäÿžäžžç§ěçŽĎçL'źæĂğåřśæŸŕæŢřçżĎåŠŇCçżŞæđĎä¡Şçśžåđ́ÑèČ¡çŽ

```
import array
nums = array.array('i', [1, 2, 3, 4])
with open('data.bin','wb') as f:
    f.write(nums)
```

è£ŹäÿłéĂĆçŤÍäžŐäżżä¡ŢåőđçŐřäžĘèćńçğřäźŃäÿžâĂİçijŞåĘšæŐěåŔčâĂİçŽĎåŕźèšąïijÑè£Źçğ■åŕźèšąäi

äžÑè£ŻåĹűæŢřæ■őçŽĎåĘźåĚěářśæŸŕè£Źçśżæş■ä¡IJäźÑäÿĂãĂĆ

å¿Ĺåď Žářźèśąę£ŸåĚĄèőÿéĂŽè£Ğä¡£çŤĺæŰĞäžűářźèśąçŽĎ `readinto()` æŰźæşŢçŻť æŐěèŕżåŔŰäžÑè£ŻåĹűæŢřæ■őåĹřåĚűåžŢåśĆçŽĎåĘĚå■Ÿäÿ■åŐżãĂĆæŕŤåęĆïijŽ

```
>>> import array
>>> a = array.array('i', [0, 0, 0, 0, 0, 0, 0, 0])
>>> with open('data.bin', 'rb') as f:
...     f.readinto(a)
...
16
>>> a
array('i', [1, 2, 3, 4, 0, 0, 0, 0])
>>>
```

ä¡ĘæŸŕä¡£çŤĺè£Źçğ■æŁĂæIJŕçŽĎæŮűåĂŹéIJĂèęĄæäijåď ŰåřŔå£ČïijŇåŻääÿžåőČéĂŽåÿÿåĚůæIJL'åź åŔŕäžěæ§ěçIJŃ5.9åřŔèŁĆäÿ■åŔęåď ŰäÿĂäÿłèŕżåŔŰäžÑè£ŻåĹűæŢřæ■őåĹřåŔŕä£őæŤźçijŞåĘšåŇžçŽĎä¿Ńå

7.5 5.5 æŰĞäžűäÿ■å■ŸåIJĺæL'■èČ¡åĘŹåĚě

éŮőécŸ

ä¡ăæČşåČŔäÿĂäÿłæŰĞäžűäÿ■åĘŹåĚěæŢřæ■őïijŇä¡ĘæŸŕåL'■æŔŘå£ĚéążæŸŕè£ŹäÿłæŰĞäžűåIJĺæŰĞ äźş§åřśæŸŕäÿ■åĚĄèöÿèęEçŻŰåůšå■ŸåIJĺçŽĎæŰĞäžűåĘĚåőźãĂĆ

èğčåĘşæŰźæąĹ

åŔŕäžěåIJĺ `open()` åĞ¡æŢřäÿ■ä¡£çŤĺ `x` æĺąåijŔæİěäžčæŻ£ `w` æĺąåijŔçŽĎæŰźæşŢæİěèğčåĘşè£ŹäÿłéŮőécŸãĂĆæŕŤåęĆïijŽ

```
>>> with open('somefile', 'wt') as f:
...     f.write('Hello\n')
...
>>> with open('somefile', 'xt') as f:
...     f.write('Hello\n')
...
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
FileExistsError: [Errno 17] File exists: 'somefile'
>>>
```

åęĆæđIJæŰĞäžűæŸŕäžÑè£ŻåĹűçŽĎïijÑä¡£çŤĺ `xb` æİěäžčæŻ£ `xt`

èőĺèőž

è£ŹäÿĂåřŔèŁĆæijŤçď žäžĘåIJĺåĘŹæŰĞäžűæŮűéĂŽåÿÿäijŽéĄĞåĹřçŽĎäÿĂäÿłéŮőécŸçŽĎåőÑç¿Őèğ äÿĂäÿłæŻ£äžčæŰźæąĹæŸŕåĚĹæţŃèŕŢè£ŹäÿłæŰĞäžűæŸŕåŘęå■ŸåIJĺïijÑåČŔäÿÑéĺćè£Źæăůïijž

```
>>> import os
>>> if not os.path.exists('somefile'):
...     with open('somefile', 'wt') as f:
...         f.write('Hello\n')
... else:
...     print('File already exists!')
...
File already exists!
>>>
```

æŸ¿èĂŇæŸŞèğĄïijŇä¡£çŤĺxæŰĞäžűæĺąåijŕæŻťåŁăçőÅå■ŢãĂĆèęĄæşĺæĎŕçŽĎæŸŕxæĺąåijŕæŸŕäÿ
open() åĞ¡æŢřçL´źæIJL´çŽĎæL´ľåśŢãĂĆ åIJĺPythonçŽĎæŮğçL´ĹæIJňæĹŰèĂĚæŸŕPythonåőđçŐřçŽĎåžŢ

7.6 5.6 å■Ůçňęäÿšç ŽĎl/OæŞ■ä¡lJ

éŮőécŸ

ä¡ăæČşä¡£çŤĺæŞ■ä¡IJcśżæŰĞäžűåŕźèśąçŽĎçÍŃåžŔæİěæŞ■ä¡IJæŰĞæIJňæĹŰäžŇè£ŻåĹűå■Ůçňęäÿšã

èğčåĘşæŰźæąĹ

ä¡£çŤĺ io.StringIO() åŠŇ io.BytesIO() cśżæİěåĹŻåžžçśżæŰĞäžűåŕźèśąæŞ■ä¡IJå■ŮçňęäÿšæŢ

```
>>> s = io.StringIO()
>>> s.write('Hello World\n')
12
>>> print('This is a test', file=s)
15
>>> # Get all of the data written so far
>>> s.getvalue()
'Hello World\nThis is a test\n'
>>>

>>> # Wrap a file interface around an existing string
>>> s = io.StringIO('Hello\nWorld\n')
>>> s.read(4)
'Hell'
>>> s.read()
'o\nWorld\n'
>>>
```

io.StringIO åŔłèČ¡çŤĺäžŐæŰĞæIJñãĂĆåęĆæđIJä¡ăèęĄæŞ■ä¡IJäžŇè£ŻåĹűæŢřæ■őïijŇèęĄä¡£çŤĺ
io.BytesIO cśżæİěäžčæŻ£ãĂĆæŕŤåęĆïijŽ

```
>>> s = io.BytesIO()
>>> s.write(b'binary data')
>>> s.getvalue()
```

```
b'binary data'
>>>
```

èőĺèőž

å¡Şä¡äæČşæĺąæŃ§äÿĂäÿłæŹőéĂŽçŽĎæŰĞäžűçŽĎæŮűåĂŹ `StringIO` åŠŇ `BytesIO` çśżæŸŕå¿ĹæIJL'çŤĺçŽĎãĂĆ æŕŤåęĆïijŇåIJĺå■ŢåĚČæţŃèŕŢäÿ■ïijŇä¡ååŔŕäžěä¡£çŤĺ `StringIO` æİěåĹŽåžžäÿĂäÿłåŇĚåŘńæţŃèŕŢæŢřæ■őçŽĎçśżæŰĞäžűåŕźĕśąïijŇè£ŹäÿłåŕźĕśąåŔŕäžěèćńäijäçżŹæ§ŘäÿłåŔĆæŢřäÿžæŹőéĂŽæŰĞäžűåŕźĕśąçŽĎåĞ¡æŢřãĂĆ

éIJĂèęĄæşĺæĎŘçŽĎæŸŕïijŇ `StringIO` åŠŇ `BytesIO` åőđä¿ŃåźűæšąæIJL'æ■čçąőçŽĎæŢt'æŢřçśżåđŃçŽĎæŰĞäžűæŔŔè£řçňęãĂĆ åŽäæ■d'ïijŇåőČäžñäÿ■èČ¡åIJĺéĆčäžŽéIJĂèęĄä¡£çŤĺçIJ§åőđçŽĎçşżçż§çžğæŰĞäžűåęĆæŰĞäžűïijŇçőąéĄŞ

7.7 5.7 èŕżåĘŹåŐŃçijľæŰĞäžű

éŮőécŸ

ä¡äæČşèŕżåĘŹäÿĂäÿłgzipæĹŰbz2æäijåijŔçŽĎåŐŃçijľæŰĞäžűãĂĆ

èğčåĘşæŰźæąĹ

`gzip` åŠŇ `bz2` æĺąåİŮåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåd'ĎçŘĘè£ŹäžŽæŰĞäžűãĂĆ äÿd'äÿłæĺąåİŮéČ¡äÿž `open()` åĞ¡æŢřæŔŘä¿ŻäžĘåŔęåd'ŰçŽĎåőđçŐřæİěèğčåĘşè£ŹäÿłéŮőécŸãĂĆ æŕŤåęĆïijŇäÿžäžĘäžěæŰĞæIJňå¡ćåijŔèŕżåŔŰåŐŃçijľæŰĞäžűïijŇåŔŕäžěè£ŹæäůåĄŽïijŽ

```
# gzip compression
import gzip
with gzip.open('somefile.gz', 'rt') as f:
    text = f.read()

# bz2 compression
import bz2
with bz2.open('somefile.bz2', 'rt') as f:
    text = f.read()
```

çśżäijijçŽĎïijŇäÿžäžĘåĘŹåĚěåŐŃçijľæŢřæ■őïijŇåŔŕäžěè£ŹæäůåĄŽïijŽ

```
# gzip compression
import gzip
with gzip.open('somefile.gz', 'wt') as f:
    f.write(text)

# bz2 compression
import bz2
```

```
with bz2.open('somefile.bz2', 'wt') as f:
    f.write(text)
```

åęĆäÿŁïijŇæL’ĂæIJL’çŽDI/OæŞ■ä¡IJéČ¡ä¡£çŤĺæŰĞæIJňæĺąåijŔåźűæL’ğèąŇUnicodeçŽDçijŰçăĄ/èğčç çśżäijijçŽDïijŇåęĆæđIJä¡ăæČşæŞ■ä¡IJäžŇè£ŻåĹűæŢřæ■őïijŇä¡£çŤĺ `rb` æĹŰèĂĚ `wb` æŰĞäżűæĺąåijŔå■şåŔřãĂĆ

èőĺèőž

åď’ğéČĺåĹĘæČĚåĘţäÿŃèŕżåĘŹåŐŃçijľ’æŢřæ■őéČ¡æŸŕå¿ĹçőĂå■ŢçŽDãĂĆä¡ĘæŸŕèęĄæşĺæĎŔçŽDæŸ åęĆæđIJä¡ääÿ■æŇĞåőŽæĺąåijŔïijŇéĆčäźĹéżŸèőď’çŽDåřśæŸŕäžŇè£ŻåĹűæĺąåijŔïijŇåęĆæđIJè£ŹæŮűåĂŹç `gzip.open()` åŠŇ `bz2.open()` æŐěåŘŮèů§åĘĚç¡őçŽD `open()` åĞ¡æŢřäÿĂæăůçŽDåŔĆæŢřïijŇ åŇĚæŃň `encoding`ïijŇ`errors`ïijŇ`newline` ç■L’ç■L’ãĂĆ

å¡ŞåĘŹåĚěåŐŃçijľ’æŢřæ■őæŮűïijŇåŔřäżěä¡£çŤĺ `compresslevel` è£ŹäÿłåŔřéĂL’çŽDåĚşéŤőå■ŮåŔĆæŢřæİěæŇĞåőŽäÿĂäÿłåŐŃçijľ’çžğåĹńãĂĆærŤåęĆïijŽ

```
with gzip.open('somefile.gz', 'wt', compresslevel=5) as f:
    f.write(text)
```

éżŸèőď’çŽDç■L’çžğæŸŕ9ïijŇäź§æŸŕæIJĂéńŸçŽDåŐŃçijľ’ç■L’çžğãĂĆç■L’çžğèűŁä¡ŐæĂğèČ¡èűŁåě¡ïij

æIJĂåŘŐäÿĂçĆźïijŇ `gzip.open()` åŠŇ `bz2.open()` è£ŸæIJL’äÿĂäÿłå¿ĹářŠèćńç§ěéĄŞçŽDçL’źæĂğïijŇ åőČäžňåŔřäżěä¡IJçŤĺåIJĺäÿĂäÿłåůšå■ŸåIJĺåźűäżěäžŇè£

```
import gzip
f = open('somefile.gz', 'rb')
with gzip.open(f, 'rt') as g:
    text = g.read()
```

è£ŹæăůářšåĚĄèőÿ `gzip` åŠŇ `bz2` æĺąåİŮåŔřäżěåůěä¡IJåIJĺèőÿåď’ŽçśżæŰĞäżűářźèśąäÿŁïijŇærŤåęĆåě

7.8 5.8 åŻžåőŽåď’gářŔèőřå¡ŢçŽDæŰĞäżűè£■äžc

éŮőécŸ

ä¡ăæČşåIJĺäÿĂäÿłåŻžåőŽéŢ£åžęèőřå¡ŢæĹŰèĂĚæŢřæ■őåİŮçŽDéŽęåŘĹäÿŁè£■äžčïijŇèĂŇäÿ■æŸŕåIJ

èğčåĘşæŰźæąĹ

éĂŽè£ĞäÿŃéİćè£ŹäÿłářŔæŁĂåůğä¡£çŤĺ `iter` åŠŇ `functools.partial()` åĞ¡æŢřijŽ

```
from functools import partial

RECORD_SIZE = 32
```

```
with open('somefile.data', 'rb') as f:
    records = iter(partial(f.read, RECORD_SIZE), b'')
    for r in records:
        ...
```

è£Źäÿłä¿Ńå■Řäÿ■çŽĎ `records` åŕźèśąæŸŕäÿĂäÿłåŔŕè£■äžčåŕźèśąïijŇåőČäijŽäÿ■æŰ■çŽĎäžğçŢ§åŻžèęĄæşĺæĎŔçŽĎæŸŕåęĆæđIJæĂżèőŕå¡Ţåď ğåřŔäÿ■æŸŕåİŮåď ğåřŔçŽĎæŢť æŢřåĂ■çŽĎèŕÏïijŇæIJĂåŘŐäÿĂ

èőÍèőž

`iter()` åĞ¡æŢřæIJL'äÿĂäÿłéšIJäÿžäžžç§ěçŽĎçL'źæĂğåřśæŸŕïijŇåęĆæđIJä¡äçżŹåőČäijåéĂŠäÿĂäÿłåI
è£Źäÿłè£■äžčåŹĺäijŽäÿĂçŻť'èřČçŢĺäijååĚěçŽĎåŔŕèřČçŢĺåŕźèśąçŻť'åĹřåőČè£ŢåŻđæăĞèőŕåÄijäÿžæ■ćïijŇè£

åIJĺä¿Ńå■Řäÿ■ïijŇ `functools.partial` çŢĺæİěåĹŻåžžäÿĂäÿłæŕŔæňąèćńèřČçŢĺæŮűäžŐæŰĞäžűä
æăĞèőŕåĂij `b''` åřśæŸŕå¡ŞåĹřè¿¿æŰĞäžűçżŞåř¿æŮűçŽĎè£ŢåŻđåÄijãĂĆ

æIJĂåŘŐåĘ■æŔŘäÿĂçĆźïijŇäÿŁéİćçŽĎä¿Ńå■Řäÿ■çŽĎæŰĞäžűæŮűäžěäžŇè£ŻåĹűæĺąåijŔæĽŞåijĂç
åęĆæđIJæŸŕèŕźåŔŰåŻžåőŽåď ğåřŔçŽĎèőŕå¡ŢïijŇè£ŹéĂŽåÿÿæŸŕæIJĂæŹőéĄ■çŽĎæČĚåĘţãĂĆ
èĂŇåŕźäžŐæŰĞæIJñæŰĞäžűïijŇäÿĂèąŇäÿĂèąŇçŽĎèŕźåŔŰ(éżŸèőď'çŽĎè£■äžčèąŇäÿž)æŻť'æŹőéĄ■çĆźã

7.9 5.9 èŕźåŔŰäžŇè£ŻåĹűæŢřæ■őåĹřåŔŕåŔŸçijŞåĘšåŇžäÿ■

éŮőécŸ

ä¡ăæČşçŻť'æŐěèŕźåŔŰäžŇè£ŻåĹűæŢřæ■őåĹřäÿĂäÿłåŔŕåŔŸçijŞåĘšåŇžäÿ■ïijŇèĂŇäÿ■éIJĂèęĄåĄŽäż
æĹŰèĂĚä¡ăæČşåŐ§åIJřä£őæŤźæŢřæ■őåźűårĘåőČåĘŹåŻđåĹřäÿĂäÿłæŰĞäžűäÿ■åŐżãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèŕźåŔŰæŢřæ■őåĹřäÿĂäÿłåŔŕåŔŸæŢřçżĎäÿ■ïijŇä¡£çŢĺæŰĞäžűåŕźèśąçŽĎ `readinto()` æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
import os.path

def read_into_buffer(filename):
    buf = bytearray(os.path.getsize(filename))
    with open(filename, 'rb') as f:
        f.readinto(buf)
    return buf
```

äÿŃéİćæŸŕäÿĂäÿłæijŤçd'žè£ŹäÿłåĞ¡æŢřä¡£çŢĺæŰźæşŢçŽĎä¿Ńå■ŘïijŽ

```
>>> # Write a sample file
>>> with open('sample.bin', 'wb') as f:
...     f.write(b'Hello World')
...
```

```
>>> buf = read_into_buffer('sample.bin')
>>> buf
bytearray(b'Hello World')
>>> buf[0:5] = b'Hallo'
>>> buf
bytearray(b'Hallo World')
>>> with open('newsample.bin', 'wb') as f:
...     f.write(buf)
...
11
>>>
```

èőÍèőž

æŰĞäżűåŕźèśąçŽĎ `readinto()` æŰźæşŢèČ¡èćńçŤÍæİěäÿžécĎåĚĹåĹĘéĚ■åĘĚå■ŸçŽĎæŢřçżĎåąńåĚ `array` æÍąåİŮæĹŰ `numpy` åžŞåĹŻåżžçŽĎæŢřçżĎãĂĆ åŠŇæŹőéĂŽ `read()` æŰźæşŢäÿ■åŘŇçŽĎæŸŕïijŇ `readinto()` åąńåĚĚåůšå■ŸåIJÍçŽĎçijŞåĘšåŇžèĂŇäÿ■æŸŕäÿžæŰřåŕźèśąéĞ åŽăæ■ď˙ïijŇä¡ăåŔŕäžěä¡£çŤĺåőČæİěéĄ£åĚ■åď˙ğéĞŔçŽĎåĘĚå■ŸåĹĘéĚ■æŞ■ä¡IJãĂĆ æŕŤåęĆïijŇåęĆæđIJä¡ăèŕžåŔŰäÿĂäÿłçŤśçŻÿåŘŇåď˙ğåřŔçŽĎèőřå¡ŢçżĎæĹŘçŽĎäžŇè£ŻåĹűæŰĞäżűæŮűïi

```
record_size = 32 # Size of each record (adjust value)

buf = bytearray(record_size)
with open('somefile', 'rb') as f:
    while True:
        n = f.readinto(buf)
        if n < record_size:
            break
        # Use the contents of buf
        ...
```

åŔęåď˙ŰæIJL˙äÿĂäÿłæIJL˙èűčçL˙źæĂğåřśæŸŕ `memoryview` ïijŇ åőČåŔŕäžěéĂŽè£ĞéŽűåď˙■åĹűçŽĎæŰźåijŔåŕźåůšå■ŸåIJÍçŽĎçijŞåĘšåŇžæL˙ğèąŇåĹĞçL˙ĞæŞ■ä¡IJïijŇçŤŽè

```
>>> buf
bytearray(b'Hello World')
>>> m1 = memoryview(buf)
>>> m2 = m1[-5:]
>>> m2
<memory at 0x100681390>
>>> m2[:] = b'WORLD'
>>> buf
bytearray(b'Hello WORLD')
>>>
```

ä¡£çŤÍ `f.readinto()` æŮűéIJĂèęĄæşÍæĎŔçŽĎæŸŕïijŇä¡ăå£ĚéążæčĂæ§ěåőČçŽĎè£ŤåŽđåĂijïijŇä

åęĆæđIJå■ŮèŁĆæŢřåŕŔäžŐçijŞåĘšåŇžåď˙ğåřŔïijŇèąÍæŸŐæŢřæ■őèćńæĹłæŰ■æĹŰèĂĚèćńçăť˙åİŔäžĘ

æIJĂåŘŐïijŇçŢŹå£ČèğĆåŕ§åĚűäžŰåĞ¡æŢřåžŞåŠŇæÍąåİŮäÿ■åŠŇ `into`

çŻÿåĚşçŽĎåĞ¡æŢř(æŕŤåęĆ recv_into() ïijŇ pack_into() ç■Ľ)ãĂĆ PythonçŽĎå¿Ĺåď ŽåĚűäżŰéČĺåĹĘåůšçżŔèČ¡æŤŕæŇĄçŻť æŐěçŽĎI/OæĹŰæŢřæ■őèő£éŮőæŞ■ä¡IJïijŇè£Źä

åĚşäžŐèğčæđŘäžŇè£ŻåĹűçżŞæđĎåŠŇ memoryviews ä¡£çŤĺæŰźæşŢçŽĎæŻť éńŸçžğä¿Ńå■ŘïijŇèŕůåŔĆèĂČ6.12åřŔèŁĆãĂĆ

7.10 5.10 åĘĚå■ŸæŸăåřĎçŽĎäžŇè£ŻåĹűæŰĞäżű

éŮőéćŸ

ä¡ăæČşåĘĚå■ŸæŸăåřĎäÿĂäÿłäžŇè£ŻåĹűæŰĞäżűåĹřäÿĂäÿłåŔŕåŔŸå■ŮèŁĆæŢřçżĎäÿ■ïijŇçŻőçŽĎåĺ

èğčåĘşæŰźæąĹ

ä¡£çŤĺ mmap æĺąåİŮæİěåĘĚå■ŸæŸăåřĎæŰĞäżűãĂĆ äÿŃéİćæŸŕäÿĂäÿłåůěåĚůåĞ¡æŢřïijŇåŘŚä¡ăæijŤçď žäžĘåęĆä¡ŢæĽŞåijĂäÿĂäÿłæŰĞäżűåźűäżěäÿĂçğ■ä¿£æ

```
import os
import mmap

def memory_map(filename, access=mmap.ACCESS_WRITE):
    size = os.path.getsize(filename)
    fd = os.open(filename, os.O_RDWR)
    return mmap.mmap(fd, size, access=access)
```

äÿžäžĘä¡£çŤĺè£ŹäÿłåĞ¡æŢřïijŇä¡ăéIJĂèęĄæIJĽäÿĂäÿłåůšåĹŻåżžåźűäÿŤåĘĚåőźäÿ■äÿžçľžçŽĎæŰĞäż äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇæŢŹä¡ăæĂŐæăůåĹİåğŃåĹŻåżžäÿĂäÿłæŰĞäżűåźűåřĘåĚűåĘĚåőźæĽľåĚĚåĹř

```
>>> size = 1000000
>>> with open('data', 'wb') as f:
...     f.seek(size-1)
...     f.write(b'\x00')
...
>>>
```

äÿŃéİćæŸŕäÿĂäÿłåĹľçŤĺ memory_map() åĞ¡æŢřçśżåĘĚå■ŸæŸăåřĎæŰĞäżűåĘĚåőźçŽĎä¿Ńå■ŘïijŽ

```
>>> m = memory_map('data')
>>> len(m)
1000000
>>> m[0:10]
b'\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
>>> m[0]
0
>>> # Reassign a slice
>>> m[0:11] = b'Hello World'
>>> m.close()
```

```
>>> # Verify that changes were made
>>> with open('data', 'rb') as f:
... print(f.read(11))
...
b'Hello World'
>>>
```

mmap() è£ŤåŻđçŽĎ mmap åŕźèśąåŘŇæăůäź§åŔŕäžěä¡IJäÿžäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹÍæÌěä¡£çŤĺïij
è£ŹæŮűůåĂŹåžŢåśÇçŽĎæŰĞäžűäijŽèćńèĞłåŁÍåĚşéŮ■ãĂĆærŤåęĆïijŽ

```
>>> with memory_map('data') as m:
...     print(len(m))
...     print(m[0:10])
...
1000000
b'Hello World'
>>> m.closed
True
>>>
```

éżŸèőd'æČĚåĘţäÿŃïijŇ memeory_map() åĞ¡æŢřæL'ŞåijĂçŽĎæŰĞäžűåŘŇæŮűæŤŕæŇĄèŕžåŠŇåĘŹ
äżżä¡ŢçŽĎä£őæŤźåĘĚåőźéČ¡äijŽåd'■åĹűåŻđåŐ§æÌěçŽĎæŰĞäžűäÿ■ãĂĆ
åęĆæđIJéIJĂèęĄåŔłèŕżçŽĎè£éŮőæÍąåijŔïijŇåŔŕäžěçżŹåŔĆæŢř access èţŃåĂijäÿž
mmap.ACCESS_READ ãĂĆærŤåęĆïijŽ

```
m = memory_map(filename, mmap.ACCESS_READ)
```

åęĆæđIJä¡ăæČşåIJÍæIJňåIJřä£őæŤźæŢřæ■őïijŇä¡ĘæŸŕåŔĹäÿ■æČşåřĘä£őæŤźåĘŹåŻđåĹřåŐ§ågŃæŰĞ
mmap.ACCESS_COPY ïijŽ

```
m = memory_map(filename, mmap.ACCESS_COPY)
```

èőÍèőž

äÿžäžĘéŽŔæIJžèő£éŮőæŰĞäžűçŽĎåĘĚåőźïijŇä¡£çŤÍ mmap
åřĘæŰĞäžűæŸăåřĎåĹřåĘĚå■ŸäÿæŸŕäÿĂäÿłéńŸæŢĹåŠŇäijŸéŻĘçŽĎæŰźæşŢãĂĆ
ä¿ŃåęĆïijŇä¡ăæŮăéIJĂæL'ŞåijĂäÿĂäÿłæŰĞäžűåźűæL'ğèąŇåď'ğéĞŔçŽĎ seek() ïijŇ
read() ïijŇ write() èřČçŤÍïijŇ åŔłéIJĂèęĄçőĂå■ŢçŽĎæŸăåřĎæŰĞäžűåźűä¡£çŤÍåĹĞçL'ĞæŞ■ä¡IJèő£é

äÿĂèĹňæÌěèőšïijŇ mmap() æL'ĂæŽť'éIJšçŽĎåĘĚå■ŸçIJŃäÿŁåŐżåřśæŸŕäÿĂäÿłäžŇè£ŻåĹűæŢřçżĎårź
ä¡ĘæŸŕïijŇä¡ăåŔŕäžěä¡£çŤÍäÿĂäÿłåĘĚå■ŸèğĘåŻ¿æÌěèğčæđŘåĚűäÿ■çŽĎæŢřæ■őãĂĆærŤåęĆïijŽ

```
>>> m = memory_map('data')
>>> # Memoryview of unsigned integers
>>> v = memoryview(m).cast('I')
>>> v[0] = 7
>>> m[0:4]
b'\x07\x00\x00\x00'
>>> m[0:4] = b'\x07\x01\x00\x00'
```

```
>>> v[0]
263
>>>
```

éIJÄèęĄåijžèřČçŽĎäÿÄçĆźæŸŕïijŇåĘĚå■ŸæŸäářĎäÿÄäÿłæŰĞäżűåźűäÿ■äijŽářijèĞť'æŢť'äÿłæŰĞäżűë
äź§åřśæŸŕèŕť'ïijŇæŰĞäżűåźűæšąæIJL'èćńåď'■åĹűåĹřåĘĚå■ŸçijŞå■ŸæĹŰæŢřçżĎäÿ■ãĂĆçŻÿåŔ■ïijŇæŞ■ä¡
å¡Şä¡ãèő£éŮőæŰĞäżűçŽĎäÿ■åŘŇåŇžå§§æŮűïijŇè£ŹäžŻåŇžå§§çŽĎåĘĚåőźæL'■æăźæ■őéIJÄèęĄèćńèŕžåŔ
èĂŇéĆčäžŻäżŐæšąèćńèő£éŮőåĹřçŽĎéČĺåĹĘè£ŸæŸŕçŢŹåIJÍçčĄçŻŸäÿŁãĂĆæL'ĂæIJL'è£ŹäžŻè£ĞçÍŃæŸ

åęĆæđIJåď'ŽäÿłpythonèğčéĞŁåŹĺåĘĚå■ŸæŸäářĎåŘŇäÿĂäÿłæŰĞäżűïijŇå¿ŮåĹřçŽĎ
`mmap` åŕźèśąèČ¡åď'§èćńçŤĺæİěåIJÍèğčéĞŁåŹĺçŻt'æŐëäžď'æ■ćæŢřæ■őãĂĆ
äź§åřśæŸŕèŕť'ïijŇæL'ĂæIJL'èğčéĞŁåŹĺéČ¡èČ¡åŘŇæŮűèŕžåĘŹæŢřæ■őïijŇåźűäÿŤåĚűäÿ■äÿĂäÿłèğčéĞŁåŹĺa
å¿ĹæŸŐæŸ¿ïijŇè£ŹéĞŇéIJÄèęĄèĂČèŹŠåŘŇæ■ěçŽĎéŮőécŸãĂĆä¡ĘæŸŕè£Źçğ■æŰźæşŢæIJL'æŮűåĂŹåŔ

è£ŹäÿĂåřŔèŁĆäÿ■åĞ¡æŢřår¡éĞŔåĘŹå¿Ůå¿ĹéĂŽçŤĺïijŇåŘŇæŮűéĂĆçŤĺäžŐUnixåŠŇWindowsåźşåŔì
èęĄæşĺæĎŔçŽĎæŸŕä¡£çŤĺ `mmap()` åĞ¡æŢřæŮűäijŽåIJÍåžŢåśĆæIJL'äÿĂäžŻåźşåŘŕçŽĎåůőåijĆæĂğãĂĆ
åŘęåď'ŰïijŇè£ŸæIJL'äÿĂäžŻéĂL'éąźåŔŕäžěçŤĺæİěåĹŻåżžåŇ£åŘ■çŽĎåĘĚå■ŸæŸäářĎåŇžå§§ãĂĆ
åęĆæđIJä¡åářźè£ŹäÿłæĎ§åĚt'èűčïijŇçąőä£Ïä¡ăäžŤçżĘçăŤèŕžäžĘPythonæŰĞæąčäÿ■
è£ŹæŰźéIćçŽĎåĘĚåőź ãĂĆ

7.11 5.11 æŰĞäżűèůŕå¿ĎåŘ■çŽĎæŞ■ä¡IJ

éŮőécŸ

ä¡ăéIJÄèęĄä¡£çŤĺèůŕå¿ĎåŘ■æİěèŐůåŔŰæŰĞäżűåŘ■ïijŇçŻőå¡ŢåŘ■ïijŇçżİåŕźèůŕå¿Ďç■L'ç■L'ãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺ `os.path` æĺąåİŮäÿ■çŽĎåĞ¡æŢřæİěæŞ■ä¡IJèůŕå¿ĎåŘ■ãĂĆ
äÿŇéIćæŸŕäÿĂäÿłäžď'äžŠåijŔä¿Ńå■ŘæİěæijŤçď'žäÿĂäžŻåĚşéŤőçŽĎçL'źæĂğïijŽ

```
>>> import os
>>> path = '/Users/beazley/Data/data.csv'

>>> # Get the last component of the path
>>> os.path.basename(path)
'data.csv'

>>> # Get the directory name
>>> os.path.dirname(path)
'/Users/beazley/Data'

>>> # Join path components together
>>> os.path.join('tmp', 'data', os.path.basename(path))
'tmp/data/data.csv'

>>> # Expand the user's home directory
>>> path = '~/Data/data.csv'
```

```
>>> os.path.expanduser(path)
'/Users/beazley/Data/data.csv'

>>> # Split the file extension
>>> os.path.splitext(path)
('~/Data/data', '.csv')
>>>
```

èőÍèőž

åŕźäžŐäżżä¡ŢçŽĎæŰĞäžűåŘ■çŽĎæŞ■ä¡IJïijŇä¡äéČ¡åžŤèŕěä¡£çŤÍ os.path æÍąåİŮïijŇèĂŇäÿ■æŸŕä¡£çŤÍæăĞåĞĘå■ŮçňęäÿšæŞ■ä¡IJæİěæđĎéĂäèĞłåůśçŽĎäżčçăĄãĂĆçL'źåĹńæŸŕäÿžäžĘåŔŕçğżæď■æĂğèĂČèŹŚçŽĎæŮűåĂŹæŻť'åžŤåęĆæ■ď'ïijŇ åŽääÿž os.path æÍąåİŮç§ěéĄŞUnixåŠŇWindowsçşżçż§äźŃéŮť'çŽĎåůőåijĆåźűäÿŤèČ¡åď'§åŔŕéİääIJřåď'ĎçŘĘçśżäijij Data/data.csv åŠŇ Data\data.csv è£ŹæăůçŽĎæŰĞäžűåŘ■ãĂĆ åĚűæňąïijŇä¡äçIJ§çŽĎäÿ■åžŤèŕěæţłèť'źæŮűéŮť'åŐżéĞ■åď'■éĂäè¡őå■ŘãĂĆéĂŽåÿÿæIJĂåě¡æŸŕçŻť'æŐëä¡£

èęĄæşÍæĎŔçŽĎæŸŕ os.path è£ŸæIJL'æŻť'åď'ŽçŽĎåŁ§èČ¡åIJÍè£ŹéĞŇåźűæšąæIJL'åĹŮäÿ¿åĞžæİěãåŔŕäžěæ§ěéŸĚåőŸæŰźæŰĞæąčæİěèŐůåŔŰæŻť'åď'ŽäÿŐæŰĞäžűæţŃèŕŢïijŇçňęåŘůéŞ¿æŐëç■L'çŽÿåĚşçŽ

7.12 5.12 æţŃèŕŢæŰĞäžűæŸŕåŘęå■ŸåIJÍ

éŮőécŸ

ä¡ăæČşæţŃèŕŢäÿĂäÿłæŰĞäžűæĹŰçŻőå¡ŢæŸŕåŘęå■ŸåIJÍãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ os.path æÍąåİŮæİěæţŃèŕŢäÿĂäÿłæŰĞäžűæĹŰçŻőå¡ŢæŸŕåŘęå■ŸåIJÍãĂĆæŕŤåęĆïijŽ

```
>>> import os
>>> os.path.exists('/etc/passwd')
True
>>> os.path.exists('/tmp/spam')
False
>>>
```

ä¡ăè£ŸèČ¡è£ŻäÿĂæ■ěæţŃèŕŢè£ŹäÿłæŰĞäžűæŮűäżĂäźĹçśżåđŃçŽĎãĂĆ åIJÍäÿŃéİćè£ŹäžŻæţŃèŕŢäÿ■ïijŇåęĆæđIJæţŃèŕŢçŽĎæŰĞäžűäÿ■å■ŸåIJÍçŽĎæŮűåĂŹïijŇçżŞæđIJéČ¡äijŽè

```
>>> # Is a regular file
>>> os.path.isfile('/etc/passwd')
True

>>> # Is a directory
>>> os.path.isdir('/etc/passwd')
```

```
False

>>> # Is a symbolic link
>>> os.path.islink('/usr/local/bin/python3')
True

>>> # Get the file linked to
>>> os.path.realpath('/usr/local/bin/python3')
'/usr/local/bin/python3.3'
>>>
```

åęĆædIJä¡ăè£ŸæČşèŐůåŔŰåĚČæŢřæ■ő(æŕŤåęĆæŰĞäżűåď˘ğářŔæĹŰèĂĚæŸŕă£őæŤźæŮěæIJ§)ïijŇäź
`os.path` æĺąåİŮæİěèğčåEşïijŽ

```
>>> os.path.getsize('/etc/passwd')
3669
>>> os.path.getmtime('/etc/passwd')
1272478234.0
>>> import time
>>> time.ctime(os.path.getmtime('/etc/passwd'))
'Wed Apr 28 13:10:34 2010'
>>>
```

èőÍèőž

ä¡£çŤÍ `os.path` æİěè£ŽèąŇæŰĞäżűæţŃèŕŢæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆ
åIJĺåĘŹè£ŹäžŽèĎŽæIJňæŮűïijŇåŔŕèČ¡åŤŕäÿĂéIJĂèęĄæşĺæĎŔçŽĎåřśæŸŕă¡ăéIJĂèęĄèĂČèŹŚæŰĞäżűæİČ

```
>>> os.path.getsize('/Users/guido/Desktop/foo.txt')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "/usr/local/lib/python3.3/genericpath.py", line 49, in␣
 ↪getsize
        return os.stat(filename).st_size
PermissionError: [Errno 13] Permission denied: '/Users/guido/
 ↪Desktop/foo.txt'
>>>
```

7.13 5.13 èŐůåŔŰæŰĞäżűåď´žäÿ■çŽĎæŰĞäżűåĹŮèąĺ

éŮőécŸ

ä¡ăæČşèŐůåŔŰæŰĞäżűçşżçż§äÿ■æ§ŘäÿłçŻőå¡ŢäÿŃçŽĎæĽĂæIJĽæŰĞäżűåĹŮèąĺãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `os.listdir()` åĞ¡æŢřæİěèŐůåŔŰæ§ŘäÿłçŻőå¡Ţäÿ■çŽĎæŰĞäžűåĹŮèąĺïijŽ

```
import os
names = os.listdir('somedir')
```

çžŞæđIJäijŽè£ŤåŻđçŻőå¡Ţäÿ■æĽĂæIJL'æŰĞäžűåĹŮèąĺïijŇåŇĚæŃňæĽĂæIJL'æŰĞäžűïijŇå■ŘçŻőå¡Ţ
åęĆæđIJä¡ăéIJĂèęĄéĂŽè£Ğæ§Řçğ■æŰźåijŔè£Ğæżď'æŢřæ■őïijŇåŔŕäžěèĂČèŹŚçżŞåŘĹ
`os.path` åžŞäÿ■çŽĎäÿĂäžŻåĞ¡æŢřæİěä¡£çŤĺåĹŮèąĺæŐĺåŕijãĂĆæŕŤåęĆïijŽ

```
import os.path

# Get all regular files
names = [name for name in os.listdir('somedir')
        if os.path.isfile(os.path.join('somedir', name))]

# Get all dirs
dirnames = [name for name in os.listdir('somedir')
        if os.path.isdir(os.path.join('somedir', name))]
```

å■Ůçňęäÿšç ŽĎ `startswith()` åŠŇ `endswith()`
æŰźæşŢåŕźäžŐè£ĞæżďäÿĂäÿłçŻőå¡ŢçŽĎåĘĚåőźäź§æŸŕå¿ĹæIJL'çŤĺçŽĎãĂĆæŕŤåęĆïijŽ

```
pyfiles = [name for name in os.listdir('somedir')
           if name.endswith('.py')]
```

åŕźäžŐæŰĞäžűåŘ■çŽĎåŇźéĚ■ïijŇä¡ăåŔŕèČ¡äijŽèĂČèŹŚä¡£çŤÍ `glob` æĹŰ `fnmatch`
æĺąåİŮãĂĆæŕŤåęĆïijŽ

```
import glob
pyfiles = glob.glob('somedir/*.py')

from fnmatch import fnmatch
pyfiles = [name for name in os.listdir('somedir')
           if fnmatch(name, '*.py')]
```

èŐĺèŐž

èŐůåŔŰçŻőå¡Ţäÿ■çŽĎåĹŮèąĺæŸŕå¿Ĺåőźæ ŸŞçŽĎïijŇä¡ĘæŸŕåĚűè£ŤåŻđçżŞæđIJåŔłæŸŕçŻőå¡Ţäÿ■åĉ
åęĆæđIJä¡ăè£ŸæČşèŐůåŔŰäĚűäžŰçŽĎåĚČä£ąæĄŕïijŇæŕŤåęĆæŰĞäžűåđ'ğåřŔïijŇä£őæŤźæŮűéŮť'ç■Ľ'ç■Ľ
ä¡ăæĹŰèőÿè£ŸéIJĂèęĄä¡£çŤĺåĹř `os.path` æĺąåİŮäÿ■çŽĎåĞ¡æŢřæĹŰçİĂ `os.stat()`
åĞ¡æŢřæİěæŤűéŻĘæŢřæ■őãĂĆæŕŤåęĆïijŽ

```
# Example of getting a directory listing

import os
import os.path
import glob
```

```
pyfiles = glob.glob('*.py')

# Get file sizes and modification dates
name_sz_date = [(name, os.path.getsize(name), os.path.
↪getmtime(name))
                for name in pyfiles]
for name, size, mtime in name_sz_date:
    print(name, size, mtime)

# Alternative: Get file metadata
file_metadata = [(name, os.stat(name)) for name in pyfiles]
for name, meta in file_metadata:
    print(name, meta.st_size, meta.st_mtime)
```

æIJÄåŘŐè£ŸæIJL'äÿĂçĆźèęĄæşÍæĎŔçŽĎåřśæŸřïijŇæIJL'æŮűåĂŹåIJÍåď'ĎçŘĘæŰĞäżűåŘ■çijŰçăĄéĂŽåÿÿæİěèőšïijŇåĞ¡æŢř `os.listdir()` è£ŤåŻđçŽĎåőđä¡ŞåĹŮèąÍäijŽæäźæ■őçşžçż§éžŸèőď'çŽĎæŰĞäą¡ĘæŸřæIJL'æŮűåĂŹäź§äijŽçćřåĹřäÿĂäžŽäÿ■èČ¡æ■čåÿÿèğčçăĄçŽĎæŰĞäžűåŘ■ãĂĆ åĚşäžŐæŰĞäżűåŘ■çŽĎåď'ĎçŘĘéŮőécŸïijŇåIJÍ5.14åŠŇ5.15åřŔèŁĆæIJL'æŻt'èŕęçżĘçŽĎèőšèğčãĂĆ

7.14 5.14 å£¡çŢěæŰĞäżűåŘ■çijŰçăĄ

éŮőécŸ

ä¡ăæČşä¡£çŤĺåŐ§ågŃæŰĞäżűåŘ■æL'ğèąŇæŰĞäžűçŽĎI/OæŞ■ä¡IJïijŇäź§åřśæŸŕèŕť'æŰĞäżűåŘ■åźűæš

èğčåĘşæŰźæąĹ

éżŸèőď'æČĚåĘţäÿŃïijŇæL'ĂæIJL'çŽĎæŰĞäżűåŘ■éČ¡äijŽæäźæ■ő `sys.getfilesystemencoding()` è£ŤåŻđçŽĎæŰĞæIJñçijŰçăĄæİěçijŰçăĄæĹŰèğčçăĄãĂĆæŕŤåęĆïijŽ

```
>>> sys.getfilesystemencoding()
'utf-8'
>>>
```

åęĆædIJåŽääÿžæ§Řçğ■åŐ§åŽää¡ăæČşå£¡çŢěè£Źçğ■çijŰçăĄïijŇåŔŕäżěä¡£çŤĺäÿĂäÿłåŐ§ågŃå■ŮèŁĆå

```
>>> # Wrte a file using a unicode filename
>>> with open('jalape\xf1o.txt', 'w') as f:
...     f.write('Spicy!')
...
6
>>> # Directory listing (decoded)
>>> import os
>>> os.listdir('.')
['jalapeÃśo.txt']
```

```
>>> # Directory listing (raw)
>>> os.listdir(b'.') # Note: byte string
[b'jalapen\xcc\x83o.txt']

>>> # Open file with raw filename
>>> with open(b'jalapen\xcc\x83o.txt') as f:
...     print(f.read())
...
Spicy!
>>>
```

æ■čåęĆä¡ăæĽĂèğĄïijŇåIJÍæIJĂåŘŐäÿď äÿłæŞ■ä¡IJäÿ■ïijŇå¡Şä¡ăçżŹæŰĞäżűçŻÿåĚşåĞ¡æŢřåęĆ open() åŠŇ os.listdir() äijăéĂŠå■ŮèŁĆå■ŮçňęäÿšæŮűïijŇæŰĞäżűåŘ■çŽĎåď ĎçŘĘæŰźåijŔäijŽçÍ

èőÍèőž

éĂŽåÿÿæİěèőšïijŇä¡ăäÿ■éIJĂèęĄæŃĚå£ČæŰĞäżűåŘ■çŽĎçijŰçăĄåŠŇèğčçăĄïijŇæŹőéĂŽçŽĎæŰĞäż ä¡ĘæŸŕïijŇæIJĽäžŻæŞ■ä¡IJçşżçż§åĚĄèőÿçŤÍæĹůéĂŽè£ĞåĄűçĎűæĹŰæĄűæĎŔæŰźåijŔåŐżåĹŻåżžåŘ■å■ è£ŹäžŻæŰĞäżűåŘ■åŔŕèČ¡äijŽçěđçğŸåIJřäÿ■æŰ■éĆčäžŻéIJĂèęĄåď ĎçŘĘåď ğéĞŔæŰĞäżűçŽĎPythonçÍŃ

èŕżåŔŰçŻőå¡ŢåźűéĂŽè£ĞåŐ§åğŃæIJłèğčçăĄæŰźåijŔåď ĎçŘĘæŰĞäżűåŘ■åŔŕäżěæIJĽæŢĹçŽĎéĄ£å ář¡çőąè£ŹæăůäijŽåÿęæİěäÿĂåőŽçŽĎçijŰçÍŃéŽ¿åžęãĂĆ

åĚşäžŐæĽŞå■řäÿ■åŔŕèğčçăĄçŽĎæŰĞäżűåŘ■ïijŇèŕůåŔĆèĂČ5.15åřŔèŁĆãĂĆ

7.15 5.15 æĽŞå■řäÿ■åŘĹæşŢçŽĎæŰĞäżűåŘ■

éŮőéćŸ

ä¡ăçŽĎçÍŃåžŔèŐůåŔŰäžĘäÿĂäÿłçŻőå¡Ţäÿ■çŽĎæŰĞäżűåŘ■åĹŮèąĺïijŇä¡ĘæŸŕå¡ŞåőČèŕŢçİĂåŐżæĽŞ åĞžçŐřäžĘ UnicodeEncodeError åijĆåÿÿåŠŇäÿĂæİąåěĞæĂłçŽĎæűĹæĄŕâĂŤâĂŤ surrogates not allowed ãĂĆ

èğčåĘşæŰźæąĹ

å¡ŞæĽŞå■řæIJłç§ěçŽĎæŰĞäżűåŘ■æŮűïijŇä¡£çŤĺäÿŃéİćçŽĎæŰźæşŢåŔŕäżěéĄ£åĚ■è£ŹæăůçŽĎéŤŹè

```
def bad_filename(filename):
    return repr(filename)[1:-1]

try:
    print(filename)
except UnicodeEncodeError:
    print(bad_filename(filename))
```

èőÍèőž

è£ŹäÿĂåřŔèŁĆèőÍèőžçŽĎæŸŕåIJÍçijŰåĘŹå£Ěéąžåď ĎçŘĘæŰĞäžűçşžçż§çŽĎçÍŃåžŔæŮűäÿĂäÿłäÿ■åd
éżŸèőď æČĚåĘţäÿŃïijŇPythonåĄĞåőŽæĽĂæIJĽæŰĞäžűåŘ■éČ¡åůšçżŔæăźæ■ő
`sys.getfilesystemencoding()` çŽĎåĂijçijŰçăĄè£ĞäžĘãĂĆ
ä¡ĘæŸŕïijŇæIJĽäÿĂäžŽæŰĞäžűçşžçż§åźűæšąæIJĽåijžåĹűèęĄæśĆè£ŹæăůåĄŽïijŇåŽăæ■ď åĚĄèőÿåĹŽåžž
è£ŹçğæČĚåĘţäÿ■åď łåÿÿèğĄïijŇä¡ĘæŸŕæĂżäijŽæIJĽäžŽçŤÍæĹůåĘŠéŹĺ è£ŹæăůåĄŽæĹŰèĂĚæŸŕæŮăæĎ
åŔŕèČ¡æŸŕåIJÍäÿĂäÿłæIJĽçijžéŹůçŽĎäžčçĂąäÿ■çżŹ `open()`
åĞ¡æŢřäijăéĂŠäžĘäÿĂäÿłäÿ■åŘĹèğĎèŇČçŽĎæŰĞäžűåŘ■)ãĂĆ

å¡ŞæĽğèąŇçśžäijij `os.listdir()` è£ŹæăůçŽĎåĞ¡æŢřæŮűïijŇè£ŹäžŽäÿ■åŘĹèğĎèŇČçŽĎæŰĞäžű
äÿĂæŰźéİćïijŇåőČäÿ■èČ¡äžĚäžĚåŔłæŸŕäÿćåijČè£ŹäžŽäÿ■åŘĹæăijçŽĎåŘ■å■ŮãĆèĂŇåŔęäÿĂæŰźéİćïij
Pythonåŕźè£ŹäÿłéŮőécŸçŽĎèğčåĘşæŰźæąĹæŸŕäžŐæŰĞäžűåŘ■äÿ■èŐůåŔŰæIJłèğčçăĄçŽĎå■ŮèŁĆåĂijæ
`\xhh` åźűåřĘåőČæŸăåřĎæĹŘUnicodeå■Ůçňę `\udchh` èąĺçď žçŽĎæĽĂèřŞçŽĎâĂİäžčçŘĘçijŰçăĄâĂİãĂĆ
äÿŃéİćäÿĂäÿłä¿Ńå■ŘæijŤçď žäžĘå¡ŞäÿĂäÿłäÿ■åŘĹæăijçŽőå¡ŢåĹŮèąĺäÿ■åŘńæIJĽäÿĂäÿłæŰĞäžűåŘ■äÿžb
1èĂŇäÿ■æŸŕUTF-8çijŰçăĄ)æŮűçŽĎæăůå■Řïijž

```
>>> import os
>>> files = os.listdir('.')
>>> files
['spam.py', 'b\udce4d.txt', 'foo.txt']
>>>
```

åęĆæđIJä¡ăæIJĽäžčçăĄéIJĂèęĄæŞ■ä¡IJæŰĞäžűåŘ■æĹŰèĂĚåřĘæŰĞäžűåŘ■äijăéĂŠçżŹ
`open()` è£ŹæăůçŽĎåĞ¡æŢřïijŇäÿĂåĹĞéČ¡èČ¡æ■čåÿÿåůëä¡IJãĂĆ
åŘłæIJĽå¡Şä¡ăæČşèęĄè¿ŞåĞžæŰĞäžűåŘ■æŮűæĽ■äijŽçćřåĹřäžŹéžžçČę(æŕŤåęĆæĽŞå■řè¿ŞåĞžåĹřåśŔåź
çĽźåĹńçŽĎïijŇå¡Şä¡ăæČşæĽŞå■řäÿŁéİćçŽĎæŰĞäžűåŘ■åĹŮèąĺæŮűïijŇä¡ăçŽĎçÍŃåžŔåřśäijŽåť ľ æžČïijž

```
>>> for name in files:
...     print(name)
...
spam.py
Traceback (most recent call last):
    File "<stdin>", line 2, in <module>
UnicodeEncodeError: 'utf-8' codec can't encode character '\udce4' in
position 1: surrogates not allowed
>>>
```

çÍŃåžŔåť ľ æžČçŽĎåŐ§åŽăářśæŸŕå■Ůçňę `\udce4` æŸŕäÿĂäÿłéİđæşŢçŽĎUni-
codeå■Ůçňęã ĂĆ åőČåĚűåőđæŸŕäÿĂäÿłèćńçğřäÿžäžčçŘĘå■Ůçňęåŕźç ŽĎåŔŇå■Ůçňęçž ĎåŘĹçŽĎåŘŐå■ŁéČ
çŤśäžŐçijžåřŚäžĘåĽ■å■ŁéČĺåĹĘïijŇåŽăæ■ď åőČæŸŕäÿłéİđæşŢçŽĎUnicodeãĂĆ
æĽĂäžěïijŇåŤŕäÿĂèČ¡æĹŘåŁ§è¿ŞåĞžçŽĎæŰźæşŢåřśæŸŕå¡ŞéĄĞåĹřäÿ■åŘĹæşŢæŰĞäžűåŘ■æŮűéĞĞåŔU
æŕŤåęĆåŔŕäžěåřĘäÿŁè£ŕäžčçăĄä£őæŤźåęĆäÿŃïijž

```
>>> for name in files:
... try:
...     print(name)
... except UnicodeEncodeError:
...     print(bad_filename(name))
...
```

```
spam.py
b\udce4d.txt
foo.txt
>>>
```

åIJÍ bad_filename() åĞ¡æŢřäÿ■æĂŐæăůåď'Ďç¡őåŔŰåĘşäžŐä¡äèĞłåůśãĂĆ åŔęåď'ŰäÿĂäÿłéĂL'æŃľ'åřśæŸŕéĂŽè£Ğæ§ŘçğæŰźåijŔéĞ■æŰřçijŰçăĄïijŇçď'žä¿ŃåęĆäÿŃïijŽ

```
def bad_filename(filename):
    temp = filename.encode(sys.getfilesystemencoding(), errors=
↪'surrogateescape')
    return temp.decode('latin-1')
```

èŕŚèĂĚæşÍ:

```
surrogateescape:
è£ŹçğæŸŕPythonåIJÍçżÍåd'ğéČÍåĹĘéÍćåŘŚOSçŽĎAPIäÿ■æL'Ăä¡£çŤÍçŽĎéŤŹèŕŕåd'ĎçŘĘåŹÍïi
åőČèČ¡äżěäÿĂçğ■äijŸéŻĚçŽĎæŰźåijŔåd'ĎçŘĘçŤśæŞ■ä¡IJçşżçż§æŔŘä¿ŻçŽĎæŢřæ■őçŽĎçijŰçăĄé
åIJÍèğčçăĄåĞžéŤŹæŮűäijŽåřĘåĞžéŤŹå■ÙèŁĆå■ŸåĆÍåĹřäÿĂäÿłå¿ĹåřŚèćńä¡£çŤÍåĹřçŽĎUnicode
åIJÍçijŰçăĄæŮűåřĘéĆčäžżéŽŘèŮŔåĂijåŔĹè£ŸåŐ§åŻđåŐ§åĚĹèğčçăĄåd'śèt'ěçŽĎå■ÙèŁĆåžŘåĹŮ
åőČäÿ■äżĚåŕźäžŐOS␣
↪APIéİđåÿÿæIJL'çŤÍïijÑäź§èČ¡å¿ĹåőźæŸŞçŽĎåd'ĎçŘĘåĚűäżŰæČĚåĘţäÿŃçŽĎçijŰçăĄéŤŹèŕŕã
```

ä¡£çŤÍè£ŹäÿłçL'ĹæIJňäžğçŤ§çŽĎè¿ŞåĞžåęĆäÿŃïijŽ

```
>>> for name in files:
...     try:
...         print(name)
...     except UnicodeEncodeError:
...         print(bad_filename(name))
...
spam.py
bÃd'd.txt
foo.txt
>>>
```

è£ŹäÿĂåřŔèŁĆäÿżécŸåŔŕèČ¡äijŽèćńåď'ğéČÍåĹĘèŕżèĂĚæL'Ăå£¡çŢěãĂĆä¡ĘæŸŕåęĆæđIJä¡ăåIJÍçijŰåĘ
åřśå£Ěéążå¿ŮèĂČèŹŚåĹřè£ŹäÿłãĂĆåŘęåĹŹä¡ăåŔŕèČ¡äijŽåIJÍæ§Řäÿłåŝ́ÍæIJńèćńåŔńåĹřåŁđåĚñåőd'åŐżèřČ

7.16 5.16 åćđåŁăæĹŰæŤźåŔŸåůšæL'ŞåijĂæŰĞäžűçŽĎçijŰçăĄ

éŮőécŸ

ä¡ăæČşåIJÍäÿ■åĚşéŮ■äÿĂäÿłåůšæL'ŞåijĂçŽĎæŰĞäžűåL'■æŔŘäÿŃåćđåŁăæĹŰæŤźåŔŸåőČçŽĎUnico

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăæČşçżŹäÿĂäÿłäžëäžŇè£ŻåĹűæĺąåijŔæL'ŞåijĂçŽĎæŰĞäžűæůżåŁăUnicodeçijŰçăĄ/èğčçăĄ
åŔŕäžëä¡£çŤĺ `io.TextIOWrapper()` åŕźèśąåŇĚèčĚåőČãĂĆæŕŤåęĆïijŽ

```
import urllib.request
import io

u = urllib.request.urlopen('http://www.python.org')
f = io.TextIOWrapper(u, encoding='utf-8')
text = f.read()
```

åęĆædIJä¡ăæČşä£őæŤźäÿĂäÿłåůšçżŔæL'ŞåijĂçŽĎæŰĞæIJňæĺąåijŔçŽĎæŰĞäžűçŽĎçijŰçăĄæŰźåijŔ
`detach()` æŰźæşŢçğżéŹď'æŐL'åůšå■ŸåIJĺçŽĎæŰĞæIJňçijŰçăĄåśĆïijŇ
åźűä¡£çŤĺæŰřçŽĎçijŰçăĄæŰźåijŔäžčæŽ£ãĂĆäÿŇéİćæŸŕäÿĂäÿłåIJĺ `sys.stdout`
äÿŁä£őæŤźçijŰçăĄæŰźåijŔçŽĎä¿Ńå■ŘïijŽ

```
>>> import sys
>>> sys.stdout.encoding
'UTF-8'
>>> sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding=
↪'latin-1')
>>> sys.stdout.encoding
'latin-1'
>>>
```

è£Źæăůå˛ŽåŔŕèČ¡äijŽäÿ■æŰ■ä¡ăçŽĎçżĹçńŕïijŇè£ŹéĞŇäžĚäžĚæŸŕäÿžäžĘæijŢçď'žèĂŇåůšãĂĆ

èőĺèőž

I/Oçşżçž§çŤśäÿĂçşżåĹŰçŽĎåśĆæňąædĎåżžèĂŇæĹŘãĂĆä¡ăåŔŕäžëèŕŢçİÄè£ŘèąŇäÿŃéİćè£ŹäÿłæŞ■ä

```
>>> f = open('sample.txt','w')
>>> f
<_io.TextIOWrapper name='sample.txt' mode='w' encoding='UTF-8'>
>>> f.buffer
<_io.BufferedWriter name='sample.txt'>
>>> f.buffer.raw
<_io.FileIO name='sample.txt' mode='wb'>
>>>
```

åIJĺè£Źäÿłä¿Ńå■Řäÿ■ïijŇ`io.TextIOWrapper` æŸŕäÿĂäÿłçijŰçăĄåŠŇèğčçăĄU-
nicodeçŽĎæŰĞæIJňåď'ĎçŘĘåśĆïijŇ `io.BufferedWriter`
æŸŕäÿĂäÿłåď'ĎçŘĘäžŇè£ŻåĹűæŢŕæ■őçŽĎåÿęçijŞåĘšçŽĎI/OåśĆïijŇ `io.FileIO`
æŸŕäÿĂäÿłèąĺçď'žæŞ■ä¡IJçşżçž§åžŢåśĆæŰĞäžűæŔŘè£rçňęçŽĎåŐ§ågŃæŰĞäžűãĂĆ
åćđåŁăæĹŰæŤźåŔŸæŰĞæIJňçijŰçăĄäijŽæűL'åŔŁåćđåŁăæĹŰæŤźåŔŸæIJĂäÿŁéİćçŽĎ
`io.TextIOWrapper` åśĆãĂĆ

äÿĂèĹňæİèèőšïijŇåČŔäÿŁéİćä¿Ńå■Řè£ŹæăůéĂŽè£Ğèŏ£éŮőåśđæĂğåĂijæİěçŻť'æŐěæŞ■ä¡IJäÿ■åŘŇç
ä¿ŃåęĆïijŇåęĆædIJä¡ăèŕŢçİÄä¡£çŤĺäÿŃéİćè£ŹæăůçŽĎæŁĂæIJŕæŤźåŔŸçijŰçăĄçIJŃçIJŃäijŽåŔŠçŢ§äžĂä

```
>>> f
<_io.TextIOWrapper name='sample.txt' mode='w' encoding='UTF-8'>
>>> f = io.TextIOWrapper(f.buffer, encoding='latin-1')
>>> f
<_io.TextIOWrapper name='sample.txt' encoding='latin-1'>
>>> f.write('Hello')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ValueError: I/O operation on closed file.
>>>
```

çżŞædIJåĞžéŤŹäžĘïijŇåŻääÿžfçŽĎåŐ§ågŃåĂijåůšçżŔèćńçăť åİŔäžĘåźűåĚşéŮ■äžĘåžŢåśĆçŽĎæŰĞäz

detach() æŰźæşŢäijŽæŰ■åijĂæŰĞäżűçŽĎæIJĂéąűåśĆåźűè£ŤåŻđçňňäžŇåśĆïijŇäźŃåŘŐæIJĂéąűå

```
>>> f = open('sample.txt', 'w')
>>> f
<_io.TextIOWrapper name='sample.txt' mode='w' encoding='UTF-8'>
>>> b = f.detach()
>>> b
<_io.BufferedWriter name='sample.txt'>
>>> f.write('hello')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ValueError: underlying buffer has been detached
>>>
```

äÿĂæŮęæŰ■åijĂæIJĂéąűåśĆåŘŐïijŇä¡ăářśåŔŕäžěçżŹè£ŤåŻđçżŞædIJæůżåŁääÿĂäÿłæŰřçŽĎæIJĂéąűå

```
>>> f = io.TextIOWrapper(b, encoding='latin-1')
>>> f
<_io.TextIOWrapper name='sample.txt' encoding='latin-1'>
>>>
```

ář¡çőąåůšçżŔåŘŚä¡ăæijŤçď žäžĘæŤźåŔŸçijŰçăĄçŽĎæŰźæşŢïijŇ
ä¡ĘæŸŕä¡ăè£ŸåŔŕäžěåĹ'çŤĺè£Źçğ■æŁĂæIJŕæİěæŤźåŔŸæŰĞäžűèąŇåď'ĎçŘĘãĂąéŤŹèŕŕæIJžåĹűäžěåŘŁæ

```
>>> sys.stdout = io.TextIOWrapper(sys.stdout.detach(), encoding=
↪'ascii',
...                                errors='xmlcharrefreplace')
>>> print('Jalape\u00f1o')
Jalape&#241;o
>>>
```

æşÍæĎŔäÿŇæIJĂåŘŐè¿ŞåĞžäÿ■çŽĎéİđASCIIå■Ůçňę Ãś æŸŕåęĆä¡Ţèćń ñ åŘŰäžčçŽĎãĂĆ

7.17 5.17 åřĘå■ÙèŁĆåĘŹåĚěæŰĞæIJňæŰĞäžű

éŮőécŸ

ä¡ăæČşåIJĺæŰĞæIJňæĺąåijŕæL'Şåij ĂçŽĎæŰĞäžűäÿ■åĘŹåĚěåŐ§ågŃçŽĎå■ÙèŁĆæŢřæ■őãĂĆ

èğčåĘşæŰźæąĹ

åřĘå■ÙèŁĆæŢřæ■őçŻť'æŐěåĘŹåĚěæŰĞäžűçŽĎçijŞåĘšåŇžå■şåŕřïijŇä¿ŃåęĆïijŽ

```
>>> import sys
>>> sys.stdout.write(b'Hello\n')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: must be str, not bytes
>>> sys.stdout.buffer.write(b'Hello\n')
Hello
5
>>>
```

çśżäijijçŽĎïijŇèČ¡åď'§éĂŽè£Ğèŕżå ŔŰæŰĞæIJňæŰĞäžűçŽĎ buffer åšđæĂğæÌěèŕżå ŔŰäžŇè£ŻåĹűæŢřæ■őãĂĆ

èőÍèőž

I/Oçşżçż§äžěåśĆçžğçżŞæđĎçŽĎå¡ćåijŔæđĎåżžèĂŇæĹŘãĂĆ
æŰĞæIJňæŰĞäžűæŸŕéĂŽè£ĞåIJĺäÿĂäÿłæŃěæIJL'çijŞåĘšçŽĎäžŇè£ŻåĹűæĺąåijŔæŰĞäžűäÿŁåćđåŁääÿĂäÿ
buffer åšđæĂğæŇĞåŘŠåŕźåžŤçŽĎåžŢåśĆæŰĞäžűãĂĆåęĆæđIJä¡ăçŻť'æŐèèő£éŮőåőĆçŽĎèŕÌåřšäijŽçżŢè

æIJňåřŔèŁĆä¿Ńå■ŘåśŢçd'žçŽĎ sys.stdout åŔŕèČ¡çIJŃètůæÌěæIJL'çĆźçL'źæőŁãĂĆ
éżŸèőd'æČĚåĘţäÿŃïijŇsys.stdout æĂżæŸŕäžěæŰĞæIJňæĺąåijŔæL'Şåij ĂçŽĎãĂĆ
ä¡ĘæŸŕåęĆæđIJä¡ăåIJĺåĘŹäÿĂäÿłéIJĂèęĄæL'Şå■řäžŇè£ŻåĹűæŢřæ■őåĹřæăĞåĞĘè¿ŞåĞžçŽĎèĎŽæIJňçŽĎ

7.18 5.18 åřĘæŰĞäžűæŔŔè£řçňęåŇĚèčĚæĹŘæŰĞäžűåŕźèśą

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłåŕźåžŤäžŐæŞ■ä¡IJçşżçż§äÿŁäÿĂäÿłåůšæL'Şåij ĂçŽĎI/OéĂŽéĄŞ(æŕŤåęĆæŰĞäžűãĂĄç
ä¡ăæČşåřĘåőČåŇĚèčĚæĹŘäÿĂäÿłæŽť'énŸåśĆçŽĎPythonæŰĞäžűåŕźèśąãĂĆ

èğčåĘşæŰźæąĹ

äÿĂäÿłæŰĞäžűæŔŔè£řçňęåŠŇäÿĂäÿłæL'Şåij ĂçŽĎæŹőéĂŽæŰĞäžűæŸŕäÿ■äÿĂæăůçŽĎãĂĆ
æŰĞäžűæŔŔè£řçňęäžĚäžĚæŸŕäÿĂäÿłçŤśæŞ■ä¡IJçşżçż§æŇĞåőŽçŽĎæŢť'æŢřijŇçŤĺæÌěæŇĞäžčæ§Řäÿłçş
åęĆæđIJä¡ăçćřåůğæIJL'è£ŻäźĹäÿĂäÿłæŰĞäžűæŔŔè£řçňęïijŇä¡ăåŔŕäžěéĂŽè£Ğä¡£çŤĺ

open() åĞ¡æŢřæİěářĘåĚűåŇĚèčĚäÿžäÿĂäÿłPythonçŽĎæŰĞäžűářźèśąąĂĆ
ä¡ăäžĚäžĚåŔłéIJĂèęĄä¡£çŤĺè£ŹäÿłæŢťæŢřåĂijçŽĎæŰĞäžűæŔŔè£řçňęä¡IJäÿžçňňäÿĂäÿłåŔĆæŢřæİěäžčæ

```
# Open a low-level file descriptor
import os
fd = os.open('somefile.txt', os.O_WRONLY | os.O_CREAT)

# Turn into a proper file
f = open(fd, 'wt')
f.write('hello world\n')
f.close()
```

å¡ŞéńŸåśĆçŽĎæŰĞäžűářźèśąèćńåĚşéŮ■æĹŰèĂĚçăť åİŔçŽĎæŮűåĂŹïijŇåžŢåśĆçŽĎæŰĞäžűæŔŔè£ì
åęĆæđIJè£Źäÿłåźűäÿ■æŸŕä¡ăæČşèęĄçŽĎçżŞæđIJïijŇä¡ăåŔŕäžěçżŹ open()
åĞ¡æŢřäijăéĂŠäÿĂäÿłåŔŕéĂĽ'çŽĎ colsefd=False ãĂĆæŕŤåęĆïijŽ

```
# Create a file object, but don't close underlying fd when done
f = open(fd, 'wt', closefd=False)
...
```

èőĺèőž

åIJÍUnixçşżçż§äÿ■ïijŇè£Źçğ■åŇĚèčĚæŰĞäžűæŔŔè£řçňęçŽĎæŁĂæIJŕåŔŕäžěå¿ĹæŰźä¿£çŽĎåřĘäÿĂä
åęĆçőąéĄŞãĂĄåěŮæŐěå■Ůç■Ľ˜ãĂĆäÿ¿ä¿ŃæİěèőšïijŇäÿŃéİćæŸŕäÿĂäÿłæŞ■ä¡IJçőąéĄŞçŽĎä¿Ńå■ŘïijŽ

```
from socket import socket, AF_INET, SOCK_STREAM

def echo_client(client_sock, addr):
    print('Got connection from', addr)

    # Make text-mode file wrappers for socket reading/writing
    client_in = open(client_sock.fileno(), 'rt', encoding='latin-1',
                     closefd=False)

    client_out = open(client_sock.fileno(), 'wt', encoding='latin-1
↪',
                      closefd=False)

    # Echo lines back to the client using file I/O
    for line in client_in:
        client_out.write(line)
        client_out.flush()

    client_sock.close()

def echo_server(address):
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(address)
    sock.listen(1)
    while True:
```

```
        client, addr = sock.accept()
        echo_client(client, addr)
```

éIJĂèęĄéĞ■çĆźåijžèřČçŽĎäÿĂçĆźæŸŕïijŇäÿŁéİćçŽĎä¿Ńå■ŘäżĚäżĚæŸŕäÿžäžĘæijŤçď žåĘĚç¡őçŽĎ open() åĞ¡æŢřçŽĎäÿĂäÿłçĽźæĂğïijŇåźűäÿŤäź§åŔłéĂĆçŤĺäžŐå§žäžŐUnixçŽĎçşżçż§ãĂĆ åęĆæđIJä¡ăæČşåřĘäÿĂäÿłçśżæŰĞäżűæŐěåŔčä¡IJçŤĺåIJĺäÿĂäÿłåěŮæŐěå■ŮåźűåÿŇæIJŻä¡ăçŽĎäżčçăĄåŔŕä makefile() æŰźæşŢãĂĆ ä¡ĘæŸŕåęĆæđIJäÿ■èĂČèŹŚåŔŕçğżæď■æĂğçŽĎèŕİïijŇéĆčäÿŁéİćçŽĎèğčåĘşæ makefile() æĂğèČ¡æŻť åě¡äÿĂçĆźãĂĆ

ä¡ăäź§åŔŕäżěä¡£çŤĺè£Źçğ■æŁĂæIJŕæİěæđĎéĂăäÿĂäÿłåĹńåŘ■ïijŇåĚĄèőÿäżěäÿ■åŘŇäžŐçňňäÿĂæňąæ ä¿ŃåęĆïijŇäÿŃéİćæijŤçď žåęĆä¡ŢåĹŻåżžäÿĂäÿłæŰĞäżűåŕźèśąïijŇåőČåĚĄèőÿä¡ăè¿ŞåĞžäžŇè£ŻåĹűæŢřæ■

```
import sys
# Create a binary-mode file for stdout
bstdout = open(sys.stdout.fileno(), 'wb', closefd=False)
bstdout.write(b'Hello World\n')
bstdout.flush()
```

åř¡çőąåŔŕäżěåřĘäÿĂäÿłåůšå■ŸåIJĺçŽĎæŰĞäżűæŔŔè£řçňęåŇĚèčĚæĹŘäÿĂäÿłæ■čåÿÿçŽĎæŰĞäżűåŕźè ä¡ĘæŸŕèęĄæşĺæĎŔçŽĎæŸŕåźűäÿ■æŸŕæĽĂæIJĽçŽĎæŰĞäżűæĺąåijŔéČ¡èćńæŤŕæŇĄïijŇåźűäÿŤæ§ŘäžŻçś (çĽźåĹńæŸŕæűĽåŔŁåĹřéŤŹèŕŕåď ĎçŘĘãĂĄæŰĞäżűçżŞåř¿æİąäżűç■Ľç■ĽçŽĎæŮűåĂŹ)ãĂĆ åIJĺäÿ■åŘŇçŽĎæŞ■ä¡IJçşżçż§äÿŁè£Źçğ■èąŇäÿžäź§æŸŕäÿ■äÿĂæăůïijŇçĽźåĹńçŽĎïijŇäÿŁéİćçŽĎä¿Ńå■Ře æĹŚèŕť äžĘè£ŹäźĹåď ŽïijŇæĎŔæĂİåřśæŸŕèőľ ä¡ăåĚĚåĹĘæţŃèŕŢèĞłåůśçŽĎåőđçŐřäżčçăĄïijŇçąőä£İåőČèĞ

7.19 5.19 åĹŻåżžäÿťæŮűæŰĞäżűåŠŇæŰĞäżűåď ź

éŮőéćŸ

ä¡ăéIJĂèęĄåIJĺçĺŃåžŔæĽğèąŇæŮűåĹŻåżžäÿĂäÿłäÿť æŮűæŰĞäżűæĹŰçŻőå¡ŢïijŇåźűåÿŇæIJŻä¡£çŤĺå

èğčåĘşæŰźæąĹ

tempfile æĺąåİŮäÿ■æIJĽå¿Ĺåď ŽçŽĎåĞ¡æŢřåŔŕäżěåőŇæĹŘè£ŹäżżåŁąãĂĆ äÿžäžĘåĹŻåżžäÿĂäÿłåŇ£åŘ■çŽĎäÿť æŮűæŰĞäżűïijŇåŔŕäżěä¡£çŤĺ tempfile. TemporaryFile ïijŽ

```
from tempfile import TemporaryFile

with TemporaryFile('w+t') as f:
    # Read/write to the file
    f.write('Hello World\n')
    f.write('Testing\n')

    # Seek back to beginning and read the data
    f.seek(0)
    data = f.read()
```

```
# Temporary file is destroyed
```

æĹŰèĂĚïijŇåęĆæđIJä¡ăåŰIJæňćïijŇä¡ăè£ŸåŔŕäžěåČŔè£Źæăůä¡£çŤĺäÿťæŮűæŰĞäžűïijŽ

```
f = TemporaryFile('w+t')
# Use the temporary file
...
f.close()
# File is destroyed
```

TemporaryFile() çŽĎçňňäÿĂäÿłåŔĆæŢřæŸŕæŰĞäžűæĺąåijŔïijŇéĂŽåÿÿæİěèőšæŰĞæIJňæĺąåijŔä w+t ïijŇäžŇè£ŻåĹűæĺąåijŔä¡£çŤĺ w+b ãĂĆ è£ŹäÿłæĺąåijŔåŘŇæŮűæŤŕæŇĄèŕżåŠŇåĘŹæŞ■ä¡IJïijŇåIJĺè£Ź TemporaryFile() åŔęåď Úè£ŸæŤŕæŇĄèů§åĘĚç¡őçŽĎ open() åĞ¡æŢřäÿĂæăůçŽĎåŔĆæŢřãĂĆæŕŤåęĆïijŽ

```
with TemporaryFile('w+t', encoding='utf-8', errors='ignore') as f:
    ...
```

åIJĺåď ğåď ŽæŢřUnixçşżçż§äÿŁïijŇéĂŽè£Ğ TemporaryFile() åĹŻåžžçŽĎæŰĞäžűéČ¡æŸŕåŇ£åŘ■çŽĎïijŇçŤŽèĞşè£đçŻőå¡ŢéČ¡æšąæIJL'ãĂĆ åęĆæđIJä¡ăæČşæL'Şçăť è£ŹäÿłéŹŘåĹűïijŇåŔŕäžěä¡£çŤĺ NamedTemporaryFile() æİěäžčæŽ£ãĂĆæŕŤåęĆïijŽ

```
from tempfile import NamedTemporaryFile

with NamedTemporaryFile('w+t') as f:
    print('filename is:', f.name)
    ...

# File automatically destroyed
```

è£ŹéĞŇïijŇèćńæL'ŞåijĂæŰĞäžűçŽĎ f.name åšđæĂğåŇĚåŔńäžĘèŕěäÿťæŮűæŰĞäžűçŽĎæŰĞäžűåŘ å¡Şä¡ăéIJĂèęĄåřĘæŰĞäžűåŘ■äijăéĂŠçżŹåĚűäžŰäžčçĂąæİěæL'ŞåijÀè£ŹäÿłæŰĞäžűçŽĎæŰűåĂŹïijŇè£Źä åŠŇ TemporaryFile() äÿĂæăůïijŇçżŞæđIJæŰĞäžűåĚşéŮ■æŮűäijŽèćńèĞłåŁĺåĹăéŹď'æŐL'ãĂĆ åęĆæđIJä¡ääÿ■æČşè£ŹäźĹåĄŽïijŇåŔŕäžěäijăéĂŠäÿĂäÿłåĚşéŤőå■ŮåŔĆæŢř delete=False å■şåŔŕãĂĆæŕŤåęĆïijŽ

```
with NamedTemporaryFile('w+t', delete=False) as f:
    print('filename is:', f.name)
    ...
```

äÿžäžĘåĹŻåžžäÿĂäÿłäÿť æŮűçŽőå¡ŢïijŇåŔŕäžěä¡£çŤĺ tempfile.TemporaryDirectory() ãĂĆæŕŤåęĆïijŽ

```
from tempfile import TemporaryDirectory

with TemporaryDirectory() as dirname:
    print('dirname is:', dirname)
    # Use the directory
```

```
    ...
# Directory and all contents destroyed
```

èőÍèőž

TemporaryFile() ãĂĄNamedTemporaryFile() åŠŇ TemporaryDirectory() åĞ¡æŢř åžŤèŕěæŸŕåď ĎçŘĘäÿť æŮűæŰĞäžűçŻőå¡ŢçŽĎæIJĂçőĂå■ŢçŽĎæŰź åIJÍäÿĂäÿłæŻťä¡ŐçŽĎçžğåĹńïijŇä¡ăåŔŕäżěä¡£çŤÍ mkstemp() åŠŇ mkdtemp() æÍěåĹŻåžžäÿť æŮűæŰĞäžűåŠŇçŻőå¡ŢãĂĆæŕŤåęĆïijŽ

```
>>> import tempfile
>>> tempfile.mkstemp()
(3, '/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-/tmp7fefhv')
>>> tempfile.mkdtemp()
'/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-/tmp5wvcv6'
>>>
```

ä¡ĘæŸŕïijŇè£ŹäžŻåĞ¡æŢřåźűäÿ■äijŽåĄŽè£ŻäÿĂæ■ěçŽĎçőąçŘĘäžĘãĂĆ ä¿ŃåęĆïijŇåĞ¡æŢř mkstemp() äżĚäżĚářśè£ŤåŻđäÿĂäÿłåŐ§ågŃçŽĎOSæŰĞäžűæŔŔè£řçňęïijŇä¡ăéIJĂèę åŘŇæåůä¡ãè£ŸéIJĂèęĄèĞłåůśæÿĚçŘĘè£ŹäžŻæŰĞäžűãĂĆ

éĂŽåÿÿæÍěèőšïijŇäÿť æŮűæŰĞäžűåIJÍçşżçż§éźŸèőď çŽĎä¡■ç¡őèćńåĹŻåžžïijŇæŕŤåęĆ /var/tmp æĹŰçśżäijijçŽĎåIJřæŰźãĂĆ äÿžäžĘèŐůåŔŰçIJ§åőđçŽĎä¡■ç¡őïijŇåŔŕäżěä¡£çŤÍ tempfile.gettempdir() åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
>>> tempfile.gettempdir()
'/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-'
>>>
```

æĽĂæIJL'åŠŇäÿť æŮűæŰĞäžűçŻÿåĚşçŽĎåĞ¡æŢřéČ¡åĚĄèőÿä¡ăéĂŽè£Ğä¡£çŤĺåĚşéŤőå■ŮåŔĆæŢř prefix ãĂĄsuffix åŠŇ dir æÍěèĞłåőŽäźĽçŽőå¡ŢäžěåŔŁåŚ¡åŘ■èğĎåĹŹãĂĆæŕŤåęĆïijŽ

```
>>> f = NamedTemporaryFile(prefix='mytemp', suffix='.txt', dir='/tmp
↪')
>>> f.name
'/tmp/mytemp8ee899.txt'
>>>
```

æIJĂåŘŐè£ŸæIJL'äÿĂçĆźïijŇår¡åŔŕèČ¡äžěæIJĂåőĽåĚÍçŽĎæŰźåijŔä¡£çŤÍ tempfile æÍąåİŮæİěåĹŻåžžäÿť æŮűæŰĞäžűãĂĆ åŇĚæŃňäžĚçżŹå¡ŞåĽ■çŤÍæĹůæŐĹæİČèő£éŮőäžěåŔŁåIJÍæŰĞäžűę èęĄæşÍæĎŔçŽĎæŸŕäÿ■åŘŇçŽĎåźşåŘřåŔŕèČ¡äijŽäÿ■äÿĂæăůãĂĆåŽăæ■ď ä¡ăæIJĂåě¡éŸĚèŕż åőŸæŰźæŰĞæąč æÍěäžĘèğčæŽť åď ŽçŽĎçżĘèŁĆãĂĆ

7.20 5.20 äÿŐäÿšèąŇçńŕåŔčçŽĎæŢřæ■őéĂŽä£ą

éŮőécŸ

ä¡ăæČşéĂŽè£ĞäÿšèąŇçńŕåŔčèŕżåĘŹæŢřæ■őïijŇåĚÿåđŃåIJžæŹŕåŕśæŸŕåŠŇäÿĂäžŻçąňäžűëő¿åd'ĞæL'Ş

èğčåĘşæŰźæąĹ

åř¡çőąä¡ăåŔŕäžëéĂŽè£Ğä¡£çŤÍPythonåĘĚç¡őçŽĎI/OæÍąåİŮæİèåőŇæĹŘè£ŹäÿłäżżåŁąïijŇä¡ĘåŕźäžŐäÿš
pySerialåŇĚ ãĂĆ è£ŹäÿłåŇĚçŽĎä¡£çŤÍéİđåÿÿçőĂå■ŢïijŇåĚĹáőL'èčĚpySerialïijŇä¡£çŤÍçśżäijijäÿŃéİćè£Źæ

```
import serial
ser = serial.Serial('/dev/tty.usbmodem641', # Device name varies
                    baudrate=9600,
                    bytesize=8,
                    parity='N',
                    stopbits=1)
```

èő¿åd'ĞåŘ■åŕźäžŐäÿ■åŘŇçŽĎèő¿åd'ĞåŠŇæŞ■ä¡IJçşżçż§æŸŕäÿ■äÿĂæăůçŽĎãĂĆ
æŕŤåęĆïijŇåIJÍWindowsçşżçż§äÿŁïijŇä¡ăåŔŕäžëä¡£çŤÍ0, 1ç■L'èąÍçd'žçŽĎäÿĂäÿłèő¿åd'ĞæİěæL'ŞåijĂéĂŽä
äÿĂæŮęçńŕåŔčæL'ŞåijĂïijŇéĆčåŕśåŔŕäžëä¡£çŤÍ `read()`ïijŇ`readline()` åŠŇ `write()`
åĞ¡æŢřèŕżåĘŹæŢřæ■őäžĘãĂĆä¿ŃåęĆïijŽ

```
ser.write(b'G1 X50 Y50\r\n')
resp = ser.readline()
```

åd'ğåd'ŽæŢřæČĚåĘţäÿŃïijŇçőĂå■ŢçŽĎäÿšåŔčéĂŽä£ąäžŐæ■d'åŔŸå¿Ůå■ĄåĹĘçőĂå■ŢãĂĆ

èőÍèőž

åř¡çőąèąÍéİćäÿŁçIJŃèţůæİěå¿ĹçőĂå■ŢïijŇåĚűåőđäÿšåŔčéĂŽä£ąæIJL'æŮűåĂŹäź§æŸŕæŇžéžżçČęçŽĎ
æŐÍè■Řä¡ää¡£çŤÍçňňäÿL'æŰźåŇĚåęĆ `pySerial` çŽĎäÿĂäÿłåŐ§åŽăæŸŕåőČæŔŘä¿ŻäžĘåŕźéńŸçžğçL'źæĂ
(æŕŤåęĆèűĚæŮűïijŇæŐğåĹűæţĄïijŇçijŞåĘšåŇžåĹůæŰřïijŇæŔąæL'Ńå■Řèőőç■L'ç■L')ãĂĆäÿ¿äÿłä¿Ńå■Řïij
`RTS-CTS` æŔąæL'Ńå■Řèőőïij Ň ä¡ăåŔłéIJĂèęĄçżŹ `Serial()` äijăéĂŠäÿĂäÿł
`rtscts=True` çŽĎåŔĆæŢřå■şåŔŕãĂĆ åĚűåőŸæŰźæŰĞæąčéİđåÿÿåőŇåŰĎïijŇåŽăæ■d'æĹŠåIJÍè£ŹéĞŇa

æŮűåĹżèőřä¡ŔæL'ĂæIJL'æűL'åŘŁåĹřäÿšåŔčçŽĎI/OéČ¡æŸŕäžŇè£ŻåĹűæÍąåijŔçŽĎãĂĆåŽăæ■d'ïijŇçą
(æĹŰæIJL'æŮűåĂŹæL'ğèąŇæŰĞæIJńçŽĎçijŰçăĄ/èğčçăĄæŞ■ä¡IJ)ãĂĆ
åŔęåd'Űå¡Şä¡ăéIJĂèęĄåĹŻåžżäžŇè£ŻåĹűçijŰçăĄçŽĎæŇĞäžd'æĹŰæŢřæ■őåŇĚçŽĎæŮűåĂŹïijŇ`struct`
æÍąåİŮäź§æŸŕéİđåÿÿæIJL'çŤÍçŽĎãĂĆ

7.21 5.21 åžŔåĹŮåŇŰPythonåŕźèšą

éŮőécŸ

ä¡ăéIJĂèęĄåřĘäÿĂäÿłPythonåŕźèšąåžŔåĹŮåŇŰäÿžäÿĂäÿłå■ŮèŁĆæţĄïijŇäžëä¿£åřĘåőČä£Íå■ŸåĹřäÿĂ

èğčåĘşæŰźæąĹ

åŕźäžŐåžŔåĹŮåŇŰæIJĂæŹőéĄ■çŽĎåĄŽæşŢåřśæŸŕä¡£çŤÍ pickle æĺąåİŮãĂĆäÿžäžĘåřĘäÿĂäÿłåŕźèśąä£Ìå■ŸåĹřäÿĂäÿłæŰĞäžűäÿ■ïijŇåŔŕäžěè£ŹæăůåĄŽïijŽ

```
import pickle

data = ... # Some Python object
f = open('somefile', 'wb')
pickle.dump(data, f)
```

äÿžäžĘåřĘäÿĂäÿłåŕźèśąè¡ňåĆĺäÿžäÿĂäÿłå■ŮçňęäÿšïijŇåŔŕäžěä¡£çŤÍ pickle.dumps() ïijŽ

```
s = pickle.dumps(data)
```

äÿžäžĘäžŐå■ŮèŁĆæţĄäÿ■æĄćåd'■äÿĂäÿłåŕźèśąïijŇä¡£çŤÍ picle.load() æĹŰ pickle.loads() åĞ¡æŢřãĂĆæŕŤåęĆïijŽ

```
# Restore from a file
f = open('somefile', 'rb')
data = pickle.load(f)

# Restore from a string
data = pickle.loads(s)
```

èŐĺèŏž

åŕźäžŐåd'ğåd'ŽæŢřåžŤçŤÍçÍŇåžŔæİěèőšïijŇdump() åŠŇ load() åĞ¡æŢřçŽĎä¡£çŤÍåřśæŸŕä¡ăæIJL'æŢĹä¡£çŤÍ pickle æĺąåİŮæL'ĂéIJĂçŽĎåĚÍéČÍíažĘãĂĆ åőČåŔŕéĂĆçŤÍäžŐçżİåd'ğéČÍåĹĘPythonæŢřæ■őçśžåđŇåŠŇçŤÍæĹůèĞłåőŽäźL'çśżçŽĎåŕźèśąåőđä¿ŃãĂĆ åęĆæđIJä¡ăçćřåĹřæ§Řäÿłåžşåŕźäžěèől'ä¡ăåIJÍæŢřæ■őåžŞäÿ■ä£Ìå■Ÿ/æĄćåd'■PythonåŕźèśąæĹŰèĂĚæŸŕéĂŽ éČčäźĹå¿ĹæIJL'åŔŕèČ¡è£ŹäÿłåžŞçŽĎåžŢåśĆåŕśä¡£çŤÍäžĘ pickle æĺąåİŮãĂĆ

pickle æŸŕäÿĂçğ■PythonçL'źæIJL'çŽĎèĞłæŔŔè£řçŽĎæŢřæ■őçijŰçăĄãĂĆ éĂŽè£ĞèĞłæŔŔè£řïijŇèćńåžŔåĹŮåŇŰåŘŐçŽĎæŢřæ■őåŇĚåŘńæŕŔäÿłåŕźèśąåijĂåğŇåŠŇçżŞæİ§äžěåŔŁåč åŽăæ■d'ïijŇä¡ăæŮăéIJĂæŃĚå£Čåŕźèśąèőŕå¡ŢçŽĎåőŽäźL'ïijŇåőČæĂžæŸŕèČ¡åůëä¡IJãĂĆ äÿ¿äÿłä¿Ńå■ŘïijŇåęĆæđIJèęĄåd'ĎçŘĘåd'ŽäÿłåŕźèśąïijŇä¡ăåŔŕäžěè£ŹæăůåĄŽïijŽ

```
>>> import pickle
>>> f = open('somedata', 'wb')
>>> pickle.dump([1, 2, 3, 4], f)
>>> pickle.dump('hello', f)
>>> pickle.dump({'Apple', 'Pear', 'Banana'}, f)
>>> f.close()
>>> f = open('somedata', 'rb')
>>> pickle.load(f)
[1, 2, 3, 4]
>>> pickle.load(f)
```

```
'hello'
>>> pickle.load(f)
{'Apple', 'Pear', 'Banana'}
>>>
```

ä¡äè£ŸèČ¡åžŔåĹŮåŇŰåĞ¡æŢřïijŇçśžïijŇè£ŸæIJL’æŐěåŔčïijŇä¡ĘæŸŕçżŞæđIJæŢřæ■őäżĚäżĚåřĘåőČä

```
>>> import math
>>> import pickle.
>>> pickle.dumps(math.cos)
b'\x80\x03cmath\ncos\nq\x00.'
>>>
```

å¡ŞæŢřæ■őåŔ■åžŔåĹŮåŇŰåŻđæİěçŽĎæŮűåĂŹïijŇäijŽåĚĹåĄĞåőŽæL’ĂæIJL’çŽĎæžŘæŢřæ■őæŮűåĘ
æĺąåİŮãĂĄçśżåŠŇåĞ¡æŢřäijŽèĞłåŁĺæŇL’éIJĂåŕijåĚèè£ŻæİěãĂĆåŕźäžŐPythonæŢřæ■õèćńäÿ■åŘŇæIJžåŹĺã
æŢřæ■őçŽĎä£Ía■ŸåŔŕèČ¡äijŽæIJL’éŮőécŸïijŇåŻääÿžæL’ĂæIJL’çŽĎæIJžåŹĺéČ¡å£Ěéążèő£éŮőåŘŇäÿĂäÿł

æşÍ

```
å■ČäÿĞäÿ■èęĄåŕźäÿ■ä£ąäżżçŽĎæŢřæ■őä¡£çŤÍpickle.load()ãĂĆ
pickleåIJÍåŁãè¡¡æŮűæIJL’äÿĂäÿłåL’ŕä¡IJçŤÍåřśæŸŕåőČäijžèĞłåŁĺåŁãè¡¡çŻÿåžŤæĺąåİŮåź
ä¡ĘæŸŕæ§ŘäÿłåİŔäžžåęĆæđIJç§ěéĄ§pickleçŽĎåůěä¡IJåŐ§çŘĘïijŇ
äżŰåřśåŔŕäżěåĹżåžžäÿĂäÿłæĄűæĎŔçŽĎæŢřæ■őåŕijèĞt’PythonæL’ğèąŇéŽŘæĎŔæŇĞåőžçŽĎçşżçż
åŻăæ■d’ïijŇäÿĂåőŽèęĄä£İèŕĄpickleåŔłåIJÍçŻÿäžŠäźŇéŮt’åŔŕäżěèőd’èŕĄåŕźæŰźçŽĎèğčæđŘ
```

æIJL’äžŻçśżåđŃçŽĎåŕźèśąæŸŕäÿ■èČ¡èćńåžŔåĹŮåŇŰçŽĎãĂĆè£ŹäžŻéĂŽåÿÿæŸŕéĆčäžŻä¿İèţŰåđ’Ůé
æŕŤåęĆæL’ŞåijĂçŽĎæŰĞäżűïijŇç¡ŚçżIJè£đæŐěïijŇçž£çÍŃïijŇè£ŻçÍŃïijŇæăĹåÿğç■L’ç■L’ãĂĆ
çŤÍæĹůèĞłåőŽäźL’çśżåŔŕäżěéĂŽè£ĞæŔŘä¿Ż __getstate__()
åŠŇ __setstate__() æŰźæşŢæİěçżŢè£Ğè£ŹäžŻéŹŘåĹűãĂĆ
åęĆæđIJåőŽäźL’äžĘè£ŹäÿđäÿłæŰźæşŢïijŇpickle.dump()
åřśäijŽèřČçŤÍ __getstate__() èŐůåŔŰåžŔåĹŮåŇŰçŽĎåŕźèśąãĂĆ
çśżäijijçŽĎïijŇ__setstate__() åIJÍåŔ■åžŔåĹŮåŇŰæŮűèćńèřČçŤÍãĂĆäÿžäžĘæijŢçđ’žè£Źäÿłåůěä¡IJåČ
äÿŇéİćæŸŕäÿĂäÿłåIJÍåĘĚéČÍåőŽäźL’äžĘäÿĂäÿłçž£çÍŃä¡Ęäż■çĎůåŔŕäżěåžŔåĹŮåŇŰåŠŇåŔ■åžŔåĹŮåŇŰç

```
# countdown.py
import time
import threading

class Countdown:
    def __init__(self, n):
        self.n = n
        self.thr = threading.Thread(target=self.run)
        self.thr.daemon = True
        self.thr.start()

    def run(self):
        while self.n > 0:
            print('T-minus', self.n)
            self.n -= 1
            time.sleep(5)
```

```
    def __getstate__(self):
        return self.n

    def __setstate__(self, n):
        self.__init__(n)
```

èŕŢçİĂè£ŘèąŉäÿŉéİćçŽĎåžŔåĹŮåŇŰèŕŢéłŇäżčçăĄïijŽ

```
>>> import countdown
>>> c = countdown.Countdown(30)
>>> T-minus 30
T-minus 29
T-minus 28
...

>>> # After a few moments
>>> f = open('cstate.p', 'wb')
>>> import pickle
>>> pickle.dump(c, f)
>>> f.close()
```

çĎűåŘŐéĂĂåĞžPythonèğčæđŘåŹĺåźűéĞ■åŘŕåŘŐåĘ■èŕŢéłŇäÿŉïijŽ

```
>>> f = open('cstate.p', 'rb')
>>> pickle.load(f)
countdown.Countdown object at 0x10069e2d0>
T-minus 19
T-minus 18
...
```

ä¡ăåŔŕäžěçIJŃåĹřçž£çĺŃåŔĹåěĞè£źèĹňçŽĎéĞ■çŤ§äžĘïijŇäżŐä¡ăçňňäÿĂæňąåžŔåĹŮåŇŰåőČçŽĎåIJ

pickle åŕźäžŐåď ğåđŃçŽĎæŢřæ■őçżŞæđĎæŕŤåęĆä¡£çŤĺ array æĹŰ numpy æĺąåİŮåĹŻåžžçŽĎäžŇè£ŻåĹűæŢřçżĎæŢĹçŐĞåźűäÿ■æŸŕäÿĂäÿłéńŸæŢĹçŽĎçijŰçăĄæŰźåijŔãĂĆ åęĆæđIJä¡ăéIJĂèęĄçğżåŁĺåď ğéĞŔçŽĎæŢřçżĎæŢřæ■őïijŇä¡ăæIJĂåě¡æŸŕåĚĹåIJĺäÿĂäÿłæŰĞäžűäÿ■årĘåĚ (éIJĂèęĄçňňäÿL'æŰźåžŞçŽĎæŤŕæŇĄ)ãĂĆ

çŤśäžŐ pickle æŸŕPythonçL'źæIJL'çŽĎåźűäÿŤéŹĎçİĂåIJĺæžŘçăĄäÿŁïijŇæL'ĂæIJL'åęĆæđIJéIJĂèę ä¿ŃåęĆïijŇåęĆæđIJæžŘçăĄåŘŸåŁĺäžĘïijŇä¡ăæL'ĂæIJL'çŽĎå■ŸåĆĺæŢřæ■őåŔŕèČ¡äijŽèćńçăt'åİŔåźűäÿŤåŘ åİęçŹ¡æİèèőšïijŇårźäžŐåIJĺæŢřæ■őåžŞåŠŇå■Ÿæąčæ ŰĞäżűäÿ■å■ŸåĆĺæŢřæ■őæŮűïijŇä¡ăæIJĂåě¡ä¡£çŤĺæŽ è£ŹäžŽçijŰçăĄæăijåijŔæŽt'æăĞåĞĘïijŇåŔŕäžěèćńäÿ■åŘŇçŽĎèŕ■èĺĂæŤŕæŇĄïijŇåźűäÿŤäź§èČ¡å¿Ĺåě¡çŽĎ

æIJĂåŘŐäÿĂçĆźèęĄæşĺæĎŔçŽĎæŸŕ pickle æIJL'åď ğéĞŔçŽĎéĚ■ç¡őéĂL'éąźåŠŇäÿĂäžŽæčŸæL'Ń åŕźäžŐæIJĂåÿÿèğĄçŽĎä¡£çŤĺåIJžæŹŕïijŇä¡ääÿ■éIJĂèęĄåŐżæŃĚå£Čè£ŹäÿłïijŇä¡ĘæŸŕåęĆæđIJä¡ăèęĄåIJĺ æIJĂåě¡åŐżæ§ěéŸĚäÿĂäÿŃ åőŸæŰźæŰĞæąč ãĂĆ

8 çňňåĚ■çńäïijŽæŢřæ■őçijŰçăĄåŠŇåď'ĎçŘĘ

è£ŹäÿĂçńääÿžèęĄèőÍèőžä¡£çŤÍPythonåď'ĎçŘĘåŘĎçğ■äÿ■åŘŇæŰźåijŔçijŰçăĄçŽĎæŢřæ■őïijŇæŕŤåę
åŠŇæŢřæ■őçżŞæđĎéĆčäÿĂçńääÿ■åŘŇçŽĎæŸŕïijŇè£Źçńääÿ■äijŽèőÍèőžçĽźæőŁçŽĎçőŮæşŢéŮőécŸïijŇè

Contents:

8.1 6.1 èŕžåĘŹCSVæŢřæ■ő

éŮőécŸ

ä¡ăæČşèŕžåĘŹäÿĂääÿłCSVæăijåijŔçŽĎæŰĞäžűãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐåď'ğåď'ŽæŢřçŽĎCSVæăijåijŔçŽĎæŢřæ■őèŕžåĘŹéŮőécŸïijŇéČ¡åŔŕäžěä¡£çŤÍ
csv åžŞãĂĆ ä¿ŃåęĆïijŽåĄĞèő¿ä¡ääIJÍäÿĂäÿłåŘ■åŔństocks.csvæŰĞäžűäÿ■æIJĽäÿĂäžŻèĆąçĕÍåÿĆåIJžæŢ

```
Symbol,Price,Date,Time,Change,Volume
"AA",39.48,"6/11/2007","9:36am",-0.18,181800
"AIG",71.38,"6/11/2007","9:36am",-0.15,195500
"AXP",62.58,"6/11/2007","9:36am",-0.46,935000
"BA",98.31,"6/11/2007","9:36am",+0.12,104800
"C",53.08,"6/11/2007","9:36am",-0.25,360900
"CAT",78.29,"6/11/2007","9:36am",-0.23,225400
```

äÿŃéIćåŘŚä¡ääśŢçď'žåęĆä¡ŢårĘè£ŹäžŻæŢřæ■őèŕžåŔŰäÿžäÿĂäÿłåĚČçżĎçŽĎåžŔåĹŮïijŽ

```
import csv
with open('stocks.csv') as f:
    f_csv = csv.reader(f)
    headers = next(f_csv)
    for row in f_csv:
        # Process row
        ...
```

åIJÍäÿŁéIćçŽĎäžčçăĄäÿ■ïijŇ `row` äijŽæŸŕäÿĂäÿłåĹŮèąÍãĂĆåŻăæ■d'ïijŇäÿžäžĘèő£éŮőæ§Răÿłå■Ůæč
`row[0]` èő£éŮőSymbolïijŇ `row[4]` èő£éŮőChangeãĂĆ

çŤśäžŐè£Źçğ■äÿŃæăĞèő£éŮőéĂŽåÿÿäijŽåijŢètůæůůæůĘïijŇä¡ääŔŕäžěèĂČèŹŚä¡£çŤÍåŚ¡åŘ■åĚČçżĎâ

```
from collections import namedtuple
with open('stock.csv') as f:
    f_csv = csv.reader(f)
    headings = next(f_csv)
    Row = namedtuple('Row', headings)
    for r in f_csv:
        row = Row(*r)
```

```
        # Process row
        ...
```

åőČåĚĄèőÿä¡ää¡£çŤÍåĹŮåŘ■åęĆ row.Symbol åŠŇ row.Change äžčæŻ£äÿŃæăĞèő£éŮőãĂĆ éĲĂèęĄæşÍæĐŘçŽĎæŸŕè£Źäÿłå̊ŔłæĲĽåĲÍåĹŮåŘ■æŸŕåŘĹæşŢçŽĎPythonæ ä¡ääŔŕèČ¡éĲĂèęĄä£őæŤźäÿŃåŐ§åğŃçŽĎåĹŮåŘ■(åęĆåŕĘéÍđæăĞèŕĘçňęå■Ůçňęæ£æ■ćæĹŘäÿŃåĹŠçž£ä

åŘęåď ŰäÿĂäÿłéĂĽæŃľåřśæŸŕåřĘæŢřæ■őèŕżåŘŰåĹřäÿĂäÿłå■ŮåĚÿåžŔåĹŮäÿ■åŐżãĂĆåŔŕäżěè£Źæă

```
import csv
with open('stocks.csv') as f:
    f_csv = csv.DictReader(f)
    for row in f_csv:
        # process row
        ...
```

åĲÍè£ŹäÿłçĽĹæĲňäÿ■ïijŇä¡ääŔŕäżěä¡£çŤÍåĹŮåŘ■åŐżèő£éŮőæŕŔäÿĂèąŇçŽĎæŢřæ■őäžĘãĂĆæŕŤåęĆ æĹŰèĂĚ row['Change']

äÿžäžĘåĘŹåĚěCSVæŢřæ■őïijŇä¡ääż■çĎűåŔŕäżěä¡£çŤÍcsvæÍąåİŮïijŇäÿ■è£Ğè£ŹæŮűåĂŹåĚĹåĹŻåżžä writer åŕźèśąãĂĆä¿Ńåę Ć:

```
headers = ['Symbol','Price','Date','Time','Change','Volume']
rows = [('AA', 39.48, '6/11/2007', '9:36am', -0.18, 181800),
         ('AIG', 71.38, '6/11/2007', '9:36am', -0.15, 195500),
         ('AXP', 62.58, '6/11/2007', '9:36am', -0.46, 935000),
       ]

with open('stocks.csv','w') as f:
    f_csv = csv.writer(f)
    f_csv.writerow(headers)
    f_csv.writerows(rows)
```

åęĆæđIJä¡ăæIJĽäÿĂäÿłå■ŮåĚÿåžŔåĹŮçŽĎæŢřæ■őïijŇåŔŕäżěåČŔè£ŹæăůåĄŽïijŽ

```
headers = ['Symbol', 'Price', 'Date', 'Time', 'Change', 'Volume']
rows = [{'Symbol':'AA', 'Price':39.48, 'Date':'6/11/2007',
        'Time':'9:36am', 'Change':-0.18, 'Volume':181800},
        {'Symbol':'AIG', 'Price': 71.38, 'Date':'6/11/2007',
        'Time':'9:36am', 'Change':-0.15, 'Volume': 195500},
        {'Symbol':'AXP', 'Price': 62.58, 'Date':'6/11/2007',
        'Time':'9:36am', 'Change':-0.46, 'Volume': 935000},
        ]

with open('stocks.csv','w') as f:
    f_csv = csv.DictWriter(f, headers)
    f_csv.writeheader()
    f_csv.writerows(rows)
```

èőÍèőž

ä¡ãåžŤèřěæĂžæŸřäijŸåĚĹéĂĽ'æŃľcsvæĺąåİŮåĹĘåĽ'šæĹŰèğčæđŘCSVæŢřæ■őãĂĆä¿ŃåęĆïijŇä¡ãåŔŕ

```
with open('stocks.csv') as f:
for line in f:
    row = line.split(',')
    # process row
    ...
```

ä¡£çŤĺè£ŹçğæŰźåijŔçŽĎäÿĂäÿłçijžçĆźåřśæŸřä¡ääž■çĎűéIJĂèęĄåŐžåď'ĎçŘĘäÿĂäžŻæčŸæĽ'ŃçŽĎç
æřŤåęĆïijŇåęĆæđIJæ§ŘäžŻå■Ůæőţå Äijèćńåij ŢåŔůåŇĚåŻť'ïijŇä¡ääÿ■å¿Ůäÿ■åŐžéŹď'è£ŹäžŻåijŢåŔůãĂĆ
åŔęåď'ŰïijŇåęĆæđIJäÿĂäÿłèćńåij ŢåŔůåŇĚåŻť'çŽĎå■ŮæőţçćřåůğåŘńæIJĽäÿĂäÿłéĂŮåŔůïijŇéĆčäźĹçĺŃåž

éżŸèőď'æČĚåĘţäÿŃïijŇcsv åžŞåŔŕèřĘåĹńMicrosoft Ex-
celæĽ'Ăä¡£çŤĺçŽĎCSVçijŰçăĄèğĎåĹŹãĂĆ è£ŹæĹŰèőÿäź§æŸřæIJĂåÿÿèğĄçŽĎå¡ćåijŔïijŇåźűäÿŤäź§äijŽ
çĎűèĂŇïijŇåęĆæđIJä¡ăæ§ěçIJŃcsvçŽĎæŰĞæąčïijŇåřśäijŽåŔŚçŐřæIJĽ'å¿Ĺåď'Žçğ■æŰźæŞţåřĘáőČåžŢçŤĺ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăæČşèŕžåŔŰäžětabåĹĘåĽ'šçŽĎæŢřæ■őïijŇåŔŕäžěè£ŹæăůåĄŽïijŽ

```
# Example of reading tab-separated values
with open('stock.tsv') as f:
    f_tsv = csv.reader(f, delimiter='\t')
    for row in f_tsv:
        # Process row
        ...
```

åęĆæđIJä¡ăæ■čåIJĺèŕžåŔŰCSVæŢřæ■őåźűåřĘåőČäžňè¡ňæ■ćäÿžåŚ¡åŘ■åĚČçżĎïijŇéIJĂèęĄæşĺæĎŔåŕź
ä¿ŃåęĆïijŇäÿĂäÿłCSVæäijåijŔæŰĞäžűæIJĽ'äÿĂäÿłåŇĚåŘńéİđæşŢæăĞèŕĘçňęçŽĎåĹŮåď't'èąŇïijŇçśžäijijž

```
StreetÂăAddress,Num-Premises,Latitude,Longitude 5412ÂăNÂăCLARK,10,
↪41.980262,-87.668452
```

è£ŹæăůæIJĂçżĹäijŽåŕijèĞt'åIJĺåĹŻåžžäÿĂäÿłåŚ¡åŘ■åĚČçżĎæŮűäžğçŤ§äÿĂäÿł
ValueError åijĆåÿÿèĂŇåď'śèt'ěãĂĆ äÿžäžEèğčåEşè£ŹéŮőécŸïijŇä¡ääŔŕèČ¡äÿ■å¿Ůäÿ■åĚĹåŐžä£őæ■čå
ä¿ŃåęĆïijŇåŔŕäžěåČŔäÿŃéİćè£ŹæăůåIJĺéİđæşŢæăĞèŕĘçňęäÿŁä¡£çŤĺäÿĂäÿłæ■čåĹŹèąĺè¿¿åijŔæŻ£æ■ćïijŽ

```
import re
with open('stock.csv') as f:
    f_csv = csv.reader(f)
    headers = [ re.sub('[^a-zA-Z_]', '_', h) for h in next(f_csv) ]
    Row = namedtuple('Row', headers)
    for r in f_csv:
        row = Row(*r)
        # Process row
        ...
```

è£ŸæIJĽ'éĞ■èęĄçŽĎäÿĂçĆźéIJĂèęĄåijžèřČçŽĎæŸřïijŇcsväžğçŤ§çŽĎæŢřæ■őéČ¡æŸřå■Ůçňęäÿšçśżå
åęĆæđIJä¡ăéIJĂèęĄåĄŽè£ŹæăůçŽĎçśžåđŃè¡ňæ■ćïijŇä¡ää£Ěéążè ĞłåůśæĽ'ŃåŁĺåŐžåőđçŐřãĂĆ
äÿŃéİćæŸřäÿĂäÿłåIJĺCSVæŢřæ■őäÿŁæĽ'ğèąŇåĚűäžŮçśžåđŃè¡ňæ■ćçŽĎä¿Ńå■ŘïijŽ

```
col_types = [str, float, str, str, float, int]
with open('stocks.csv') as f:
    f_csv = csv.reader(f)
    headers = next(f_csv)
    for row in f_csv:
        # Apply conversions to the row items
        row = tuple(convert(value) for convert, value in zip(col_
↪types, row))
        ...
```

åŘęåď ŰïijŇäÿŃéÍćæŸŕäÿĂäÿłè¡ňæ■ćå■ŮåĚÿäÿ■çL’ź åőŽå■Ůæőţç ŽĎä¿Ńå■ŘïijŽ

```
print('Reading as dicts with type conversion')
field_types = [ ('Price', float),
                ('Change', float),
                ('Volume', int) ]

with open('stocks.csv') as f:
    for row in csv.DictReader(f):
        row.update((key, conversion(row[key]))
                   for key, conversion in field_types)
        print(row)
```

éĂŽåÿÿæÍěëőšïijŇä¡ăåŘŕèČ¡åźűäÿ■æČşè£ĞåďŽåŐżèĂČèŹŚè£ŹäžŻè¡ňæ■ćéŮőécŸãĂĆ åIJĺåőđéŹĚæČĚåĘţäÿ■ïijŇCSVæŰĞäżűéČ¡æĹŰåď ŽæĹŰåřŚæIJL’äžŽçijžåď śçŽĎæŢřæ■őïijŇèćńçăť åİŘçŽĎ åŻäæ■d’ïijŇéŹd’éÍđä¡ăçŽĎæŢřæ■őçąőåőđæIJL’ä£ÍéŽIJæŸŕåĞEçąőæŮäèŕŕçŽĎïijŇåŘęåĹŹä¡åå£Ěéążè ĂČèŹ

æIJĂåŘŐïijŇåęĆæđIJä¡ăèŕżåŘŰCSVæŢřæ■őçŽĎçŽőçŽĎæŸŕåĄŽæŢřæ■őåĹĘæđŘåŠŇçż§èőąçŽĎèŕÍïij ä¡ăåŘŕèČ¡éIJĂèęĄçIJŃäÿĂçIJŃ `Pandas` åŇĚãĂĆPandas åŇĚåŘńäžĘäÿĂäÿłéÍđåÿÿæŰźä¿£çŽĎåĞ¡æŢřåŘń `pandas.read_csv()` ïijŇ åőČåŘŕäžěåŁäè¡¡CSVæŢřæ■őåĹřäÿĂäÿł `DataFrame` åŕźèśąäÿ■åŐżãĂĆ çĎűåŘŐåĹl’çŤÍè£Źäÿłåŕźèśąä¡ăåřśåŘŕäžěçŤ§æĹŘåŘĎçğ■å¡ćåijŘçŽĎçż§èőąãĂĄè£Ğæżď’æŢřæ■őäžěåŘŁæI åIJÍ6.13åřŘèŁĆäÿ■äijŽæIJL’è£ŹæăůäÿĂäÿłä¿Ńå■ŘãĂĆ

8.2 6.2 èŕżåĘŹJSONæŢřæ■ő

éŮőécŸ

ä¡ăæČşèŕżåĘŹJSON(JavaScript Object Notation)çijŰçăĄæäijåijŘçŽĎæŢřæ■őãĂĆ

èğčåĘşæŰźæąĹ

`json` æĺąåİŮæŘŘä¿ŻäžĘäÿĂçğ■å¿ĹçőĂå■ŢçŽĎæŰźåijŘæİěçijŰçăĄåŠŇèğčçăĄJSONæŢřæ■őãĂĆ åĚűäÿ■äÿďäÿłäÿżèęĄçŽĎåĞ¡æŢřæŸŕ `json.dumps()` åŠŇ `json.loads()` ïijŇ èęĄæŕŤåĚűäżŰåžŘåĹŮåŇŰåĞ¡æŢřåžŞåęĆpickleçŽĎæŐěåŘčářŠå¿Ůåď ŽãĂĆ äÿŃéÍćæijŤçď žåęĆä¡ŢářĘäÿĂäÿłPythonæŢřæ■őçżŞædĎè¡ňæ■ćäÿžJSONïijŽ

```
import json

data = {
    'name' : 'ACME',
    'shares' : 100,
    'price' : 542.23
}

json_str = json.dumps(data)
```

äÿŃéÍćæijŤçď'žåęĆä¡ŢåřĘäÿĂäÿłJSONçijŰçăĄçŽĎå■Ůçňęäÿšè¡ňæ■ćåŻđäÿĂäÿłPythonæŢřæ■őçżŞæđÌ

```
data = json.loads(json_str)
```

åęĆæđIJä¡ăèęĄåď'ĎçŘĘçŽĎæŸŕæŰĞäžűèĂŇäÿ■æŸŕå■ŮçňęäÿšïijŇä¡ăåŔŕäžěä¡£çŤÍ `json.dump()` åŠŇ `json.load()` æÍěçijŰçăĄåŠŇèğčçăĄJSONæŢřæ■őãĆĆä¿ŃåęĆïijŽ

```
# Writing JSON data
with open('data.json', 'w') as f:
    json.dump(data, f)

# Reading data back
with open('data.json', 'r') as f:
    data = json.load(f)
```

èőÍèőž

JSONçijŰçăĄæŤŕæŇĄçŽĎå§žæIJňæŢřæ■őçśžåđŃäÿž `None` ïijŇ `bool` ïijŇ `int` ïijŇ `float` åŠŇ `str` ïijŇ äžěåŔŁåŇĚåŘńè£ŹäžŻçśžåđŃæŢřæ■őçŽĎlistsïijŇtuplesåŠŇdictionariesãĆĆ åŕźäžŐdictionariesïijŇkeyséIJĂèęĄæŸŕå■ŮçňęäÿšçśžåđŃ(å■ŮåĚÿäÿ■äżžä¡ŢéÍđå■ŮçňęäÿšçśžåđŃçŽĎkeyåI äÿžäžĘéĄţå¿ŁJSONèğĎèŇČïijŇä¡ăåžŤèŕěåŔŁçijŰçăĄPythonçŽĎlistsåŠŇdictionariesãĆĆ èĂŇäÿŤïijŇåIJÍwebåžŤçŤÍçÍŇåžŔäÿ■ïijŇéąűåśĆåŕźèśąèćńçijŰçăĄäÿžäÿĂäÿłå■ŮåĚÿæŸŕäÿĂäÿłæăĞåĞĘåA

JSONçijŰçăĄçŽĎæăijåijŔåŕźäžŐPythonèŕ■æşŢèĂŇåůšåĞääźŐæŸŕåőŇåĚÍäÿĂæăůçŽĎïijŇéŹď'äžĘäÿĂ æŕŤåęĆïijŇTrueäijŽèćńæŸăåřĎäÿžtrueïijŇFalseèćńæŸăåřĎäÿž-falseïijŇèĂŇNoneäijŽèćńæŸăåřĎäÿžnullãĆĆ äÿŃéÍćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇæijŤçď'žäžĘçijŰçăĄåŘŐçŽĎå■

```
>>> json.dumps(False)
'false'
>>> d = {'a': True,
...      'b': 'Hello',
...      'c': None}
>>> json.dumps(d)
'{"b": "Hello", "c": null, "a": true}'
>>>
```

åęĆæđIJä¡ăèŕŢçİÅåŐżæčÄæ§ěJSONèğčçăĄåŘŐçŽĎæŢřæ■őïijŇä¡ăéĂŽåÿÿå¿ĹéŽ¿éĂŽè£ĞçőÅå■ŢçŽĹ çL'źåĹńæŸŕå¡ŞæŢřæ■őçŽĎåţŇåěŮçżŞæđĎåśĆæňąå¿ĹæůśæĹŰèĂĚåŇĚåŘńåď'ğéĞŘçŽĎå■ŮæőţæŮűãĆĆ äÿžäžĘèğčåĘşè£ŹäÿłéŮőécŸïijŇåŔŕäžěèĂČèŹŠä¡£çŤÍpprintæÍąåİŮçŽĎ

pprint() åĞ¡æŢřæİěäžčæŻ£æŹőéĂŽçŽĎ print() åĞ¡æŢřãĂĆ
åőČäijŽæŇL'çĚğkeyçŽĎå■Ůæí■éąžåžŔåźűäžěäÿĂçğ■æŻť'åŁăç¿ŐèğĆçŽĎæŰźåijŔè¿ŞåĞžãĂĆ
äÿŃéÍćæŸŕäÿĂäÿłæijŤçď'žåęĆä¡ŢæijĆäžőçŽĎæL'Şå■řè¿ŞåĞžTwitteräÿŁæŘIJçť'ćçżŞædIJçŽĎä¿Ńå■ŘïijŽ

```
>>> from urllib.request import urlopen
>>> import json
>>> u = urlopen('http://search.twitter.com/search.json?q=python&
 ↪rpp=5')
>>> resp = json.loads(u.read().decode('utf-8'))
>>> from pprint import pprint
>>> pprint(resp)
{'completed_in': 0.074,
'max_id': 264043230692245504,
'max_id_str': '264043230692245504',
'next_page': '?page=2&max_id=264043230692245504&q=python&rpp=5',
'page': 1,
'query': 'python',
'refresh_url': '?since_id=264043230692245504&q=python',
'results': [{'created_at': 'Thu, 01 Nov 2012 16:36:26 +0000',
             'from_user': ...
             },
             {'created_at': 'Thu, 01 Nov 2012 16:36:14 +0000',
             'from_user': ...
             },
             {'created_at': 'Thu, 01 Nov 2012 16:36:13 +0000',
             'from_user': ...
             },
             {'created_at': 'Thu, 01 Nov 2012 16:36:07 +0000',
             'from_user': ...
             }
             {'created_at': 'Thu, 01 Nov 2012 16:36:04 +0000',
             'from_user': ...
             }],
'results_per_page': 5,
'since_id': 0,
'since_id_str': '0'}
>>>
```

äÿĂèĹñæİěèőšïijŇJSONèğčçăĄäijŽæăźæ■őæŔŘä¿ŻçŽĎæŢřæ■őåĹŻåžždictsæĹŰlistsãĂĆ
åęĆæđIJä¡ăæČşèęĄåĹŻåžžåĚűäžŰçśžåđŇçŽĎåŕźèšąïijŇåŔŕäžěçżŹ json.
loads() äijăéĂŠobject_pairs_hookæĹŰobject_hookåŔĆæŢřãĂĆ
ä¿ŃåęĆïijŇäÿŃéÍćæŸŕæijŤçď'žåęĆä¡Ţèğčçă ĄJSONæŢřæ■őåźűåIJĺäÿĂäÿłOrderedDictäÿ■ä£ÍçŢŹåĚűéąžåžŘ

```
>>> s = '{"name": "ACME", "shares": 50, "price": 490.1}'
>>> from collections import OrderedDict
>>> data = json.loads(s, object_pairs_hook=OrderedDict)
>>> data
OrderedDict([('name', 'ACME'), ('shares', 50), ('price', 490.1)])
>>>
```

äÿŃéÍćæŸŕåęĆä¡ŢåřĘäÿĂäÿłJSONå■ŮåĚÿè¡ñæ■ćäÿžäÿĂäÿłPythonåŕźèšąä¿Ńå■ŘïijŽ

```
>>> class JSONObject:
...     def __init__(self, d):
...         self.__dict__ = d
...
>>>
>>> data = json.loads(s, object_hook=JSONObject)
>>> data.name
'ACME'
>>> data.shares
50
>>> data.price
490.1
>>>
```

æIJÄåŘŐäÿĂäÿłä¿Ńå■Řäÿ■ïijÑJSONèğčçăĄåŘŐçŽĎå■ŮåĚÿä¡IJäÿžäÿĂäÿłå■Ţäÿłå
__init__() ãĂĆ çĎűåŘŐïijÑä¡ăåřśåŔŕäżěéŽŔå£ČæĽĂæňšçŽĎä¡£çŤĺåőČäžĘïijÑæŕŤåęĆä¡IJäÿžäÿĂäÿłå

åIJÍçijŰçăĄJSONçŽĎæŮűåĂŹïijÑè£ŸæIJĽäÿĂäžŻéĂĽéąźå¿ĹæIJĽçŤĺãĂĆ
åęĆædIJä¡ăæČşèŐůå¿ŮæijĆäžőçŽĎæăijåijŔåŇŰå■ŮçňęäÿšåŘŐè¿ŞåĞžïijÑåŔŕäżěä¡£çŤĺ
json.dumps() çŽĎindentåŔĆæŢřãĂĆ åőČäijŽä¡£å¿Ůè¿ŞåĞžåŠÑpprint()åĞ¡æŢřæŢĹædIJçśżäijijãĂĆæŕŤ

```
>>> print(json.dumps(data))
{"price": 542.23, "name": "ACME", "shares": 100}
>>> print(json.dumps(data, indent=4))
{
    "price": 542.23,
    "name": "ACME",
    "shares": 100
}
>>>
```

åŕźèśąåőđä¿ŃéĂŽåÿÿåźűäÿ■æŸŕJSONåŔŕåžŔåĹŮåŇŰçŽĎãĂĆä¿ŃåęĆïijŽ

```
>>> class Point:
...     def __init__(self, x, y):
...         self.x = x
...         self.y = y
...
>>> p = Point(2, 3)
>>> json.dumps(p)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "/usr/local/lib/python3.3/json/__init__.py", line 226, in␣
↪dumps
        return _default_encoder.encode(obj)
    File "/usr/local/lib/python3.3/json/encoder.py", line 187, in␣
↪encode
        chunks = self.iterencode(o, _one_shot=True)
    File "/usr/local/lib/python3.3/json/encoder.py", line 245, in␣
↪iterencode
        return _iterencode(o, 0)
```

```
    File "/usr/local/lib/python3.3/json/encoder.py", line 169, in␣
↪default
        raise TypeError(repr(o) + " is not JSON serializable")
TypeError: <__main__.Point object at 0x1006f2650> is not JSON␣
↪serializable
>>>
```

åęĆædIJä¡ăæČşåžŘåĹŮåŇŰářźèśąåőđä¿ŃïijŇä¡ăåŘŕäžěæŔŘä¿ŻäÿĂäÿłåĞ¡æŢřïijŇåőČçŽĎè¿ŞåĚěæŸŕ

```
def serialize_instance(obj):
    d = { '__classname__' : type(obj).__name__ }
    d.update(vars(obj))
    return d
```

åęĆædIJä¡ăæČşåŔ■è£ĞæİěèŐůåŔŰè£Źäÿłåőđä¿ŃïijŇåŘŕäžěè£ŹæăůåĄŽïijŽ

```
# Dictionary mapping names to known classes
classes = {
    'Point' : Point
}

def unserialize_object(d):
    clsname = d.pop('__classname__', None)
    if clsname:
        cls = classes[clsname]
        obj = cls.__new__(cls) # Make instance without calling __
↪init__
        for key, value in d.items():
            setattr(obj, key, value)
        return obj
    else:
        return d
```

äÿŃéİćæŸŕåęĆä¡Ţä¡£çŤĺè£ŹäžŻåĞ¡æŢřçŽĎä¿Ńå■ŘïijŽ

```
>>> p = Point(2,3)
>>> s = json.dumps(p, default=serialize_instance)
>>> s
'{"__classname__": "Point", "y": 3, "x": 2}'
>>> a = json.loads(s, object_hook=unserialize_object)
>>> a
<__main__.Point object at 0x1017577d0>
>>> a.x
2
>>> a.y
3
>>>
```

json æĺąåİŮè£ŸæIJL’å¿Ĺåd’ŽåĚűäžŰéĂL’éąźæİěæŐğåĹűæŻt’ä¡ŐçžğåĹńçŽĎæŢřå■ŮãĂĄçL’źæőŁåĂi
åŘŕäžěåŘĆèĂČåőŸæŰźæŰĞæąčèŐůåŘŰæŻt’åd’ŽçżĘèŁĆãĂĆ

8.3 6.3 èğčæđŘçőĂå■ŢçŽĎXMLæŢřæ■ő

éŮőécŸ

ä¡ăæČşäżŐäÿĂäÿłçőĂå■ŢçŽĎXMLæŰĞæąčäÿ■æŔŘåŔŰæŢřæ■őãĂĆ

èğčåĘşæŰźæąĹ

åŘŕäžěä¡£çŤÍ xml.etree.ElementTree æĺąåİŮäżŐçőĂå■ŢçŽĎXMLæŰĞæąčäÿ■æŔŘåŔŰæŢřæ■őãĂĆ äÿžäžĘæijŤçd'žïijŇåĄĞèő¿ä¡ăæČşèğčæđŘPlanet PythonäÿŁçŽĎRSSæžŘãĂĆäÿŃéİćæŸŕçŻÿåžŤçŽĎäžčçăĄïijŽ

```
from urllib.request import urlopen
from xml.etree.ElementTree import parse

# Download the RSS feed and parse it
u = urlopen('http://planet.python.org/rss20.xml')
doc = parse(u)

# Extract and output tags of interest
for item in doc.iterfind('channel/item'):
    title = item.findtext('title')
    date = item.findtext('pubDate')
    link = item.findtext('link')

    print(title)
    print(date)
    print(link)
    print()
```

è£ŘèąŇäÿŁéİćçŽĎäžčçăĄïijŇè¿ŞåĞžçżŞæđIJçśżäijijè£ŹæăůïijŽ

```
Steve Holden: Python for Data Analysis
Mon, 19 Nov 2012 02:13:51 +0000
http://holdenweb.blogspot.com/2012/11/python-for-data-analysis.html

Vasudev Ram: The Python Data model (for v2 and v3)
Sun, 18 Nov 2012 22:06:47 +0000
http://jugad2.blogspot.com/2012/11/the-python-data-model.html

Python Diary: Been playing around with Object Databases
Sun, 18 Nov 2012 20:40:29 +0000
http://www.pythondiary.com/blog/Nov.18,2012/been-...-object-
 ↪databases.html

Vasudev Ram: Wakari, Scientific Python in the cloud
Sun, 18 Nov 2012 20:19:41 +0000
http://jugad2.blogspot.com/2012/11/wakari-scientific-python-in-
 ↪cloud.html
```

```
Jesse Jiryu Davis: Toro: synchronization primitives for Tornado␣
↪coroutines
Sun, 18 Nov 2012 20:17:49 +0000
http://feedproxy.google.com/~r/EmptysquarePython/~3/_DOZT2Kd0hQ/
```

å¿ÍæŸ¿çĎűïijŇåęĆæđIJä¡ăæČşåĄŽè£ŻäÿĂæ■ěçŽĎåď ĎçŘĘïijŇä¡ăéIJĂèęĄæŻ£æ■ć `print()` èí■åŔěæİěåőŇæÍŘåĚűäžŰæIJL'èűčçŽĎäžŇãĂĆ

èőÍèőž

åIJÍå¿Íåď ŽåžŤçŤÍçÍŃåžŔäÿ■åď ĎçŘĘXMLçijŰçăĄæăijåijŔçŽĎæŢřæ■őæŸŕå¿Íåÿÿè§ĄçŽĎãĂĆäÿ■äžĚåŻääÿžXMLåIJÍInternetäÿŁéİćåůšçżŔèćńäź£æşŽåžŤçŤÍäžŐæŢřæ■őäžď æ■ćïijŇåŘŇæŮűåőČäź§æŸŕäÿĂçğ■å■ŸåĆÍåžŤçŤÍçÍŃåžŔæŢřæ■őçŽĎåÿÿçŤÍæăijåijŔ(æŕŤåęĆå■Ůåď ĎçŘĘïijŇé§şäæŐěäÿŃæİěçŽĎèőÍèőžäijŽåĚÍåĄĞåőŽèŕżèĂĚåůšçżŔåŕźXMLå§žçąĂæŕŤè¿ČçĘ§æĆL'äžĘãĂĆ

åIJÍå¿Íåď ŽæČĚåĘţäÿŃïijŇå¡Şä¡£çŤÍXMLæİěäżĚäżĚå■ŸåĆÍæŢřæ■őçŽĎæŮűåĂŹïijŇåŕźåžŤçŽĎæŰĞä¿ŃåęĆïijŇäÿŁéİćä¿Ńå■Řäÿ■çŽĎRSSèőćéŸĚæžŘçśżäijijäžŐäÿŃéİćçŽĎæăijåijŔïijŽ

```
<?xml version="1.0"?>
<rss version="2.0" xmlns:dc="http://purl.org/dc/elements/1.1/">
    <channel>
        <title>Planet Python</title>
        <link>http://planet.python.org/</link>
        <language>en</language>
        <description>Planet Python - http://planet.python.org/</
↪description>
        <item>
            <title>Steve Holden: Python for Data Analysis</title>
            <guid>http://holdenweb.blogspot.com/...-data-analysis.
↪html</guid>
            <link>http://holdenweb.blogspot.com/...-data-analysis.
↪html</link>
            <description>...</description>
            <pubDate>Mon, 19 Nov 2012 02:13:51 +0000</pubDate>
        </item>
        <item>
            <title>Vasudev Ram: The Python Data model (for v2 and␣
↪v3)</title>
            <guid>http://jugad2.blogspot.com/...-data-model.html</
↪guid>
            <link>http://jugad2.blogspot.com/...-data-model.html</
↪link>
            <description>...</description>
            <pubDate>Sun, 18 Nov 2012 22:06:47 +0000</pubDate>
        </item>
        <item>
            <title>Python Diary: Been playing around with Object␣
↪Databases</title>
```

```
            <guid>http://www.pythondiary.com/...-object-databases.
↪html</guid>
            <link>http://www.pythondiary.com/...-object-databases.
↪html</link>
            <description>...</description>
            <pubDate>Sun, 18 Nov 2012 20:40:29 +0000</pubDate>
        </item>
        ...
    </channel>
</rss>
```

xml.etree.ElementTree.parse() åĞ¡æŢřèğčæđŘæŢť äÿłXMLæŰĞæąčåźűåřĘåĚűè¡ňæ■ćæĹŘ çĎűåŘŐïijŇä¡ăåřśèČ¡ä¡£çŤÍ find() ãĂĄiterfind() åŠŇ findtext() ç■Ľ æŰźæşŢæİěæŘIJçť ćçĽ źåőŽçŽĎXMLåĚČçť ääžĘãĂĆ è£ŹäžŻåĞ¡æŢřçŽĎåŔĆæŢřåřśæŸŕæ§ŘäÿłæŇĞåőŽçŽĎæăĞç■¿åŘ■ïijŇä¿ŃåęĆ channel/item æĹŰ title ãĂĆ

æŕŔæňąæŇĞåőŽæ§ŘäÿłæăĞç■¿æŮűïijŇä¡ăéIJĂèęĄéĄ■åŐĘæŢť äÿłæŰĞæąčçżŞæđĎãĂĆæŕŔæňąæŘIJ åŘŇæăůïijŇæŕŔæňąæŞ■ä¡IJæĽ ĂæŇĞåőŽçŽĎæăĞç■¿åŘ■äź§æŸŕèţůåğŃåĚČçť ăçŽĎçŻÿåŕźèůŕå¿ĎãĂĆ ä¿ŃåęĆïijŇæĽ ğèąŇ doc.iterfind('channel/item') æİěæŘIJçť ćæĽ ĂæIJĽ åIJÍ channel åĚČçť ăäÿŃéİćçŽĎ item åĚČçť ăãĂĆ doc äżčèąĺæŰĞæąčçŽĎæIJĂéąűåśĆ(äź§åřśæŸŕçňňäÿĂçžğçŽĎ rss åĚČçť ă)ãĂĆ çĎűåŘŐæŐěäÿŃæİěçŽĎèřČçŤÍ item.findtext() äijŽäżŐåůšæĽ¿åĹřçŽĎ item åĚČçť ăä¡■ç¡őåijĂåğŃæŘIJçť ćãĂĆ

ElementTree æĺąåİŮäÿ■çŽĎæŕŔäÿłåĚČçť ăæIJĽ äÿĂäžŻéĞ■èęĄçŽĎåśđæĂğåŠŇæŰźæşŢïijŇåIJÍèğč tag åśđæĂğåŇĚåŘńäžĘæăĞç■¿çŽĎåŘ■å■ŮïijŇtext åśđæĂğåŇĚåŘńäžĘåĘĚéČĺçŽĎæŰĞæIJňïijŇèĂŇ get() æŰźæşŢèČ¡èŐůåŔŰåśđæĂğåĂijãĂĆä¿ŃåęĆïijŽ

```
>>> doc
<xml.etree.ElementTree.ElementTree object at 0x101339510>
>>> e = doc.find('channel/title')
>>> e
<Element 'title' at 0x10135b310>
>>> e.tag
'title'
>>> e.text
'Planet Python'
>>> e.get('some_attribute')
>>>
```

æIJĽ äÿĂçĆźèęĄåijžèřČçŽĎæŸŕ xml.etree.ElementTree åźűäÿ■æŸŕXM-LèğčæđŘçŽĎåŤŕäÿĂæŰźæşŢãĂĆ åŕźäžŐæŻť éńŸçžğçŽĎåžŤçŤĺçĺŃåžŔïijŇä¡ăéIJĂèęĄèĂČèŹŚä¡£çŤÍ lxml ãĂĆ åőČä¡£çŤĺäžĘåŠŇElementTreeåŘŇæăůçŽĎçijŰçĺŃæŐěåŔčïijŇåŻăæ■ď äÿŁéİćçŽĎä¿Ńå■ŘåŘŇæ ä¡ăåŔłéIJĂèęĄåřĘåĹŽåijĂåğŃçŽĎimportèŕ■åŔěæ■ćæĹŘ from lxml.etree import parse åřśèąŇäžĘãĂĆ lxml åőŇåĚĺéĄţå¿łXMLæăĞåĞĘïijŇåźűäÿŤéĂ§åžęäź§éİđåÿÿå£ńïijŇåŘŇæŮűè£Ÿæ

8.4 6.4 åćđéĞŔåijŔèğčædŘåď'ğåđŃXMLæŰĞäžű

éŮőécŸ

ä¡äæČşä¡£çŤĺår¡åŔŕèČ¡ářŚçŽĎåĘĚå■ŸäżŐäÿĂäÿłèűĚåď'ğçŽĎXMLæŰĞæąčäÿ■æŔŘåŔŰæŢřæ■őãĂĆ

èğčåĘşæŰźæąĹ

äżżä¡ŢæŮűåĂŹåŔłèęĄä¡äéĄĞåĹřåćđéĞŔåijŔçŽĎæŢřæ■őåď'ĎçŘĘæŮűïijŇçňňäÿĂæŮűéŮť'åřśåžŤèŕěa
äÿŃéİćæŸŕäÿĂäÿłå¿ĹçőĂå■ŢçŽĎåĞ¡æŢřïijŇåŔłä¡£çŤĺå¿ĹářŚçŽĎåĘĚå■ŸåřśèČ¡åćđéĞŔåijŔçŽĎåď'ĎçŘĘä

```
from xml.etree.ElementTree import iterparse

def parse_and_remove(filename, path):
    path_parts = path.split('/')
    doc = iterparse(filename, ('start', 'end'))
    # Skip the root element
    next(doc)

    tag_stack = []
    elem_stack = []
    for event, elem in doc:
        if event == 'start':
            tag_stack.append(elem.tag)
            elem_stack.append(elem)
        elif event == 'end':
            if tag_stack == path_parts:
                yield elem
                elem_stack[-2].remove(elem)
            try:
                tag_stack.pop()
                elem_stack.pop()
            except IndexError:
                pass
```

äÿžäžĘæţŃèŕŢè£ŹäÿłåĞ¡æŢřïijŇä¡äéIJĂèęĄåĚĹæIJL'äÿĂäÿłåď'ğåđŃçŽĎXMLæŰĞäžűãĂĆ
éĂŽåÿÿä¡ääŔŕäžěåIJÍæŤ£åžIJç¡ŚçńŹæĹŰåĚňåĚśæŢřæ■őç¡ŚçńŹäÿŁæL'¿åĹřè£Źæăůç˘ĎæŰĞäžűãĂĆ
ä¿ŃåęĆïijŇä¡ääŔŕäžěäÿŃè¡¡XMLæăijåijŔçŽĎèŁİåŁääŞěå§ŐåÿĆéĄŞèůŕåİŚæť'ijæŢřæ■őåžŞãĂĆ
åIJĺåĘŹè£ŹæIJñäźęçŽĎæŮűåĂŹïijŇäÿŃè¡¡æŰĞäžűåůšçżŔåŇĚåŘńèűĚè£Ğ100,000èąŇæŢřæ■őïijŇçijŰçăĄ

```
<response>
    <row>
        <row ...>
            <creation_date>2012-11-18T00:00:00</creation_date>
            <status>Completed</status>
            <completion_date>2012-11-18T00:00:00</completion_date>
            <service_request_number>12-01906549</service_request_
 ↪number>
            <type_of_service_request>Pot Hole in Street</type_of_
 ↪service_request>
```

```
            <current_activity>Final Outcome</current_activity>
            <most_recent_action>CDOT Street Cut ... Outcome</most_
↪recent_action>
            <street_address>4714 S TALMAN AVE</street_address>
            <zip>60632</zip>
            <x_coordinate>1159494.68618856</x_coordinate>
            <y_coordinate>1873313.83503384</y_coordinate>
            <ward>14</ward>
            <police_district>9</police_district>
            <community_area>58</community_area>
            <latitude>41.808090232127896</latitude>
            <longitude>-87.69053684711305</longitude>
            <location latitude="41.808090232127896"
            longitude="-87.69053684711305" />
        </row>
        <row ...>
            <creation_date>2012-11-18T00:00:00</creation_date>
            <status>Completed</status>
            <completion_date>2012-11-18T00:00:00</completion_date>
            <service_request_number>12-01906695</service_request_
↪number>
            <type_of_service_request>Pot Hole in Street</type_of_
↪service_request>
            <current_activity>Final Outcome</current_activity>
            <most_recent_action>CDOT Street Cut ... Outcome</most_
↪recent_action>
            <street_address>3510 W NORTH AVE</street_address>
            <zip>60647</zip>
            <x_coordinate>1152732.14127696</x_coordinate>
            <y_coordinate>1910409.38979075</y_coordinate>
            <ward>26</ward>
            <police_district>14</police_district>
            <community_area>23</community_area>
            <latitude>41.91002084292946</latitude>
            <longitude>-87.71435952353961</longitude>
            <location latitude="41.91002084292946"
            longitude="-87.71435952353961" />
        </row>
    </row>
</response>
```

åĄĞèő¿ä¡ăæČşåĘźäÿĂäÿłèĎŽæIJňæİěæŇL’çĔğåİŚæt’ijæŁěåŚŁæŢřéĞŔæŐŠåĹŮéĆőçijŰůŔůçăĄãĂĆä

```
from xml.etree.ElementTree import parse
from collections import Counter

potholes_by_zip = Counter()

doc = parse('potholes.xml')
for pothole in doc.iterfind('row/row'):
```

```
    potholes_by_zip[pothole.findtext('zip')] += 1
for zipcode, num in potholes_by_zip.most_common():
    print(zipcode, num)
```

è£Źäÿłè ĎŽæIJňåŤŕäÿĂçŽĎéŮőécŸæŸŕåőČäijŽåĚĹåřĘæŢťäÿłXMLæŰĞäžűåŁäè¡¡åĹřåĘĚå■Ÿäÿ■çĎűå IJĺæĹŠçŽĎæIJžåŹĺäÿŁïijŇäÿžäžĘè£ŘèąŇè£ŹäÿłçĺŇåžŔéIJĂèęĄçŤĺåĹř450MBåůęåŔşçŽĎåĘĚå■Ÿçľ žéŮť'ă åęĆædIJä¡£çŤĺåęĆäÿŇäžčçăĄïijŇçĺŇåžŔåŔłéIJĂèęĄä£őæŤźäÿĂçĆźçĆźïijŽ

```
from collections import Counter

potholes_by_zip = Counter()

data = parse_and_remove('potholes.xml', 'row/row')
for pothole in data:
    potholes_by_zip[pothole.findtext('zip')] += 1
for zipcode, num in potholes_by_zip.most_common():
    print(zipcode, num)
```

çżŞædIJæŸŕïijŽè£ŹäÿłçL'ĹæIJňçŽĎäžčçăĄè£ŘèąŇæŮűåŔłéIJĂèęĄ7MBçŽĎåĘĚå■Ÿ–åď'ğåď'ğèŁĆçžę

èőĺèőž

è£ŹäÿĂèŁĆçŽĎæŁĂæIJŕäijŽä¿ÌèţŰ `ElementTree` æĺąåİŮäÿ■çŽĎäÿď'äÿłæăÿå£ČåŁ§èČ¡ãĂĆ çňňäÿĂïijŇiterparse() æŰźæşŢåĚĄèőÿåŕźXMLæŰĞæąčè£ŻèąŇåćđéĞŔæŞ■ä¡IJãĂĆ ä¡£çŤĺæŮűïijŇä¡äéIJĂèęĄæŔŘä¿ŻæŰĞäžűåŘ■åŠŇäÿĂäÿłåŇĚåŘńäÿŃéİćäÿĂçğ■æĹŮåď'Žçğ■çśžåđŇçŽĎä `start` , `end`, `start-ns` åŠŇ `end-ns` ãĂĆ çŤś `iterparse()` åĹŻåžžçŽĎè£■äžčåŹĺäijŽäžğçŤ§å¡ćåęĆ `(event, elem)` çŽĎåĚČçżĎïijŇ åĚűäÿ■ `event` æŸŕäÿŁè£ŕäžŃäžűåĹŮèąĺäÿ■çŽĎæ§ŘäÿĂäÿłïijŇèĂŇ `elem` æŸŕçŻÿåžŤçŽĎXM-LåĚČçť'ãăĂĆä¿ŃåęĆïijŽ

```
>>> data = iterparse('potholes.xml',('start','end'))
>>> next(data)
('start', <Element 'response' at 0x100771d60>)
>>> next(data)
('start', <Element 'row' at 0x100771e68>)
>>> next(data)
('start', <Element 'row' at 0x100771fc8>)
>>> next(data)
('start', <Element 'creation_date' at 0x100771f18>)
>>> next(data)
('end', <Element 'creation_date' at 0x100771f18>)
>>> next(data)
('start', <Element 'status' at 0x1006a7f18>)
>>> next(data)
('end', <Element 'status' at 0x1006a7f18>)
>>>
```

`start` äžŃäžűåIJĺæ§ŘäÿłåĚČçť'ăçňňäÿĂæňąèćńåĹŻåžžåźűäÿŤè£ŸæšąæIJL'èćńæŔŠåĚěåĚűäžŮæŢřæ■ èĂŇ `end` äžŃäžűåIJĺæ§ŘäÿłåĚČçť'ăåůšçżŔåőŇæĹŘæŮűèćńåĹŻåžžãĂĆ

åř¡çőąæšąæIJL'åIJÍä¿Ńå■Řäÿ■æijŤçď žïijŃ `start-ns` åŠŃ `end-ns` äžŃäžűèćńçŤÍæÍěåď'ĎçŘĘXMLæŰĞæąčåŚ¡åŘ■çl'žéÚť'çŽĎåčřæŸŐãĂĆ

è£ŹæIJńèŁĆä¿Ńå■Řäÿ■ïijŃ `start` åŠŃ `end` äžŃäžűèćńçŤÍæÍěçőąçŘĘåĚČçt'ăåŠŃæăĞç■¿æăĹãĂĆ æăĹäžčèąÍäžĘæŰĞæąčèćńèğčæđŘæŮűçŽĎåśĆæňąçżŞæđĎïijŃ è£ŸèćńçŤÍæÍěåĹď'æŰ■æ§Řäÿ łåĚČçt'ăæŸŕåŘęåŃźéĚ■äijäçżŹåĞ¡æŢř `parse_and_remove()` çŽĎèůŕå¿ĎãĂĆ åęĆæđIJåŃźéĚ■ïijŃåřśåĹl'çŤÍ `yield` èŕ■åŘěåŘŚèřČçŤÍèĂĚè£ŤåŻđè£ŹäÿłåĚČçt'ãăĂĆ

åIJÍ `yield` äźŃåŘŐçŽĎäÿŃéÍćè£Źäÿłèŕ■åŘěæL'■æŸŕä¡£å¿ŮçÍŃåžŘå■ăçŤÍæđĄåřŚåĘĚå■ŸçŽĎElemen

```
elem_stack[-2].remove(elem)
```

è£Źäÿłèŕ■åŘěä¡£å¿ŮäźŃåL'■çŤś `yield` äžğçŤ§çŽĎåĚČçt'ăäżŐåőČçŽĎçĹűèŁĆçĆźäÿ■åĹäéŹđ'æŐL'ãA åĄĞèő¿åůšçżŔæšąæIJL'åĚűåőČçŽĎåIJřæŰźåijŢçŤÍè£ŹäÿłåĚČçt'ăäžĘïijŃéĆčäźLè£ŹäÿłåĚČçt'ăåřśèćńéŤĂæ

åŕźèŁĆçĆźçŽĎè£■äżčåijŔèğčæđŘåŠŃåĹäéŹđ'çŽĎæIJĂçżĹæŢĹæđIJåřśæŸŕäÿĂäÿłåIJÍæŰĞæąčäÿŁéńŸ æŰĞæąčæăŚçżŞæđĎäžŐåğŃèĞłçżĹæšąèćńåőŇæŢt'çŽĎåĹŻåžžè£ĞãĂĆår¡çőąåęĆæ■đ'ïijŃè£ŸæŸŕèČ¡éĂŽè

è£ŹçğæŰźæąĹçŽĎäÿżèęĄçijžéŹůåřśæŸŕåőČçŽĎè£ŘèąŇæĂğèČ¡äžĘãĂĆ æĹŚèĞłåůśætŃèŕŢçŽĎçżŞæđIJæŸŕŕïijŃèŕżåŘŰæŢt'äÿłæŰĞæąčåĹřåĘĚå■ŸäÿçŽĎçL'ĹæIJńçŽĎè£ŘèąŇéĂ§ ä¡ĘæŸŕåőČå■t'ä¡£çŤÍäžĘèúĚè£ĞåŘŐèĂĚ60åĂçŽĎåĘĚå■ŸãĂĆ åŻăæ■đ'ïijŃåęĆæđIJä¡ăæŻt'åĚşå£ČåĘĚå■Ÿä¡£çŤÍéĞŘçŽĎèŕĬïijŃéĆčäźĹåćđéĞŘåijŘçŽĎçL'ĹæIJńåőŃèČIJ

8.5 6.5 åřĘå■ŮåĚÿè¡ňæ■ćäÿžXML

éŮőécŸ

ä¡ăæČşä¡£çŤÍäÿĂäÿłPythonå■ŮåĚÿå■ŸåĆÍæŢřæ■őïijŃåźűåřĘåőČè¡ňæ■ćæĹŘXMLæăijåijŘãĂĆ

èğčåĘşæŰźæąĹ

åř¡çőą `xml.etree.ElementTree` åžŞéĂŽåÿÿçŤÍæÍěåĄŽèğčæđŘåůëä¡IJïijŃåĚűåőđåőČäź§åŔŕäžëåÍ ä¿ŃåęĆïijŃèĂČèŹŚåęĆäÿŃè£ŹäÿłåĞ¡æŢřïijŽ

```
from xml.etree.ElementTree import Element

def dict_to_xml(tag, d):
'''
Turn a simple dict of key/value pairs into XML
'''
elem = Element(tag)
for key, val in d.items():
    child = Element(key)
    child.text = str(val)
    elem.append(child)
return elem
```

äÿŃéÍćæŸŕäÿĂäÿłä¡£çŤÍä¿Ńå■ŘïijŽ

```
>>> s = { 'name': 'GOOG', 'shares': 100, 'price':490.1 }
>>> e = dict_to_xml('stock', s)
>>> e
<Element 'stock' at 0x1004b64c8>
>>>
```

è¡ňæ■ćçżŞæđIJæŸŕäÿĂäÿł `Element` åőđä¿ŃãĂĆåŕźäžŐI/OæŞ■ä¡IJïijŇä¡£çŤÍ `xml.etree.ElementTree` äÿ■çŽĎ `tostring()` åĞ¡æŢřå¿ĹåőźæŸŞåřśèČ¡åřĘåőČè¡ňæ■ćæĹŘäÿĂäÿłå■ŮèŁ

```
>>> from xml.etree.ElementTree import tostring
>>> tostring(e)
b'<stock><price>490.1</price><shares>100</shares><name>GOOG</name></
↪stock>'
>>>
```

åęĆæđIJä¡ăæČşçżŹæ§ŘäÿłåĚČçť'ăæůżåŁåášđæĂğåĂijïijŇåŔŕäžěä¡£çŤÍ `set()` æŰźæşŢïijŽ

```
>>> e.set('_id','1234')
>>> tostring(e)
b'<stock _id="1234"><price>490.1</price><shares>100</shares><name>
↪GOOG</name>
</stock>'
>>>
```

åęĆæđIJä¡ăè£ŸæČşä£ÍæŇĄåĚČçť'ăçŽĎéąžåžŔïijŇåŔŕäžěèĂČèŹŚæđĎéĂääÿĂäÿł `OrderedDict` æÍěäžčæŻ£äÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿãĂĆèŕůåŔĆèĂČ1.7åřŔèŁĆãĂĆ

èőÍèőž

å¡ŞåĹŻåžžXMLçŽĎæŮűåĂŹïijŇä¡ăèćńéŹŘåĹűåŔłèČ¡æđĎéĂää■ŮçňęäÿšçśžåđŃçŽĎåĂijãĂĆä¿ŃåęĆï

```
def dict_to_xml_str(tag, d):
    '''
    Turn a simple dict of key/value pairs into XML
    '''
    parts = ['<{}>'.format(tag)]
    for key, val in d.items():
        parts.append('<{0}>{1}</{0}>'.format(key,val))
    parts.append('</{}>'.format(tag))
    return ''.join(parts)
```

éŮőécŸæŸŕåęĆæđIJä¡ăæL'ŃåŁÍçŽĎåŐżæđĎéĂăçŽĎæŮűåĂŹåŔŕèČ¡äijŽçćřåĹřäÿĂäžŽéžżçČęãĂĆä¿Ń

```
>>> d = { 'name' : '<spam>' }

>>> # String creation
>>> dict_to_xml_str('item',d)
'<item><name><spam></name></item>'
```

```
>>> # Proper XML creation
>>> e = dict_to_xml('item',d)
>>> tostring(e)
b'<item><name>&lt;spam&gt;</name></item>'
>>>
```

æşÍæĎŔåĹřçÍŃåžŔçŽĎåŘŐéÍćéĆčäÿłä¿Ńå■Řäÿ■ïijŇå■Ůçňę âĂŸ<âĂŹ åŠŇ âĂŸ>âĂŹ èćńæŻ£æ■ćæĹŘäžĘ `<` åŠŇ `>`

äÿŃéÍćäżĚä¿ŻåŔĆèĂČïijŇåęĆæđIJä¡ăéIJĂèęĄæĽ'ŃåŁĺåŐżè¡ňæ■ćè£ŹäžŻå■Ůçňęïij Ň åŘŕäžĕä¡£çŤÍ `xml.sax.saxutils` äÿ■çŽĎ `escape()` åŠŇ `unescape()` åĞ¡æŢřãĂĆä¿ŃåęĆïijŽ

```
>>> from xml.sax.saxutils import escape, unescape
>>> escape('<spam>')
'&lt;spam&gt;'
>>> unescape(_)
'<spam>'
>>>
```

éŹđ'äžĘèČ¡åĹŻåžžæ■čçąőçŽĎè¿ŞåĞžåđ'ŰïijŇè£ŸæIJL'åŔęåđ'ŰäÿĂäÿłåŐ§åŻăæŐÍè■Řä¡ääĹŻåžž `Element` åőđä¿ŃèĂŇäÿ■æŸŕå■Ůçňęäÿšïij Ń éĆčářśæŸŕä¡£çŤÍå■ŮçňęäÿšçżĎåŘĹæđĎéĂäÿĂäÿłæŻt'åđ'ğç èĂŇ `Element` åőđä¿ŃåŔŕäžĕäÿ■çŤÍèĂČèŹŚèğčæđŘXMLæŰĞæIJňçŽĎæČĚåĘţäÿŃéĂŽè£Ğåđ'Žçğ■æŰźåi äź§ářśæŸŕèŕt'ïijŇä¡ääŔŕäžĕäIJÍäÿĂäÿłéńŸçžğæŢřæ■őçżŞæđĎäÿŁåőŇæĹŘä¡ăæĽ'ĂæIJL'çŽĎæŞ■ä¡IJïijŇåźű

8.6 6.6 èğčæđŘåŠŇä£őæŤźXML

éŮőécŸ

ä¡ăæČşèŕžåŘŰäÿĂäÿłXMLæŰĞæąčïijŇåŕźåőČæIJĂäÿĂäžŻä£őæŤźïijŇçĎűåŘŐåřĘçżŞæđIJåĘŹåŻđXM

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `xml.etree.ElementTree` æÍąåİŮåŔŕäžĕå¿ĹåőźæŸŞçŽĎåđ'ĎçŘĘè£ŹäžŻäżżåŁąãĂĆ çňňäÿĂæ■ěæŸŕäžĕéĂŽåÿÿçŽĎæŰźåijŔæİèèğčæđŘè£ŹäÿłæŰĞæąčãĂĆä¿ŃåęĆïijŇåĄĞèő¿ä¡ăæIJL'äÿĂäÿłå `pred.xml` çŽĎæŰĞæąčïijŇçśżäijijäÿŃéÍćè£Źæăůïij Ž

```
<?xml version="1.0"?>
<stop>
    <id>14791</id>
    <nm>Clark &amp; Balmoral</nm>
    <sri>
        <rt>22</rt>
        <d>North Bound</d>
        <dd>North Bound</dd>
    </sri>
    <cr>22</cr>
```

```
    <pre>
        <pt>5 MIN</pt>
        <fd>Howard</fd>
        <v>1378</v>
        <rn>22</rn>
    </pre>
    <pre>
        <pt>15 MIN</pt>
        <fd>Howard</fd>
        <v>1867</v>
        <rn>22</rn>
    </pre>
</stop>
```

äÿŃéÌćæŸŕäÿĂäÿłåĹľçŤÍ ElementTree æÌěèŕžåŔŰè£ŹäÿłæŰĞæąčåźűåŕźåőČåĄŽäÿĂäžŽä£őæŤźçŽ

```
>>> from xml.etree.ElementTree import parse, Element
>>> doc = parse('pred.xml')
>>> root = doc.getroot()
>>> root
<Element 'stop' at 0x100770cb0>

>>> # Remove a few elements
>>> root.remove(root.find('sri'))
>>> root.remove(root.find('cr'))
>>> # Insert a new element after <nm>...</nm>
>>> root.getchildren().index(root.find('nm'))
1
>>> e = Element('spam')
>>> e.text = 'This is a test'
>>> root.insert(2, e)

>>> # Write back to a file
>>> doc.write('newpred.xml', xml_declaration=True)
>>>
```

åď ĎçŘEçżŞæđIJæŸŕäÿĂäÿłåČŔäÿŃéÌćè£ŹæăůæŰřçŽĎXMLæŰĞäžűïijŽ

```
<?xml version='1.0' encoding='us-ascii'?>
<stop>
    <id>14791</id>
    <nm>Clark &amp;amp; Balmoral</nm>
    <spam>This is a test</spam>
    <pre>
        <pt>5 MIN</pt>
        <fd>Howard</fd>
        <v>1378</v>
        <rn>22</rn>
    </pre>
    <pre>
        <pt>15 MIN</pt>
```

```
        <fd>Howard</fd>
        <v>1867</v>
        <rn>22</rn>
    </pre>
</stop>
```

èőÍèőž

ä£őæŤźäÿĂäÿłXMLæŰĞæąčçžŞæđĎæŸŕå¿Ĺåőźæŷ§çŽĎïijŇä¡ĘæŸŕä¡ăå£Ěéążç L'ćèőřçŽĎæŸŕæL'ĂæI
åřĘåőČä¡IJäÿžäÿĂäÿłåĹŮèąÍæİěåď'ĎçŘĘãĂĆä¿ŃåęĆïijŇåęĆæđIJä¡ăåĹăéŹď'æ§Řäÿłå ĚČçt'ăïijŇéĂŽè£ĞěřČ
`remove()` æŰźæşŢäżŐåőČçŽĎçŻť'æŐěçĹűèŁĆçĆźäÿ■åĹăéŹď'ãĂĆ
åęĆæđIJä¡ăæŔŠåĚěæĹŰåćđåŁăæŰřçŽĎåĚČçt'ăïijŇä¡ăåŘŇæăůä¡£çŤĺçĹűèŁĆçĆźåĚČçt'ăçŽĎ
`insert()` åŠŇ `append()` æŰźæşŢãĂĆ è£ŸèČ¡åŕźåĚČçt'ăä¡£çŤĺçt'ćåijŢåŠŇåĹĞçL'ĞæŞ■ä¡IJïijŇæŕŤåęĆ
`element[i]` æĹŰ `element[i:j]`

åęĆæđIJä¡ăéIJĂèęĄåĹŻåżžæŰřçŽĎåĚČçt'ăïijŇåŔŕäżěä¡£çŤĺæIJňèŁĆæŰźæąĹäÿ■æijŤçd'žçŽĎ
`Element` çśżãĂĆæĹŠäżňåIJĺ6.5åřŔèŁĆåůšçżŔèŕęçżĘèőÍèőžè£ĞäžĘãĂĆ

8.7 6.7 åĹľçŤĺåŚ¡åŘ■çľžéŮť'èğčæđŘXMLæŰĞæąč

éŮőécŸ

ä¡ăæČşèğčæđŘæ§ŘäÿłXMLæŰĞæąčïijŇæŰĞæąčäÿ■ä¡£çŤĺäžĘXMLåŚ¡åŘ■çľžéŮť'ãĂĆ

èğčåĘşæŰźæąĹ

èĂČèŹŠäÿŃéIćè£Źäÿłä¡£çŤĺäžĘåŚ¡åŘ■çľžéŮť'çŽĎæŰĞæąčïijŽ

```
<?xml version="1.0" encoding="utf-8"?>
<top>
    <author>David Beazley</author>
    <content>
        <html xmlns="http://www.w3.org/1999/xhtml">
            <head>
                <title>Hello World</title>
            </head>
            <body>
                <h1>Hello World!</h1>
            </body>
        </html>
    </content>
</top>
```

åęĆæđIJä¡ăèğčæđŘè£ŹäÿłæŰĞæąčåźűæL'ğèąŇæŹőéĂŽçŽĎæ§ěèŕćïijŇä¡ăäijŽåŔŚçŐřè£Źäÿłåźűäÿ■æŸ

```
>>> # Some queries that work
>>> doc.findtext('author')
'David Beazley'
>>> doc.find('content')
<Element 'content' at 0x100776ec0>
>>> # A query involving a namespace (doesn't work)
>>> doc.find('content/html')
>>> # Works if fully qualified
>>> doc.find('content/{http://www.w3.org/1999/xhtml}html')
<Element '{http://www.w3.org/1999/xhtml}html' at 0x1007767e0>
>>> # Doesn't work
>>> doc.findtext('content/{http://www.w3.org/1999/xhtml}html/head/
↪title')
>>> # Fully qualified
>>> doc.findtext('content/{http://www.w3.org/1999/xhtml}html/'
... '{http://www.w3.org/1999/xhtml}head/{http://www.w3.org/1999/
↪xhtml}title')
'Hello World'
>>>
```

ä¡ăåŔŕäżěéĂŽè£ĞåřĘåŚ¡åŘ■çľ'žéŮť'åď'ĎçŘĘéĂżè¿ŚåŇĚèčĚäÿžäÿĂäÿłåůěåĚůçśżæİěçőĂåŃŰè£Źäÿłè£

```
class XMLNamespaces:
    def __init__(self, **kwargs):
        self.namespaces = {}
        for name, uri in kwargs.items():
            self.register(name, uri)
    def register(self, name, uri):
        self.namespaces[name] = '{'+uri+'}'
    def __call__(self, path):
        return path.format_map(self.namespaces)
```

éĂŽè£ĞäÿŃéİćçŽĎæŰźåijŔä¡£çŤĺè£ŹäÿłçśżïijŽ

```
>>> ns = XMLNamespaces(html='http://www.w3.org/1999/xhtml')
>>> doc.find(ns('content/{html}html'))
<Element '{http://www.w3.org/1999/xhtml}html' at 0x1007767e0>
>>> doc.findtext(ns('content/{html}html/{html}head/{html}title'))
'Hello World'
>>>
```

èőĺèőž

èğčæđŘåŘńæIJL'åŚ¡åŘ■çľ'žéŮť'çŽĎXMLæŰĞæąčäijŽæŕŤè¿ČçźĄçŘŘãĂĆ äÿŁéİćçŽĎ `XMLNamespaces` äżĚäżĚæŸŕåĚĄèőÿä¡ăä¡£çŤĺçijľçŢěåŘ■äżčæŻ£åőŇæŢť'çŽĎURIåřĘåĚűåŔŸå¿Űçĺ■å¿őç

å¿Láÿ■åźÿçŽĎæŸŕïijŇåIJĺå§žæIJňçŽĎ `ElementTree` èğčæđŘäÿ■æšąæIJL'äżżä¡ŢéĂŤå¿ĎèŐůåŔŰåŚ¡åŘ■çľ'žéŮť'çŽĎä£ąæAŕãĂĆ ä¡ĘæŸŕïijŇåęĆæđIJä¡ăä¡£çŤĺ `iterparse()` åĞ¡æŢřçŽĎèŕİåřśåŔŕäżěèŐůåŔŰæŻť'åď'ŽåĚşäžŐåŚ¡åŘ■çľ'žé

```
>>> from xml.etree.ElementTree import iterparse
>>> for evt, elem in iterparse('ns2.xml', ('end', 'start-ns', 'end-
 ↪ns')):
... print(evt, elem)
...
end <Element 'author' at 0x10110de10>
start-ns ('', 'http://www.w3.org/1999/xhtml')
end <Element '{http://www.w3.org/1999/xhtml}title' at 0x1011131b0>
end <Element '{http://www.w3.org/1999/xhtml}head' at 0x1011130a8>
end <Element '{http://www.w3.org/1999/xhtml}h1' at 0x101113310>
end <Element '{http://www.w3.org/1999/xhtml}body' at 0x101113260>
end <Element '{http://www.w3.org/1999/xhtml}html' at 0x10110df70>
end-ns None
end <Element 'content' at 0x10110de68>
end <Element 'top' at 0x10110dd60>
>>> elem # This is the topmost element
<Element 'top' at 0x10110dd60>
>>>
```

æIJĂåŘŐäÿĂçĆźïijŇåęĆæđIJä¡ãèęĄåď'ĎçŘEçŽĎXMLæŰĞæIJňéŹď'äžEèęĄä¡£çŤĺåĹřåĚűäžŰéńŸçžğ
åžžèőőä¡ãæIJĂåě¡æŸŕä¡£çŤÍ `lxml` åĞ¡æŢřåžŞæİěäžčæŻ£ `ElementTree` ãĂĆ
ä¿ŇåęĆïijŇ`lxml`åŕźåĹľçŤÍDTDéłŇèŕĄæŰĞæąčãĂĄæŻť'åě¡çŽĎXPathæŤŕæŇĄåŠŇäÿĂäžŽåĚűäžŰéńŸçžğ
è£ŹäÿĂåřŔèŁĆåĚűåőđåŔłæŸŕæŢŹä¡ãåęĆä¡Ţèőľ'XMLèğčæđŘçĺ■å¿őçőĂå■ŢäÿĂçĆźãĂĆ

8.8 6.8 äÿŐåĚşçşżåđŃæŢřæ■őåžŞçŽĎäžď'äžŠ

éŮőécŸ

ä¡ăæČşåIJĺåĚşçşżåđŃæŢřæ■őåžŞäÿ■æ§ěèŕćãĂĄåćđåŁăæĹŰåĹăéŹď'èőřå¡ŢãĂĆ

èğčåĘşæŰźæąĹ

Pythonäÿ■èąĺçď'žåď'ŽèąŇæŢřæ■őçŽĎæăĞåĞĘæŰźåijŔæŸŕäÿĂäÿłçŤśåĚČçżĎæđĎæĹŘçŽĎåžŔåĹŮãĂ

```
stocks = [
    ('GOOG', 100, 490.1),
    ('AAPL', 50, 545.75),
    ('FB', 150, 7.45),
    ('HPQ', 75, 33.2),
]
```

ä¿Iæ■őPEP249ïijŇéĂŽè£Ğè£Źçğ■å¡ćåijŔæŔŘä¿ŻæŢřæ■őïijŇ
åŔŕäžěå¿ĹåőźæŸ§çŽĎä¡£çŤÍPythonæăĞåĞĘæŢřæ■őåžŞAPIåŠŇåĚşçşżåđŃæŢřæ■őåžŞè£ŻèąŇäžď'äžŠãĂĆ
æĽĂæIJĽæŢřæ■őåžŞäÿŁçŽĎæŞ■ä¡IJéČ¡éĂŽè£ĞSQLæ§ěèŕćèŕ■åŘěæİěåőŇæĹŘãĂĆæŕŔäÿĂèąŇè¿ŞåĚěè¿Ş

äÿžäžĘæijŤçď'žèŕť'æŸŐïijŇä¡ăåŔŕäžěä¡£çŤÍPythonæăĞåĞĘåžŞäÿ■çŽĎ `sqlite3`
æĺąåİŮãĂĆ åęĆæđIJä¡ää¡£çŤÍçŽĎæŸŕäÿĂäÿłäÿ■åŘŇçŽĎæŢřæ■őåžŞ(æŕŤåęĆMySqlãĂĄPostgresqlæĹŰèĂ

è£Ÿå¿Ůåő L’èčĚçŻÿåžŤçŽĎçňňäÿL’æŰźæÍąåİŮæİěæŔŘä¿ŻæŤŕæŇĄãĂĆ
äÿ■è£ĞçŻÿåžŤçŽĎçijŰçÍŃæŐěåŔčåĞääźŐéČ¡æŸŕäÿĂæăůçŽĎïijŇéŹd’äžĘäÿĂçĆźçĆźçżĘå¿őåůőåĹńåd’Ůã

çňňäÿĂæ■ěæŸŕè£đæŐěåĹřæŢřæ■őåžŞãĂĆéĂŽåÿÿä¡äèęĄæL’ğèąŇ `connect()`
åĞ¡æŢřïijŇ çżŹåőČæŔŘä¿ŻäÿĂäžŻæŢřæ■őåžŞåŘ■ãĂĄäÿżæIJžãĂĄçŤÍæĹůåŘ■ãĂĄåŕĘçăĄåŠŇåĚűäżŰå£Ě

```
>>> import sqlite3
>>> db = sqlite3.connect('database.db')
>>>
```

äÿžäžĘåd’ĎçŘĘæŢřæ■őïijŇäÿŃäÿĂæ■ěä¡äéIJĂèęĄåĹŻåżžäÿĂäÿłæÿÿæăĞãĂĆ
äÿĂæŮęä¡äæIJL’äžĘæÿÿæăĞïijŇéĆčäźĹä¡ăåřšåŔŕäžěæL’ğèąŇSQLæ§ěèŕćèŕ■åŘěäžĘãĂĆæŕŤåęĆïijŽ

```
>>> c = db.cursor()
>>> c.execute('create table portfolio (symbol text, shares integer,␣
 ↪price real)')
<sqlite3.Cursor object at 0x10067a730>
>>> db.commit()
>>>
```

äÿžäžĘåŘŚæŢřæ■őåžŞèąĺäÿ■æŘŠåĚěåd’Žæİąèőŕå¡ŢïijŇä¡£çŤĺçśżäijijäÿŃéİćè£ŹæăůçŽĎèŕ■åŘěïijŽ

```
>>> c.executemany('insert into portfolio values (?,?,?)', stocks)
<sqlite3.Cursor object at 0x10067a730>
>>> db.commit()
>>>
```

äÿžäžĘæL’ğèąŇæ§Řäÿłæ§ěèŕćïijŇä¡£çŤĺåČŔäÿŃéİćè£ŹæăůçŽĎèŕ■åŘěïijŽ

```
>>> for row in db.execute('select * from portfolio'):
...     print(row)
...
('GOOG', 100, 490.1)
('AAPL', 50, 545.75)
('FB', 150, 7.45)
('HPQ', 75, 33.2)
>>>
```

åęĆæđIJä¡ăæČşæŐěåŔŮçŤĺæĹůè¿ŞåĚëä¡IJäÿžåŔĆæŢřæİěæL’ğèąŇæ§ěèŕćæŞ■ä¡IJïijŇå£Ěéążçąőä£Ía¡ăä

```
>>> min_price = 100
>>> for row in db.execute('select * from portfolio where price >= ?
 ↪',
                          (min_price,)):
...     print(row)
...
('GOOG', 100, 490.1)
('AAPL', 50, 545.75)
>>>
```

èőÍèőž

åIJÍæŕŤè¿Čä¡ŐçŽĎçžğåĹńäÿŁåŠŇæŢřæ■őåžŞäžď äžŠæŸŕéÍđåÿÿçőĂå■ŢçŽĎãĂĆ
ä¡ăåŘłéIJĂæŘŘä¿ŻSQLèŕ■åŘěåźűèřČçŤÍçŽÿåžŤçŽĎæĺąåÍŮåřśåŘŕäžěæŽť æŰřæĹŰæŘŘåŘŰæŢřæ■őäžĘãĂ
èŹ¡èŕť åęĆæ■ď ïijŇè£ŸæŸŕæIJL äÿĂäžŻæŕŤè¿ČæčŸæĽŃçŽĎçżĘèŁĆéŮőécŸéIJĂèęĄä¡ăéĂŘäÿłåĹŮåĞžåĆ

äÿĂäÿłéŽ¿çĆźæŸŕæŢřæ■őåžŞäÿ■çŽĎæŢřæ■őåŠŇPythonçśžåđŃçŽť æŐěçŽĎæŸăåřĎãĂĆ
åŕźäžŐæŮěæIJ§çśžåđŃïijŇéĂŽåÿÿåŘŕäžěä¡£çŤÍ `datetime` æĺąåÍŮäÿ■çŽĎ `datetime`
åőđä¿ŃïijŇ æĹŰèĂĚåŘŕèČ¡æŸŕ `time` æĺąåÍŮäÿ■çŽĎçşżçż§æŮűéŮť æĹşãĂĆ
åŕźäžŐæŢřå■ŮçśžåđŃïijŇçĽźåĹńæŸŕä¡£çŤÍåĹřåŕŘæŢřçŽĎéĞŚèđ■æŢřæ■őïijŇåŘŕäžěçŤÍ
`decimal` æĺąåÍŮäÿ■çŽĎ `Decimal` åőđä¿ŃæÍěèąĺçď žãĂĆ
äÿ■åźÿçŽĎæŸŕïijŇåŕźäžŐäÿ■åŘŇçŽĎæŢřæ■őåžŞèĂŇèÍĂåĚůä¡ŞæŸăåřĎèğĎåĹŹæŸŕäÿ■äÿĂæăůçŽĎïijŇä¡

åŘęåď ŰäÿĂäÿłæŻť åŁăåď ■æİĆçŽĎéŮőécŸåřśæŸŕSQLèŕ■åŘěå■ŮçňęäÿšçŽĎæđĎéĂããĂĆ
ä¡ăå■ČäÿĞäÿ■èęĄä¡£çŤÍPythonå■ŮçňęäÿšæăijåijŔåŇŰæŞ■ä¡IJçňę(åęĆ%)æĹŰèĂĚ
`.format()` æŰźæşŢæİěåĹŻåżžè£ŹæăůçŽĎå■Ůçňęäÿšã ĂĆ
åęĆæđIJäijăéĂŠçżŹè£ŹäžŻæăijåijŔåŇŰæŞ■ä¡IJçňęçŽĎåĂijæİěèĞłäžŐçŤÍæĹůçŽĎè¿ŞåĚěïijŇéĆčäźĹä¡äçŽ
http://xkcd.com/327)ãĂĆ æ§ěèŕćèŕ■åŘěäÿ■çŽĎéĂŽéĚ■çňę ?
æŇĞçď žåŘŐåŘŕæŢřæ■őåžŞä¡£çŤÍåőČèĞłåůśçŽĎå■ŮçňęäÿšæŻ£æ■ćæIJžåĹűïijŇè£ŹæăůæŻť åŁäçŽĎåőL åł

äÿ■åźÿçŽĎæŸŕïijŇäÿ■åŘŇçŽĎæŢřæ■őåžŞåŘŐåŘřåŕźäžŐéĂŽéĚ■çňęçŽĎä¡£çŤÍæŸŕäÿ■äÿĂæăůçŽĎãĂ
`?` æĹŰ `%s` ïijŇ è£ŸæIJL åĚűäžŰäÿĂäžŻä¡£çŤÍäžĘäÿ■åŘŇçŽĎçňęåŘůïijŇærŤåęĆ:0æĹŰ:1æİěæŇĞçď žåŘĆ
åŘŇæăůçŽĎïijŇä¡ăè£ŸæŸŕå¿ŮåŐżåŘĆèĂĆä¡ăä¡£çŤÍçŽĎæŢřæ■őåžŞæĺąåÍŮçŻÿåžŤçŽĎæŰĞæąčãĂĆ
äÿĂäÿłæŢřæ■őåžŞæĺąåÍŮçŽĎ `paramstyle` åšđæĂğåŇĚåŘńäžĘåŘĆæŢřåijŢçŤÍéčŐæäijçŽĎä£ąæĄŕãĂĆ

åŕźäžŐçőĂå■ŢçŽĎæŢřæ■őåžŞæŢřæ■őçŽĎèŕżåĘŹéŮőécŸïijŇä¡£çŤÍæŢřæ■őåžŞAPIéĂŽåÿÿéİđåÿÿçőĂ
åęĆæđIJä¡ăèęĄåď ĎçŘĘæŽť åĹăåď ■æİĆçŽĎéŮőécŸïijŇåżżèőőä¡ăä¡£çŤÍæŽť åĹăéńŸçžğçŽĎæŐěåŘčïijŇæí
çśžäijij `SQLAlchemy` è£ŹæăůçŽĎåžŞåĚĄèőÿä¡ăä¡£çŤÍPythonçśžæİěèąÍçď žäÿĂäÿłæŢřæ■őåžŞèąĺïijŇ
åźűäÿŤèČ¡åIJÍéŽŘèŮŔåžŢåśĆSQLçŽĎæČĚåĘţäÿŇåődçŐřåŘĎçğ■æŢřæ■őåžŞçŽĎæŞ■ä¡IJãĂĆ

8.9 6.9 çijŰçăĄåŠŇèğčçăĄå■ĄåĚ■è£ŻåĹűæŢř

éŮőécŸ

ä¡ăæČşåřĘäÿĂäÿłå■ĄåĚ■è£ŻåĹűå■ŮçňęäÿšèğčçăĄæĹŘäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšæĹŰèĂĚåřĘäÿĂäÿłå

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăåŘłæŸŕçőĂå■ŢçŽĎèğčçăĄæĹŰçijŰçăĄäÿĂäÿłå■ĄåĚ■è£ŻåĹűçŽĎåŐ§ågŃå■ŮçňęäÿšïijŇå
æĺąåÍŮãĂĆä¿ŃåęĆïijŽ

```
>>> # Initial byte string
>>> s = b'hello'
>>> # Encode as hex
>>> import binascii
>>> h = binascii.b2a_hex(s)
>>> h
b'68656c6c6f'
```

```
>>> # Decode back to bytes
>>> binascii.a2b_hex(h)
b'hello'
>>>
```

çśżäijijçŽĎåŁ§èČ¡åŘŇæăůåŔŕäżěåIJĺ `base64` æĺąåİŮäÿ■æĽ¿åĹřãĂĆä¿ŃåęĆïijŽ

```
>>> import base64
>>> h = base64.b16encode(s)
>>> h
b'68656C6C6F'
>>> base64.b16decode(h)
b'hello'
>>>
```

èőĺèőž

åď ğéČĺåĹĘæČĚåĘţäÿŃïijŇéĂŽè£Ğä¡£çŤĺäÿŁè£řçŽĎåĞ¡æŢřæİěè¡ňæ■ćå■ĄåĚ■è£ŻåĹűæŸŕå¿ĹçőĂå■Ţäÿ
Łéİćäÿďçğ■æŁĂæIJŕçŽĎäÿżèęĄäÿ■åŘŇåIJĺäžŐåď ğářŔåĘŹçŽĎåď ĎçŘĘãĂĆ
åĞ¡æŢř `base64.b16decode()` åŠŇ `base64.b16encode()`
åŔłèČ¡æŞ■ä¡IJåď ğåĘŹå¡ćåijŔçŽĎå■ĄåĚ■è£ŻåĹűå■Ůæŕ■ïijŇ èĂŇ `binascii`
æĺąåİŮäÿ■çŽĎåĞ¡æŢřåď ğářŔåĘŹéČ¡èČ¡åď ĎçŘĘãĂĆ

è£ŸæIJL'äÿĂçĆźéIJĂèęĄæşĺæĎŔçŽĎæŸŕçijÚçăĄåĞ¡æŢřæL'Ăäžğçŧ§çŽĎè¿ŞåĞžæĂżæŸŕäÿĂäÿłå■Ů
åęĆæđIJæČşåijžåĹűäžěUnicodeå¡ćåijŔè¿ŞåĞžïijŇä¡ăéIJĂèęĄåćđåŁääÿĂäÿłéćİåď ŰçŽĎçŢŇéİćæ■ěéłď ãĂĆ

```
>>> h = base64.b16encode(s)
>>> print(h)
b'68656C6C6F'
>>> print(h.decode('ascii'))
68656C6C6F
>>>
```

åIJĺèğčçăĄå■ĄåĚ■è£ŻåĹűæŢřæŮűïijŇåĞ¡æŢř `b16decode()`
åŠŇ `a2b_hex()` åŔŕäžěæŐěåŔŮå■ŮèŁĆæĹŰunicodeå■Ůçňęäÿšã ĂĆ
ä¡ĘæŸŕïijŇunicodeå■Ůçňęäÿšå£ĚéążäžĚäžĚåŔłåŇĚåŘńASCIIçijŰçăĄçŽĎå■ĄåĚ■è£ŻåĹűæŢřãĂĆ

8.10 6.10 çijŰçăĄèğčçăĄBase64æŢřæ■ő

éŮőécŸ

ä¡ăéIJĂèęĄä¡£çŤĺBase64æăijåijŔèğčçăĄæĹŰçijŰçăĄäžŇè£ŻåĹűæŢřæ■őãĂĆ

èğčåĘşæŰźæąĹ

base64 æĺąåİŮäÿ■æIJL'äÿd'äÿłåĞ¡æŢř b64encode() and b64decode() åŔŕäžěåÿőä¡ăèğčåĘşè£ŹäÿłéŮőécŸãĂĆä¿ŃåęĆ;

```
>>> # Some byte data
>>> s = b'hello'
>>> import base64

>>> # Encode as Base64
>>> a = base64.b64encode(s)
>>> a
b'aGVsbG8='

>>> # Decode from Base64
>>> base64.b64decode(a)
b'hello'
>>>
```

èőÍèőž

Base64çijŰçăĄäžĚäžĚçŤÍäžŐéÍćåŘŚå■ŮèŁĆçŽĎæŢřæ■őærŤåęĆå■ŮèŁĆå■ŮçňęäÿšåŠŇå■ŮèŁĆæŢřçż
æ■d'åd'ŰïijŇçijŰçăĄåd'ĎçŘĘçŽĎè¿ŞåĞžçżŞæđIJæĂżæŸŕäÿĂäÿłå■ŮèŁĆå■Ůçňęäÿšã ĂĆ
åęĆæđIJä¡ăæČşæůůåŘĹä¡£çŤÍBase64çijŰçăĄçŽĎæŢřæ■őåŠŇUnicodeæŰĞæIJñïijŇä¡ăå£ĚéążæůżåŁääÿĂä

```
>>> a = base64.b64encode(s).decode('ascii')
>>> a
'aGVsbG8='
>>>
```

å¡Şèğčçă ĄBase64çŽĎæŮűåĂŹïijŇå■ŮèŁĆå■Ůçňęäÿšå ŠŇUnicodeæŰĞæIJñéČ¡åŔŕäžěä¡IJäÿžåŔĆæŢřă
ä¡ĘæŸŕïijŇUnicodeå■Ůçňęäÿšå ŘłèČ¡åŇĚåŘńASCIIå■Ůçňęã ĂĆ

8.11 6.11 èŕžåĘŹäžŇè£ŻåĹűæŢřçżĎæŢřæ■ő

éŮőécŸ

ä¡ăæČşèŕžåĘŹäÿĂäÿłäžŇè£ŻåĹűæŢřçżĎçŽĎçżŞæđĎåŇŰæŢřæ■őåĹřPythonåĚČçżĎäÿ■ãĂĆ

èğčåĘşæŰźæąĹ

åŔŕäžěä¡£çŤÍ struct æĺąåİŮåd'ĎçŘĘäžŇè£ŻåĹűæŢřæ■õã ĂĆ
äÿŃéÍćæŸŕäÿĂæőţçd'žä¿ŃäżčçăĄåřĘäÿĂäÿłPythonåĚČçżĎåĹŮèąĺåĘŹåĚěäÿĂäÿłäžŇè£ŻåĹűæŰĞäżűïijŇåž
struct åřĘæŕŘäÿłåĚČçżĎçijŰçăĄäÿžäÿĂäÿłçżŞæđĎä¡ŞãĂĆ

```
from struct import Struct
def write_records(records, format, f):
    '''
    Write a sequence of tuples to a binary file of structures.
    '''
    record_struct = Struct(format)
    for r in records:
        f.write(record_struct.pack(*r))

# Example
if __name__ == '__main__':
    records = [ (1, 2.3, 4.5),
                (6, 7.8, 9.0),
                (12, 13.4, 56.7) ]
    with open('data.b', 'wb') as f:
        write_records(records, '<idd', f)
```

æIJL'å¿Ĺåd'Žçğ■æŰźæşŢæİèèŕžåŔŰè£ŹäýłæŰĞäżűåźűè£ŤåŻđäÿĂäýłåĚČçżĎåĹŮèąĺãĂĆ
éęŰåĚĹïijŇåęĆæđIJä¡ăæĽ'ŞçőŮäžěåİŮçŽĎå¡ćåijŔåćđéĞŔèŕžåŔŰæŰĞäżűïijŇä¡ăåŔŕäžěè£ŹæăůåĄŽïijŽ

```
from struct import Struct

def read_records(format, f):
    record_struct = Struct(format)
    chunks = iter(lambda: f.read(record_struct.size), b'')
    return (record_struct.unpack(chunk) for chunk in chunks)

# Example
if __name__ == '__main__':
    with open('data.b','rb') as f:
        for rec in read_records('<idd', f):
            # Process rec
            ...
```

åęĆæđIJä¡ăæČşåřĘæŢťäýłæŰĞäżűäÿĂæňąæĂğèŕžåŔŰåĹřäÿĂäýłå■ŮèŁĆå■Ůçňęäÿšäÿ■ïijŇçĎűåŘŐåĽ

```
from struct import Struct

def unpack_records(format, data):
    record_struct = Struct(format)
    return (record_struct.unpack_from(data, offset)
            for offset in range(0, len(data), record_struct.size))

# Example
if __name__ == '__main__':
    with open('data.b', 'rb') as f:
        data = f.read()
    for rec in unpack_records('<idd', data):
        # Process rec
        ...
```

äÿd'çğ■æČĚåĘţäÿŃçŽĎçżŞæđIJéČ¡æŸŕäÿĂäÿłåŔŕè£ŤåŻđçŤĺæİěåĹŻåżžèŕěæŰĞäžűçŽĎåŐ§ågŃåĚČçž

èőÍèőž

åŕźäžŐéIJĂèęĄçijŰçăĄåŠŇèğčçăĄäžŇè£ŻåĹűæŢřæ■őçŽĎçÍŃåžŔèĂŇèÍĂïijŇéĂŽåÿÿäijŽä¡£çŤÍ `struct` æÍąåİŮãĂĆ äÿžäžĘåčřæŸŐäÿĂäÿłæŰřçŽĎçżŞæđĎä¡Şïij ŇåŔłéIJĂèęĄåČŔè£Źæăůå ĹŻåżžäÿĂäÿł `Struct` åőđä¿Ńå■şåŔŕïijŽ

```
# Little endian 32-bit integer, two double precision floats
record_struct = Struct('<idd')
```

çżŞæđĎä¡ŞéĂŽåÿÿäijŽä¡£çŤÍäÿĂäžŻçżŞæđĎçăĄåĂiji, d, fç■L' [åŔĆèĂČ PythonæŰĞæąč]ãĂĆ è£ŹäžŻäžčçăĄåĹĘåĹńäžčèąÍæ§ŘäÿłçL'źåőŽçŽĎäžŇè£ŻåĹűæŢřæ■őçśżåđŃåęĆ32ä¡■a çňňäÿĂäÿłå■Ůçňę < æŇĞåőŽäžEå■ŮèŁĆéąžåžŔãĂĆåIJÍè£ŹäÿłäżŃå■Řäÿ■ïijŇåőČèąÍçđ'žâĂÍä¡Őä¡■åIJÍåL'■ æŻť'æŤźè£Źäÿłå■Ůçňęäÿž > èąÍçđ'žéńŸä¡■åIJÍåL'■ïijŇæĹŰèĂĚæŸŕ ! èąÍçđ'žç¡ŚçżIJå■ŮèŁĆéąžåžŔãĂĆ

äžğçŤ§çŽĎ `Struct` åőđä¿ŃæIJL'å¿Ĺåđ'Žåśđæ ĂğåŠŇæŰźæşŢçŤÍæÍěæŞ■ä¡IJçŻÿåžŤçśżåđŃçŽĎçżŞæđ `size` åśđæĂğåŇĚåŘńäžEçżŞæđĎçŽĎå■ŮèŁĆæŢřïijŇè£ŹåIJÍI/OæŞ■ä¡IJæŮűéİđåÿÿæIJL'çŤÍãĂĆ `pack()` åŠŇ `unpack()` æŰźæşŢèćńçŤÍæÍěæL'ŞåŇĚåŠŇèğčåŇĚæŢřæ■õãĂĆæŕŤåęĆïijŽ

```
>>> from struct import Struct
>>> record_struct = Struct('<idd')
>>> record_struct.size
20
>>> record_struct.pack(1, 2.0, 3.0)
b
 ↪'\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00@\x00\x00\x00\x00\x00\x00\x08@
 ↪'
>>> record_struct.unpack(_)
(1, 2.0, 3.0)
>>>
```

æIJL'æŮűåĂŹä¡àè£ŸäijŽçIJŃåĹř `pack()` åŠŇ `unpack()` æŞ■ä¡IJäžěæÍąåİŮçžğåĹńåĞ¡æŢřèćńěřČçŤÍïijŇçśžäijijäÿŃéİćè£ŹæăůïijŽ

```
>>> import struct
>>> struct.pack('<idd', 1, 2.0, 3.0)
b
 ↪'\x01\x00\x00\x00\x00\x00\x00\x00\x00\x00@\x00\x00\x00\x00\x00\x00\x08@
 ↪'
>>> struct.unpack('<idd', _)
(1, 2.0, 3.0)
>>>
```

è£ŹæăůåŔŕäžěåůěä¡IJïijŇä¡ĘæŸŕæĎ§èğL'æšąæIJL'åőđä¿ŃæŰźæşŢéĆčäźĹäijŸéŻĚïijŇçL'źåĹńæŸŕåIJÍà éĂŽè£ĞåĹŻåžžäÿĂäÿł `Struct` åőđä¿ŃïijŇæäijåijŔäžčçăĄåŔłäijŽæŇĞåőŽäÿĂäěňąåźűäÿŤæL'ĂæIJL'çŽĎæ è£ŹæăůäÿĂæİěäžčçăĄçżť'æŁď'åŕśåŔŸå¿ŮæŻť'åŁăçőĂå■ŢäžĘ(åŽääÿžä¡ăåŔłéIJĂèęĄæŤźåŔŸäÿĂåď'ĎäžčçăĄ

èŕžåŔŰäžŇè£ŻåĹűçžŞæđĎçŽĎäžčçăĄèęĄçŤÍåĹřäÿĂäžŻéİđåÿÿæIJL'èűčèĂŇäijŸç¿ŐçŽĎçijŰçÍŃæŁĂ

åIJÍåĞ¡æŢřãĂĂread_recordsäÿ■ïijÑiter()èćńçŢÍæİěåĹŻåżžäÿĂäÿłè£ŤåŻđåŻžåőŽåď ğåřŔæŢřæ■őå
è£ŹäÿłèŁ■äžčåŹÍäijŽäÿ■æŰ■çŽĎěřČçŢÍäÿĂäÿłçŢÍæĹůæŔŘä¿ŻçŽĎåŔŕèřČçŢÍåŕźèśą(æŕŤåęĆ
lambda: f.read(record_struct.size))ïijÑ çŻť åĹřåőČè£ŤåŻđäÿĂäÿłçL'źæőŁçŽĎåĂij(åęĆbâ

```
>>> f = open('data.b', 'rb')
>>> chunks = iter(lambda: f.read(20), b'')
>>> chunks
<callable_iterator object at 0x10069e6d0>
>>> for chk in chunks:
... print(chk)
...
b'\x01\x00\x00\x00ffffff\x02@\x00\x00\x00\x00\x00\x00\x12@'
b'\x06\x00\x00\x00333333\x1f@\x00\x00\x00\x00\x00\x00"@'
b'\x0c\x00\x00\x00\xcd\xcc\xcc\xcc\xcc\xcc*@\x9a\x99\x99\x99\x99YL@'
>>>
```

åęĆä¡ăæL'ĂèğĄïijÑåĹŻåżžäÿĂäÿłåŔŕè£■äžčåŕźèśąçŽĎäÿĂäÿłåŐ§åŻăæŸŕåőČèČ¡åĚĄèőÿä¡£çŤÍäÿĂäÿ
åęĆæđIJä¡ăäÿ■ä¡£çŤÍè£ŹçğæŁĂæIJŕïijÑéĆčäźĹäžčçăĄåŔŕèČ¡äijŽåČŔäÿÑéİćè£ŹæăůïijŽ

```
def read_records(format, f):
    record_struct = Struct(format)
    while True:
        chk = f.read(record_struct.size)
        if chk == b'':
            break
        yield record_struct.unpack(chk)
```

åIJÍåĞ¡æŢř unpack_records() äÿ■ä¡£çŤÍäžĘåŔęåď ŰäÿĂçğ■æŰźæşŢ
unpack_from() ãĂĆunpack_from() åŕźäžŐäżŐäÿĂäÿłåď ğåđÑäžŇè£ŻåĹűæŢřçŽäÿ■æŔŘåŔŰäžŇè
åŽăäÿžåőČäÿ■äijŽäžğçŢ§äżżä¡ŢçŽĎäÿť æŮűåŕźèśąæĹŰèĂĚè£ŻèąŇåĘĚå■Ÿåď ■åĹűæŞ■ä¡IJãĂĆ
ä¡ăåŘłéIJĂèęĄçżŹåőČäÿĂäÿłå■ŮèŁĆå■Ůçňęäÿš(æĹŰæŢřçžĎ)åŠŇäÿĂäÿłå■ŮèŁĆåĄŔçğżéĞŕïijÑåőČäijŽ

åęĆæđIJä¡ăä¡£çŤÍ unpack() æİěäžčæŻ£ unpack_from() ïijÑ
ä¡ăéIJĂèęĄä£őæŤźäżčçăĄæİěæđĎéĂăåď ğéĞŔçŽĎåřŔçŽĎåĹĞçL'Ğäžěåŕŀè£ŻèąŇåĄŔçğżéĞŔçŽĎèőąçőŮ

```
def unpack_records(format, data):
    record_struct = Struct(format)
    return (record_struct.unpack(data[offset:offset + record_struct.
↪size])
            for offset in range(0, len(data), record_struct.size))
```

è£Źçğ■æŰźæąĹéŹď'äžĘäżčçăĄçIJÑäÿŁåŐżå¿Ĺåď ■æİĆåď ŰïijÑè£Ÿå¿ŮåĄŽå¿Ĺåď ŽéćÍåď ŰçŽĎåůěä¡
åď ■åĹűæŢřæ■őäžěåŔŁædĎéĂăåřŔçŽĎåĹĞçL'ĞåŕźèśąãĆ åęĆæđIJä¡ăåĞĘåď ĞäżŐèŕżåŔŰåĹŕçŽĎäÿĂäÿłå
äijŽèąĺçŐřçŽĎæŻť åĞžèL'šãĂĆ

åIJÍèğčåŇĚçŽĎæŮűåĂŹïijÑcollections æĺąåİŮäÿ■çŽĎåŚ¡åŘ■åĚČçżĎåŕźèśąæĹŰèőÿæŸŕä¡ăæČşè
åőČåŔŕäžěèől'ä¡ăçżŹè£ŤåŻđåĚČçżĎèő¿ç¡őåśđæĂğåŘ■çğřãĆä¿ŃåęĆïijŽ

```
from collections import namedtuple

Record = namedtuple('Record', ['kind','x','y'])
```

```
with open('data.p', 'rb') as f:
    records = (Record(*r) for r in read_records('<idd', f))

for r in records:
    print(r.kind, r.x, r.y)
```

åęĆædIJä¡ăçŽĎçÍŃåžŔéIJĂèęĄåď'ĎçŘĘåď'ğéĞŔçŽĎäžŇè£ŻåĹűæŢřæ■őïijŇä¡ăæIJĂåě¡ä¡£çŤÍ
numpy æÍąåİŮãĂĆ ä¿ŃåęĆïijŇä¡ăåŔŕäžěårĘäÿĂäÿłäžŇè£ŻåĹűæŢřæ■őèŕżåŔŰåĹřäÿĂäÿłçżŞædĎåŇŰæŢřçż

```
>>> import numpy as np
>>> f = open('data.b', 'rb')
>>> records = np.fromfile(f, dtype='<i,<d,<d')
>>> records
array([(1, 2.3, 4.5), (6, 7.8, 9.0), (12, 13.4, 56.7)],
dtype=[('f0', '<i4'), ('f1', '<f8'), ('f2', '<f8')])
>>> records[0]
(1, 2.3, 4.5)
>>> records[1]
(6, 7.8, 9.0)
>>>
```

æIJĂåŘŐæŔŘäÿĂçĆźïijŇåęĆædIJä¡ăéIJĂèęĄäżŐåůšç§ěçŽĎæŰĞäżűæăijåijŔ(åęĆåŻ¿çL'ĞæăijåijŕïijŇ
åĚĹæčĂæ§ěçIJŃçIJŃPythonæŸŕäÿ■æŸŕåůšçżŔæŔŘä¿ŻäžEçŐŕå■ŸçŽĎæĺąåİŮãĂĆåŻăäÿžäÿ■åĹřäÿĞäÿ■å¿

8.12 6.12 èŕżåŔŰåţŇåěŮåŠŇåŔŕåŔŸéŢ£äžŇè£ŻåĹűæŢřæ■ő

éŮőécŸ

ä¡ăéIJĂèęĄèŕżåŔŰåŇĚåŘńåţŇåěŮæĹŰèĂĚåŔŕåŔŸéŢ£èőŕå¡ŢéŻĘåŘĹçŽĎåď'■æİĆäžŇè£ŻåĹűæăijåijŔ

èğčåĘşæŰźæąĹ

struct æÍąåİŮåŔŕèćńçŤĺæİěçijŰçăĄ/èğčçăĄåĞääźŐæL'ĂæIJL'çśżåđŃçŽĎäžŇè£ŻåĹűçŽĎæŢřæ■őçż
æİěèąÍçď'žäÿĂäÿłçżĎæĹŘäÿĂçşżåĹŮåď'Žè¿źå¡ćçŽĎçĆźçŽĎéŻĘåŘĹïijŽ

```
polys = [
    [ (1.0, 2.5), (3.5, 4.0), (2.5, 1.5) ],
    [ (7.0, 1.2), (5.1, 3.0), (0.5, 7.5), (0.8, 9.0) ],
    [ (3.4, 6.3), (1.2, 0.5), (4.6, 9.2) ],
]
```

çŐřåIJĺåĄĞèő¿è£ŹäÿłæŢřæ■őèćńçijŰçăĄåĹřäÿĂäÿłäżěäÿŇåĹŮåď't'éČĺåijĂåğŇçŽĎäžŇè£ŻåĹűæŰĞäż

```
+------+--------+-----------------------------------+
|Byte  | Type   |  Description                      |
+======+========+===================================+
|0     | int    |  æŰĞäżűäżčçăĄïijĹ0x1234ïijŇåřŔçńŕïijL'          |
```

```
+------+--------+-------------------------------------+
|4     | double |  x çŽĎæIJĂåřŔåĂijïijĹåřŔçńŕïijL’                 |
+------+--------+-------------------------------------+
|12    | double |  y çŽĎæIJĂåřŔåĂijïijĹåřŔçńŕïijL’                 |
+------+--------+-------------------------------------+
|20    | double |  x çŽĎæIJĂåd’ğåĂijïijĹåřŔçńŕïijL’                 |
+------+--------+-------------------------------------+
|28    | double |  y çŽĎæIJĂåd’ğåĂijïijĹåřŔçńŕïijL’                 |
+------+--------+-------------------------------------+
|36    | int    |  äÿL’èğŠå¡ćæŢřéĞŔïijĹåřŔçńŕïijL’                 |
+------+--------+-------------------------------------+
```

çť ğèů§çİĂåď ť éČĺæŸŕäÿĂçşżåĹŮçŽĎåď Žè¿źå¡ćèőřå¡ŢïijŇçijŰçăĄæăijåijŔåęĆäÿŃïijŽ

```
+------+--------+-------------------------------------------+
|Byte  | Type   |  Description                              |
+======+========+===========================================+
|0     | int    |  èőřå¡ŢéŢ£åžęïijĹNå■ŮèŁĆïijL’                 ␣
↪  |
+------+--------+-------------------------------------------+
|4-N   | Points |  (X,Y) åİŘæăĞïijŇäżěæţőçĆźæŢřèąĺçd’ž                ␣
↪   |
+------+--------+-------------------------------------------+
```

äÿžäžĘåĘŹè£ŹæăůçŽĎæŰĞäżűïijŇä¡ăåŔŕäżěä¡£çŤĺåęĆäÿŃçŽĎPythonäżčçăĄïijŽ

```python
import struct
import itertools

def write_polys(filename, polys):
    # Determine bounding box
    flattened = list(itertools.chain(*polys))
    min_x = min(x for x, y in flattened)
    max_x = max(x for x, y in flattened)
    min_y = min(y for x, y in flattened)
    max_y = max(y for x, y in flattened)
    with open(filename, 'wb') as f:
        f.write(struct.pack('<iddddi', 0x1234,
                            min_x, min_y,
                            max_x, max_y,
                            len(polys)))
        for poly in polys:
            size = len(poly) * struct.calcsize('<dd')
            f.write(struct.pack('<i', size + 4))
            for pt in poly:
                f.write(struct.pack('<dd', *pt))
```

åřĘæŢřæ■őèŕżåŔŰåŻđæİěçŽĎæŮűåĂŹïijŇåŔŕäżěåĹľçŤĺåĞ¡æŢř struct.unpack() ïijŇäżčçăĄå¿ĹçŻÿäijijïijŇå§žæIJňåřśæŸŕäÿŁéİćåĘŹæŞ■ä¡IJçŽĎéĂĘåžŔãĂĆåęĆäÿŃïijŽ

```python
def read_polys(filename):
    with open(filename, 'rb') as f:
        # Read the header
        header = f.read(40)
        file_code, min_x, min_y, max_x, max_y, num_polys = \
            struct.unpack('<iddddi', header)
        polys = []
        for n in range(num_polys):
            pbytes, = struct.unpack('<i', f.read(4))
            poly = []
            for m in range(pbytes // 16):
                pt = struct.unpack('<dd', f.read(16))
                poly.append(pt)
            polys.append(poly)
    return polys
```

åř¡çőąè£ŹäÿłäżčçăĄåŔŕäżěåůěä¡IJïijŇä¡ĘæŸŕéĞŇéİćæůůæİĆäžĘå¿Ĺåď ŽèŕżåŔŰãĂĄèğčåŇĚæŢřæ■őçż
éĆčæIJłåĚ■äź§åď łçźĄæİĆäžĘçĆźãĂĆåŻăæ■ď å¿ĹæŸ¿çĎűåžŤèŕěæIJĽåŔęäÿĂçğ■èğčåĘşæŰźæşŢåŔŕäżěçő

åIJĺæIJňåřŔèŁĆæŐěäÿŃæİěçŽĎéČĺåĹĘïijŇæĹŚäijŽéĂŘæ■ěæijŤçď žäÿĂäÿłæŻť åŁăäijŸçğĂçŽĎèğčæ
çŻőæăĞæŸŕåŔŕäżěçżŹçĺŃåžŔåŚŸæŔŘä¿ŻäÿĂäÿłéńŸçžğçŽĎæŰĞäżűæăijåijŔåŇŰæŰźæşŢïijŇåźűçőĂåŇŰ
æIJňåřŔèŁĆæŐěäÿŃæİěçŽĎéČĺåĹĘäżčçăĄåžŤèŕěæŸŕæŢť æIJňäźęäÿ■æIJĂåď ■æİĆæIJĂéńŸçžğçŽĎä¿Ńå■
äÿĂåőŽèęĄäżŤçżĘçŽĎéŸĚèŕżæĹŚäżňçŽĎèőĺèőžéČĺåĹĘïijŇåŔęåď Űäź§èęĄåŔĆèĂČäÿŃåĚűäżŰçńăèŁĆåĘ

éęŰåĚĹïijŇå¡ŞèŕżåŔŰå■ŮèŁĆæŢřæ■őçŽĎæŮűåĂŹïijŇéĂŽåÿÿåIJĺæŰĞäżűåijĂåğŃéČĺåĹĘäijŽåŇĚåŘ
åř¡çőąstructæĺąåİŮåŔŕäżěèğčåŇĚè£ŹäžŻæŢřæ■őåĹřäÿĂäÿłåĚČçżĎäÿ■åŐżïijŇåŔęåď ŰäÿĂçğ■èąĺçď žè£Źçğ
åřśåČŔäÿŃéİćè£ŹæăůïijŽ

```python
import struct

class StructField:
    '''
    Descriptor representing a simple structure field
    '''
    def __init__(self, format, offset):
        self.format = format
        self.offset = offset
    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            r = struct.unpack_from(self.format, instance._buffer,␣
↪self.offset)
            return r[0] if len(r) == 1 else r

class Structure:
    def __init__(self, bytedata):
        self._buffer = memoryview(bytedata)
```

è£ŹéĞŇæĹŚäżňä¡£çŤĺäžĘäÿĂäÿłæŔŔè£řåŹĺæİěèąĺçď žæŕŔäÿłçżŞæđĎå■ŮæőţïijŇæŕŔäÿłæŔŔè£řåŹĺåŇ
å■ŸåĆĺåIJĺåĘĚéČĺçŽĎåĘĚå■ŸçijŞåĘšäÿ■ãĂĆåIJĺ `__get__()` æŰźæşŢäÿ■ïijŇ`struct`.

unpack_from() åĞ¡æŢřèćńçŤĺæİěäżŐçijŞåĘšäÿ■èğčåŇĚäÿĂäÿłåĂijïijŇçIJĄåŐżäžĘéćİåď˙ŰçŽĎåĹĘçL’Č

Structure çśżåřśæŸŕäÿĂäÿłå§žçąĂçśżïijŇæŐěåŔŮå■ŮèŁĆæŢřæ■őåźűå■ŸåĆĺåIJĺåĘĚéČĺçŽĎåĘĚå■ StructField æŔŔè£řåŹĺä¡£çŤĺãĂĆ è£ŹéĞŇä¡£çŤĺäžĘ memoryview() ïijŇæĹŚäżñäijŽåIJĺåŘŐéİćèřęçżĘèőšèğčåőČæŸŕçŤĺæİěåźšåŸŽçŽĎãĂĆ

ä¡£çŤĺè£ŹäÿłäżčçăĄïijŇä¡ăçŐřåIJĺåřśèČ¡åőŽäźĽäÿĂäÿłénŸåśĆæňąçŽĎçżŞæđĎåŕźèśąæİěèąĺçď˙žäÿŁéİ

```
class PolyHeader(Structure):
    file_code = StructField('<i', 0)
    min_x = StructField('<d', 4)
    min_y = StructField('<d', 12)
    max_x = StructField('<d', 20)
    max_y = StructField('<d', 28)
    num_polys = StructField('<i', 36)
```

äÿŃéİćçŽĎä¿Ńå■ŘåĹl’çŤĺè£ŹäÿłçśżæİěèŕżåŔŰäźŃåĽ’■æĹŚäżňåĘŹåĚěçŽĎåď˙Žè¿źå¡ćæŢřæ■őçŽĎåď˙t’

```
>>> f = open('polys.bin', 'rb')
>>> phead = PolyHeader(f.read(40))
>>> phead.file_code == 0x1234
True
>>> phead.min_x
0.5
>>> phead.min_y
0.5
>>> phead.max_x
7.0
>>> phead.max_y
9.2
>>> phead.num_polys
3
>>>
```

è£Źäÿłå¿ĹæIJL’èűčïijŇäÿ■è£Ğè£ŹçğæŰźåijŔè£ŸæŸŕæIJL’äÿĂäžŻçČęäžžçŽĎåIJřæŰźãĂĆéęŰåĚĹïijŇä¡ĘæŸŕè£ŹäÿłäżčçăĄè£ŸæŸŕæIJL’çĆźèĞČèĆ£ïijŇè£ŸéIJĂèęĄä¡£çŤĺèĂĚæŇĞåőŽå¿Ĺåď˙ŽåžŢåśĆçŽĎçżĘ StructFieldïijŇæŇĞåőŽåĄŔçğžéĞŘç■L’)ãĂĆ åŔęåď˙ŰïijŇè£ŤåŻđçŽĎçżŞæđIJçśżåŘŇæăůçąőåđđäÿĂä

äżżä¡ŢæŮűåĂŹåŔłèęĄä¡ăéĄĞåĹřäžĘåČŔè£ŹæăůåĘŮä¡ŹçŽĎçśżåőŽäźĽïijŇä¡ăåžŤèŕěèĂČèŹŚäÿŃä¡£ç åĚČçśżæIJL’äÿĂäÿłçL’źæĂğåŕśæŸŕåőČèČ¡åď§ècńçŤĺæİěåąńåĚĚèőÿåď˙Žä¡ŐåśĆçŽĎåőđçŐřçżĘèŁĆïijŇäžŐ äÿŃéİćæĹŚæİěäÿ¿äÿłä¿Ńå■ŘïijŇä¡£çŤĺåĚČçśżçĺ■å¿őæŤźéĂääÿŃæĹŚäžńçŽĎ Structure çśżïijŽ

```
class StructureMeta(type):
    '''
    Metaclass that automatically creates StructField descriptors
    '''
    def __init__(self, clsname, bases, clsdict):
        fields = getattr(self, '_fields_', [])
        byte_order = ''
        offset = 0
        for format, fieldname in fields:
```

```
            if format.startswith(('<','>','!','@')):
                byte_order = format[0]
                format = format[1:]
            format = byte_order + format
            setattr(self, fieldname, StructField(format, offset))
            offset += struct.calcsize(format)
        setattr(self, 'struct_size', offset)

class Structure(metaclass=StructureMeta):
    def __init__(self, bytedata):
        self._buffer = bytedata

    @classmethod
    def from_file(cls, f):
        return cls(f.read(cls.struct_size))
```

ä¡£çŤĺæŰřçŽĎ `Structure` çśžïijŇä¡ååŔŕäžěåČŔäÿŃéİćè£ŹæăůåőŽäźL’äÿĂäÿłçżŞæđĎïijŽ

```
class PolyHeader(Structure):
    _fields_ = [
        ('<i', 'file_code'),
        ('d', 'min_x'),
        ('d', 'min_y'),
        ('d', 'max_x'),
        ('d', 'max_y'),
        ('i', 'num_polys')
    ]
```

æ■čåęĆä¡ăæL’ĂèğĄïijŇè£ŹæăůåĘŹåřśçőĂå■Ţåd’ŽäžĘãĂĆæĹŚäžňæůżåŁăçŽĎçśżæŰźæşŢ `from_file()` èőľ’æĹŚäžňåIJĺäÿ■éIJĂèęĄç§ěéĄŞäžżä¡ŢæŢřæ■őçŽĎåd’ğářŔåŠŇçżŞæđĎçŽĎæČĚåĘţäÿŃă

```
>>> f = open('polys.bin', 'rb')
>>> phead = PolyHeader.from_file(f)
>>> phead.file_code == 0x1234
True
>>> phead.min_x
0.5
>>> phead.min_y
0.5
>>> phead.max_x
7.0
>>> phead.max_y
9.2
>>> phead.num_polys
3
>>>
```

äÿĂæŮęä¡ăåijĂåğŃä¡£çŤĺäžĘåĚČçśżïijŇä¡ăåřśåŔŕäžěèőľ’åőČåŔŸå¿ŮæŻt’åŁăæŹžèČ¡ãĂĆä¿ŃåęĆïijŇå
äÿŃéİćæŸŕåŕźåL’■éİćåĚČçśżçŽĎäÿĂäÿłåřŔçŽĎæŤźè£ŻïijŇæŔŘä¿ŻäžĘäÿĂäÿłæŰřçŽĎè¿ĚåŁl’æŔŔè£řåŹĺa

```
class NestedStruct:
    '''
    Descriptor representing a nested structure
    '''
    def __init__(self, name, struct_type, offset):
        self.name = name
        self.struct_type = struct_type
        self.offset = offset

    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            data = instance._buffer[self.offset:
                                    self.offset+self.struct_type.struct_
↪size]
            result = self.struct_type(data)
            # Save resulting structure back on instance to avoid
            # further recomputation of this step
            setattr(instance, self.name, result)
            return result

class StructureMeta(type):
    '''
    Metaclass that automatically creates StructField descriptors
    '''
    def __init__(self, clsname, bases, clsdict):
        fields = getattr(self, '_fields_', [])
        byte_order = ''
        offset = 0
        for format, fieldname in fields:
            if isinstance(format, StructureMeta):
                setattr(self, fieldname,
                        NestedStruct(fieldname, format, offset))
                offset += format.struct_size
            else:
                if format.startswith(('<','>','!','@')):
                    byte_order = format[0]
                    format = format[1:]
                format = byte_order + format
                setattr(self, fieldname, StructField(format,␣
↪offset))
                offset += struct.calcsize(format)
        setattr(self, 'struct_size', offset)
```

åIJÍè£ŹæőţäžčçăĄäÿ■ïijŇNestedStruct æŔŔè£řåŹÍèćńçŤÍæÌěåŔäåŁäåŔęåd'ŰäÿĂäÿłåőŽäźL'åIJÍæ§
åőČéĂŽè£ĞåřĘåŐ§åğŃåĘĚå■ŸçijŞåĘšè£ŻèąŇåĹĞçL'ĞæŞ■ä¡IJåŘŐåőđä¿ŃåŇŰçżŹåőŽçŽĎçżŞæđĎçśžåđŇ
æL'Ääžěè£Źçğ■åĹĞçL'ĞæŞ■ä¡IJäÿ■äijŽåijŢåŔŚäżżä¡ŢçŽĎéćÌåd'ŰçŽĎåĘĚå■Ÿåd'■åĹűãĂĆçŻÿåŔ■ïijŇåőČ
åŔęåd'ŰïijŇäÿžäžĘéŸšæ■ćéĞ■åd'■åőđä¿ŃåŇŰïijŇéĂŽè£Ğä¡£çŤÍåŠŇ8.10åřŔèŁĆåŘŇæăůçŽĎæŁĂæIJřïijŇ

ä¡£çŤÍè£ŹäÿłæŰřçŽĎä£őæ■čçL'ĹïijŇä¡ăářšåŔřäžěåČŔäÿŃéÍćè£ŹæăůçijŰåĘŹïijŽ

```
class Point(Structure):
    _fields_ = [
        ('<d', 'x'),
        ('d', 'y')
    ]

class PolyHeader(Structure):
    _fields_ = [
        ('<i', 'file_code'),
        (Point, 'min'), # nested struct
        (Point, 'max'), # nested struct
        ('i', 'num_polys')
    ]
```

äžd'äžžæČŁèőűçŽĎæŸŕřijŇåőČäź§èČ¡æŇL'çĚğéćĎæIJ§çŽĎæ■čåÿÿåůëä¡IJïijŇæĹŚäžňåőđéŹĚæŞ■ä¡IJ

```
>>> f = open('polys.bin', 'rb')
>>> phead = PolyHeader.from_file(f)
>>> phead.file_code == 0x1234
True
>>> phead.min # Nested structure
<__main__.Point object at 0x1006a48d0>
>>> phead.min.x
0.5
>>> phead.min.y
0.5
>>> phead.max.x
7.0
>>> phead.max.y
9.2
>>> phead.num_polys
3
>>>
```

åĹřçŽőåL'■äÿžæ■ćïijŇäÿĂäÿłåď'ĎçŘĘåőŽéŢ£èőřå¡ŢçŽĎæąĘæđűåůšçżŔåĘŹåě¡äžĘãĂĆä¡ĘæŸŕåęĆæđI
æŕŤåęĆïijŇåď'Žè¿źå¡ćæŰĞäžűåŇĚåŘńåŔŸéŢ£çŽĎéČĺåĹĘãĂĆ

äÿĂçğ■æŰźæąĹæŸŕåĘŹäÿĂäÿłçśżæİěèąÍçď'žå■ŮèŁĆæŢřæ■őïijŇåŘŇæŮűåĘŹäÿĂäÿłåůěåĚůåĞ¡æŢřæ

```
class SizedRecord:
    def __init__(self, bytedata):
        self._buffer = memoryview(bytedata)

    @classmethod
    def from_file(cls, f, size_fmt, includes_size=True):
        sz_nbytes = struct.calcsize(size_fmt)
        sz_bytes = f.read(sz_nbytes)
        sz, = struct.unpack(size_fmt, sz_bytes)
        buf = f.read(sz - includes_size * sz_nbytes)
        return cls(buf)
```

```
    def iter_as(self, code):
        if isinstance(code, str):
            s = struct.Struct(code)
            for off in range(0, len(self._buffer), s.size):
                yield s.unpack_from(self._buffer, off)
        elif isinstance(code, StructureMeta):
            size = code.struct_size
            for off in range(0, len(self._buffer), size):
                data = self._buffer[off:off+size]
                yield code(data)
```

çśżæŰźæşŢ `SizedRecord.from_file()` æŸŕäÿĂäÿłåůěåĚůïijŇçŤĺæİěäżŐäÿĂäÿłæŰĞäžűäÿ■èŕżå
è£Źäź§æŸŕå¿Ĺåď˙ŽæŰĞäžűæäijåijŔåÿÿçŤĺçŽĎæŰźåijŔãĂĆä¡IJäÿžè¿ŞåĚěïijŇåőČæŐěåŔŮäÿĂäÿłåŇĚåŘńå
åŔŕéĂĽçŽĎ `includes_size` åŔĆæŢřæŇĞåőŽäžĘå■ŮèŁĆæŢřæŸŕåŘęåŇĚåŘńåď˙ťéČĺåď˙ğářŔãĂĆ
äÿŇéİćæŸŕäÿĂäÿłä¿Ńå■ŘæŢŹä¡äæĂŐæăůä¡£çŤĺäžŐåď˙Žè¿źå¡ćæŰĞäžűäÿ■èŕżåŔŰå■ŢçŇńçŽĎåď˙Žè¿źå¡ća

```
>>> f = open('polys.bin', 'rb')
>>> phead = PolyHeader.from_file(f)
>>> phead.num_polys
3
>>> polydata = [ SizedRecord.from_file(f, '<i')
...              for n in range(phead.num_polys) ]
>>> polydata
[<__main__.SizedRecord object at 0x1006a4d50>,
<__main__.SizedRecord object at 0x1006a4f50>,
<__main__.SizedRecord object at 0x10070da90>]
>>>
```

åŔŕäżěçIJŃåĞžïijŇ`SizedRecord`åőđä¿ŃçŽĎåĘĚåőźè£ŸæšąæIJL’èćńèğčæđŘåĞžæİěãĂĆ
åŔŕäżěä¡£çŤĺ `iter_as()` æŰźæşŢæİěè¿¿åĹřçŻőçŽĎïijŇè£ŹäÿłæŰźæşŢæŐěåŔŮäÿĂäÿłçżŞæđĎæăijåijŔål
`Structure` çśżä¡IJäÿžè¿ŞåĚěãĂĆ è£ŹæăůåŘåŔŕäżěå¿ĹçĄţæť˙žçŽĎåŐżèğčæđŘæŢřæ■őïijŇä¿ŃåęĆïijŽ

```
>>> for n, poly in enumerate(polydata):
...     print('Polygon', n)
...     for p in poly.iter_as('<dd'):
...         print(p)
...
Polygon 0
(1.0, 2.5)
(3.5, 4.0)
(2.5, 1.5)
Polygon 1
(7.0, 1.2)
(5.1, 3.0)
(0.5, 7.5)
(0.8, 9.0)
Polygon 2
(3.4, 6.3)
(1.2, 0.5)
(4.6, 9.2)
```

```
>>>

>>> for n, poly in enumerate(polydata):
...     print('Polygon', n)
...     for p in poly.iter_as(Point):
...         print(p.x, p.y)
...
Polygon 0
1.0 2.5
3.5 4.0
2.5 1.5
Polygon 1
7.0 1.2
5.1 3.0
0.5 7.5
0.8 9.0
Polygon 2
3.4 6.3
1.2 0.5
4.6 9.2
>>>
```

åřĘæL’ĂæIJL’è£ŹäžŻçżŞåŘĹèţůæİěïijŇäÿŃéİćæŸŕäÿĂäÿł `read_polys()` åĞ¡æŢřçŽĎåŔęåď’ŰäÿĂäÿłä£őæ■čçL’Ĺïij Ž

```
class Point(Structure):
    _fields_ = [
        ('<d', 'x'),
        ('d', 'y')
    ]

class PolyHeader(Structure):
    _fields_ = [
        ('<i', 'file_code'),
        (Point, 'min'),
        (Point, 'max'),
        ('i', 'num_polys')
    ]

def read_polys(filename):
    polys = []
    with open(filename, 'rb') as f:
        phead = PolyHeader.from_file(f)
        for n in range(phead.num_polys):
            rec = SizedRecord.from_file(f, '<i')
            poly = [ (p.x, p.y) for p in rec.iter_as(Point) ]
            polys.append(poly)
    return polys
```

èőÍèőž

è£ŹäÿĂèŁĆåŘŚä¡ăáśŢçd’žäžĘèőÿåd’ŽéńŸçžğçŽĎçijŰçÍŃæŁĂæIJŕïijŇåŇĚæŃňæŔŔè£řåŹÍïijŇåżűè£§ë
çĎűèĂŇïijŇåőČäžňéČ¡äÿžäžĘåŘŇäÿĂäÿłçĽźåőŽçŽĎçŽőæăĞæIJ■åŁąãĂĆ

äÿŁéİćçŽĎåőđçŐřçŽĎäÿĂäÿłäÿżèęĄçĽźå¿ĄæŸřåőČæŸřå§žäžŐæĞŠèğčåŇĚçŽĎæĂİæČşãĂĆå¡ŞäÿĂä
`Structure` åőđä¿ŃèćńåĹŻåžžæŮűïijŇ `__init__()` äžĚäžĚåŔłæŸřåĹŻåžžäÿĂäÿłå■ŮèŁĆæŢřæ■őçŽĎå
çĽźåĹńçŽĎïijŇè£ŹæŰűåĂŹåźűæšąæIJL’äżżä¡ŢçŽĎèğčåŇĚæĹŰèĂĚåĚűäžŰäÿŐçżŞæđĎçŻÿåĚşçŽĎæŞ■ä¡
è£ŹæăůåĄŽçŽĎäÿĂäÿłåŁĺæIJżæŸřä¡ăåŔřèČ¡äžĚäžĚåŔłåŕźäÿĂäÿłå■ŮèŁĆèőřå¡ŢçŽĎæ§ŘäÿĂářŔéČĺåĹĘæ

äÿžäžĘåőđçŐřæĞŠèğčåŇĚåŠŇæĽ’ŞåŇĚïijŇéIJĂèęĄä¡£çŤÍ
`StructField` æŔŔè£řåŹÍçśżãĂĆ çŤÍæĹůåIJÍ `_fields_`
äÿ■åĹŮåĞžæİèçŽĎæŕŔäÿłåśđæĂğéČ¡äijŽèćńè¡ňåŇŰæĹŘäÿĂäÿł `StructField`
æŔŔè£řåŹÍïijŇ åőČářĘçŻÿåĚşçżŞæđĎæăijåijŔçăĄåŠŇåĄŘçğżåĂijä£Ía■ŸåĹřå■ŸåĆÍçijŞå■Ÿäÿ■ãĆåĚČçś
`StructureMeta` åIJÍåď’ŽäÿłçżŞæđĎçśżèćńåőŽäźĽ’æŮűèĞłåŁÍåĹŽåžžäžĘè£ŹäžŽæŔŔè£řåŹÍãĂĆ
æĹŚäžňä¡£çŤÍåĚČçśżçŽĎäÿĂäÿłäÿżèęĄåŐ§åŽăæŸřåőČä¡£å¿ŮçŤÍæĹůéİđåÿÿæŰźä¿£çŽĎéĂŽè£ĞäÿĂäÿłér

`StructureMeta` çŽĎäÿĂäÿłå¿Ĺå¿őåęŹçŽĎåIJřæŰźářśæŸřåőČäijŽåŽžåőŽå■ŮèŁĆæŢřæ■őéąžåžŔãĂ
äź§ářśæŸřèřť’ïijŇåęĆæđIJäżżæĎŔçŽĎåśđæĂğæŇĞåőŽäžĘäÿĂäÿłå■ŮèŁĆéąžåžŔ(<èąĺçd’žä¡Öä¡■äijŸåĚĹ
æĹŰèĂĚ >èąĺçd’žéńŸä¡■äijŸåĚĹ)ïijŇ éĆčåŘŐéİćæĽ’ĂæIJL’å■Ůæőţç ŽĎéąžåžŔéČ¡äžèè£ŹäÿłéąžåžŔäÿžåĞE
æŕŤåęĆïijŇä¡ăåŔřèČ¡æIJL’äÿĂäžŽæŕŤè¿Čåď’■æİĆçŽĎçżŞæđĎïijŇářśåČŔäÿŇéİćè£ŹæăůïijŽ

```
class ShapeFile(Structure):
    _fields_ = [ ('>i', 'file_code'), # Big endian
         ('20s', 'unused'),
         ('i', 'file_length'),
         ('<i', 'version'), # Little endian
         ('i', 'shape_type'),
         ('d', 'min_x'),
         ('d', 'min_y'),
         ('d', 'max_x'),
         ('d', 'max_y'),
         ('d', 'min_z'),
         ('d', 'max_z'),
         ('d', 'min_m'),
         ('d', 'max_m') ]
```

äźŃåĽ’■æĹŚäžňæŔŔåĹřè£ĞïijŇmemoryview() çŽĎä¡£çŤÍåŔŕäžěåÿőåŁľæĹŚäžňéĄ£åĚ■åĘĚå■ŸçŽĎ
å¡ŞçżŞæđĎå■ŸåIJÍåţŇåěŮçŽĎæŮűåĂŹïijŇmemoryviews åŔŕäžěåŔăåŁăåŘŇäÿĂäĘĚå■ŸåŇžå§§äÿŁåőŽä
è£ŹäÿłçĽźæĂğæŕŤè¿Čå¿őåęŹïijŇä¡ĘæŸřåőČåĚşæşÍçŽĎæŸřåĘĚå■ŸègĘåŻ¿äÿŐæŹőéĂŽå■ŮèŁĆæŢřçżĎç
åęĆæđIJä¡ååIJÍäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšæĹŰå■ŮèŁĆæŢřçżĎäÿŁæĽ’ğèąŇåĹĞçĽ’ĞæŞ■ä¡IJïijŇä¡ăéĂŽåÿÿä
èĂŇåĘĚå■ŸègĘåŻ¿åĹĞçĽ’Ğäÿ■æŸřè£ŹæăůçŽĎïijŇåőČäžĚäžĚæŸřåIJÍåůšå■ŸåIJÍçŽĎåĘĚå■ŸäÿŁéİćåŔăă

è£ŸæIJL’å¿Ĺåď’ŽçŻÿåĚşçŽĎçńăèŁĆåŔŕäžěåÿőåŁľæĹŚäžňæL’ľåśŢè£ŹéĞŇèőÍèőžçŽĎæŰźæąĹãĂĆ
åŔĆèĂČ8.13åřŔèŁĆä¡£çŤÍæŔŔè£řåŹÍædĎåžžäÿĂäÿłçśżåđŃçşżçż§ãĂĆ
8.10åřŔèŁĆæIJL’æŻť’åď’ŽåĚşäžŐåžűè£§èőąçőŮåśđæĂğåĂijçŽĎèőÍèőžïijŇåźűäÿŤèů§NestedStructæŔŔè£řå
9.19åřŔèŁĆæIJL’äÿĂäÿłä¡£çŤÍåĚČçśżæİěåĹİågŃåŇŰçśżæĹŘåŚŸçŽĎä¿Ńå■ŘïijŇåŠŇ
`StructureMeta` çśźéİđåÿÿçŻÿäijijãĂĆ PythonçŽĎ `ctypes`
æžŘçăĄåŘŇæăůäź§å¿ĹæIJL’èűčïijŇåőČæŔŔä¿ŻäžĘåŕźåőŽäźĽ’æŢřæ■őçżŞæđĎãĂĄæŢřæ■őçżŞæđĎåţŇåěŮ

8.13 6.13 æŢřæ■őçŽĎçťŕåŁääÿŐçż§èőąæŞ■ä¡IJ

éŮőécŸ

ä¡ăéIJĂèęĄåď ĎçŘĘäÿĂäÿłå¿Ĺåď ğçŽĎæŢřæ■őéŻĘåźűéIJĂèęĄèőąçŐŮæŢřæ■őæĂżåŠŇæĹŰåĚűäżŰçż

èğčåĘşæŰźæąĹ

åŕźäžŐäżżä¡Ţæű L’åŘŁåĹřçż§èőąãĂĄæŮűéŮť’åžŔåĹŮäżěåŘŁåĚűäżŰçŻÿåĚşæŁĂæIJŕçŽĎæŢřæ■őåĹĘ
PandasåžŞ ãĂĆ

äÿžäžĘèőľä¡ăåĚĹä¡ŞéłŇäÿŃïijŇäÿŃéİćæŸŕäÿĂäÿłä¡£çŤÍPandasæİěåĹĘædŘèŁİåŁăåŞěå§ŐåÿĆçŽĎ
èĂĄéijåăŠŇåŢőé¡£çśżåŁÍçĽľ’æŢřæ■őåžŞ çŽĎä¿Ńå■ŘãĂĆ åIJĺæĹŠåĘŹè£ŹçŕĞæŰĞçńăçŽĎæŮűåĂŹïijŇè£Ź

```
>>> import pandas

>>> # Read a CSV file, skipping last line
>>> rats = pandas.read_csv('rats.csv', skip_footer=1)
>>> rats
<class 'pandas.core.frame.DataFrame'>
Int64Index: 74055 entries, 0 to 74054
Data columns:
Creation Date 74055 non-null values
Status 74055 non-null values
Completion Date 72154 non-null values
Service Request Number 74055 non-null values
Type of Service Request 74055 non-null values
Number of Premises Baited 65804 non-null values
Number of Premises with Garbage 65600 non-null values
Number of Premises with Rats 65752 non-null values
Current Activity 66041 non-null values
Most Recent Action 66023 non-null values
Street Address 74055 non-null values
ZIP Code 73584 non-null values
X Coordinate 74043 non-null values
Y Coordinate 74043 non-null values
Ward 74044 non-null values
Police District 74044 non-null values
Community Area 74044 non-null values
Latitude 74043 non-null values
Longitude 74043 non-null values
Location 74043 non-null values
dtypes: float64(11), object(9)

>>> # Investigate range of values for a certain field
>>> rats['Current Activity'].unique()
array([nan, Dispatch Crew, Request Sanitation Inspector],␣
 ↪dtype=object)
>>> # Filter the data
```

```
>>> crew_dispatched = rats[rats['Current Activity'] == 'Dispatch␣
↪Crew']
>>> len(crew_dispatched)
65676
>>>

>>> # Find 10 most rat-infested ZIP codes in Chicago
>>> crew_dispatched['ZIP Code'].value_counts()[:10]
60647 3837
60618 3530
60614 3284
60629 3251
60636 2801
60657 2465
60641 2238
60609 2206
60651 2152
60632 2071
>>>

>>> # Group by completion date
>>> dates = crew_dispatched.groupby('Completion Date')
<pandas.core.groupby.DataFrameGroupBy object at 0x10d0a2a10>
>>> len(dates)
472
>>>

>>> # Determine counts on each day
>>> date_counts = dates.size()
>>> date_counts[0:10]
Completion Date
01/03/2011 4
01/03/2012 125
01/04/2011 54
01/04/2012 38
01/05/2011 78
01/05/2012 100
01/06/2011 100
01/06/2012 58
01/07/2011 1
01/09/2012 12
>>>

>>> # Sort the counts
>>> date_counts.sort()
>>> date_counts[-10:]
Completion Date
10/12/2012 313
10/21/2011 314
09/20/2011 316
```

```
10/26/2011 319
02/22/2011 325
10/26/2012 333
03/17/2011 336
10/13/2011 378
10/14/2011 391
10/07/2011 457
>>>
```

åŮŕïijŇçIJŃæăůå■Ř2011åźť10æIJĹ7æŮěåŕźèĂĄéijääžňæİěèŕť'æŸŕäÿłå¿Ĺå£ŹçćŇçŽĎæŮěå■ŘåŢŁïijĄ

èőĺèőž

PandasæŸŕäÿĂäÿłæŃěæIJL'å¿Ĺåď'ŽçL'źæĂğçŽĎåď'ğåđŃåĞ¡æŢřåžŞïijŇæĹŚåIJĺè£ŹéĞŇäÿ■åŔŕèČ¡äżŃ
ä¡ĘæŸŕåŔłèęĄä¡ăéIJĂèęĄåŐżåĹĘæđŘåď'ğåđŃæŢřæ■őéŽĘåŘĹãĂĄåŕźæŢřæ■őåĹĘçżĎãĂĄèőąçŐŮåŘĎçğ■

9 çňňäÿČçńäïijŽåĞ¡æŢř

ä¡£çŤĺ def èŕ■åŘěåőŽäźL'åĞ¡æŢřæŸŕæL'ĂæIJL'çĺŃåžŔçŽĎå§žçąĂãĂĆ
æIJňçńäçŽĎçŻőæăĞæŸŕèőšèğčäÿĂäžŽæŻť'åŁăéńŸçžğåŠŇäÿ■åÿÿèğĄçŽĎåĞ¡æŢřåőŽäźL'äÿŐä¡£çŤĺæĺąåijŘ
æűL'åŘŁåĹřçŽĎåĘĚåőźåŇĚæŃňéżŸèőď'åŔĆæŢřãĂĄäżżæĎŔæŢřéĞŔåŔĆæŢřãĂĄåijžåĹűåĚşéŤőå■ŮåŔĆ
åŘęåď'ŰïijŇäÿĂäžŽéńŸçžğçŽĎæŐğåĹűæţĄåŠŇåĹl'çŤĺåŻđèřČåĞ¡æŢřäijăéĂŠæŢřæ■őçŽĎæŁĂæIJŕåIJĺè£Ź

Contents:

9.1 7.1 åŘŕæŐěåŘŮäżżæĎŔæŢřéĞŔåŔĆæŢřçŽĎåĞ¡æŢř

éŮőécŸ

ä¡ăæČşæđĎéĂääÿĂäÿłåŘŕæŐěåŘŮäżżæĎŔæŢřéĞŔåŔĆæŢřçŽĎåĞ¡æŢřãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡èől'äÿĂäÿłåĞ¡æŢřæŐěåŘŮäżżæĎŔæŢřéĞŔçŽĎä¡■ç¡őåŔĆæŢřïijŇåŘŕäžěä¡£çŤĺäÿĂäÿł*åŔĆ

```
def avg(first, *rest):
    return (first + sum(rest)) / (1 + len(rest))

# Sample use
avg(1, 2) # 1.5
avg(1, 2, 3, 4) # 2.5
```

åIJĺè£Źäÿłä¿Ńå■Řäÿ■ïijÑrestæŸŕçŤśæL'ĂæIJL'åĚűäžŰä¡■ç¡őåŔĆæŢřçżĎæĹŘçŽĎåĚČçżĎãĂĆçĎűåŘ

äÿžäžĘæŐěåŘŮäżżæĎŔæŢřéĞŔçŽĎåĚşéŤőå■ŮåŔĆæŢřïijŇä¡£çŤĺäÿĂäÿłäžě**åijĂåď't'çŽĎåŔĆæŢřã

```
import html

def make_element(name, value, **attrs):
    keyvals = [' %s="%s"' % item for item in attrs.items()]
    attr_str = ''.join(keyvals)
    element = '<{name}{attrs}>{value}</{name}>'.format(
                name=name,
                attrs=attr_str,
                value=html.escape(value))
    return element

# Example
# Creates '<item size="large" quantity="6">Albatross</item>'
make_element('item', 'Albatross', size='large', quantity=6)

# Creates '<p>&lt;spam&gt;</p>'
make_element('p', '<spam>')
```

åIJÍè£ŹéĞŇïijŇattrsæŸŕäÿĂäÿłåŇĚåŘńæĽĂæIJĽèćńäijååĚěè£ŻæİěçŽĎåĚşéŤőå■ŮåŔĆæŢřçŽĎå■ŮåĚ

åęĆæđIJä¡ăè£ŸåÿŇæIJŻæ§ŘäÿłåĞ¡æŢřèČ¡åŘŇæŮűæŐěåŔŮäżżæĎŔæŢřéĞŔçŽĎä¡■ç¡őåŔĆæŢřåŠŇål

```
def anyargs(*args, **kwargs):
    print(args) # A tuple
    print(kwargs) # A dict
```

ä¡£çŤÍè£ŹäÿłåĞ¡æŢřæŮűïijŇæĽĂæIJĽä¡■ç¡őåŔĆæŢřäijŽèćńæŤ¿åĹřargsåĚČçżĎäÿ■ïijŇæĽĂæIJĽåĚş

èőÍèőž

äÿĂäÿł*åŔĆæŢřåŔłèČ¡åĞžçŐřåIJÍåĞ¡æŢřåőŽäźĽäÿ■æIJĂåŘŐäÿĂäÿłä¡■ç¡őåŔĆæŢřåŘŐéİćïijŇèĂŇ
**åŔĆæŢřåŔłèČ¡åĞžçŐřåIJÍæIJĂåŘŐäÿĂäÿłåŔĆæŢřãĂĆ æIJĽäÿĂçĆźèęĄæşÍæĎŔçŽĎæŸŕïijŇåIJÍ*åŔĆæ

```
def a(x, *args, y):
    pass

def b(x, *args, y, **kwargs):
    pass
```

è£Źçğ■åŔĆæŢřåřśæŸŕæĹŚäžňæĽĂèŕť'çŽĎåijžåĹűåĚşéŤőå■ŮåŔĆæŢřïijŇåIJÍåŘŐéİć7.2åřŔèŁĆè£Ÿäij

9.2 7.2 åŘłæŐěåŔŮåĚşéŤőå■ŮåŔĆæŢřçŽĎåĞ¡æŢř

éŮőécŸ

ä¡ăåÿŇæIJŻåĞ¡æŢřçŽĎæ§ŘäžŻåŔĆæŢřåijžåĹűä¡£çŤĺåĚşéŤőå■ŮåŔĆæŢřäijåéĂŠ

èğčåĘşæŰźæąĹ

åřĘåijžåĹűåĚşéŤőå■ŮåŔĆæŢřæŤ¿åĹřæ§Řäÿł*åŔĆæŢřæĹŰèĂĚå■Ţäÿł*åŘŐéİćåřśèČ¡è¿¿åĹřè£Źçğ■æ

```
def recv(maxsize, *, block):
    'Receives a message'
    pass

recv(1024, True) # TypeError
recv(1024, block=True) # Ok
```

åĹľçŤĺè£Źçğ■æŁĂæIJřïijŇæĹŚäžňè£ŸèČ¡åIJÍæŐěåŘŮäżżæĎŔåď˙Žäÿłä¡■ç¡őåŔĆæŢřçŽĎåĞ¡æŢřäÿ■æ

```
def mininum(*values, clip=None):
    m = min(values)
    if clip is not None:
        m = clip if clip > m else m
    return m

minimum(1, 5, 2, -5, 10) # Returns -5
minimum(1, 5, 2, -5, 10, clip=0) # Returns 0
```

èŐÍèŐž

å¿Ĺåď˙ŽæČĚåĘţäÿŃïijŇä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŔĆæŢřäijŽæŕŤä¡£çŤĺä¡■ç¡őåŔĆæŢřèąÍæĎŔæŻť˙åŁă
ä¿ŇåęĆïijŇèĂČèŹŚäÿŇåęĆäÿŇäÿĂäÿłåĞ¡æŢřèřČçŤĺïijŽ

```
msg = recv(1024, False)
```

åęĆæđIJèřČçŤĺèĂĚårźrecvåĞ¡æŢřåźűäÿ■æŸŕå¿ĹçĘ§æĆL˙ïijŇéĆčäžŰèĆŕåőŽäÿ■æŸŐçŹ¡éĆčäÿłFalseå
ä¡ĘæŸŕïijŇåęĆæđIJäžčçăĄåŘŸæĹŘäÿŃéİćè£Źæăůå■ŘçŽĎèŕİåřśæÿĚæěŽåď˙ŽäžĘïijŽ

```
msg = recv(1024, block=False)
```

åŘęåď˙ŰïijŇä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŔĆæŢřäź§äijŽæŕŤä¡£çŤĺ**kwargsåŔĆæŢřæŻť˙åě¡ïijŇåŻääÿžåIJ

```
>>> help(recv)
Help on function recv in module __main__:
recv(maxsize, *, block)
    Receives a message
```

åijžåĹűåĚşéŤőå■ŮåŔĆæŢřåIJÍäÿĂäžŽæŻť˙éńŸçžğåIJžåŘĹåŘŇæăůäź§å¿ĹæIJL˙çŤĺãĂĆ
ä¿ŇåęĆïijŇåőČäžňåŔŕäžěèćńçŤĺæİěåIJÍä¡£çŤĺ*argsåŠŇ**kwargsåŔĆæŢřä¡IJäÿžè¿ŞåĚěçŽĎåĞ¡æŢřäÿ■æŘŠ

9.3 7.3 çżŹåĞ¡æŢřåŔĆæŢřåćđåŁääĚČä£qæĄŕ

éŮőécŸ

ä¡ääĘŹåě¡äžĘäÿĂäÿłåĞ¡æŢřijŇçĎűåŘŐæČşäÿžè£ŹäÿłåĞ¡æŢřçŽĎåŔĆæŢřåćđåŁääÿĂäžŻécİåď˙ŰçŽĎ

èğčåĘşæŰźæąĹ

ä¡£çŤĺåĞ¡æŢřåŔĆæŢřæşĺèğčæŸŕäÿĂäÿłå¿Ĺåě¡çŽĎåŁđæşŢijŇåőČèČ¡æŔŘçď˙žçÍŇåžŔåŚŸåžŤèŕěæĂŐ
ä¿ŇåęĆijŇäÿŇéİćæIJL˙äÿĂäÿłècńæşĺèğčäžĘçŽĎåĞ¡æŢřijŽ

```
def add(x:int, y:int) -> int:
    return x + y
```

pythonèğčéĞŁåŹĺäÿ■äijŽåŕźè£ŹäžŻæşĺèğčæůżåŁääżżä¡ŢçŽĎèŕ■äźL˙ãĂĆåőČäżñäÿ■äijŽèćńçśżåđŇæčĂ
çĎűèĂŇijŇåŕźäžŐéĆčäžŻéŸĚèŕżæžŘçăĄçŽĎäžżæİěèőšåřšå¿ĹæIJL˙åÿőåŁľ˙åŢęãĂĆçňñäÿL˙æŰźåůěåĚůåŠŇ

```
>>> help(add)
Help on function add in module __main__:
add(x: int, y: int) -> int
>>>
```

åř¡çőąä¡ääŔŕäžëä¡£çŤĺäżżæĎŘçśżåđŇçŽĎåŕźèśąçżŹåĞ¡æŢřæůżåŁääşĺèğč(ä¿ŇåęĆæŢřå■ŮïijŇå■Ůçňęä

èőĺèőž

åĞ¡æŢřæşĺèğčåŔłå■ŸåĆĺåIJĺåĞ¡æŢřçŽĎ `__annotations__`
åšđæĂğäÿ■ãĂĆä¿ŇåęĆijŽ

```
>>> add.__annotations__
{'y': <class 'int'>, 'return': <class 'int'>, 'x': <class 'int'>}
```

åř¡çőąæşĺèğčçŽĎä¡£çŤĺæŰźæşŢåŔŕèČ¡æIJL˙å¿Ĺåď˙Žçğ■ïijŇä¡ĘæŸŕåőČäżňçŽĎäÿżèęĄçŤĺéĂŤè£ŸæŸŕ
åŻääÿžpythonåźűæšąæIJL˙çśżåđŇåčřæŸŐïijŇéĂŽåÿÿæİěèőšäžĚäžĚéĂŽè£ĞéŸĚèŕżæžŘçăĄå¿ĹéŽ¿ç§ěéĄŞå
è£ŹæŮűåĂŹä¡£çŤĺæşĺèğčåřśèČ¡çżŹçĺŇåžŔåŚŸæŻť˙åď˙ŽçŽĎæŔŘçď˙žïijŇèőľ˙äžŰäžňåŔŕäžëæ■čçąőçŽĎä¡£ç

åŔĆèĂČ9.20åřŘèŁĆçŽĎäÿĂäÿłæŻť˙åŁäéńŸçžğçŽĎä¿Ňå■ŘïijŇæijŤçď˙žäžĘåęĆä¡ŢåĹľ˙çŤĺæşĺèğčæİěåć

9.4 7.4 è£ŤåŻđåď˙ŽäÿłåĂijçŽĎåĞ¡æŢř

éŮőécŸ

ä¡ääÿŇæIJŻæđĎéĂääÿĂäÿłåŔŕäžëè£ŤåŻđåď˙ŽäÿłåĂijçŽĎåĞ¡æŢř

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡è£ŤåŻđåď'ŽäÿłåĂijïijŇåĞ¡æŢřçŻt'æŐěreturnäÿĂäÿłåĚČçżĎářśèąŇäžĘãĂĆä¿ŃåęĆïijŽ

```
>>> def myfun():
... return 1, 2, 3
...
>>> a, b, c = myfun()
>>> a
1
>>> b
2
>>> c
3
```

èŐĺèŐž

åř¡çőąmyfun()çIJŃäÿŁåŐżè£ŤåŻđäžĘåď'ŽäÿłåĂijïijŇåőđéŹĚäÿŁæŸŕåĚĹåĹŻåżžäžĘäÿĂäÿłåĚČçżĎçĎ
è£Źäÿłèń■æşŢçIJŃäÿŁåŐżæŕŤè¿ČåěĞæĂłïijŇåőđéŹĚäÿŁæĹŚäžňä¡£çŤĺçŽĎæŸŕéĂŮåŔůæİěçŤ§æĹŘäÿĂäÿ

```
>>> a = (1, 2) # With parentheses
>>> a
(1, 2)
>>> b = 1, 2 # Without parentheses
>>> b
(1, 2)
>>>
```

å¡ŞæĹŚäžňèřČçŤĺè£ŤåŻđäÿĂäÿłåĚČçżĎçŽĎåĞ¡æŢřçŽĎæŮűåĂŹ
ïijŇéĂŽåÿÿæĹŚäžňäijŽåřĘçżŞæđIJèţŃåĂijçżŹåď'ŽäÿłåŔŸéĞŔïijŇåřśåČŔäÿŁéİćçŽĎéĆčæăůãĂĆ
åĚűåőđè£ŹåřśæŸŕ1.1åřŔèŁĆäÿ■æĹŚäžňæĽĂèŕť'çŽĎåĚČçżĎèğčåŇĚãĂĆè£ŤåŻđçżŞæđIJäź§åŔŕäžěèţŃåĂij
è£ŹæŮűåĂŹè£ŹäÿłåŔŸéĞŔåĂijåřśæŸŕåĞ¡æŢřè£ŤåŻđçŽĎéĆčäÿłåĚČçżĎæIJňèžńäžĘïijŽ

```
>>> x = myfun()
>>> x
(1, 2, 3)
>>>
```

9.5 7.5 åőŽäźĽælJĽéżŸèőď'åŔĆæŢřçŽĎåĞ¡æŢř

éŮőécŸ

ä¡ăæČşåőŽäźĽäÿĂäÿłåĞ¡æŢřæĹŰèĂĚæŰźæşŢïijŇåőČçŽĎäÿĂäÿłæĹŰåď'ŽäÿłåŔĆæŢřæŸŕåŔŕéĂĽ'çŽ

èğčåĘşæŰźæąĹ

åőŽäźĽäÿĂäÿłæIJĽåŔŕéĂĽåŔĆæŢřçŽĎåĞ¡æŢřæŸŕéİđåÿÿçőĂå■ŢçŽĎïijŇçŻť'æŐěåIJĺåĞ¡æŢřåőŽäźĽ

```
def spam(a, b=42):
    print(a, b)

spam(1) # Ok. a=1, b=42
spam(1, 2) # Ok. a=1, b=2
```

åęĆæđIJéžŸèőď'åŔĆæŢřæŸŕäÿĂäÿłåŔŕä£őæŤźçŽĎåőźåŹĺæŕŤåęĆäÿĂäÿłåĹŮèąĺãĂĄéŻĘåŘĹæĹŰèĂĚ

```
# Using a list as a default value
def spam(a, b=None):
    if b is None:
        b = []
    ...
```

åęĆæđIJäj¡ăåźűäÿ■æČşæŔŘä¿ŻäÿĂäÿłéžŸèőď'åĂijïijŇèĂŇæŸŕæČşäżĚäżĚæţŃèŕŢäÿŃæ§ŘäÿłéžŸèőď'å

```
_no_value = object()

def spam(a, b=_no_value):
    if b is _no_value:
        print('No b value supplied')
    ...
```

æĹŚäžňæţŃèŕŢäÿŃè£ŹäÿłåĞ¡æŢřijŽ

```
>>> spam(1)
No b value supplied
>>> spam(1, 2) # b = 2
>>> spam(1, None) # b = None
>>>
```

äżŤçżĘèğĆåŕ§åŔŕäžěåŔŚçŐřåĹřäijăéĂŠäÿĂäÿłNoneåĂijåŠŇäÿ■äijăåĂijäÿd'çğ■æČĚåĘţæŸŕæIJĽåůőå

èőÍèőž

åőŽäźĽåÿęéžŸèőď'åĂijåŔĆæŢřçŽĎåĞ¡æŢřæŸŕå¿ĹçőĂå■ŢçŽĎïijŇä¡ĘçżÍäÿ■äžĚäżĚåŔłæŸŕè£ŹäÿłïijŇ

éęŰåĚĹïijŇéžŸèőď'åŔĆæŢřçŽĎåĂijäžĚäżĚåIJĺåĞ¡æŢřåőŽäźĽçŽĎæŮűåĂŹèţŃåĂijäÿĂæňąãĂĆèŕŢçİ

```
>>> x = 42
>>> def spam(a, b=x):
...     print(a, b)
...
>>> spam(1)
1 42
>>> x = 23 # Has no effect
```

```
>>> spam(1)
1 42
>>>
```

æşÍæĎŔåĹřå¡ŞæĹŚäžňæŤźåŔŸxçŽĎåĂijçŽĎæŮűåĂŹåŕźéżŸèőď'åŔĆæŢřåĂijåźűæšąæIJL'å¡śåŞ■ïijŇè

åĚűæňąïijŇéżŸèőď'åŔĆæŢřçŽĎåĂijåžŤèŕěæŸŕäÿ■åŔŕåŔŸçŽĎåŕźèśąïijŇæŕŤåęĆNoneãĂĄTrueãĂĄFal
çL'źåĹńçŽĎïijŇå■ČäÿĞäÿ■èęĄåČŔäÿŃéİćè£ŹæăůåEŹäžčçăĄïijŽ

```
def spam(a, b=[]): # NO!
    ...
```

åęĆæđIJä¡ăè£ŹäźĹåĄŽäžĘïijŇå¡ŞéżŸèőď'åĂijåIJÍåĚűäżŮåIJřæŰźèćńä£őæŤźåŘŐä¡ăářĘäijŽéĄĞåĹřåŘ

```
>>> def spam(a, b=[]):
...     print(b)
...     return b
...
>>> x = spam(1)
>>> x
[]
>>> x.append(99)
>>> x.append('Yow!')
>>> x
[99, 'Yow!']
>>> spam(1) # Modified list gets returned!
[99, 'Yow!']
>>>
```

è£Źçğ■çżŞæđIJåžŤèŕěäÿ■æŸŕä¡ăæČşèęĄçŽĎãĂĆäÿžäžĘéĄ£åĚ■è£Źçğ■æČĚåĘţçŽĎåŔŞçŤ§ïijŇæIJĂå
çĎůåŘŐåIJÍåĞ¡æŢřéĞŇéİćæčĂæ§ěåőČïijŇåL'■éİćçŽĎä¿Ńå■ŘåřśæŸŕè£ŹæăůåĄŽçŽĎãĂĆ

åIJÍæţŃèŕŢNoneåĂijæŮűä¡£çŤÍ is æŞ■ä¡IJçňęæŸŕå¿ĹéĞ■èęĄçŽĎïijŇäź§æŸŕè£Źçğ■æŰźæąĹçŽĎåĚş
æIJL'æŮűåĂŹåđ'ğåőűäijŽçŁŕäÿŃäÿŃéİćè£ŹæăůçŽĎéŤŹèŕŕïijŽ

```
def spam(a, b=None):
    if not b: # NO! Use 'b is None' instead
        b = []
    ...
```

è£ŹäźĹåĘŹçŽĎéŮőécŸåIJÍäžŐår¡çőąNoneåĂijçąőåőđæŸŕècńå¡ŞæĹŘFalseïijŇ
ä¡ĘæŸŕè£ŸæIJL'åĚűäżŰçŽĎårźèśą(æŕŤåęĆéŢ£åžęäÿž0çŽĎå■Ůçňęäÿšã ĂĄåĹÙèąÍãĂĄåĚČçżĎãĂĄå■ŮåĚÿ
åŽăæ■d'ïijŇäÿŁéİćçŽĎäžčçăĄäijŽèŕŕářĘäÿĂäžŻåĚűäżŰè¿ŞåĚěäź§å¡ŞæĹŘæŸŕæšąæIJL'è¿ŞåĚěãĆæŕŤåęĆ

```
>>> spam(1) # OK
>>> x = []
>>> spam(1, x) # Silent error. x value overwritten by default
>>> spam(1, 0) # Silent error. 0 ignored
>>> spam(1, '') # Silent error. '' ignored
>>>
```

æIJÄåŘŐäÿĂäÿłéŮőécŸæŕŤè¿Čå¿őåęŹïijŇéĆčářśæŸŕäÿĂäÿłåĞ¡æŢřéIJÄèęĄæţŃèŕŢæ§ŘäÿłåŔŕéĂĽ'åŔ
è£ŹæŮűåĂŹéIJÄèęĄåřŔå£ČçŽĎæŸŕä¡ääÿ■èČ¡çŤĺæ§ŘäÿłéżŸèőď'åĂijæŕŤåęĆNoneãĂĄ
0æĹŰèĂĚFalseåĂijæĭěæţŃèŕŢçŤĺæĹůæŔŘä¿ŻçŽĎåĂij(åŽääÿžè£ŹäžŽåĂijéČ¡æŸŕåŘĹæşŢçŽĎåĂijïijŇæŸ
åŽäæ■d'ïijŇä¡äéIJÄèęĄåĚűäžŰçŽĎèğčåEşæŰźæąĹäžĘãĂĆ

äÿžäžĘèğčåEşè£ŹäÿłéŮőécŸïijŇä¡ääŔŕäżěåĹŽåžžäÿĂäÿłçŃňäÿĂæŮääžŇçŽĎçğĄæIJL'åŕźèśąåőđä¿Ńïij
åIJĺåĞ¡æŢřéĞŇéİćïijŇä¡ääŔŕäżěéĂŽè£ĞæčÄæ§ěèćńäijäéĂŠåŔĆæŢřåĂijèů§è£Źäÿłåőđä¿ŃæŸŕåĘäÿĂæăůä
è£ŹéĞŇçŽĎæĂİèůŕæŸŕçŤĺæĹůäÿ■åŔŕèČ¡åŐżäijäéĂŠè£ŹäÿŁ_no_valueåőđä¿Ńä¡IJäÿžè¿ŞåĚěãĂĆ
åŽäæ■d'ïijŇè£ŹéĞŇéĂŽè£ĞæčÄæ§ěè£ŹäÿłåĂijåŕśèČ¡çąőåőŽæ§ŘäÿłåŔĆæŢřæŸŕåĘèćńäijäéĂŠè£ŽæİěäžĘ

è£ŹéĞŇåŕź `object()` çŽĎä¡£çŤĺçIJŃäÿŁåŐżæIJL'çĆźäÿ■åd'łåÿÿèğĄãĂĆ`object`
æŸŕpythonäÿ■æL'ÄæIJL'çśżçŽĎå§žçśżãĂĆ ä¡ääŔŕäżěåĹŽåžž `object`
çśżçŽĎåőđä¿ŃïijŇä¡ĘæŸŕè£ŹäžŽåőđä¿ŃæšąäžÄäźĹåőđéŹĚçŤĺåd'ĎïijŇåŽääÿžåőČåźűæšąæIJL'äżżä¡ŢæIJL'
äź§æšąæIJL'äżżä¡Ţåőđä¿ŃæŢræ■ő(åŽääÿžåőČæšąæIJL'äżżä¡ŢçŽĎåőđä¿Ńå■ŮåĚÿïijŇä¡äçŤŽèĞşéČ¡äÿ■èČ¡è
ä¡ääŤŕäÿĂèČ¡åĄŽçŽĎåŕśæŸŕæţŃèŕŢåŘŇäÿĂæĂğãĂĆè£ŹäÿłåĹŽåě¡çňęåŔĹæĹŚçŽĎèęĄæśĆïijŇåŽääÿžæĹ

9.6 7.6 åőŽäźĽåŇ£åŘ■æĹŰåĘĚèĄŤåĞ¡æŢř

éŮőécŸ

ä¡äæČşäÿž `sort()` æŞ■ä¡IJåĹŽåžžäÿĂäÿłå¿Łç§■çŽĎåŽđèŘČåĞ¡æŢřïijŇä¡ĘåŔĹäÿ■æČşçŤĺ
`def` åŐżåĘŹäÿĂäÿłå■ŢèąŇåĞ¡æŢřïijŃ èĂŇæŸŕåÿŇæIJŽéĂŽè£Ğæ§ŘäÿłåŁńæ■ůæŰźåijŔäžěåĘĚèĄŤæŰźåi

èğčåEşæŰźæąĹ

å¡ŞäÿĂäžŽåĞ¡æŢřå¿ŁçőĂå■ŢïijŇäżĚäžĚåŔłæŸŕèőąçőŮäÿĂäÿłèąĺè¿¿åijŔçŽĎåĂijçŽĎæŮűåĂŹïijŇårś

```
>>> add = lambda x, y: x + y
>>> add(2,3)
5
>>> add('hello', 'world')
'helloworld'
>>>
```

è£ŹéĞŇä¡£çŤĺçŽĎlambdaèąĺè¿¿åijŔèů§äÿŇéİćçŽĎæŢĹæđIJæŸŕäÿĂæăůçŽĎïijŽ

```
>>> def add(x, y):
...     return x + y
...
>>> add(2,3)
5
>>>
```

lambdaèąĺè¿¿åijŔåĚÿåđŇçŽĎä¡£çŤĺåIJžæŹŕæŸŕæŐŠåžŔæĹŰæŢræ■őreduceç■L'ïijŽ

```
>>> names = ['David Beazley', 'Brian Jones',
...          'Raymond Hettinger', 'Ned Batchelder']
>>> sorted(names, key=lambda name: name.split()[-1].lower())
```

```
['Ned Batchelder', 'David Beazley', 'Raymond Hettinger', 'Brian␣
↪Jones']
>>>
```

èőÍèőž

åř¡çőąlambdaèąÍè¿¿åijŔåĚĄèőÿä¡ăåőŽäźL’çőĂå■ŢåĞ¡æŢřïijŇä¡ĘæŸŕåőČçŽĎä¡£çŤÍæŸŕæIJL’éŹŘåĹűç
ä¡ăåŔłèČ¡æŇĞåőŽå■Ţäÿłèąĺè¿¿åijŔïijŇåőČçŽĎåĂijåřśæŸŕæIJĂåŘŐçŽĎè£ŤåŽđåĂijãĂĆäź§åřśæŸŕèŕť äÿ■è
åŇĚæŃňåď’Žäÿłèŕ■åŔěãĂĄæİąäžűèąÍè¿¿åijŔãĂĄè£■äžčäžěåŔŁåijĆåÿÿåď’ĎçŘĘç■L’ç■L’ãĂĆ

ä¡ăåŔŕäžěäÿ■ä¡£çŤÍlambdaèąÍè¿¿åijŔåřśèČ¡çijŰåĘŹåď’ğéČĺåĹĘpythonäžčçăĄãĂĆ
ä¡ĘæŸŕijŇå¡ŞæIJL’äžžçijŰåĘŹåď’ğéĞŔèőąçőÙèąÍè¿¿åijŔåĂijçŽĎç§■åřŔåĞ¡æŢřæĹŰèĂĚéIJĂèęĄçŤÍæĹůa
ä¡ăåřśäijŽçIJŃåĹřlambdaèąÍè¿¿åijŔçŽĎèžńå¡śäžĘãĂĆ

9.7 7.7 åŇ£åŘ■åĞ¡æŢřæ■ŢèŐůåŔŸéĞŔåĂij

éŮőécŸ

ä¡ăçŤÍlambdaåőŽäźL’äžĘäÿĂäÿłåŇ£åŘ■åĞ¡æŢřïijŇåźűæČşåIJÍåőŽäźL’æŮűæ■ŢèŐůåĹřæ§ŘäžŽåŔŸéĞ

èğčåĘşæŰźæąĹ

åĚĹçIJŃäÿŃäÿŃéİćäžčçăĄçŽĎæŢĹæđIJïijŽ

```
>>> x = 10
>>> a = lambda y: x + y
>>> x = 20
>>> b = lambda y: x + y
>>>
```

çŐřåIJÍæĹŚéŮőä¡ăïijŇa(10)åŠŇb(10)è£ŤåŽđçŽĎçżŞæđIJæŸŕäžĂäźĹïij§åęĆæđIJä¡ăèőď’äÿžçżŞæđIJæŸ

```
>>> a(10)
30
>>> b(10)
30
>>>
```

è£ŹåĚűäÿ■çŽĎåěěåęŹåIJÍäžŐlambdaèąÍè¿¿åijŔäÿ■çŽĎxæŸŕäÿĂäÿłèĞłçŤšåŔŸéĞŔïijŇ
åIJÍè£ŘèąŇæŮűçżŚåőŽåĂijïijŇèĂŇäÿ■æŸŕåőŽäźL’æŮűářçżŚåőŽïijŇè£ŹèůšåĞ¡æŢřçŽĎéżŸèőď’åĂijåŔĆæ
åŽăæ■d’ïijŇåIJÍèřĆçŤÍè£ŹäÿłlambdaèąÍè¿¿åijŔçŽĎæŮűåĂŹïijŇxçŽĎåĂijæŸŕæL’ğèąŇæŮűçŽĎåĂijãĂĆä¿Ň

```
>>> x = 15
>>> a(10)
25
>>> x = 3
```

```
>>> a(10)
13
>>>
```

åęĆædIJä¡ăæČşèőľæ§ŘäÿłåŇ£åŘ■åĞ¡æŢřåIJĺåőŽäźĽæŮűåřśæ■ŢèŐůåĹřåĂijïijŇåŔŕäžěářĘéĆčäÿłåŔĆ

```
>>> x = 10
>>> a = lambda y, x=x: x + y
>>> x = 20
>>> b = lambda y, x=x: x + y
>>> a(10)
20
>>> b(10)
30
>>>
```

èőÍèőž

åIJÍè£ŹéĞŇåĹŮåĞžæİěçŽĎéŮőécŸæŸŕæŰřæĽŃå¿Ĺåőźæ ŸŞçŁŕçŽĎéŤŹèŕŕïijŇæIJĽäžŻæŰřæĽŃåŔŕè
ærŤåęĆïijŇéĂŽè£ĞåIJÍäÿĂäÿłå¿łçŐŕæĹŰåĹŮèąĺæŐĺåŕijäÿ■åĹŻåżžäÿĂäÿłlambdaèąĺè¿¿åijŔåĹŮèąĺïijŇåźűa

```
>>> funcs = [lambda x: x+n for n in range(5)]
>>> for f in funcs:
... print(f(0))
...
4
4
4
4
4
>>>
```

ä¡ĘæŸŕåőđéŹĚæŢĹæđIJæŸŕè£ŘèąŇæŸŕnçŽĎåĂijäÿžè£■äżčçŽĎæIJĂåŘŐäÿĂäÿłåĂijãĂĆçŐřåIJÍæĹŚă

```
>>> funcs = [lambda x, n=n: x+n for n in range(5)]
>>> for f in funcs:
... print(f(0))
...
0
1
2
3
4
>>>
```

éĂŽè£Ğä¡£çŤĺåĞ¡æŢřéżŸèőď åĂijåŔĆæŢřå¡ćåijŔïijŇlambdaåĞ¡æŢřåIJĺåőŽäźĽæŮűåřśèČ¡çżŚåőŽåĹřå

9.8 7.8 åĞŔåřŚåŔŕèřČçŤĺåŕźèśąçŽĎåŔĆæŢřäÿłæŢř

éŮőéćŸ

ä¡ăæIJL'äÿĂäÿłèćńåĚűäżŰpythonäżčçăĄä¡£çŤĺçŽĎcallableåŕźèśąïijŇåŔŕèČ¡æŸŕäÿĂäÿłåŻđèřČåĞ¡æŢřä¡ĘæŸŕåőČçŽĎåŔĆæŢřåd'łåď'ŽäžĘïijŇåŕijèĞt'èřČçŤĺæŮűåĞžéŤŹãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJéIJĂèęĄåĞŔåřŚæ§ŘäÿłåĞ¡æŢřçŽĎåŔĆæŢřäÿłæŢřïijŇä¡ăåŔŕäżěä¡£çŤĺ `functools.partial()` ãĂĆ `partial()` åĞ¡æŢřåĚĄèőÿä¡ăçżŹäÿĂäÿłæĹŰåd'ŽäÿłåŔĆæŢřèő¿ç¡őåŻžäÿžäžĘæijŤçd'žæÿĚæěŽïijŇåĄĞèő¿ä¡ăæIJL'äÿŃéİćè£ŹæăůçŽĎåĞ¡æŢřïijŽ

```
def spam(a, b, c, d):
    print(a, b, c, d)
```

çŐřåIJĺæĹŚäżňä¡£çŤĺ `partial()` åĞ¡æŢřæİěåŻžåőŽæ§ŘäžŻåŔĆæŢřåĂijïijŽ

```
>>> from functools import partial
>>> s1 = partial(spam, 1) # a = 1
>>> s1(2, 3, 4)
1 2 3 4
>>> s1(4, 5, 6)
1 4 5 6
>>> s2 = partial(spam, d=42) # d = 42
>>> s2(1, 2, 3)
1 2 3 42
>>> s2(4, 5, 5)
4 5 5 42
>>> s3 = partial(spam, 1, 2, d=42) # a = 1, b = 2, d = 42
>>> s3(3)
1 2 3 42
>>> s3(4)
1 2 4 42
>>> s3(5)
1 2 5 42
>>>
```

åŔŕäżěçIJŃåĞž `partial()` åŻžåőŽæ§ŘäžŻåŔĆæŢřåźűè£ŤåŻđäÿĂäÿłæŰřçŽĎcallableåŕźèśąãĂĆè£Źã çĎűåŘŐèů§äźŃåL'■åůšçżŔèţŃåĂijè£ĞçŽĎåŔĆæŢřåŘĹåźűèţůæİěïijŇæIJĂåŘŐåřĘæL'ĂæIJL'åŔĆæŢřäijäéĂ

èőĺèőž

æIJňèŁĆèęĄèğčåĘşçŽĎéŮőéćŸæŸŕèőľ'åŐ§æIJňäÿ■åĚijåőźçŽĎäžčçăĄåŔŕäżěäÿĂèţůåůěä¡IJãĂĆäÿŃéİ

çňňäÿĂäÿłä¿Ńå■ŘæŸŕïijŇåĄĞèő¿ä¡ăæIJL'äÿĂäÿłçĆźçŽĎåĹŮèąĺæİěèąĺçd'ž(x,y)åİŘæăĞåĚĈçżĎãĂĆ ä¡ăåŔŕäżěä¡£çŤĺäÿŃéİćçŽĎåĞ¡æŢřæİěèőąçőŮäÿd'çĆźäźŃéŮt'çŽĎèůİçężïijŽ

```
points = [ (1, 2), (3, 4), (5, 6), (7, 8) ]

import math
def distance(p1, p2):
    x1, y1 = p1
    x2, y2 = p2
    return math.hypot(x2 - x1, y2 - y1)
```

çŐřåIJÍåĄĞèő¿ä¡ăæČşäžěæ§Řäył̨çĆźäÿžå§žçĆźïijŇæăźæ■őçĆźåŠŇå§žçĆźäźŃéŮťçŽĎèůİçężæİěæŐŠå
åĹŮèąÍçŽĎ `sort()` æŰźæşŢæŐěåŔŮäÿĂäÿłåĚşéŤőå■ŮåŔĆæŢřæİěèĞłåőŽäźĽæŐŠåžŔéĂżè¿ŚïijŇ
ä¡ĘæŸŕåőČåŔłèČ¡æŐěåŔŮäÿĂäÿłå■ŢäÿłåŔĆæŢřçŽĎåĞ¡æŢř(distance()å¿ĹæŸŐæŸ¿æŸŕäÿ■çňęåŘĹæİąäżű
çŐřåIJÍæĹŚäżňåŔŕäžěéĂŽè£Ğä¡£çŤÍ `partial()` æİěèğčåĘşè£ŹäÿłéŮőécŸïijŽ

```
>>> pt = (4, 3)
>>> points.sort(key=partial(distance,pt))
>>> points
[(3, 4), (1, 2), (5, 6), (7, 8)]
>>>
```

æŻť'è£ŻäÿĂæ■ěïijŇ`partial()` éĂŽåÿÿèćńçŤÍæİěå¿őèřČåĚűäžŮåžŞåĞ¡æŢřæĽĂä¡£çŤÍçŽĎåŻđèřČå
ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂæőţäžčçăĄïijŇä¡£çŤÍ `multiprocessing`
æİěåijĆæ■ěèőąçőŮäÿĂäÿłçżŞæđIJåĂijïijŇ çĎűåŘŐè£ŹäÿłåĂijèćńäijąéĂŠçżŹäÿĂäÿłæŐěåŘŮäÿĂäÿłresultåŹ

```
def output_result(result, log=None):
    if log is not None:
        log.debug('Got: %r', result)

# A sample function
def add(x, y):
    return x + y

if __name__ == '__main__':
    import logging
    from multiprocessing import Pool
    from functools import partial

    logging.basicConfig(level=logging.DEBUG)
    log = logging.getLogger('test')

    p = Pool()
    p.apply_async(add, (3, 4), callback=partial(output_result,␣
↪log=log))
    p.close()
    p.join()
```

å¡ŞçżŹ `apply_async()` æŔŘä¿ŻåŻđèřČåĞ¡æŢřæŮűïijŇéĂŽè£Ğä¡£çŤÍ
`partial()` äijąéĂŠéćİåď'ŰçŽĎ `logging` åŔĆæŢřãĂĆ èĂŇ `multiprocessing`
åŕźè£ŹäžŻäÿĂæŮăæĽĂç§ěâĂŤâĂŤåőČäżĚäżĚåŘłæŸŕä¡£çŤÍå■ŢäÿłåĂijæİěèřČçŤÍåŻđèřČåĞ¡æŢřãĂĆ

ä¡IJäÿžäÿĂäÿłçśžäijijçŽĎä¿Ńå■ŘïijŇèĂČèŹŚäÿŃçijŰåĘŹç¡ŚçżIJæIJ■åŁąåŹÍçŽĎéŮőécŸïijŇ`sockets`
æÍąåİŮèőĺ'åőČåŘŸå¿Ůå¿Ĺåőźæ ŸŞãĂĆ äÿŃéİćæŸŕäÿłçőĂå■ŢçŽĎechoæIJ■åŁąåŹÍïijŽ

```
from socketserver import StreamRequestHandler, TCPServer

class EchoHandler(StreamRequestHandler):
    def handle(self):
        for line in self.rfile:
            self.wfile.write(b'GOT:' + line)

serv = TCPServer(('', 15000), EchoHandler)
serv.serve_forever()
```

äÿ■è£ĞïijŇåĄĞèő¿ä¡ăæČşçżŹEchoHandlerråćđåŁääÿĂäÿłåŔŕäžěæŐěåŔŮåĚűäžŰéĚ■ç¡őéĂL’éąźçŽĎ `__init__` æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
class EchoHandler(StreamRequestHandler):
    # ack is added keyword-only argument. *args, **kwargs are
    # any normal parameters supplied (which are passed on)
    def __init__(self, *args, ack, **kwargs):
        self.ack = ack
        super().__init__(*args, **kwargs)

    def handle(self):
        for line in self.rfile:
            self.wfile.write(self.ack + line)
```

è£ŹäźĹä£őæŤźåŘŐïijŇæĹŚäžňåřśäÿ■éIJĂèęĄæŸ¿åijŔåIJřåIJÍTCPServerçśžäÿ■æůžåŁääL’■çijĂäžĘãĂĆä¡ĘæŸŕä¡ăåĘ■æňąè£ŘèąŇçÍŃåžŔåŘŐäijŽæŁěçśžäijijäÿŃéİćçŽĎéŤŹèŕŕïijŽ

```
Exception happened during processing of request from ('127.0.0.1',␣
↪59834)
Traceback (most recent call last):
...
TypeError: __init__() missing 1 required keyword-only argument: 'ack
↪'
```

åĹİçIJŃèţůæİěåě¡åČŔå¿ĹéŽ¿ä£őæ■čè£ŹäÿłéŤŹèŕŕïijŇéŹď äžĘä£őæŤź `socketserver` æĺąåİŮæžŘäžčçăĄæĹŰèĂĚä¡£çŤĺæ§ŘäžŽåěĞæĂłçŽĎæŰźæşŢäźŃåď ŰãĂĆä¡ĘæŸŕïijŇåęĆæđIJä¡£çŤĺ `partial()` åŕśèČ¡å¿Ĺè¡żæİ¿çŽĎèğčåĘşâĂŤâĂŤçžŹåőČäijăéĂŠ `ack` åŔĆæŢřçŽĎåÄijæİěåĹİåğŃåŇŰå■şåŔŕïijŇåęĆäÿŃïijŽ

```
from functools import partial
serv = TCPServer(('', 15000), partial(EchoHandler, ack=b'RECEIVED:
↪'))
serv.serve_forever()
```

åIJĺè£Źäÿłä¿Ńå■Řäÿ■ïijŇ`__init__()` æŰźæşŢäÿ■çŽĎack-åŔĆæŢřåčřæŸŐæŰźåijŔçIJŃäÿŁåŐżå¿ĹæIJL’èűčïijŇåĚűåőđåŕśæŸŕåčřæŸŐackäÿžäÿÄäÿłåijžåĹűåĚşéŤőå■ŮåĚşäžŐåijžåĹűåĚşéŤőå■ŮåŔĆæŢřéŰőécŸæĹŚäžňåIJĺ7.2åŕŔèŁĆæĹŚäžňåůšçžŔèőĺèőžè£ĞäžĘïijŇèŕźèĂĚåŔ

å¿Ĺåď ŽæŰűåĂŹ `partial()` èČ¡åőđçŐřçŽĎæŢĹæđIJïijŇlambdaèąĺè¿¿åijŔäź§èČ¡åőđçŐřãĂĆæŕŤåęĆ

```
points.sort(key=lambda p: distance(pt, p))
p.apply_async(add, (3, 4), callback=lambda result: output_
↪result(result,log))
serv = TCPServer(('', 15000),
        lambda *args, **kwargs: EchoHandler(*args, ack=b'RECEIVED:',
↪ **kwargs))
```

è£ŹæăůåĘŹäź§èČ¡åőđçŐřåŘŇæăůçŽĎæŢĹæđIJïijŇäÿ■è£ĞçŻÿæŕŤèĂŇåůšäijŽæŸ¿å¿ŮæŕŤè¿ČèĞČèĆ
è£ŹæŮűůåĂŹä¡£çŤĺ `partial()` åŔŕäžěæŽť˙åŁăçŽť˙èğĆçŽĎèąĺè¿¿ä¡äçŽĎæĎŔåŻ¿(çżŹæ§ŘäžŽåŔĆæŢřécĎ

9.9 7.9 åřĘå■ŢæŰźæşŢçŽĎçśżè¡ňæ■ćäÿžåĞ¡æŢř

éŮőécŸ

ä¡äæIJL˙äÿĂäÿłéŹđ˙ `__init__()` æŰźæşŢåđ˙ŰåŔłåőŽäźL˙äžĘäÿĂäÿłæŰźæşŢçŽĎçśżãĂĆäÿžäžĘçőĂ

èğčåĘşæŰźæąĹ

åđ˙ğåđ˙ŽæŢřæČĚåĘţäÿŃïijŇåŔŕäžěä¡£çŤĺéŮ■åŇĚæİěårĘå■ŢäÿłæŰźæşŢçŽĎçśżè¡ňæ■ćæĹŘåĞ¡æŢřãĂ
äÿ¿äÿłä¿Ńå■ŘïijŇäÿŃéİćçđ˙žä¿Ńäÿ■çŽĎçśżåĚĄèőÿä¡£çŤĺèĂĚæăźæ■őæ§Řäÿłæĺąæİ£æŰźæąĹæİěèŐůåŔŰå

```
from urllib.request import urlopen

class UrlTemplate:
    def __init__(self, template):
        self.template = template

    def open(self, **kwargs):
        return urlopen(self.template.format_map(kwargs))

# Example use. Download stock data from yahoo
yahoo = UrlTemplate('http://finance.yahoo.com/d/quotes.csv?s={names}
↪&f={fields}')
for line in yahoo.open(names='IBM,AAPL,FB', fields='sl1c1v'):
    print(line.decode('utf-8'))
```

è£ŹäÿłçśżåŕäžěèćńäÿĂäÿłæŽť˙çőĂå■ŢçŽĎåĞ¡æŢřæİěäžčæŻ£ïijŽ

```
def urltemplate(template):
    def opener(**kwargs):
        return urlopen(template.format_map(kwargs))
    return opener

# Example use
yahoo = urltemplate('http://finance.yahoo.com/d/quotes.csv?s={names}
↪&f={fields}')
for line in yahoo(names='IBM,AAPL,FB', fields='sl1c1v'):
    print(line.decode('utf-8'))
```

èőÍèőž

åď ğéČÍåĹĘæČĚåĘţäÿŃïijŇä¡ăæŃěæIJL’äÿĂäÿłå■ŢæŰźæşŢçśżçŽĎåŐ§åŻăæŸŕéIJĂèęĄå■ŸåĆÍæ§ŘäžæŕŤåęĆïijŇåőŽäźL’UrlTemplateçśżçŽĎåŤŕäÿĂçŻőçŽĎåřśæŸŕåĚĹåIJÍæ§Řäÿłå IJřæŰźå■ŸåĆÍæÍąæİ£åĂijïijŇ

ä¡£çŤÍäÿĂäÿłåĘĚéČÍåĞ¡æŢřæĹŰèĂĚéŮ■åŇĚçŽĎæŰźæąĹéĂŽåÿÿäijŽæŻť’äijŸéŻĚäÿĂäžŽãĂĆçőĂå■ åŘłäÿ■è£ĞåIJÍåĞ¡æŢřåĘĚéČÍåÿęäÿŁäžĘäÿĂäÿłéćİåď’ŰçŽĎåŔŸéĞŘçŐŕåćČãĂĆéŮ■åŇĚåĚşéŤőçL’źçĆźåřśå ŻĂæ■ď’ïijŇåIJÍæĹŚäžňçŽĎèğčåĘşæŰźæąĹäÿ■ïijŇopener() åĞ¡æŢřèőřä¡ŔäžĘ template åŔĆæŢřçŽĎåĂijïijŇåźűåIJÍæŐěäÿŃæİěçŽĎèřĆçŤÍäÿ■ä¡£çŤÍåőĆãĂĆ

äżżä¡ŢæŮűåĂŹåŘłèęĄä¡ăçćřåĹřéIJĂèęĄçżŹæ§Řäÿłå Ğ¡æŢřåćđåŁăéćİåď’ŰçŽĎçŁűæĂĄä£ąæĄŕçŽĎéŮ çŻÿæŕŤåřĘä¡ăçŽĎåĞ¡æŢřè¡ñæ■ćæĹŘäÿĂäÿłçśżèĂŇèÍĂïijŇéŮ■åŇĚéĂŽåÿÿæŸŕäÿĂçğ■æŻť åŁăçőĂæť ĄåŠ

9.10 7.10 åÿęécİåď’ŰçŁűæĂĄä£ąæĄŕçŽĎåŻđèřČåĞ¡æŢř

éŮőécŸ

ä¡ăçŽĎäżčçăĄäÿ■éIJĂèęĄä¿İèţŰåĹřåŻđèřČåĞ¡æŢŕçŽĎä¡£çŤÍ(æŕŤåęĆäžŃäżűåď’ĎçŘĘåŹÍãĂĄç■L’å¿Ě åźűäÿŤä¡äè£ŸéIJĂèęĄèől’åŻđèřČåĞ¡æŢřæŇěæIJL’écİåď’ŰçŽĎçŁűæĂĄåĂijïijŇäžëä¿£åIJÍåőČçŽĎåĘĚéČÍä

èğčåĘşæŰźæąĹ

è£ŹäÿĂářŔèŁĆäÿżèęĄèőÍèőžçŽĎæŸŕéĆčäžŻåĞžçŐřåIJÍå¿Ĺåď’ŽåĞ¡æŢřåžŞåŠŇæąĘæđűäÿ■çŽĎåŻđèřČ äÿžäžĘæijŤçď’žäÿŐæţŃèŕŢïijŇæĹŚäżňåĚĹåőŽäźL’åęĆäÿŇäÿĂäÿłéIJĂèęĄèřČçŤÍåŻđèřČåĞ¡æŢřçŽĎåĞ¡æŢ

```
def apply_async(func, args, *, callback):
    # Compute the result
    result = func(*args)

    # Invoke the callback with the result
    callback(result)
```

åőđéŹĚäÿŁïijŇè£Źæőţäžčçă ĄåŔŕäžěåĄŽäżżä¡ŢæŻť’éńŸçžğçŽĎåď’ĎçŘĘïijŇåŇĚæŃńçž£çÍŃãĂĄè£Żçİ æĹŚäžňäżĚäżĚåŘłéIJĂèęĄåĚşæşĺåŻđèřČåĞ¡æŢřçŽĎèřČçŤÍãĂĆäÿŃéİćæŸŕäÿĂäÿłæijŤçď’žæĂŐæăůä¡£çŤÍä

```
>>> def print_result(result):
...     print('Got:', result)
...
>>> def add(x, y):
...     return x + y
...
>>> apply_async(add, (2, 3), callback=print_result)
Got: 5
>>> apply_async(add, ('hello', 'world'), callback=print_result)
Got: helloworld
>>>
```

æşÍæĎŔåĹř `print_result()` åĞ¡æŢřäžĚäžĚåŔłæŐěåŔŮäÿĂäÿłåŔĆæŢř `result` ãĂĆäÿ■èČ¡åĘ■äijååĚěåĚűäžŰä£qæĄŕãĂĆ èĂŇå¡Şä¡ăæČşèőľ åŻđèřČåĞ¡æŢřèő£éŮőåĚűäžŰåŔŸéĞŔæĹŰè

äÿžäžĘèőľåŻđèřČåĞ¡æŢřèő£éŮőåď ŰéČĺä£qæĄřijŇäÿĂçğ■æŰźæşŢæŸŕä¡£çŤĺäÿĂäÿłçżŚáőŽæŰźæşŢ æŕŤåęĆïijŇäÿŇéÍćè£ŹäÿłçśžäijŽä£Ía■ŸäÿĂäÿłåĘĚéČĺåžŔåĹŮåŔůïijŇæŕŔæňqæŐěæŤűåĹřäÿĂäÿł `result` çŽĎæŮűåĂŹåžŔåĹŮåŔůåŁă1ïijŽ

```
class ResultHandler:

    def __init__(self):
        self.sequence = 0

    def handler(self, result):
        self.sequence += 1
        print('[{}] Got: {}'.format(self.sequence, result))
```

ä¡£çŤĺè£ŹäÿłçśžçŽĎæŮűåĂŹïijŇä¡ăåĚĹåĹŻåžžäÿĂäÿłçśžçŽĎåőđä¿ŇïijŇçĎűåŘŐçŤĺåőČçŽĎ `handler()` çżŚáőŽæŰźæşŢæÍěåĄŽäÿžåŻđèřČåĞ¡æŢřijŽ

```
>>> r = ResultHandler()
>>> apply_async(add, (2, 3), callback=r.handler)
[1] Got: 5
>>> apply_async(add, ('hello', 'world'), callback=r.handler)
[2] Got: helloworld
>>>
```

çñňäžŇçğ■æŰźåijŔïijŇä¡IJäÿžçśžçŽĎæŻ£äžčïijŇåŕŕäžěä¡£çŤĺäÿĂäÿłéŮ■åŇĚæ■ŢèŐůçŁűæĂĄåĂijïijŇ

```
def make_handler():
    sequence = 0
    def handler(result):
        nonlocal sequence
        sequence += 1
        print('[{}] Got: {}'.format(sequence, result))
    return handler
```

äÿŇéÍćæŸŕä¡£çŤĺéŮ■åŇĚæŰźåijŔçŽĎäÿĂäÿłä¿Ńå■ŘïijŽ

```
>>> handler = make_handler()
>>> apply_async(add, (2, 3), callback=handler)
[1] Got: 5
>>> apply_async(add, ('hello', 'world'), callback=handler)
[2] Got: helloworld
>>>
```

è£ŸæIJL'åŔęåď ŰäÿĂäÿłæŻť'éńŸçžğçŽĎæŰźæşŢïijŇåŕŕäžěä¡£çŤĺå■ŔçÍŃæÍěåőŇæĹŘåŘŇæăůçŽĎäžŇ

```
def make_handler():
    sequence = 0
    while True:
```

```
        result = yield
        sequence += 1
        print('[{}] Got: {}'.format(sequence, result))
```

åŕźäžŐå■ŔçÍŃïijŇä¡ăéIJĂèęĄä¡£çŤÍåőČçŽĎ send() æŰźæşŢä¡IJäÿžåŻđèřČåĞ¡æŢřïijŇåęĆäÿŃæĽĂç

```
>>> handler = make_handler()
>>> next(handler) # Advance to the yield
>>> apply_async(add, (2, 3), callback=handler.send)
[1] Got: 5
>>> apply_async(add, ('hello', 'world'), callback=handler.send)
[2] Got: helloworld
>>>
```

èőÍèőž

å§žäžŐåŻđèřČåĞ¡æŢřçŽĎè¡ŕäžűéĂŽåÿÿéČ¡æIJL'åŔŕèČ¡åŔŸå¿ŮéIđåÿÿåd'■æİĆãĂĆäÿĂéČÍåĹEåŐ§åŻ
åŻăæ■d'ïijŇèŕůæśĆæL'ğèąŇåŠŇåd'ĎçŘEçżŞæđIJäźŃéŮt'çŽĎæL'ğèąŇçŐŕáćČåőđéŹĚäÿŁåůšçżŔäÿćåd'śäžE
éĆčä¡ăåŕšå£ĚéążåŐżèğčåEşåęĆä¡Ţä£Íå■ŸåŠŇæĄćåd'■çŻÿåĚşçŽĎçŁűæĂĄä£ąæĄŕäžEąĂĆ

èĞşåřŚæIJL'äÿd'çğ■äÿżèęĄæŰźåijŔæÍěæ■ŢèŐůåŠŇä£Íå■ŸçŁűæĂĄä£ąæĄŕïijŇä¡ăåŔŕäžěåIJÍäÿĂäÿłåŕz
äÿd'çğ■æŰźåijŔçŻÿæŕŤïijŇéŮ■åŇĚæĹŰèőÿæŸŕæŻt'åŁăè¡żéĞŔçžğåŠŇèĞłçĎűäÿĂçĆźïijŇåŻääÿžåőČäżňă
åőČäżňè£ŸèČ¡èĞłåŁÍæ■ŢèŐůæL'ĂæIJL'è¢ńä¡£çŤÍåĹřçŽĎåŔŸéĞŔãĂĆåŻäæ■d'ïijŇä¡ăæŮăéIJĂåŐżæŇĚå£(

åęĆæđIJä¡£çŤÍéŮ■åŇĚïijŇä¡ăéIJĂèęĄæşÍæĎŔåŕźéĆčäžŻåŔŕä£őæŤźåŔŸéĞŔçŽĎæŞ■ä¡IJãĂĆåIJÍäÿŁé
nonlocal åčřæŸŐèŕ■åŔęçŤÍæÍěæŇĞçd'żæŐěäÿŃæÍěçŽĎåŔŸéĞŔäijŽåIJÍåŻđèřČåĞ¡æŢřäÿ■è¢ńä£őæŤźãA

èĂŇä¡£çŤÍäÿĂäÿłå■ŔçÍŃæÍěä¡IJäÿžäÿĂäÿłåŻđèřČåĞ¡æŢřåřśæŻt'æIJL'è΍čäžĘïijŇåőČèů§éŮ■åŇĚæŰźa
æ§Řçğ■æĎŔäźL'äÿŁæÍèèőšïijŇåőČæŸ¿å¿ŮæŻt'åŁăçőĂæt'ĄïijŇåŻääÿžæĂżåĚśåřśäÿĂäÿłåĞ¡æŢřèĂŇåůšãĂ
åźűäÿŤïijŇä¡ăåŔŕäžěå¿ĹèĞłçŤśçŽĎä£őæŤźåŔŸéĞŔèĂŇæŮăéIJĂåŐżä¡£çŤÍ nonlocal
åčřæŸŐãĂĆ è£Źçğ■æŰźåijŔåŤŕäÿĂçijżçĆźåřśæŸŕçŻÿåŕźäžŐåĚűäžÚPythonæŁĂæIJŕèĂŇèÍĂæĹŰèőÿæŕŤè¿
åŔęåd'Ůè£ŸæIJL'äÿĂäžŻæŕŤè¿ČéŽ¿æĞĆçŽĎéČĺåĹĘïijŇæŕŤåęĆä¡£çŤÍäźŇåL'■éIJĂèęĄèŕČçŤÍ
next() ïijŇåőđéŹĚä¡£çŤÍæŮűè£Źäÿłæ■ěéłd'å¿Ĺåőźæ ŸŞècńå£Ÿèőřã ĂĆ
år¡çőąåęĆæ■d'ïijŇå■ŔçÍŃè£ŸæIJL'åĚűäžŰçŤÍåd'ĎïijŇæŕŤåęĆä¡IJäÿžäÿĂäÿłåĘĚèĄŤåŻđèřČåĞ¡æŢřçŽĎåőŻ

åęĆæđIJä¡ăäżĚäżĚåŔłéIJĂèęĄçżŹåŻđèřČåĞ¡æŢřäijăéĂŠécÍåd'ŰçŽĎåĂijçŽĎèŕÍïijŇè£ŸæIJL'äÿĂçğ■ä
partial() çŽĎæŰźåijŔäźš§å¿ĹæIJL'çŤÍãĂĆ åIJÍæšąæIJL'ä¡£çŤÍ partial()
çŽĎæŮűåĂŹïijŇä¡ăåŔŕèČ¡çżŔåÿÿçIJŃåĹřäÿŃéÍćè£Źçğ■ä¡£çŤÍlambdaèąÍè¿¿åijŔçŽĎåd'■æİĆäžčçăĄïijŽ

```
>>> apply_async(add, (2, 3), callback=lambda r: handler(r, seq))
[1] Got: 5
>>>
```

åŔŕäžěåŔĆèĂČ7.8åŕŔèŁĆçŽĎåĞääÿłçd'žä¿ŃïijŇæŢŹä¡ăåęĆä¡Ţä¡£çŤÍ partial()
æÍěæŻt'æŤźåŔĆæŢřç■¿åŘ■æÍěçőĂåŇŰäÿŁè£ŕäžčçăĄãĂĆ

9.11 7.11 åĘĚèĄŤåŻđèřČåĞ¡æŢř

éŮőéćŸ

å¡Şä¡äçijŰåĘŹä¡£çŤĺåŻđèřČåĞ¡æŢřçŽĎäżčçăĄçŽĎæŮűåĂŹïijŇæŃĚå£Čå¿Ĺåď ŽåřŔåĞ¡æŢřçŽĎæĽľ å
ä¡ăåÿŇæIJŻæĽ¿åĹřæ§ŘäÿłæŰźæşŢæİěèőľäżčçăĄçIJŃäÿŁåŐżæŻť åČŔæŸŕäÿĂäÿłæŹőéĂŽçŽĎæĽğèąŇåžĘ

èğčåĘşæŰźæąĹ

éĂŽè£Ğä¡£çŤĺçŤ§æĹŘåŹĺåŠŇå■ŔçĺŃåŔŕäżěä¡£å¿ŮåŻđèřČåĞ¡æŢřåĘĚèĄŤåIJĺæ§ŘäÿłåĞ¡æŢřäÿ■ãĂĆ
äÿžäžĘæijŤçď žèŕť æŸŐïijŇåĄĞèő¿ä¡ăæIJĽåęĆäÿŃæĽĂçď žçŽĎäÿĂäÿłæĽğèąŇæ§Řçğ■èőąçőŮäżżåŁąçĎű

```
def apply_async(func, args, *, callback):
    # Compute the result
    result = func(*args)

    # Invoke the callback with the result
    callback(result)
```

æŐěäÿŃæİěèőľæĹŚäżňçIJŃäÿĂäÿŃäÿŃéİćçŽĎäżčçăĄïijŇåőČåŇĚåŘńäžĘäÿĂäÿł
`Async` çśżåŠŇäÿĂäÿł `inlined_async` èčĚéěřåŹĺïijŽ

```
from queue import Queue
from functools import wraps

class Async:
    def __init__(self, func, args):
        self.func = func
        self.args = args

def inlined_async(func):
    @wraps(func)
    def wrapper(*args):
        f = func(*args)
        result_queue = Queue()
        result_queue.put(None)
        while True:
            result = result_queue.get()
            try:
                a = f.send(result)
                apply_async(a.func, a.args, callback=result_queue.
↪put)
            except StopIteration:
                break
    return wrapper
```

è£Źäÿď äÿłäżčçăĄçĽĞæőţåĚĄèőÿä¡ăä¡£çŤĺ `yield` èŕ■åŔěåĘĚèĄŤåŻđèřČæ■ěéłď ãĂĆæŕŤåęĆïijŽ

```
def add(x, y):
    return x + y

@inlined_async
def test():
    r = yield Async(add, (2, 3))
    print(r)
    r = yield Async(add, ('hello', 'world'))
    print(r)
    for n in range(10):
        r = yield Async(add, (n, n))
        print(r)
    print('Goodbye')
```

åęĆædIJä¡ăèřČçŤÍ `test()` ïijŇä¡ăäijŽå¿ŮåĹřçśżäijijåęĆäÿŃçŽĎè¿ŞåĞžïijŽ

```
5
helloworld
0
2
4
6
8
10
12
14
16
18
Goodbye
```

ä¡ăäijŽåŔŚçŐřïijŇéŹď äžĘéĆčäÿłçL’źåĹńçŽĎèčĚéěřåŹĺåŠŇ `yield` èŕ■åŔěåď ŰïijŇåĚűäžŰåIJřæŰźåźűæšąæIJL’åĞžçŐřäżżä¡ŢçŽĎåŻđèřČåĞ¡æŢř(åĚűåőđæŸŕåIJĺåŘŐåŔřåőŽäź

èőÍèőž

æIJňářŔèŁĆäijŽåőđåőđåIJĺåIJÍçŽĎæţŃèřŢä¡ăåĚşäžŐåŻđèřČåĞ¡æŢřãĂĄçŤ§æĹŘåŹĺåŠŇæŐğåĹűæţĄç

éęŰåĚĹïijŇåIJÍéIJĂèęĄä¡£çŤĺåĹřåŻđèřČçŽĎäžčçãĄäÿ■ïijŇåĚşéŤőçĆźåIJÍäžŐå¡ŞåL’■èőąçőŮåůëä¡IJäi
å¡Şèőąçő Ůé Ğ■åŘræŮűïijŇåŻđèřČåĞ¡æŢřèćńèřČçŤÍæIěçžğçż■åď ĎçŘĘçżŞæđIJãĂĆ`apply_async()`
åĞ¡æŢřæijŤçď žäžĘæL’ğèąŇåŻđèřČçŽĎåőđéŹĚéĂżè¿Śïij Ň år¡çőąåőđéŹĚæČĚåĘţäÿ■åőČåŘŕèČ¡äijŽæŻť’åŁă

èőąçőŮçŽĎæŽĆåĄIJäÿŐéĞ■åŘræĂIèůŕèů§çŤ§æĹŘåŹĺåĞ¡æŢřçŽĎæL’ğèąŇæĺąåđŃäÿ■èřŃèĂŇåŘĹãĂ
åĚůä¡ŞæIěèőšïijŇ`yield` æŞ■ä¡IJäijŽä¡£äÿĂäÿłçŤ§æĹŘåŹĺåĞ¡æŢřäžğçŤ§äÿĂäÿłåĂijåźűæŽĆåĄIJãĂĆ
æŐěäÿŃæIěèřČçŤĺçŤ§æĹŘåŹĺçŽĎ `__next__()` æĹŰ `send()`
æŰźæşŢåŘĹäijŽèőľ’åőČäžŐæŽĆåĄIJåď ĎçżğçżæL’ğèąŇãĂĆ

æăźæ■őè£ŹäÿłæĂIèůŕïijŇè£ŹäÿĂåřŔèŁĆçŽĎæăÿå£Čåřšå IJĺ `inline_async()`
èčĚéěřåŹĺåĞ¡æŢřäÿ■äžĘãĂĆ åĚşéŤőçĆźåřšæŸŕïijŇèčĚéěřåŹĺäijŽéĂŘæ■ěéĄ■åŐEçŤ§æĹŘåŹĺåĞ¡æŢřçŽĎæ
`yield` èŕ■åŘěïijŇæŕŘäÿĂæňąäÿĂäÿłãĂĆ äÿžäžĘè£ŹæăůåĄŽïijŇåĹŽåijĂåğŃçŽĎæŮűåĂŹåĹŻåžžäžĘäÿĂä
`result` éŸ§åĹŮåźűåŘŠéĞŇéIćæŤ¿åĚěäÿĂäÿł `None` åĂijãĂĆ

çĎűåŘŐåijĂåğŃäÿĂäÿłå¿łçŐŕæŞ■ä¡IJïijŇäżŐéŸ§åĹŮäÿ■åŔŰåĞžçżŞæđIJåĂijåźűåŔŚéĂĄçżŹçŢ§æĹŘåŹĺï
`yield` èŕ■åŔěïijŇ åIJĺè£ŹéĞŇäÿĂäÿł `Async` çŽĎåőđä¿Ńèćńæ Őěå ŘŮåĹřãĂĆçĎűåŘŐå¿łçŐŕåijĂåğŃæčĂæ
`apply_async()` ãĂĆ çĎűèĂŇïijŇè£ŹäÿłèőąçőŮæIJL'äÿłæIJĂèŕąåijĆéČĺåĹĘæŸŕåőČåźűæšąæIJL'ä¡£çŢĺäÿ
`put()` æŰźæşŢæİěåŻđèřČãĂĆ

è£ŹæŮűůåĂŹïijŇæŸŕæŮűůåĂŹèŕęçżĘèğčéĞŁäÿŃåĹřåžŢåŔŚçŢ§äžĘäżĂäźĹäžĘãĂĆäÿżå¿łçŐŕçńŃå■şè£
`get()` æŞ■ä¡IJãĂĆ åęĆæđIJæŢřæ■őå■ŸåIJĺïijŇåőČäÿĂåőŽæŸŕ `put()`
åŻđèřČå■ŸæŢ¿çŽĎçżŞæđIJãĂĆåęĆæđIJæšąæIJL'æŢřæ■őïijŇéĆčäźĹåĚĹæŽĆåĄIJæŞ■ä¡IJåźűç■L'å¿ĚçżŞæ
è£ŹäÿłåĚůä¡ŞæĂŐæăůåőđçŐŕæŸŕçŢś `apply_async()` åĞ¡æŢřæİěåĘşåőŽçŽĎãĂĆ
åęĆæđIJä¡ääÿ■çŻÿä£ąäijŽæIJL'è£ŹäźĹçěđåěĞçŽĎäžŃæČĚïijŇä¡ååŕŕäžěä¡£çŢĺ
`multiprocessing` åžŞæİèèŕŢäÿĂäÿŃïijŇ åIJĺå■ŢçŃňçŽĎè£ŻçĺŃäÿ■æL'ğèąŇåijĆæ■ěèőąçőŮæŞ■ä¡IJïijŇ

```
if __name__ == '__main__':
    import multiprocessing
    pool = multiprocessing.Pool()
    apply_async = pool.apply_async

    # Run the test function
    test()
```

åőđéŹĚäÿŁä¡ääijŽåŘŚçŐřè£ŹäÿłçIJ§çŽĎåřśæŸŕè£ŹæăůçŽĎïijŇä¡ĘæŸŕèęĄèğčéĞŁæÿĚæěŽåĚůä¡ŞçŽĺ

åřĘåd'■æİĆçŽĎæŐğåĹűæţĄéŽŘèŮŕåĹřçŢ§æĹŘåŹĺåĞ¡æŢřèČŇåŘŐçŽĎä¿Ńå■ŘåIJĺæăĞåĞĘåžŞåŠŇç
æŕŤåęĆïijŇåIJĺ `contextlib` äÿ■çŽĎ `@contextmanager`
èčĚéěřåŹĺä¡£çŢĺäžĘäÿĂäÿłäżd'äžžèť źèğčçŽĎæŁĂåůğïijŇ éĂŽè£ĞäÿĂäÿł `yield`
èŕ■åŔěåřĘè£ŽåĚěåŠŇçężåijĂäÿŁäÿŃæŰĞçőąçŘĘåŹĺçšŸåŘĹåIJĺäÿĂèţůãĂĆ
åŔęåd'ŰéİđåÿÿæţĄèąŇçŽĎ `Twisted` åŇĚäÿ■äź§åŇĚåŘńäžĘéİđåÿÿçśżäijijçŽĎåĘĚèĄŤåŻđèřČãĂĆ

9.12 7.12 èő£éŮőéŮ■åŇĚäÿ■åőŽäźĽçŽĎåŔŸéĞŔ

éŮőécŸ

ä¡ăæČşèęĄæĽl'åśŢåĞ¡æŢřäÿ■çŽĎæ§Řäÿłé Ů■åŇĚïijŇåĚĄèőÿåőČèČ¡èő£éŮőåŠŇä£őæŤźåĞ¡æŢřçŽĎå

èğčåĘşæŰźæąĹ

éĂŽåÿÿæİěèőšïijŇéŮ■åŇĚçŽĎåĘĚéČĺåŔŸéĞŔåŕźäžŐåd'ŰçŢŇæİěèőšæŸŕåőŇåĚĺéŽŘèŮŔçŽĎãĂĆ
ä¡ĘæŸŕïijŇä¡ăåŔŕäžěéĂŽè£ĞçijŰåĘŹèő£éŮőåĞ¡æŢřåźűåřĘåĚűä¡IJäÿžåĞ¡æŢřåśđæĂğçżŚåőŽåĹřéŮ■åŇĚäÿ

```
def sample():
    n = 0
    # Closure function
    def func():
        print('n=', n)

    # Accessor methods for n
    def get_n():
        return n
```

```
    def set_n(value):
        nonlocal n
        n = value

    # Attach as function attributes
    func.get_n = get_n
    func.set_n = set_n
    return func
```

äÿŃéÌćæŸŕä¡£çŤÍçŽĎä¿Ńå■Ř:

```
>>> f = sample()
>>> f()
n= 0
>>> f.set_n(10)
>>> f()
n= 10
>>> f.get_n()
10
>>>
```

èőÍèőž

äÿžäžĘèŕť æŸŐæÿĚæěŽåőČåęĆä¡Ţåůëä¡IJçŽĎïijŇæIJL’äÿd’çĆźéIJĂèęĄèğčéĞŁäÿĂäÿŃãĂĆéęŰåĚĹïij
åčřæŸŐåŘŕäżěèőľ æĹŚäžňçijŰåĘŹåĞ¡æŢřæİěä£őæŤźåĘĚéČĺåŔŸéĞŘçŽĎåĂijãĂĆ
åĚűæňąïijŇåĞ¡æŢřåšđæĂğåĚĄèőÿæĹŚäžňçŤĺäÿĂçğ■å¿ĹçőĂå■ŢçŽĎæŰźåijŔåřĘèő£éŮőæŰźæşŢçżŚåőŽåĹ

è£ŸåŘŕäżěè£ŻäÿĂæ■ěçŽĎæĽľ åśŢïijŇèőľéŮ■åŇĚæĺąæŃ§çśżçŽĎåőđä¿ŃãĂĆä¡ăèęĄåĄŽçŽĎäżĚäżĚa

```
import sys
class ClosureInstance:
    def __init__(self, locals=None):
        if locals is None:
            locals = sys._getframe(1).f_locals

        # Update instance dictionary with callables
        self.__dict__.update((key,value) for key, value in locals.
↪items()
                             if callable(value) )
    # Redirect special methods
    def __len__(self):
        return self.__dict__['__len__']()

# Example use
def Stack():
    items = []
    def push(item):
        items.append(item)
```

```
    def pop():
        return items.pop()

    def __len__():
        return len(items)

    return ClosureInstance()
```

äÿŃéÌćæŸŕäÿĂäÿłäžd'äžŠåijŔäijŽèŕÌæÌěæijŤçd'žåőČæŸŕåęĆä¡Ţåůëä¡IJçŽĎïijŽ

```
>>> s = Stack()
>>> s
<__main__.ClosureInstance object at 0x10069ed10>
>>> s.push(10)
>>> s.push(20)
>>> s.push('Hello')
>>> len(s)
3
>>> s.pop()
'Hello'
>>> s.pop()
20
>>> s.pop()
10
>>>
```

æIJL'èűčçŽĎæŸŕïijŇè£ŹäÿłäžčçăĄè£ŘèąŇèţůæÌěäijŽæŕŤäÿĂäÿłæŹőéĂŽçŽĎçśżåőŽäźL'èęĄå£ńå¿Ĺåd'

```
class Stack2:
    def __init__(self):
        self.items = []

    def push(self, item):
        self.items.append(item)

    def pop(self):
        return self.items.pop()

    def __len__(self):
        return len(self.items)
```

åęĆædIJè£ŹæăůåĄŽïijŇä¡ăäijŽå¿ŮåĹřçśżäijijåęĆäÿŃçŽĎçżŞædIJïijŽ

```
>>> from timeit import timeit
>>> # Test involving closures
>>> s = Stack()
>>> timeit('s.push(1);s.pop()', 'from __main__ import s')
0.9874754269840196
>>> # Test involving a class
>>> s = Stack2()
```

```
>>> timeit('s.push(1);s.pop()', 'from __main__ import s')
1.0707052160287276
>>>
```

çżŞæđIJæŸ¿çd'žïijŇéŮ■åŇĚçŽĎæŰźæąĹè£ŘèąŇètůæİěèęĄå£ńåd'ğæęĆ8%ïijŇåd'ğéČĺåĹĘåŐ§åŻăæŸ
éŮ■åŇĚæŻt'å£ńæŸŕåŻääÿžäÿ■äijŽæűL'åŔŁåĹřéćİåd'ŰçŽĎselfåŔŸéĞŔãĂĆ

Raymond Hettingerårźäž Őè£ŹäÿłéŮőécŸèő¿èőąåĞžäžĘæŻt'åŁäéŽ¿äžěçŘĘèğčçŽĎæŤźè£ŻæŰźæąĹãĂ
èĂŇäÿŤåőČåŔłæŸŕçIJ§åőđçšżçŽĎäÿĂäÿłåěĞæĂłçŽĎæŻ£æ■ćèĂŇåůšïijŇä¿ŃåęĆïijŇçšżçŽĎäÿżèęĄçĹ'źæ
åźűäÿŤä¡äèęĄåĄŽäÿĂäžŻåĚűäžŰçŽĎåůěä¡IJæĽ'■èČ¡èől'äÿĂäžŻçĽ'źæőŁæŰźæşŢçŤ§æŢĹ(æŕŤåęĆäÿŁéIć
`ClosureInstance` äÿ■éĞ■åĘŹè£ĞçŽĎ `__len__()` åőđçŐřãĂĆ)

æIJĂåŘŐïijŇä¡ăåŔŕèČ¡è£ŸäijŽèől'åĚűäžŰéŸĚèŕżä¡ääžčçăĄçŽĎäžžæĎ§åĹřçŰŚæČŚïijŇäÿžäžĂäźĹåőČ
(å¡ŞçĎűïijŇäżŰäžňäź§æČşç§ěéĄŞäÿžäžĂäźĹåőČè£ŘèąŇètůæİěäijŽæŻt'å£ń)ãĂĆår¡çőąåęĆæ■d'ïijŇè£Źårźä

æĂżä¡ŞäÿŁèőšïijŇåIJĺéĚ■ç¡őçŽĎæŮűåĂŹçżŹéŮ■åŇĚæůżåŁăæŰźæşŢäijŽæIJL'æŻt'åd'ŽçŽĎåőđçŤĺåŁ
ærŤåęĆä¡äéIJĂèęĄéĞ■ç¡őåĘĚéČĺçŁűæĂĄãĂĄåĹůæŰřçijŞåĘšåŇžãĂĄæÿĚéŹd'çijŞå■ŸæĹŰåĚűäžŰçŽĎåŔ

10 çňňåĚńçńäïijŽçśżäÿŐårźèšą

æIJňçńääÿżèęĄåĚşæşĺçĆźçŽĎæŸŕåŠŇçśżåőŽäźL'æIJL'åĚşçŽĎåÿÿèğĄçijŰçĺŃæĺąåđŃãĂĆåŇĚæŃňèől'
çśżåřĄèčĚæŁĂæIJŕãĂĄçžğæL'£ãĂĄåĘĚå■ŸçőąçŘĘäżèåŔŁæIJL'çŤĺçŽĎèő¿èőąæĺąåijŔãĂĆ

Contents:

10.1 8.1 æŤźåŔŸårźèšąçŽĎå■ŮçňęäÿšæŸ¿çd'ž

éŮőécŸ

ä¡ăæČşæŤźåŔŸårźèšąåőđä¿ŃçŽĎæL'Şå■řæĹŰæŸ¿çd'žè¿ŞåĞžïijŇèől'åőČäžňæŻt'åĚůåŔŕèŕżæĂğãĂĆ

èğčåĘşæŰźæąĹ

èęĄæŤźåŔŸäÿĂäÿłåőđä¿ŃçŽĎå■Ůçňęäÿšèąĺçd'žïijŇåŔŕéĞ■æŰřåőŽäźL'åőČçŽĎ
`__str__()` åŠŇ `__repr__()` æŰźæşŢãĂĆä¿ŃåęĆïijŽ

```
class Pair:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __repr__(self):
        return 'Pair({0.x!r}, {0.y!r})'.format(self)

    def __str__(self):
        return '({0.x!s}, {0.y!s})'.format(self)
```

__repr__() æŰźæşŢè£ŤåŻđäÿĂäÿłåőđä¿ŃçŽĎäžčçăĄèąĺçď'žå¡ćåijŕïijŇéĂŽåÿÿçŤĺæİěéĞ■æŰřæđĎ
åĘĚç¡őçŽĎ repr() åĞ¡æŢřè£ŤåŻđè£Źäÿłå■Ůçňęäÿšïij Ňèů§æĹŚäžňä¡£çŤĺäžď'äžŠåijŕèğčéĞŁåŹĺæŸ¿çď'žç
__str__() æŰźæşŢåřĘåőđä¿Ńè¡ňæ■ćäÿžäÿĂäÿłå■Ůçňęäÿšïij Ňä¡£çŤĺ str() æĹŰ
print() åĞ¡æŢřäijŽè¿ŞåĞžè£ŹäÿłåB■ŮçňęäÿšãĂĆærŤåęĆïijŽ

```
>>> p = Pair(3, 4)
>>> p
Pair(3, 4) # __repr__() output
>>> print(p)
(3, 4) # __str__() output
>>>
```

æĹŚäžňåIJĺè£ŹéĞŇè£ŸæijŤçď'žäžĘåIJĺæăijåijŕåŇŰçŽĎæŮűåĂŹæĂŐæăůä¡£çŤĺäÿ■åŘŇçŽĎå■Ůçňęäÿ
çL'źåĹńæİěèőšïijŇ!r æăijåijŕåŇŰäžčçăĄæŇĞæŸŐè¿ŞåĞžä¡£çŤĺ
__repr__() æİěäžčæŻ£éżŸèőď'çŽĎ __str__() ãĂĆ
ä¡ăåŕŕäžěçŤĺåL'■éĺćçŽĎçśżæİěèŕŢçİĂæţŃèŕŢäÿŃïijŽ

```
>>> p = Pair(3, 4)
>>> print('p is {0!r}'.format(p))
p is Pair(3, 4)
>>> print('p is {0}'.format(p))
p is (3, 4)
>>>
```

èőĺèőž

èĞłåőŽäźL' __repr__() åŠŇ __str__() éĂŽåÿÿæŸŕå¿Ĺåě¡çŽĎäźăæČŕïijŇåŻăäÿžåőČèČ¡çőĂåŇŰ
ä¿ŃåęĆïijŇåęĆæđIJäżĚäżĚåŔłæŸŕæL'Şå■řè¿ŞåĞžæĹŰæŮěå£Ůè¿ŞåĞžæ§Ŕäÿłåőđä¿ŃïijŇéĆčäźĹçĺŃåžŔåŠ

__repr__() çŤ§æĹŘçŽĎæŰĞæIJňå■ŮçňęäÿšæăĞåĞĘåĄŽæşŢæŸŕéIJĂèęĄèől'
eval(repr(x)) == x äÿžçIJ§ãĂĆ åęĆæđIJåőđåIJĺäÿ■èČ¡è£ŹæăůåB■ŘåĄŽïijŇåžŤèŕěåĹŽåžžäÿĂäÿłæIJL
< åŠŇ > æŃňètůæİěãĂĆærŤåęĆïijŽ

```
>>> f = open('file.dat')
>>> f
<_io.TextIOWrapper name='file.dat' mode='r' encoding='UTF-8'>
>>>
```

åęĆæđIJ __str__() æšąæIJL'èćńåőŽäźL'ïijŇéĆčäźĹåřšäijŽä¡£çŤĺ __repr__()
æİěäžčæŻ£è¿ŞåĞžãĂĆ

äÿŁéĺćçŽĎ format() æŰźæşŢçŽĎä¡£çŤĺçIJŃäÿŁåŐžå¿ĹæIJL'èűčïijŇæăijåijŕåŇŰäžčçăĄ
{0.x} åŕźåžŤçŽĎæŸŕçňň1äÿłåŔĆæŢřçŽĎxåšđæĂğãĂĆ åŻăæ■d'ïijŇåIJĺäÿŃéIćçŽĎåĞ¡æŢřäÿ■ïijŇ0åőđéŹ
self æIJňèžńïijŽ

```
def __repr__(self):
    return 'Pair({0.x!r}, {0.y!r})'.format(self)
```

ä¡IJäÿžè£Źçğ■åőđçŐřçŽĎäÿĂäÿłæŻ£äžčïijŇä¡ăäź§åŕŕäžěä¡£çŤĺ %
æŞ■ä¡IJçňęïijŇåřšåČŔäÿŃéIćè£ŹæăůïijŽ

```
def __repr__(self):
    return 'Pair(%r, %r)' % (self.x, self.y)
```

10.2 8.2 èĞłåőŽäźĽå■Ůçňęäÿšç ŽĎæăijåijŔåŇŰ

éŮőécŸ

ä¡ăæČşéĂŽè£Ğ format() åĞ¡æŢřåŠŇå■Ůçňęäÿšæ ŰźæşŢä¡£å¿ŮäÿĂäÿłåŕźèśąèČ¡æŤŕæŇĄèĞłåőŽäźĽ

èğčåĘşæŰźæąĹ

äÿžäžĘèĞłåőŽäźĽå■Ůçňęäÿšç ŽĎæăijåijŔåŇŰïijŇæĹŚäžňéIJĂèęĄåIJĺçśżäÿŁéĺćåőŽäźĽ __format__() æŰźæşŢãĂĆä¿ŃåęĆïijŽ

```
_formats = {
    'ymd' : '{d.year}-{d.month}-{d.day}',
    'mdy' : '{d.month}/{d.day}/{d.year}',
    'dmy' : '{d.day}/{d.month}/{d.year}'
    }

class Date:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

    def __format__(self, code):
        if code == '':
            code = 'ymd'
        fmt = _formats[code]
        return fmt.format(d=self)
```

çŐřåIJĺ Date çśżçŽĎåőđä¿ŃåŔŕäžěæŤŕæŇĄæăijåijŔåŇŰæŞ■ä¡IJäžĘïijŇåęĆåŘŇäÿŃéĺćè£ŹæăůïijŽ

```
>>> d = Date(2012, 12, 21)
>>> format(d)
'2012-12-21'
>>> format(d, 'mdy')
'12/21/2012'
>>> 'The date is {:ymd}'.format(d)
'The date is 2012-12-21'
>>> 'The date is {:mdy}'.format(d)
'The date is 12/21/2012'
>>>
```

èőÍèőž

__format__() æŰźæşŢçżŹPythonçŽĎå■ŮçňęäÿšæäijåijŔåŇŰåŁ§èČ¡æŔŘä¿ŻäžĘäÿĂäÿłéŠľå■ŘãĂ
è£ŹéĞŇéIJĂèęĄçİĂéĞ■åijžèřČçŽĎæŸŕæäijåijŔåŇŰäžčçăĄçŽĎèğčæđŘåůëä¡IJåőŇåĚÍçŤśçźèĞłåůśåĘşåőŽ
ä¿ŃåęĆïijŇåŔĆèĂČäÿŃéÍćæİěèĞł datetime æĺąåİŮäÿ■çŽĎäžčçăĄïijŽ

```
>>> from datetime import date
>>> d = date(2012, 12, 21)
>>> format(d)
'2012-12-21'
>>> format(d,'%A, %B %d, %Y')
'Friday, December 21, 2012'
>>> 'The end is {:%d %b %Y}. Goodbye'.format(d)
'The end is 21 Dec 2012. Goodbye'
>>>
```

åŕźäžŐåĘĚç¡őçśžåđŃçŽĎæäijåijŔåŇŰæIJL'äÿĂäžŻæăĞåĞĘçŽĎçžęåőŽãĂĆ
åŔŕäžěåŔĆèĂČ stringæĺąåİŮæŰĞæąč èŕť'æŸŐãĂĆ

10.3 8.3 èőľåŕźèśąæŤŕæŇĄäÿŁäÿŃæŰĞçőąçŘĘå■Ŕèőő

éŮőécŸ

ä¡ăæČşèőľä¡ăçŽĎåŕźèśąæŤŕæŇĄäÿŁäÿŃæŰĞçőąçŘĘå■Ŕèőő(withèí■åŔě)ãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèőľäÿĂäÿłåŕźèśąåĚijåőź with èí■åŔěïijŇä¡ăéIJĂèęĄåőđçŐř __enter__()
åŠŇ __exit__() æŰźæşŢãĂĆ ä¿ŃåęĆïijŇèĂČèŹŚåęĆäÿŃçŽĎäÿĂäÿłçśžïijŇåőČèČ¡äÿžæĹŚäžňåĹŻåžžäÿ

```
from socket import socket, AF_INET, SOCK_STREAM

class LazyConnection:
    def __init__(self, address, family=AF_INET, type=SOCK_STREAM):
        self.address = address
        self.family = family
        self.type = type
        self.sock = None

    def __enter__(self):
        if self.sock is not None:
            raise RuntimeError('Already connected')
        self.sock = socket(self.family, self.type)
        self.sock.connect(self.address)
        return self.sock

    def __exit__(self, exc_ty, exc_val, tb):
```

```
        self.sock.close()
        self.sock = None
```

è£Źäÿłçśżç ŽĎåĚşéŤőçL’źçĆźåIJÍäžŐåőČèąĺçď’žäžĘäÿĂäÿłç¡ŚçżIJè£đæŐěïijŇä¡ĘæŸŕåĹİåğŃåŇŰçŽĎa
è£đæŐěçŽĎåžžçńŃåŠŇåĚşéŮ■æŸŕä¡£çŤÍ with èŕ■åŔěèĞłåŁĺåőŇæĹŘçŽĎïijŇä¿ŃåęĆïijŽ

```
from functools import partial

conn = LazyConnection(('www.python.org', 80))
# Connection closed
with conn as s:
    # conn.__enter__() executes: connection open
    s.send(b'GET /index.html HTTP/1.0\r\n')
    s.send(b'Host: www.python.org\r\n')
    s.send(b'\r\n')
    resp = b''.join(iter(partial(s.recv, 8192), b''))
    # conn.__exit__() executes: connection closed
```

èőÍèőž

çijŰåĘŹäÿŁäÿŃæŰĞçőąçŘĘåŹÍçŽĎäÿżèęĄåŐ§çŘĘæŸŕä¡ăçŽĎäžčçăĄäijŽæŤ¿åĹř
with èŕ■åŔěåİŮäÿ■æĽ’ğèąŇãĂĆ å¡ŞåĞžçŐř with èŕ■åŔěçŽĎæŮűåĂŹïijŇåŕźèśąçŽĎ
__enter__() æŰźæşŢèćńèğęåŔŚïijŇ åőČè£ŤåŻđçŽĎåĂij(åęĆæđIJæIJĽ’çŽĎèŕÍ)äijŽèćńèţŃåĂijçżŹ
as åčřæŸŐçŽĎåŔŸéĞŔãĂĆçĎűåŘŐïijŇwith èŕ■åŔěåİŮéĞŇéÍćçŽĎäžčçăĄåijĂågŃæĽ’ğèąŇãĂĆ
æIJĂåŘŐïijŇ__exit__() æŰźæşŢèćńèğęåŔŚè£ŻèąŇæÿĚçŘĘåůěä¡IJãĂĆ

äÿ■çőą with äžčçăĄåİŮäÿ■åŔŚçŤ§äžĂäźĹïijŇäÿŁéÍćçŽĎæŐğåĹűæţĄéČ¡äijŽæĽ’ğèąŇåőŇïijŇåřśçőŮ
äžŃåőđäÿŁïijŇ__exit__() æŰźæşŢçŽĎçńňäÿĽäÿłåŔĆæŢřåŇĚåŘńäžĘåijĆåÿÿçśżåđŃãĂĄåijĆåÿÿåĂijåŠ
__exit__() æŰźæşŢèČ¡èĞłåůśåĘşåőŽæĂŐæăůåĹ’çŤÍè£ŹäÿłåijĆåÿÿä£ąæĄŕïijŇæĹŰèĂĚå£¡çŢěåőČåźűë
åęĆæđIJ __exit__() è£ŤåŻđ True ïijŇéĆčäźĹåijĆåÿÿäijŽèćńæÿĚçĽ’žïijŇåřśåě¡åČŔäžĂäźĹéČ¡æšąåŘŚçŤ
with èŕ■åŔěåŘŐéÍćçŽĎçÍŃäžŘçżğçż■åIJÍæ■čåÿÿæĽ’ğèąŇãĂĆ

è£ŸæIJĽäÿĂäÿłçżĘèŁĆéŮőécŸåřśæŸŕ LazyConnection
çśżæŸŕåŘęåĚĄèőÿåđ’ŽäÿŁ with èŕ■åŔěæİěåţŇåěŮä¡£çŤÍè£đæŐěãĂĆ
å¿ĹæŸ¿çĎűïijŇäÿŁéÍćçŽĎåőŽäźĽäÿ■äÿĂæňąåŔŁèČ¡åĚĄèőÿäÿĂäÿłsocketè£đæŐěïijŇåęĆæđIJæ■čåIJÍä¡£çŢ
with èŕ■åŔěïijŇ åŕśäijŽäžğçŤ§äÿĂäÿłåijĆåÿÿäžĘãĂĆäÿ■è£Ğä¡ăåŔŕäžěåČŔäÿŃéÍćè£ŹæăůäŁőæŤźäÿŇäÿŁé

```
from socket import socket, AF_INET, SOCK_STREAM

class LazyConnection:
    def __init__(self, address, family=AF_INET, type=SOCK_STREAM):
        self.address = address
        self.family = family
        self.type = type
        self.connections = []

    def __enter__(self):
        sock = socket(self.family, self.type)
        sock.connect(self.address)
```

```
        self.connections.append(sock)
        return sock

    def __exit__(self, exc_ty, exc_val, tb):
        self.connections.pop().close()

# Example use
from functools import partial

conn = LazyConnection(('www.python.org', 80))
with conn as s1:
    pass
    with conn as s2:
        pass
        # s1 and s2 are independent sockets
```

åIJÍçňňäžŇäÿłçL'ĹæIJňäÿ■ïijŇLazyConnectionçśžåŔŕäžěèćńçIJŃńåĄŽæŸŕæ§Řäÿłè£đæŐěåůěåŐĆãæŕŔæňą__enter__()æŰźæşŢæL'ğèąŇçŽĎæŮűåĂŹïijŇåőČåď'■åĹűåĹŻåžžäÿĂäÿłæŰřçŽĎè£đæŐěåźűåì__exit__()æŰźæşŢçőĂå■ŢçŽĎäžŐæăĹäÿ■åijźåĞžæIJĂåŘŐäÿĂäÿłè£đæŐěåźűåĚşéŮ■åőČãĂĆè£ŹéĞŇçĺ■å¿őæIJL'çĆźéŽ¿çŘĘèğčïijŇäÿ■è£ĞåőČèČ¡åĚĄèőÿåţŇåěŮä¡£çŤÍ with èŕ■åŔěåĹŻåžžåď'Žäÿłè£đæŐěïijŇåřšåęĆäÿŁéİćæijŤçď'žçŽĎéĆčæăůãĂĆ

åIJĺéIJĂèęĄçőąçŘĘäÿĂäžŻèţĎæžŘæŕŤåęĆæŰĞäżűãĂĄç¡ŚçżIJè£đæŐěåŠŇéŤĄçŽĎçijŰçĺŃçŐŕáćČäÿ■è£ŹäžŻèţĎæžŘçŽĎäÿĂäÿłäÿżèęĄçL'źå¿ĄæŸŕåőČäżňå£ĚéążèćńæL'ŃåŁĺçŽĎåĚşéŮ■æĹŰéĞŁæŤ¿æİěçąőä£ä¿ŃåęĆïijŇåęĆæđIJä¡ăèŕůæśĆäžĘäÿĂäÿłéŤĄïijŇéĆčäźĹä¡ăå£Ěéążçąőä£ĺäźŃåŘŐéĞŁæŤ¿äžĘåőČïijŇåŘęåĹéĂŽè£ĞåőđçŐř __enter__() åŠŇ __exit__() æŰźæşŢåźűä¡£çŤÍ with èŕ■åŔěåŕŕäžěå¿ĹåőźæŸŞçŽĎéĄ£åĚ■è£ŹäžŻéŮőécŸïijŇ åŻääÿz __exit__() æŰźæşŢåŕŕäžěèőľä¡ăæŮăéIJĂæŃĚå£Cè£ŹäžŻäžĘãĂĆ

åIJÍ contextmanager æĺąåİŮäÿ■æIJL'äÿĂäÿłæăĞåĞĘçŽĎäÿŁäÿŃæŰĞçőąçŘĘæŰźæąĹæĺąæİ£ïijŇåřŇæŮűåIJÍ12.6åřŔèŁĆäÿ■è£ŸæIJL'äÿĂäÿłåŕźæIJňèŁĆçď'žä¿ŃçÍŃåžŘçŽĎçž£çÍŃåőL'åĚĺçŽĎä£őæŤźçL'Ĺ

10.4 8.4 åĹŻåžžåď'ğéĞŔåŕźèśąæŮűèŁĆçIJĄåĘĚå■ŸæŰźæşŢ

éŮőécŸ

ä¡ăçŽĎçÍŃåžŔèęĄåĹŻåžžåď'ğéĞŔ(åŔŕèČ¡äÿŁçŹ¿äÿĞ)çŽĎåŕźèśąïijŇåŕijèĞť'å■ăçŤÍå¿Ĺåď'ğçŽĎåĘĚå■

èğčåĘşæŰźæąĹ

åŕźäžŐäÿżèęĄæŸŕçŤÍæİěå¡ŞæĹŘçőĂå■ŢçŽĎæŢřæ■őçżŞæđĎçŽĎçśżèĂŇèÍĂïijŇä¡ăåŔŕäžěéĂŽè£ĞçżŹ__slots__ åśđæĂğæİěæđĄåď'ğçŽĎåĞŔåŕŠåőđä¿ŃæL'Ăå■ăçŽĎåĘĚå■ŸãĂĆæŕŤåęĆïijŽ

```
class Date:
    __slots__ = ['year', 'month', 'day']
    def __init__(self, year, month, day):
        self.year = year
```

```
        self.month = month
        self.day = day
```

å¡Şä¡ăåőŽäźL' __slots__ åŘŐïijŇPythonåřśäijŽäÿžåőđä¿Ńä¡£çŤĺäÿĂçğ■æŻť'åŁăçť'ğåĞŚçŽĎåĘĚéČ
åőđä¿ŃéĂŽè£ĞäÿĂäÿłå¿ĹåřŔçŽĎåŻžåőŽåđ'ğåřŔçŽĎæŢřçżĎæİěæđĎåżžïijŇèĂŇäÿ■æŸŕäÿžæŕŔäÿłåőđä¿Ń
åIJĺ __slots__ äÿ■åĹŮåĞžçŽĎåśđæĂğåŘ■åIJĺåĘĚéČĺèćńæŸăărĎåĹrè£ŹäÿłæŢřçżĎçŽĎæŇĞåőŽårŔæăČ
ä¡£çŤĺslotsäÿĂäÿłäÿ■åě¡çŽĎåIJræŰźårśæŸŕæĹŚäżňäÿ■èČ¡åĘ■çżŹåőđä¿ŃæůżåŀăæŰřçŽĎåśđæĂğäžĘïijŇåĺ
__slots__ äÿ■åőŽäźL'çŽĎéĆčäžŽåśđæĂğåŘ■ãĂĆ

èőĺèőž

ä¡£çŤĺslotsåŘŐèŁĆçIJAçŽĎåĘĚå■ŸäijŽèů§å■ŸåĆĺåśđæĂğçŽĎæŢřéĞŔåŠŇçśżåđŃæIJL'åĚşãĂĆ
äÿ■è£ĞïijŇäÿĂèĹňæİěèőšïijŇä¡£çŤĺåĹrçŽĎåĘĚå■ŸæĂżéĞŔåŠŇårĘæŢřæ■őå■ŸåĆĺåIJĺäÿĂäÿłåĚĆçżĎäÿ■å
äÿžäžĘçżŹä¡ăäÿĂäÿłçŻť'èğĆèőď'èŕĘïijŇåĄĞèő¿ä¡ăäÿ■ä¡£çŤĺslotsçŻť'æŐĕå■ŸåĆĺäÿĂäÿłDateåőđä¿ŃïijŇ
åIJĺ64ä¡■çŽĎPythonäÿŁéİćèęĄå■ăçŤĺ428å■ŮèŁĆïijŇèĂŇåęĆæđIJä¡£çŤĺäžĘslotsïijŇåĘĚå■Ÿå■ăçŤĺäÿŃéŹ■
åęĆæđIJçĺŃåžŔäÿ■éIJĂèęĄåŘŇæŮűåĹŻåżžåď'ğéĞŔçŽĎæŮĕæIJ§åőđä¿ŃïijŇéĆčäźLè£Źäÿłårśè Č¡æđĄåď'ğ

år¡çőąslotsçIJŃäÿŁåŐżæŸŕäÿĂäÿłå¿ĹæIJL'çŤĺçŽĎçL'źæĂğïijŇå¿Ĺåď'ŽæŮűåĂŹä¡ăè£ŸæŸŕå¿ŮåĞŔårś
PythonçŽĎå¿Ĺåď'ŽçL'źæĂğéČ¡ä¿İètŰäžŐæŹőéĂŽçŽĎå§žäžŐå■ŮåËÿçŽĎåőđçŐřãĂĆ
åŔęåď'ŰïijŇåőŽäźL'äžĘslotsåŘŐçŽĎçśżäÿ■åĘ■æŤŕæŇĄäÿĂäžŽæŹőéĂŽçśżçL'źæĂğäžĘïijŇærŤåęĆåď'Žçżğ
åď'ğåď'ŽæŢřæČĚåĘţäÿŃïijŇä¡ăåžŤèŕěåŔłåIJĺéĆčäžŽçżŔåÿÿèćńä¡£çŤĺåĹrçŽĎçŤĺä¡IJæŢřæ■őçżŞædĎçŽĎçś
(ærŤåęĆåIJĺçĺŃåžŔäÿ■éIJĂèęĄåĹŻåżžæ§ŘäÿłçśżçŽĎåĞăçŹ¿äÿĞäÿłåőđä¿Ńårźèśą)ãĂĆ

åĚşäžŐ __slots__ çŽĎäÿĂäÿłåÿÿèğĄèŕŕåŇžæŸŕåőČåŔŕäžěä¡IJäÿžäÿĂäÿłårĄèčĚåůěåĚůæİěéŸšæ■ć
år¡çőąä¡£çŤĺslotsåŔŕäžěè¿¿åĹrè£ŹæăůçŽĎçŽőçŽĎïijŇä¡ĘæŸŕè£Źäÿłåźűäÿ■æŸŕåőČçŽĎåĹİèąůãĂĆ
__slots__ æŻť'åď'ŽçŽĎæŸŕçŤĺæİěä¡IJäÿžäÿĂäÿłåĘĚå■ŸäijŸåŇŰåůěåĚůãĂĆ

10.5 8.5 åIJĺçśżäÿ■årĄèčĚåśđæĂğåŘ■

éŮőécŸ

ä¡ăæČşårĄèčĚçśżçŽĎåőđä¿ŃäÿŁéİćçŽĎâĂIJçğĄæIJL'âĂİæŢřæ■őïijŇä¡ĘæŸŕPythonèŕ■èĺĂåźűæšąæIJ

èğčåĘşæŰźæąĹ

PythonçĺŃåžŔåŚŸäÿ■åŐżä¿İètŰèŕ■èĺĂçL'źæĂğåŐżårĄèčĚæŢřæ■őïijŇèĂŇæŸŕéĂŽè£ĞéĄţå¿łäÿĂåőŽ
çňäÿĂäÿłçžęåőŽæŸŕäżżä¡Ţäżěå■ŢäÿŃåĹŠçž£_åijĂåď'ť'çŽĎåŘ■å■ŮéČ¡åžŤèŕěæŸŕåĘĚéČĺåőđçŐřãĂĆærŤå

```
class A:
    def __init__(self):
        self._internal = 0 # An internal attribute
        self.public = 1 # A public attribute

    def public_method(self):
        '''
        A public method
```

```
        '''
        pass

    def _internal_method(self):
        pass
```

Pythonåźűäÿ■äijŽçIJ§çŽĎéŸżæ■ćåĹńäžžèő£éŮőåĘĚéČĺåŘ■çğřãĂĆä¡ĘæŸŕåęĆæđIJä¡ăè£ŹäźĹåĄŽèĆŕ
åŘŇæŮűè£ŸèęĄæşĺæĎŔåĹřïijŇä¡£çŤĺäÿŃåĹŠçž£åijĂåď'çŽĎçžęåőŽåŘŇæăůéĂĆçŤĺäžŐæĺąåİŮåŘ■åŠŇæ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăçIJŃåĹřæ§ŘäÿłæĺąåİŮåŘ■äžěå■ŢäÿŃåĹŠçž£åijĂåď'(æŕŤåęĆ_socket)ïijŇéĆčåőČåř
çśżäijijçŽĎïijŇæĺąåİŮçžğåĹńåĞ¡æŢřæŕŤåęĆ `sys._getframe()`
åIJĺä¡£çŤĺçŽĎæŮűåĂŹåřśå¿ŮåŁăåĂ■åřŔå£ČäžĘãĂĆ

ä¡ăè£ŸåŔŕèČ¡äijŽéĄĞåĹřåIJĺçśżåőŽäźL'äÿ■ä¡£çŤĺäÿd'äÿłäÿŃåĹŠçž£(__)åijĂåď'çŽĎåŚ¡åŘ■ãĂĆæŕŤå

```
class B:
    def __init__(self):
        self.__private = 0

    def __private_method(self):
        pass

    def public_method(self):
        pass
        self.__private_method()
```

ä¡£çŤĺåŘŇäÿŃåĹŠçž£åijĂåğŃäijŽåŕijèĞt'èő£éŮőåŘ■çğřåŔŸæĹŘåĚűäżŰå¡ćåijŔãĂĆ
æŕŤåęĆïijŇåIJĺåL'■éĺćçŽĎçśżBäÿ■ïijŇçğĄæIJL'åśđæĂğäijŽèćńåĹĘåĹńéĞ■åŚ¡åŘ■äÿž
`_B__private` åŠŇ `_B__private_method` ãĂĆ è£ŹæŮűåĂŹä¡ăåŔŕèČ¡äijŽéŮőè£ŹæăůéĞ■åŚ¡åŘ■çŽĎ

```
class C(B):
    def __init__(self):
        super().__init__()
        self.__private = 1 # Does not override B.__private

    # Does not override B.__private_method()
    def __private_method(self):
        pass
```

è£ŹéĞŇïijŇçğĄæIJL'åŘ■çğř `__private` åŠŇ `__private_method`
èćńéĞ■åŚ¡åŘ■äÿž `_C__private` åŠŇ `_C__private_method`
ïijŇè£ŹäÿłèůšçĹűçśżBäÿ■çŽĎåŘ■çğřæŸŕåőŃåĚĺäÿ■åŘŇçŽĎãĂĆ

èőĺèőž

äÿŁéĺćæŔŘåĹřæIJL'äÿd'çğ■äÿ■åŘŇçŽĎçijŰçăĄçžęåőŽ(å■ŢäÿŃåĹŠçž£åŠŇåŔŇäÿŃåĹŠçž£)æĬěåŚ¡åŘ
åd'ğåd'ŽæŢřèĂŇèĺĂïijŇä¡ăåžŤèŕěèől'ä¡ăçŽĎéĬđåĚňåĚśåŘ■çğřäžěå■ŢäÿŃåĹŠçž£åijĂåď'ťãĂĆä¡ĘæŸŕïijŇåę
åźűäÿŤæIJL'ăžŽåĘĚéČĺåśđæĂğåžŤèŕěåIJĺå■Řçśżäÿ■éŽŘèŮŘèţůæĬěïijŇéĆčäźĹæL'■èĂČèŹŚä¡£çŤĺåŔŇäÿŃå

è£ŸæIJL'äÿĂçĆźèęĄæşĺæĎŔçŽĎæŸŕïijŇæIJL'æŮűåĂŹä¡ăåőŽäźL'çŽĎäÿĂäÿłåŔŸéĞŔåŠŇæ§Řäÿłä£Ĭç

```
lambda_ = 2.0 # Trailing _ to avoid clash with lambda keyword
```

è£ŹéĞŇæĹŚäżňåźűäÿ■ä¡£çŤĺå■ŢäÿŃåĹŠçž£åL’■çijĂçŽĎåŐ§åŻăæŸŕåőČéĄ£åĚ■èŕŕèğčåőČçŽĎä¡£çŤĺ(åęĆä¡£çŤĺå■ŢäÿŃåĹŠçž£åL’■çijĂçŽĎçŻőçŽĎæŸŕäÿžäžĘéŸšæ■ćåŚ¡åŘ■åĘšçłĄèĂŇäÿ■æŸŕæŇĞæŸŐè£ŹéĂŽè£Ğä¡£çŤĺå■ŢäÿŃåĹŠçž£åŘŐçijĂåŔŕäżěèğčåĘşè£ŹäÿłéŮőécŸãĂĆ

10.6 8.6 åĹŻåżžåŔŕçőąçŘĘçŽĎåśđæĂğ

éŮőécŸ

ä¡ăæČşçżŹæ§Řäÿłåőđä¿ŃattributeaćđåŀăéŹd’èő£éŮőäÿŐä£őæŤźäźŃåd’ŰçŽĎåĚűäżŰåd’ĎçŘĘéĂżè¿Śi

èğčåĘşæŰźæąĹ

èĞłåőŽäźL’æ§ŘäÿłåśđæĂğçŽĎäÿĂçğ■çőĂå■ŢæŰźæşŢæŸŕåřĘáőČåőŽäźL’äÿžäÿĂäÿłpropertyãĂĆä¿ŃåęĆïijŇäÿŃéİćçŽĎäżčçăĄåőŽäźL’äžĘäÿĂäÿłpropertyïijŇåćđåŀăářźäÿĂäÿłåśđæĂğçőĂå■ŢçŽĎçśžåđŃæ

```
class Person:
    def __init__(self, first_name):
        self.first_name = first_name

    # Getter function
    @property
    def first_name(self):
        return self._first_name

    # Setter function
    @first_name.setter
    def first_name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._first_name = value

    # Deleter function (optional)
    @first_name.deleter
    def first_name(self):
        raise AttributeError("Can't delete attribute")
```

äÿŁè£řäżčçăĄäÿ■æIJL’äÿL’äÿłçŻÿåĚşèĄŢçŽĎæŰźæşŢïijŇè£ŹäÿL’äÿłæŰźæşŢçŽĎåŘ■å■ŮéČ¡å£ĚéąžäÿçňňäÿĂäÿłæŰźæşŢæŸŕäÿĂäÿł `getter` åĞ¡æŢřïijŇåőČä¡£å¿Ů `first_name` æĹŘäÿžäÿĂäÿłåśđæĂğãĂĆ åĚűäżŰäÿd’äÿłæŰźæşŢçżŹ `first_name` åśđæĂğæůžåŁäžĘ `setter` åŠŇ `deleter` åĞ¡æŢřãĂĆ éIJĂèęĄåijžèřČçŽĎæŸŕåŔłæIJL’åIJÍ `first_name` åśđæĂğècńåĹŻåżžåŘŐïijŇ åŘŐéİćçŽĎäÿd’äÿłèčĚéěřåŹÍ `@first_name.setter` åŠŇ `@first_name.deleter` æL’■èČ¡èćńåőŽäźL’ãĂĆ

propertyçŽĎäÿĂäÿłåĚşéŤőçL’źå¿ĄæŸŕåőČçIJŃäÿŁåŐżèů§æŹőéĂŽçŽĎattributeæšąäżĂäźĹäÿd’æăůïijŇä¡ĘæŸŕèő£éŮőåőČçŽĎæŮűåĂŹäijŽèĞłåŁÍèğęåŔŚ `getter` ãĂĄ`setter` åŠŇ `deleter` æŰźæşŢãĂĆä¿ŃåęĆïijŽ

```
>>> a = Person('Guido')
>>> a.first_name # Calls the getter
'Guido'
>>> a.first_name = 42 # Calls the setter
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "prop.py", line 14, in first_name
        raise TypeError('Expected a string')
TypeError: Expected a string
>>> del a.first_name
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: can`t delete attribute
>>>
```

åIJÍåőđçŐřäÿĂäÿłpropertyçŽĎæŮűåĂŹïijŇåžŢåśĆæŢřæ■ő(åęĆæđIJæIJL'çŽĎèŕİ)äż■çĎűéIJĂèęĄå■Ÿå
åŻăæ■d'ïijŇåIJÍgetåŠŇsetæŰźæşŢäÿ■ïijŇä¡ăäijŽçIJŃåĹřåŕź _first_name
åśđæĂğçŽĎæŞ■ä¡IJïijŇè£Źäź§æŸŕåőđéŹĚæŢřæ■őä£İå■ŸçŽĎåIJřæŰźãĂĆ
åŔęåd'ŰïijŇä¡ăåŔŕèČ¡è£ŸäijŽéŮőäÿžäżĂäźĹ __init__() æŰźæşŢäÿ■èő¿ç¡őäžĘ
self.first_name èĂŇäÿ■æŸŕ self._first_name ãĂĆ
åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇæĹŚäżňåĹŻåżžäÿĂäÿłpropertyçŽĎçŻőçŽĎåřśæŸŕåIJÍèő¿ç¡őattributeçŽĎæŮűåĂŹ
åŻăæ■d'ïijŇä¡ăåŔŕèČ¡æČşåIJÍåĹİåğŃåŇŰçŽĎæŮűåĂŹäź§è£ŻèąŇè£Źçğ■çśżåđŃæčĂæ§ěãĂĆéĂŽè£Ğèő¿ç¡
self.first_name ïijŇèĞłåŁÍèřČçŤÍ setter æŰźæşŢïijŇ
è£ŹäÿłæŰźæşŢéĞŇéİćäijŽè£ŻèąŇåŔĆæŢřçŽĎæčĂæ§ěïijŇåŘęåĹŹåřśæŸŕçŻť'æŐěèő£éŮő
self._first_name äžĘãĂĆ

è£ŸèČ¡åIJÍåůšå■ŸåIJÍçŽĎgetåŠŇsetæŰźæşŢå§žçąĂäÿŁåőŽäźL'propertyãĂĆä¿ŃåęĆïijŽ

```
class Person:
    def __init__(self, first_name):
        self.set_first_name(first_name)

    # Getter function
    def get_first_name(self):
        return self._first_name

    # Setter function
    def set_first_name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._first_name = value

    # Deleter function (optional)
    def del_first_name(self):
        raise AttributeError("Can't delete attribute")

    # Make a property from existing get/set methods
    name = property(get_first_name, set_first_name, del_first_name)
```

èőÍèőž

äÿĂäÿłpropertyåśđæĂğåĚűåőđåřśæŸŕäÿĂçşżåĹŮçŻÿåĚşçżŚåőŽæŰźæşŢçŽĎéŻĘåŘĹãĂĆåęĆæđIJä¡ãå
åřśäijŽåŔŚçŐřpropertyæIJňèžńçŽĎfgetãĂĄfsetåŠŇfdelåśđæĂğåřśæŸŕçśżéĞŇéÍćçŽĎæŹőéĂŽæŰźæşŢãĂĆ

```
>>> Person.first_name.fget
<function Person.first_name at 0x1006a60e0>
>>> Person.first_name.fset
<function Person.first_name at 0x1006a6170>
>>> Person.first_name.fdel
<function Person.first_name at 0x1006a62e0>
>>>
```

éĂŽåÿÿæÍěèőšïijŇä¡ääÿ■äijŽçŻť æŐěåŔŰèřČçŤÍfgetæĹŰèĂĚfsetïijŇåőČäžňäijŽåIJÍèő£éŮőpropertyçŽ

åŘłæIJL’å¡Şä¡ăçąőåőđéIJĂèęĄåŕźattributeæL’ĝèąŇåĚűäżŰéćÍåd’ŰçŽĎæŞ■ä¡IJçŽĎæŮűåĂŹæL’■åžŤèŕě
æIJL’æŮűåĂŹäÿĂäžŻäżŐåĚűäżŰçijŰçÍŃèŕ■èÍĂ(æŕŤåęĆJava)è£ĞæÍěçŽĎçÍŃåžŔåŚŸæĂżèőd’äÿžæL’ĂæIJL
æL’ĂäżěäżŰäžňèőd’äÿžäżčçăĄåžŤèŕěåČŔäÿŃéÍćè£ŹæăůåĘŹijŽ

```
class Person:
    def __init__(self, first_name):
        self.first_name = first_name

    @property
    def first_name(self):
        return self._first_name

    @first_name.setter
    def first_name(self, value):
        self._first_name = value
```

äÿ■èęĄåĘŹè£Źçğ■æšąæIJL’åĄŽäżżä¡ŢåĚűäżŰéćÍåd’ŰæŞ■ä¡IJçŽĎpropertyãĂĆ
éęŰåĚĹïijŇåőČäijŽèől’ä¡ăçŽĎäžčçăĄåŔŸå¿Ůå¿ĹèĞČèĆ£ïijŇåźűäÿŤè£ŸäijŽè£ůæČŚéŸĚèŕżèĂĚãĂĆ
åĚűæňąïijŇåőČè£ŸäijŽèől’ä¡ăçŽĎçÍŃåžŔè£ŘèąŇètůæÍěåŔŸæĚćå¿Ĺåd’ŽãĂĆ
æIJĂåŘŐïijŇè£ŹæăůçŽĎèő¿èőąĄźűæšąæIJL’åÿęæÍěäżżä¡ŢçŽĎåě¡åd’ĎãĂĆ
çL’źåĹńæŸŕå¡Şä¡ăäżěåŘŐæČşçżŹæŹőéĂŽattributeèő£éŮőæůżåŁăéćÍåd’ŰçŽĎåd’ĎçŘĘéĂżè¿ŚçŽĎæŮűåĂŹ
ä¡ăåŔŕäžěåřĘåőČåŔŸæĹŘäÿĂäÿłpropertyèĂŇæŮăéIJĂæŤźåŔŸåŐ§æÍěçŽĎäžčçăĄãĂĆ
åŽääÿžèő£éŮőattributeçŽĎäžčçăĄè£ŸæŸŕä£ÍæŇĄåŐ§æăůãĂĆ

Propertiesè£ŸæŸŕäÿĂçğ■åőŽäźL’åŁĹæĂĄèőąçŐůattributeçŽĎæŰźæşŢãĂĆ
è£Źçğ■çśżåđŃçŽĎattributesåźűäÿ■äijŽècńåőđéŹĚçŽĎå■ŸåĆÍïijŇèĂŇæŸŕåIJĺéIJĂèęĄçŽĎæŮűåĂŹèőąçőŮ

```
import math
class Circle:
    def __init__(self, radius):
        self.radius = radius

    @property
    def area(self):
        return math.pi * self.radius ** 2
```

```
    @property
    def diameter(self):
        return self.radius * 2

    @property
    def perimeter(self):
        return 2 * math.pi * self.radius
```

åIJÍè£ŹéĞŇïijŇæĹŚäžňéĂŽè£Ğä¡£çŤÍpropertiesïijŇåřĘæL'ĂæIJL'çŽĎèő£éŮőæŐěåŔčå¡ćåijŔçż§äÿĂèţ
åŕźå■Łå¿ĎãĂĄçŻť'å¿ĎãĂĄåŚÍéŢ£åŠŇéÍćçğŕçŽĎèő£éŮőéČ¡æŸŕéĂŽè£ĞåšđæĂğèő£éŮőïijŇåŕśèů§èő£éŮőç
åęĆæđIJäÿ■è£ŹæăůåĄŽçŽĎèŕİïijŇéĆčäźĹåŕśèęĄåIJÍäžčçăĄäÿ■æůůåŘĹä¡£çŤÍçőĂå■ŢåšđæĂğèő£éŮőåŠŇæ
äÿŇéÍćæŸŕä¡£çŤÍçŽĎåőđä¿ŃïijŽ

```
>>> c = Circle(4.0)
>>> c.radius
4.0
>>> c.area  # Notice lack of ()
50.26548245743669
>>> c.perimeter  # Notice lack of ()
25.132741228718345
>>>
```

år¡çőąpropertiesåŔŕäžěåőđçŐřäijŸéŻĚçŽĎçijŰçÍŃæŐěåŔčïijŇä¡ĘæIJL'äžŻæŮűåĂŹä¡ăè£ŸæŸŕäijŽæČ

```
>>> p = Person('Guido')
>>> p.get_first_name()
'Guido'
>>> p.set_first_name('Larry')
>>>
```

è£Źçğ■æČĚåĘţçŽĎåĞžçŐřéĂŽåÿÿæŸŕåŻääÿžPythonäżčçăĄèćńéŻĘæĹŘåĹřäÿĂäÿłåď'ğåđŇå§žçąĂáźş
ä¿ŇåęĆïijŇæIJL'åŔŕèČ¡æŸŕäÿĂäÿłPythonçśżåĞĘåď'ĞåŁăåĚěåĹřäÿĂäÿłå§žäžŐè£IJçÍŃè£ĞçÍŃèřČçŤÍçŽĎåć
è£Źçğ■æČĚåĘţäÿŃïijŇçŻť'æŐěä¡£çŤÍget/setæŰźæşŢ(æŹőéĂŽæŰźæşŢèřČçŤÍ)èĂŇäÿ■æŸŕpropertyæĹŰèőÿ

æIJĂåŘŐäÿĂçĆźïijŇäÿ■èęĄåČŔäÿŇéÍćè£ŹæăůåĘŹæIJL'åď'ğéĞŔéĞ■åď'■äžčçăĄçŽĎpropertyåőŽäźL'ï

```
class Person:
    def __init__(self, first_name, last_name):
        self.first_name = first_name
        self.last_name = last_name

    @property
    def first_name(self):
        return self._first_name

    @first_name.setter
    def first_name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._first_name = value
```

```
    # Repeated property code, but for a different name (bad!)
    @property
    def last_name(self):
        return self._last_name

    @last_name.setter
    def last_name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._last_name = value
```

éĞ■åď■äžčçăĄäijŽåŕijèĞť'èĞČèĆ£ãĂĄæŸŞåĞžéŤŹåŠŇäýŚéŹŃçŽĎçÍŃåžŔãĂĆåě¡æűĹæĄŕæŸŕïijŇéÀ
åŔŕäžěåŔĆèĂČ8.9åŠŇ9.21åřŔèŁĆçŽĎåĘĚåőźãĂĆ

10.7 8.7 èřČçŤÍçĹűçśżæŰźæşŢ

éŮőécŸ

ä¡ăæČşåIJÍå■Řçśżäÿ■èřČçŤÍçĹűçśżçŽĎæ§ŘäÿłåůšçżŔèćńèęĘçŻŰçŽĎæŰźæşŢãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘèřČçŤÍçĹűçśż(èűĚçśż)çŽĎäÿĂäÿłæŰźæşŢïijŇåŔŕäžěä¡£çŤÍ `super()` åĞ¡æŢřïijŇærŤåęĆïijŽ

```
class A:
    def spam(self):
        print('A.spam')

class B(A):
    def spam(self):
        print('B.spam')
        super().spam()  # Call parent spam()
```

`super()` åĞ¡æŢřçŽĎäÿĂäÿłåÿÿèğĄçŤÍæşŢæŸŕåIJÍ `__init__()` æŰźæşŢäÿ■çąőä£ÍçĹűçśżèćńæ■čçąőçŽĎåĹİåğŇåŇŰäžĘïijŽ

```
class A:
    def __init__(self):
        self.x = 0

class B(A):
    def __init__(self):
        super().__init__()
        self.y = 1
```

`super()` çŽĎåŔęåď'ŰäÿĂäÿłåÿÿèğĄçŤÍæşŢåĞžçŐŕåIJÍèęĘçŻŰPythonçL'źæőŁæŰźæşŢçŽĎäžčçăĄäÿ

```
class Proxy:
    def __init__(self, obj):
        self._obj = obj

    # Delegate attribute lookup to internal obj
    def __getattr__(self, name):
        return getattr(self._obj, name)

    # Delegate attribute assignment
    def __setattr__(self, name, value):
        if name.startswith('_'):
            super().__setattr__(name, value) # Call original __
↪setattr__
        else:
            setattr(self._obj, name, value)
```

åIJÍäÿŁéİćäżčçăĄäÿ■ïijŇ__setattr__() çŽĎåőđçŐřåŇĚåŘńäÿĂäÿłåŘ■å■ŮæčĂæ§ěãĂĆ åęĆæđIJæ§ŘäÿłåśđæĂğåŘ■äżěäÿŇåĹŞçž£(_)åijĂåď't'ïijŇåřśéĂŽè£Ğ super() èřČçŤĺåŐ§ågŃçŽĎ __setattr__() ïijŇ åŘęåĹŹçŽĎèŕĺåřśågŤæt'¿çżŹåĘĚéČĺçŽĎäżčçŘĘåŕźèśą self._obj åŐżåď'ĎçŘĘãĂĆ è£ŹçIJŃäÿŁåŐżæIJL'çĆźæĎŔæĂİïijŇåŻääÿžåřśçőŮæšąæIJL'æŸ¿åijŘçŽĎæ super() äż■çĎűåŔŕäžěæIJL'æŢĹçŽĎåůěä¡IJãĂĆ

èőĺèőž

åőđéŹĚäÿŁïijŇåď'ğåőűåŕźäžŐåIJĺPythonäÿ■åęĆä¡Ţæ■čçąőä¡£çŤĺ super() åĞ¡æŢřæŹőéĄ■ç§ěäźŃçŤŽåřŠãĂĆ ä¡ăæIJL'æŮűåĂŹäijŽçIJŃåĹřåČŔäÿŃéİćè£Źæăůç Żt'æŐèèřČçŤĺçĹűçšżç

```
class Base:
    def __init__(self):
        print('Base.__init__')

class A(Base):
    def __init__(self):
        Base.__init__(self)
        print('A.__init__')
```

åř¡çőąåŕźäžŐåď'ğéČĺåĹĘäžčçăĄèĂŇèĺĂè£ŹäźĹåĄŽæšąäżĂäźĹéŮőécŸïijŇä¡ĘæŸŕåIJĺæŻt'åď'■æİĆçŽĎ ærŤåęĆïijŇèĂČèŹŠåęĆäÿŃçŽĎæČĚåĘţïijŽ

```
class Base:
    def __init__(self):
        print('Base.__init__')

class A(Base):
    def __init__(self):
        Base.__init__(self)
        print('A.__init__')

class B(Base):
```

```
    def __init__(self):
        Base.__init__(self)
        print('B.__init__')

class C(A,B):
    def __init__(self):
        A.__init__(self)
        B.__init__(self)
        print('C.__init__')
```

åęĆædIJä¡ăè£ŘèąŇè£ŹæőţäżčçăĄåřśäijŽåŔŚçŐř `Base.__init__()` èćńèřČçŤĺäÿď æňąïijŇåęĆäÿŃæĽĂçď žïijŽ

```
>>> c = C()
Base.__init__
A.__init__
Base.__init__
B.__init__
C.__init__
>>>
```

åŔŕèČ¡äÿď æňąèřČçŤĺ `Base.__init__()` æšąäżĂäźĹåİŔåď ĎïijŇä¡ĘæIJL'æŮűåĂŹå■t'äÿ■æŸŕãĂĆ åŔęäÿĂæŰźéÍćïijŇåĄĞèő¿ä¡ăåIJĺäżčçăĄäÿ■æ■ćæĹŘä¡£çŤĺ `super()` ïijŇçżŞædIJåřśå¿Ĺåő Ňç¿ŐäžĘïijŽ

```
class Base:
    def __init__(self):
        print('Base.__init__')

class A(Base):
    def __init__(self):
        super().__init__()
        print('A.__init__')

class B(Base):
    def __init__(self):
        super().__init__()
        print('B.__init__')

class C(A,B):
    def __init__(self):
        super().__init__()  # Only one call to super() here
        print('C.__init__')
```

è£ŘèąŇè£ŹäÿłæŰřçĽĹæIJňåŘŐïijŇä¡ăäijŽåŔŚçŐřæŕŔäÿł `__init__()` æŰźæşŢåŔłäijŽèćńèřČçŤĺäÿĂæňąäžĘïijŽ

```
>>> c = C()
Base.__init__
B.__init__
```

```
A.__init__
C.__init__
>>>
```

äÿžäžĘåijĎæÿĚåőČçŽĎåŐ§çŘĘïijŇæĹŚäžňéIJĂèęĄèŁśçĆźæŮűéŮťèğčéĞŁäÿŃPythonæŸŕåęĆä¡Ţåőđ
åŕźäžŐä¡ăåőŽäźĽçŽĎæŕŔäÿĂäÿłçśżïijŇPythonäijŽèőąçőŮåĞžäÿĂäÿłæĽĂèřŞçŽĎæŰźæşŢèğčæđŘéąžäžŔ(
è£ŹäÿłMROåĹŮèąĺåřśæŸŕäÿĂäÿłçőĂå■ŢçŽĎæĽĂæIJĽå§žçśżçŽĎçž£æĂğéąžäžŔèąĺãĂĆä¿ŃåęĆïijŽ

```
>>> C.__mro__
(<class '__main__.C'>, <class '__main__.A'>, <class '__main__.B'>,
<class '__main__.Base'>, <class 'object'>)
>>>
```

äÿžäžĘåőđçŐřçżğæĽ£ïijŇPythonäijŽåIJÍMROåĹŮèąĺäÿŁäžŐåůęåĹřåŔşåijĂåğŃæ§ěæĽ¿å§žçśżïijŇçŻť

èĂŇè£ŹäÿłMROåĹŮèąĺçŽĎæđĎéĂăæŸŕéĂŽè£ĞäÿĂäÿłC3çž£æĂğåŇŰçőŮæşŢæİěåőđçŐřçŽĎãĂĆ
æĹŚäžňäÿ■åŐżæůśçľűè£ŹäÿłçőŮæşŢçŽĎæŢřå■ęåŐ§çŘĘïijŇåőČåőđéŹĚäÿŁåřśæŸŕåŘĹåźűæĽĂæIJĽçĹűçş

- å■Řçśżäij ŽåĚĹäžŐçĹűçśżèćńæčĂæ§ě
- åd'ŽäÿłçĹűçśżäij Žæăźæ■őåőČäžňåIJĺåĹŮèąĺäÿ■çŽĎéąžäžŔèćńæčĂæ§ě
- ąęĆæđIJåŕźäÿŃäÿĂäÿłçśżå■ŸåIJĺäÿd'äÿłåŘĹæşŢçŽĎéĂĽ'æŃl'ïijŇéĂĽ'æŃl'çňňäÿĂäÿłçĹűçśż

èĂĄåőđèŕť'ïijŇä¡ăæĽĂèęĄç§ěéĄŞçŽĎåřśæŸŕMROåĹŮèąĺäÿ■çŽĎçśżéąžäžŔäijŽèőľ'ä¡ăåőŽäźĽçŽĎäżz

å¡Şä¡ää¡£çŤĺ `super()` åĞ¡æŢřæŰűïijŇPythonäijŽåIJÍMROåĹŮèąĺäÿŁçżğçż■æŘIJçt'ćäÿŃäÿĂäÿłçśżãĂ
åŘłèęĄæŕŔäÿłéĞ■åőŽäźĽçŽĎæŰźæşŢçż§äÿĂä¡£çŤĺ `super()`
åźűåŘłèřČçŤĺåőČäÿĂæňąïijŇ éĆčäźĹæŐğåĹűæţĄæIJĂçżĹäijŽéĄ■åŐĘåőŇæŢt'äÿłM-
ROåĹŮèąĺïijŇæŕŔäÿłæŰźæşŢäź§åŘłäijŽèćńèřČçŤĺäÿĂæňąãĂĆ
è£Žäź§æŸŕäÿžäžĂäźĹåIJĺçňňäžŇäÿłä¿Ńå■Řäÿ■ä¡ääÿ■äijŽèřČçŤĺäÿd'æňą `Base.__init__()` çŽĎåŐ§åŻãăĂĆ

`super()` æIJĽäÿłäžd'äžžåŘČæČŁçŽĎåIJřæŰźæŸŕåőČåźűäÿ■äÿĂåőŽåŐżæ§ěæĽ¿æ§Řäÿłçśżå IJÍMRO
ä¡ăçŤŽèĞşåŘŕäžěåIJÍäÿĂäÿłæšąæIJĽçŻť'æŐěçĹűçśżçŽĎçśżäÿ■ä¡£çŤĺåőČãĂĆä¿ŃåęĆïijŇèĂČèŹŠåęĆäÿŃè

```
class A:
    def spam(self):
        print('A.spam')
        super().spam()
```

åęĆæđIJä¡ăèŕŢçİĂçŻť'æŐěä¡£çŤĺè£ŹäÿłçśżåřšäijŽåĞžéŤŹïijŽ

```
>>> a = A()
>>> a.spam()
A.spam
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "<stdin>", line 4, in spam
AttributeError: 'super' object has no attribute 'spam'
>>>
```

ä¡ĘæŸŕïijŇåęĆæđIJä¡ăä¡£çŤĺåd'ŽçżğæĽ£çŽĎèŕİçIJŃçIJŃäijŽåŘŠçŤ§äžĂäźĹïijŽ

```
>>> class B:
...     def spam(self):
...         print('B.spam')
...
>>> class C(A,B):
...     pass
...
>>> c = C()
>>> c.spam()
A.spam
B.spam
>>>
```

ä¡ăåŔŕäžěçIJŃåĹřåIJĺçśżAäÿ■ä¡£çŤĺ super().spam() åőđéŹĚäÿŁèřČçŤĺçŽĎæŸŕèů§çśżAæŕńæŮăåĚşçşżçŽĎçśżBäÿ■çŽĎ spam() æŰźæşŢãĂĆ è£ŹäÿłçŤĺçśżCçŽĎMROåĹŮèąĺářśåŔŕäžěåőŇåĚĺèğčéĞŁæÿĚæěŽäžĘïijŽ

```
>>> C.__mro__
(<class '__main__.C'>, <class '__main__.A'>, <class '__main__.B'>,
<class 'object'>)
>>>
```

åIJĺåőŽäźL'æůůåĚěçśżçŽĎæŮűåĂŹè£Źæăůä¡£çŤĺ super() æŸŕå¿ĹæŹőéĄ■çŽĎãĂĆåŔŕäžěåŔĆèĂČ8.13åŠŇ8.18ářŔèŁĆãĂĆ

çĎűèĂŇïijŇçŤśäžŐ super() åŔŕèČ¡äijŽěřČçŤĺäÿ■æŸŕä¡ăæČşèęĄçŽĎæŰźæşŢïijŇä¡ăåžŤèŕěéĄţå¿łäÿ éęŰåĚĹïijŇçąőä£ÍåIJĺçżğæL'£ä¡Şçşżäÿ■æL'ĂæIJL'çŻÿåŘŇåŘ■å■ŮçŽĎæŰźæşŢæŃěæIJL'åŔŕåĚijåőźçŽĎåŔ è£ŹæăůåŔŕäžěçąőä£Í super() èřČçŤĺäÿĂäÿłéİđçŻt'æŐěçĹűçśżæŰźæşŢæŮűäÿ■äijŽåĞžéŤŹãĂĆ åĚűæňąïijŇæIJĂåě¡çąőä£ÍæIJĂéąűåśĆçŽĎçśżæŔŘä¿ŻäžĘè£ŹäÿłæŰźæşŢçŽĎåőđçŐřïijŇè£ŹæăůçŽĎèŕÍåIJĺ

åIJĺPythonçď¿åŇžäÿ■åŕźäžŐ super() çŽĎä¡£çŤĺæIJL'æŮűåĂŹäijŽåijŢæÍěäÿĂäžŻäžL'èőőãĂĆ åř¡çőąåęĆæ■d'ïijŇåęĆædIJäÿĂåĹĞéąžåĹl'çŽĎèŕÍïijŇä¡ăåžŤèŕěåIJĺä¡ăæIJĂæŰřäžčçăĄäÿ■ä¡£çŤĺåőČãĂĆ Raymond Hettingeräÿžæ■d'åĘŹäžĘäÿĂçŕĞéİđåÿÿåě¡çŽĎæŰĞçńă âĂIJPythonâĂŹs super() Considered Super!âĂİ ïijŇ éĂŽè£Ğåd'ğéĞŘçŽĎä¿Ńå■ŘåŘśæĹŚäžňèğčéĞŁäžĘäÿžäžĂäźĹ super() æŸŕædĄåě¡çŽĎãĂĆ

10.8 8.8 å■Řçśżäÿ■æL'ľåśŢproperty

éŮőécŸ

åIJĺå■Řçśżäÿ■ïijŇä¡ăæČşèęĄæL'ľåśŢåőŽäźL'åIJĺçĹűçśżäÿ■çŽĎpropertyçŽĎåŁ§èČ¡ãĂĆ

èğčåĘşæŰźæąĹ

èĂČèŹŚåęĆäÿŃçŽĎäžčçăĄïijŇåőČåőŽäźL'äžĘäÿĂäÿłpropertyïijŽ

```
class Person:
    def __init__(self, name):
```

```
        self.name = name

    # Getter function
    @property
    def name(self):
        return self._name

    # Setter function
    @name.setter
    def name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._name = value

    # Deleter function
    @name.deleter
    def name(self):
        raise AttributeError("Can't delete attribute")
```

äÿŃéİćæŸŕäÿĂäÿłçď žä¿ŃçśżïijŇåőČçżğæĽ£èĞłPersonåźűæĽľåśŢäžĘ name åśđæĂğçŽĎåŁ§èČ¡ïijŽ

```
class SubPerson(Person):
    @property
    def name(self):
        print('Getting name')
        return super().name

    @name.setter
    def name(self, value):
        print('Setting name to', value)
        super(SubPerson, SubPerson).name.__set__(self, value)

    @name.deleter
    def name(self):
        print('Deleting name')
        super(SubPerson, SubPerson).name.__delete__(self)
```

æŐěäÿŃæİěä¡£çŤĺè£ŹäÿłæŰřçśżïijŽ

```
>>> s = SubPerson('Guido')
Setting name to Guido
>>> s.name
Getting name
'Guido'
>>> s.name = 'Larry'
Setting name to Larry
>>> s.name = 42
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 16, in name
```

```
        raise TypeError('Expected a string')
TypeError: Expected a string
>>>
```

åęĆædIJä¡ăäżĔäżĔåŔłæČşæL'l'åśŢpropertyçŽĎæ§ŘäÿĂäÿłæŰźæşŢïijŇéĆčäźĹåŔŕäżěåČŔäÿŃéİćè£Źæ

```
class SubPerson(Person):
    @Person.name.getter
    def name(self):
        print('Getting name')
        return super().name
```

æĹŰèĂĔïijŇä¡ăåŔłæČşä£őæŤźsetteræŰźæşŢïijŇåřśè£ŹäźĹåĘŹïijŽ

```
class SubPerson(Person):
    @Person.name.setter
    def name(self, value):
        print('Setting name to', value)
        super(SubPerson, SubPerson).name.__set__(self, value)
```

èőÍèőž

åIJÍå■Řçśżäÿ■æL'l'åśŢäÿĂäÿłpropertyåŔŕèČ¡äijŽåijŢètůå¿Ĺåd'Žäÿ■æŸŞår§èğL'çŽĎéŮőécŸïijŇ åŻăäÿżäÿĂäÿłpropertyåĔűåődæŸŕ getterãĂĄsetter åŠŇ deleter æŰźæşŢçŽĎéŻĘåŘĹïijŇèĂŇäÿ■æŸŕå■ŢäÿłæŰźæşŢãĂĆ åŻăæ■d'ïijŇå¡Şä¡ăæL'l'åśŢäÿĂäÿłpropertyçŽĎæŮűåĂŹïijŇä¡ăéIJÄèęĄåĚĹçąőåőŽä¡ăæŸŕåŘęèęĄéĞ■æŰřåő

åIJÍçňňäÿĂäÿłä¿Ńå■Řäÿ■ïijŇæL'ĂæIJL'çŽĎpropertyæŰźæşŢéČ¡èćńéĞ■æŰřåőŽäźL'ãĂĆ åIJÍæŕŘäÿĂäÿłæŰźæşŢäÿ■ïijŇä¡£çŤĺäžĘ super() æİěèřČçŤĺçĹűçśżçŽĎåőđçŐřãĂĆ åIJÍ setter åĞ¡æŢřäÿ■ä¡£çŤÍ super(SubPerson, SubPerson).name.__set__(self, value) çŽĎèŕ■åŘěæŸŕæšąæIJL'éŤŹçŽĎãĂĆ äÿžäžĘåğŤæL'ŸçżŹäźŃåL'■åőŽäźL'çŽĎsetteræŰźæşŢïijŇéIJÄèęĄåřĘæŐğåĹűæİČäijäéĂŠçżŹäźŃåL'■åőŽäź __set__() æŰźæşŢãĂĆ äÿ■è£ĞïijŇèŐůåŘŰè£ŹäÿłæŰźæşŢçŽĎåŤŕäÿĂéĂŤå¿ĎæŸŕä¡£çŤĺçśżåŘŸéĞŘèĂ è£Źäź§æŸŕäÿžäžĂäźĹæĹŚäžňèęĄä¡£çŤÍ super(SubPerson, SubPerson) çŽĎåŐ§åŻăãĂĆ

åęĆædIJä¡ăåŔłæČşéĞ■åőŽäźL'åĔűäÿ■äÿĂäÿłæŰźæşŢïijŇéĆčåŔłä¡£çŤÍ @property æIJňèžńæŸŕäÿ■åd'§çŽĎãĂĆæŕŤåęĆïijŇäÿŃéİćçŽĎäžčçăĄåřśæŮăæşŢåůëä¡IJïijŽ

```
class SubPerson(Person):
    @property  # Doesn't work
    def name(self):
        print('Getting name')
        return super().name
```

åęĆædIJä¡ăèŕŢçİÄè£ŘèąŇäijŽåŔŚçŐřsetteråĞ¡æŢřæŢt'äÿłæűĹåd'śäžĘïijŽ

```
>>> s = SubPerson('Guido')
Traceback (most recent call last):
```

```
    File "<stdin>", line 1, in <module>
    File "example.py", line 5, in __init__
        self.name = name
AttributeError: can't set attribute
>>>
```

ä¡ăåžŤèřěåČŔäźŃåL’■èŕť’è£ĞçŽĎéĆčæăůä£őæŤźäžčçăĄïijŽ

```
class SubPerson(Person):
    @Person.name.getter
    def name(self):
        print('Getting name')
        return super().name
```

è£ŹäźĹåĘŹåŘŐïijŇpropertyäźŃåL’■åůšçżŘåőŽäźL’è£ĞçŽĎæŰźæşŢäijŽèćńåď’■åĹűè£ĞæİěïijŇèĂŇget

```
>>> s = SubPerson('Guido')
>>> s.name
Getting name
'Guido'
>>> s.name = 'Larry'
>>> s.name
Getting name
'Larry'
>>> s.name = 42
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 16, in name
        raise TypeError('Expected a string')
TypeError: Expected a string
>>>
```

åIJÍè£ŹäÿłçĽźåĹńçŽĎèğčåĘşæŰźæąĹäÿ■ïijŇæĹŚäżňæšąåŁđæşŢä¡£çŤÍæŻť’åŁäéĂŽçŤÍçŽĎæŰźåijŔåČ
`Person` çśżåŘ■ãĂĆ åęĆæđIJä¡ääÿ■ç§ěéĄŞåĹřåžŢæŸŕåŞläÿłå§žçśżåőŽäźĽäžĘpropertyïijŇ
éĆčä¡ăåŔłèČ¡éĂŽè£ĞéĞ■æŰřåőŽäźĽæĽĂæIJL’propertyåźűä¡£çŤÍ `super()`
æİěåřĘæŐğåĹűæİČäijăéĂŠçżŹåĽ’■éİćçŽĎåőđçŐřãĂĆ

åĂijçŽĎæşĺæĎŔçŽĎæŸŕäÿŁéİćæijŤçď’žçŽĎçňňäÿĂçğ■æŁĂæIJŕè£ŸåŔŕäžěèćńçŤÍæİěæĽĺ’åśŢäÿĂäÿłą

```
# A descriptor
class String:
    def __init__(self, name):
        self.name = name

    def __get__(self, instance, cls):
        if instance is None:
            return self
        return instance.__dict__[self.name]

    def __set__(self, instance, value):
        if not isinstance(value, str):
```

```
            raise TypeError('Expected a string')
        instance.__dict__[self.name] = value

# A class with a descriptor
class Person:
    name = String('name')

    def __init__(self, name):
        self.name = name

# Extending a descriptor with a property
class SubPerson(Person):
    @property
    def name(self):
        print('Getting name')
        return super().name

    @name.setter
    def name(self, value):
        print('Setting name to', value)
        super(SubPerson, SubPerson).name.__set__(self, value)

    @name.deleter
    def name(self):
        print('Deleting name')
        super(SubPerson, SubPerson).name.__delete__(self)
```

æIJÄåŘŐåĂijçŽĎæşÍæĎŘçŽĎæŸŕïijŇèŕżåĹřè£ŹéĞŇæŮűïijŇä¡ääžŤèŕěäijŽåŔŠçŐřå■Řçśżå
setter åŠŇ deleter æŰźæşŢåĚűåőđæŸŕå¿ĹçőÅå■ŢçŽĎãĂĆ
è£ŹéĞŇæijŤçď žçŽĎèğčåĘşæŰźæąĹåŘŇæăůéĂĆçŤĺïijŇä¡ĘæŸŕåIJÍ PythonçŽĎissueéąţéÍć
æŁěåŚŁçŽĎäÿĂäÿłbugïijŇæĹŰèőÿäijŽä¡£å¿ŮåřĘæÍěçŽĎPythonçL'ĹæIJñäÿ■åĞžçŐřäÿĂäÿłæŻť'åŁăçőĂæť

10.9 8.9 åĹŻåżžæŰřçŽĎçśżæĹŰåőđä¿ŃåśđæĂğ

éŮőécŸ

ä¡ăæČşåĹŻåżžäÿĂäÿłæŰřçŽĎæŇěæIJL'äÿĂäžŻécÍåď'ŰåŁ§èČ¡çŽĎåőđä¿ŃåśđæĂğçśżåđŃïijŇæŕŤåęĆç

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşåĹŻåżžäÿĂäÿłåĚÍæŰřçŽĎåőđä¿ŃåśđæĂğïijŇåŔŕäžěéĂŽè£ĞäÿĂäÿłæŔŔè£řåŹĺçśżçŽĎ

```
# Descriptor attribute for an integer type-checked attribute
class Integer:
    def __init__(self, name):
        self.name = name
```

```
    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            return instance.__dict__[self.name]

    def __set__(self, instance, value):
        if not isinstance(value, int):
            raise TypeError('Expected an int')
        instance.__dict__[self.name] = value

    def __delete__(self, instance):
        del instance.__dict__[self.name]
```

äÿĂäÿłæŔŔè£řåŹÍåřśæŸŕäÿĂäÿłåðđçŐřäžĘäÿL'äÿłæäÿå£ČçŽĎåšđæĂğèő£éŮőæŞ■ä¡IJ(get, set, delete)çŽĎçśżïijŇ åĹĘåĹńäÿž __get__() ãĂĄ__set__() åŠŇ __delete__() è£ŹäÿL'äÿłçL'źæőŁçŽĎæŰźæşŢãĂĆ è£ŹäžŻæŰźæşŢæŐěåŔŮäÿĂäÿłåðđä¿Ńä¡IJäÿžè¿ŞåĚëïijŇäźŃåŘŐçŻÿäžŤçŽĎæŞ■ä¡IJåðđä¿ŃåžŢåśĆçŽĎå■

äÿžäžĘä¡£çŤÍäÿĂäÿłæŔŔè£řåŹÍïijŇéIJĂářĘè£ŹäÿłæŔŔè£řåŹÍçŽĎåðđä¿Ńä¡IJäÿžçśżåšđæĂğæŤ¿åĹřäÿĂ

```
class Point:
    x = Integer('x')
    y = Integer('y')

    def __init__(self, x, y):
        self.x = x
        self.y = y
```

å¡Şä¡ăè£ŹæăůåĄŽåŘŐïijŇæL'ĂæIJL'åŕźæŔŔè£řåŹÍåšđæĂğ(æŕŤåęĆxæĹŰy)çŽĎèő£éŮőäijŽècń __get__() ãĂĄ__set__() åŠŇ __delete__() æŰźæşŢæ■ŢèŐůåĹřãĂĆä¿ŃåęĆïijŽ

```
>>> p = Point(2, 3)
>>> p.x # Calls Point.x.__get__(p,Point)
2
>>> p.y = 5 # Calls Point.y.__set__(p, 5)
>>> p.x = 2.3 # Calls Point.x.__set__(p, 2.3)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "descrip.py", line 12, in __set__
        raise TypeError('Expected an int')
TypeError: Expected an int
>>>
```

ä¡IJäÿžè¿ŞåĚëïijŇæŔŔè£řåŹÍçŽĎæŕŘäÿĂäÿłæŰźæşŢäijŽæŐěåŔŮäÿĂäÿłæŞ■ä¡IJåðđä¿ŃãĂĆ äÿžäžĘåðđçŐřèŕůæśĆæŞ■ä¡IJïijŇäijŽçŻÿäžŤçŽĎæŞ■ä¡IJåðđä¿ŃåžŢåśĆçŽĎå■ŮåĚÿ(__dict__åšđæĂğ)ãĂĆ æŔŔè£řåŹÍçŽĎ `self.name` åšđæĂğå■ŸåĆĺäžĘåIJÍåðđä¿Ńå■ŮåĚÿäÿ■èćńåðđéŹĚä¡£çŤĺåĹřçŽĎkeyãĂĆ

èőÍèőž

æŔŔè£řåŹÍåŔŕåőđçŐřåď ğéČÍåĹĘPythonçśžçL’źæĂğäÿ■çŽĎåžŢåśĆé■ŤæşŢïijŇ åŇĚæŃň `@classmethod` ãĂĄ`@staticmethod` ãĂĄ`@property` ïijŇçŤŽèĞşæŸŕ `__slots__` çL’źæĂğãĂĆ

éĂŽè£ĞåőŽäźL’äÿĂäÿłæŔŔè£řåŹÍïijŇä¡ăåŔŕäžěåIJÍåžŢåśĆæ■ŢèŐůæäÿå£ČçŽĎåőđä¿ŃæŞ■ä¡IJ(get, set, delete)ïijŇåźűäÿŤåŔŕåőŇåĚÍèĞłåőŽäźL’åőČäžňçŽĎèąŇäÿžãĂĆ è£ŹæŸŕäÿĂäÿłåijžåď ğçŽĎåůěåĚůïijŇæIJL’äžĘåőČä¡ăåŔŕäžěåőđçŐřå¿Ĺåď ŽénŸçžğåŁ§èČ¡ïijŇåźűäÿŤåőČä

æŔŔè£řåŹÍçŽĎäÿĂäÿłæŕŤè¿ČåŽřæČŚçŽĎåIJřæŰźæŸŕåőČåŔłèČ¡åIJÍçśžçžğåĹńèćńåőŽäźL’ïijŇèĂŇäÿ■

```
# Does NOT work
class Point:
    def __init__(self, x, y):
        self.x = Integer('x') # No! Must be a class variable
        self.y = Integer('y')
        self.x = x
        self.y = y
```

åŘŇæŮűïijŇ`__get__()` æŰźæşŢåőđçŐřèţůæİěæŕŤçIJŃäÿŁåŐżèęĄåď ■æİĆå¿Ůåď ŽïijŽ

```
# Descriptor attribute for an integer type-checked attribute
class Integer:

    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            return instance.__dict__[self.name]
```

`__get__()` çIJŃäÿŁåŐżæIJL’çĆźåď ■æİĆçŽĎåŐ§åŻăą¡ŠçżŞäžŐåőđä¿ŃåŔŸéĞŔåŠŇçśžåŔŸéĞŔçŽĎ åęĆæđIJäÿĂäÿłæŔŔè£řåŹÍèćńå¡ŞåĄŽäÿĂäÿłçśžåŔŸéĞŔæİěëő£éŮőïijŇéĆčäźĹ `instance` åŔĆæŢřèćńèő¿ç¡őæĹŘ `None` ãĂĆ è£Źçğ■æČĚåĘţäÿŇïijŇæăĞåĞĘåĄŽæşŢåřśæŸŕçőĂå■ŢçŽĎè£ŤåŻđè£Źä

```
>>> p = Point(2,3)
>>> p.x # Calls Point.x.__get__(p, Point)
2
>>> Point.x # Calls Point.x.__get__(None, Point)
<__main__.Integer object at 0x100671890>
>>>
```

æŔŔè£řåŹÍéĂŽåÿÿæŸŕéĆčäźŻä¡£çŤÍåĹřečĚéěřåŹÍæĹŰåĚČçśžçŽĎåď ğåđŇæąĘæđűäÿ■çŽĎäÿĂäÿłçżĎ äÿ¿äÿłä¿Ńå■ŘïijŇäÿŃéÍćæŸŕäÿĂäźŽæŻť énŸçžğçŽĎå§žäźŐæŔŔè£řåŹÍçŽĎäžčçĂąïijŇåźűæűL’åŔŁåĹřäÿĂ

```
# Descriptor for a type-checked attribute
class Typed:
    def __init__(self, name, expected_type):
        self.name = name
        self.expected_type = expected_type
    def __get__(self, instance, cls):
```

```
        if instance is None:
            return self
        else:
            return instance.__dict__[self.name]

    def __set__(self, instance, value):
        if not isinstance(value, self.expected_type):
            raise TypeError('Expected ' + str(self.expected_type))
        instance.__dict__[self.name] = value
    def __delete__(self, instance):
        del instance.__dict__[self.name]

# Class decorator that applies it to selected attributes
def typeassert(**kwargs):
    def decorate(cls):
        for name, expected_type in kwargs.items():
            # Attach a Typed descriptor to the class
            setattr(cls, name, Typed(name, expected_type))
        return cls
    return decorate

# Example use
@typeassert(name=str, shares=int, price=float)
class Stock:
    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
        self.price = price
```

æIJĂåŘŐèęĄæŇĞåĞžçŽĎäÿĂçĆźæŸŕïijŇåęĆæđIJä¡ăåŔłæŸŕæČşçőĂå■ŢçŽĎèĞłåőŽäźL'æ§Řäÿłçśžç
è£Źçğ■æČĚåĘţäÿŃä¡£çŤĺ8.6åřŔèŁĆäżŃçż■çŽĎpropertyæŁĂæIJŕäijŽæŻt'åŁăåőźæŸŞãĂĆ
å¡ŞçÍŇåžŔäÿ■æIJL'å¿Ĺåd'ŽéĞ■åd'■äžčçăĄçŽĎæŮűåĂŹæŔŔè£řåŹĺářšå¿ĹæIJL'çŤĺäžĘ
(æŕŤåęĆä¡ăæČşåIJĺä¡ăäžčçăĄçŽĎå¿Ĺåd'ŽåIJřæŰźä¡£çŤĺæŔŔè£řåŹĺæŔŔä¿ŻçŽĎåŁ§èČ¡æĹŰèĂĚářĘåőČä¡I

10.10 8.10 ä¡£çŤĺäžűè£§èőąçőŮåśđæĂğ

éŮőécŸ

ä¡ăæČşåřĘäÿĂäÿłåŔłèŕžåśđæĂğåőŽäźL'æĹŘäÿĂäÿłpropertyïijŇåźűäÿŤåŔłåIJĺèő£éŮőçŽĎæŮűåĂŹæL'n
ä¡ĘæŸŕäÿĂæŮęèćńèő£éŮőåŘŐïijŇä¡ăåÿŇæIJŻçżŞæđIJåĂijèćńçijŞå■ŸètůæİěïijŇäÿ■çŤĺærŔæňąéČ¡åŐżèőą

èğčåĘşæŰźæąĹ

åőŽäźL'äÿĂäÿłåžűè£§åśđæĂğçŽĎäÿĂçğ■énŸæŢĹæŰźæşŢæŸŕéĂŽè£Ğä¡£çŤĺäÿĂäÿłæŔŔè£řåŹĺçśžïijŇ

```
class lazyproperty:
    def __init__(self, func):
```

```
        self.func = func

    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            value = self.func(instance)
            setattr(instance, self.func.__name__, value)
            return value
```

ä¡ăéIJĂèęĄåČŔäÿŃéİćè£ŹæăůåIJĺäÿĂäÿłçśżäÿ■ä¡£çŤĺåőČïijŽ

```
import math

class Circle:
    def __init__(self, radius):
        self.radius = radius

    @lazyproperty
    def area(self):
        print('Computing area')
        return math.pi * self.radius ** 2

    @lazyproperty
    def perimeter(self):
        print('Computing perimeter')
        return 2 * math.pi * self.radius
```

äÿŃéİćåIJĺäÿĂäÿłäžd'äžŠçŐŕåćČäÿ■æijŤçd'žåőČçŽĎä¡£çŤĺïijŽ

```
>>> c = Circle(4.0)
>>> c.radius
4.0
>>> c.area
Computing area
50.26548245743669
>>> c.area
50.26548245743669
>>> c.perimeter
Computing perimeter
25.132741228718345
>>> c.perimeter
25.132741228718345
>>>
```

äžŤçżĘèğĆåŕ§ä¡ăäijŽåŔŚçŐřæűĹæĄŕ `Computing area` åŠŇ `Computing perimeter` äżĚäžĚåĞžçŐřäÿĂæňąãĂĆ

èőÍèőž

å¿Ĺåď ŽæŮűåĂŹïijŇæđĎéĂääÿĂäÿłåžűè£§èőąçőŮåšđæĂğçŽĎäÿżèęĄçŽőçŽĎæŸŕäÿžäžĘæŔŘå■ĞæĂ
ä¿ŇåęĆïijŇä¡ääŔŕäżěéĄ£åĚ■èőąçőŮè£ŹäžŽåšđæĂğåĂijïijŇéŹď éİđä¡äçIJ§çŽĎéIJĂèęĄåőČäżñãĂĆ
è£ŹéĞŇæijŤçď žçŽĎæŰźæąĹåřśæŸŕçŤĺæİěåőđçŐřè£ŹæăůçŽĎæŢĹæđIJçŽĎïijŇ
åŔłäÿ■è£ĞåőČæŸŕéĂŽè£ĞäžěéİđåÿÿéńŸæŢĹçŽĎæŰźåijŔä¡£çŤĺæŔŔè£řåŹĺçŽĎäÿĂäÿłçš¿åęŹçĽźæĂğæİě

æ■čåęĆåIJĺåĚűäżŰåřŔèŁĆ(åęĆ8.9åřŔèŁĆ)æĽĂèőšçŽĎéĆčæăůïijŇå¡ŞäÿĂäÿłæŔŔè£řåŹĺèćńæŤ¿åĚěäÿ
æŕŔæňąèő£éŮőåšđæĂğæŮűåőČçŽĎ `__get__()` ãĂĄ`__set__()` åŠŇ `__delete__()`
æŰźæşŢåřśäijŽèćńèğęåŔŚãĂĆ äÿ■è£ĞïijŇåęĆæđIJäÿĂäÿłæŔŔè£řåŹĺäżĚäžĚåŔłåőŽäźĽäžĘäÿĂäÿł
`__get__()` æŰźæşŢçŽĎèŕİïijŇåőČæŕŤéĂŽåÿÿçŽĎåĚůæIJĽæŻťåijśçŽĎçżŚåőŽãĂĆ
çĽźåĹńåIJŕïijŇåŔłæIJĽå¡Şèćńèő£éŮőåšđæĂğäÿ■åIJĺåőđä¿ŇåžŢåśĆçŽĎå■ŮåĚÿäÿ■æŮű
`__get__()` æŰźæşŢæĽ■äijŽèćńèğęåŔŚãĂĆ

`lazyproperty` çśżåĹľçŤĺè£ŹäÿĂçĆźïijŇä¡£çŤĺ `__get__()`
æŰźæşŢåIJĺåőđä¿Ňäÿ■å■ŸåĆĺèőąçőŮåĞžæİěçŽĎåĂijïijŇ è£Źäÿłåőđä¿Ňä¡£çŤĺçŻÿåŘŇçŽĎåŘ■å■Ůä¡IJäÿž
è£ŹæăůäÿĂæİěïijŇçżŞæđIJåĂijèćńå■ŸåĆĺåIJĺåőđä¿Ňå■ŮåĚÿäÿ■åźűäÿŤäžěåŘŐářśäÿ■éIJĂèęĄåĘ■åŐžèőąçő
ä¡ääŔŕäžěåřİèŕŢæŻťæůśåĚěçŽĎä¿Ňå■ŘæİěèğĆář§çżŞæđIJïijŽ

```
>>> c = Circle(4.0)
>>> # Get instance variables
>>> vars(c)
{'radius': 4.0}

>>> # Compute area and observe variables afterward
>>> c.area
Computing area
50.26548245743669
>>> vars(c)
{'area': 50.26548245743669, 'radius': 4.0}

>>> # Notice access doesn't invoke property anymore
>>> c.area
50.26548245743669

>>> # Delete the variable and see property trigger again
>>> del c.area
>>> vars(c)
{'radius': 4.0}
>>> c.area
Computing area
50.26548245743669
>>>
```

è£Źçğ■æŰźæąĹæIJĽäÿĂäÿłåřŔçijžéŹůåřśæŸŕèőąçőŮåĞžçŽĎåĂijèćńåĹŻåżžåŘŐæŸŕåŔŕäžěèćńä£őæŤ

```
>>> c.area
Computing area
50.26548245743669
>>> c.area = 25
>>> c.area
```

```
25
>>>
```

åęĆædIJä¡ăæŃĚå£Čè£ŹäÿłéŮőécŸïijŇéĆčäźĹåŔŕäżěä¡£çŤĺäÿĂçğ■çĺ■å¿őæšąéĆčäźĹéńŸæŢĹçŽĎåőđç

```
def lazyproperty(func):
    name = '_lazy_' + func.__name__
    @property
    def lazy(self):
        if hasattr(self, name):
            return getattr(self, name)
        else:
            value = func(self)
            setattr(self, name, value)
            return value
    return lazy
```

åęĆædIJä¡ăä¡£çŤĺè£ŹäÿłçL'ĹæIJñïijŇåřśäijŽåŔŚçŐřçŐřåIJĺä£őæŤźæŞ■ä¡IJåůšçżŔäÿ■èćńåĚĄèőÿäžĘïij

```
>>> c = Circle(4.0)
>>> c.area
Computing area
50.26548245743669
>>> c.area
50.26548245743669
>>> c.area = 25
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: can't set attribute
>>>
```

çĎűèĂŃïijŇè£Źçğ■æŰźæąĹæIJL'äÿĂäÿłçijžçĆźåřśæŸŕæL'ĂæIJL'getæŞ■ä¡IJéČ¡å£ĚéążèćńåőŽåŘŚåĹřă
getter åĞ¡æŢřäÿŁåŐżãĂĆ è£ŹäÿłèůšäźŃåL'■çőÅå■ŢçŽĎåIJĺåőđä¿Ńå■ŮåĚÿäÿ■æšěæL'¿åĂijçŽĎæŰźæą
åęĆædIJæČşèŐůåŔŰæŻt'åď'ŽåĚşäžŐpropertyåŠŇåŔŕçőąçŘĘåśđæĂğçŽĎä£qæĄŕïijŇåŔŕäżěåŔĆèĂČ8.6åřŔ

10.11 8.11 çőĂåŇŰæŢřæ■őçżŞæđĎçŽĎåĹlåğŃåŇŰ

éŮőécŸ

ä¡ăåĘŹäžĘå¿Ĺåď'ŽäźĚäżĚçŤĺä¡IJæŢřæ■őçżŞæđĎçŽĎçśżïijŇäÿ■æČşåĘŹåď'łåď'ŽçČęäžžçŽĎ
__init__() åĞ¡æŢř

èğčåĘşæŰźæąĹ

åŔŕäżěåIJĺäÿĂäÿłå§žçśżäÿ■åĘŹäÿĂäÿłåĚñçŤĺçŽĎ __init__() åĞ¡æŢřïijŽ

```
import math

class Structure1:
    # Class variable that specifies expected fields
    _fields = []

    def __init__(self, *args):
        if len(args) != len(self._fields):
            raise TypeError('Expected {} arguments'.format(len(self.
 ↪_fields)))
        # Set the arguments
        for name, value in zip(self._fields, args):
            setattr(self, name, value)
```

çĎűåŘŐä¡£ä¡ăçŽĎçśżçżğæL'£èĞłè£Źäÿłå§žçśż:

```
# Example class definitions
class Stock(Structure1):
    _fields = ['name', 'shares', 'price']

class Point(Structure1):
    _fields = ['x', 'y']

class Circle(Structure1):
    _fields = ['radius']

    def area(self):
        return math.pi * self.radius ** 2
```

ä¡£çŤĺè£ŹäžŻçśżçŽĎçđ'žä¿ŃïijŽ

```
>>> s = Stock('ACME', 50, 91.1)
>>> p = Point(2, 3)
>>> c = Circle(4.5)
>>> s2 = Stock('ACME', 50)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "structure.py", line 6, in __init__
        raise TypeError('Expected {} arguments'.format(len(self._
 ↪fields)))
TypeError: Expected 3 arguments
```

åęĆæđIJè£ŸæČşæŤŕæŇĄåĚşéŤőå■ŮåŔĆæŢřïijŇåŔŕäžěářĘåĚşéŤőå■ŮåŔĆæŢřèő¿ç¡őäÿžåőđä¿Ńåśđæ

```
class Structure2:
    _fields = []

    def __init__(self, *args, **kwargs):
        if len(args) > len(self._fields):
            raise TypeError('Expected {} arguments'.format(len(self.
 ↪_fields)))
```

```
        # Set all of the positional arguments
        for name, value in zip(self._fields, args):
            setattr(self, name, value)

        # Set the remaining keyword arguments
        for name in self._fields[len(args):]:
            setattr(self, name, kwargs.pop(name))

        # Check for any remaining unknown arguments
        if kwargs:
            raise TypeError('Invalid argument(s): {}'.format(','.
↪join(kwargs)))
# Example use
if __name__ == '__main__':
    class Stock(Structure2):
        _fields = ['name', 'shares', 'price']

    s1 = Stock('ACME', 50, 91.1)
    s2 = Stock('ACME', 50, price=91.1)
    s3 = Stock('ACME', shares=50, price=91.1)
    # s3 = Stock('ACME', shares=50, price=91.1, aa=1)
```

ä¡ăè£ŸèČ¡åřĘäÿ■åIJĺ `_fields` äÿ■çŽĎåŘ■çğřåŁăåĚěåĹřåśđæĂğäÿ■åŐżïijŽ

```
class Structure3:
    # Class variable that specifies expected fields
    _fields = []

    def __init__(self, *args, **kwargs):
        if len(args) != len(self._fields):
            raise TypeError('Expected {} arguments'.format(len(self.
↪_fields)))

        # Set the arguments
        for name, value in zip(self._fields, args):
            setattr(self, name, value)

        # Set the additional arguments (if any)
        extra_args = kwargs.keys() - self._fields
        for name in extra_args:
            setattr(self, name, kwargs.pop(name))

        if kwargs:
            raise TypeError('Duplicate values for {}'.format(','.
↪join(kwargs)))

# Example use
if __name__ == '__main__':
    class Stock(Structure3):
```

```
        _fields = ['name', 'shares', 'price']

    s1 = Stock('ACME', 50, 91.1)
    s2 = Stock('ACME', 50, 91.1, date='8/2/2012')
```

èőÍèőž

å¡Şä¡ăéIJÄèęÄä¡£çŤÍåď ğéĞŔå¿ĹåřŔçŽĎæŢřæ■őçżŞæđĎçśżçŽĎæŮűåĂŹïijŇ çŻÿæŕŤæĽŃåůëäÿĂäÿłäÿłåőŽäźĽ `__init__()` æŰźæşŢèĂŇåůšïijŇä¡£çŤÍè£ŹçğæŰźåijŔåŔŕäžëåď ğåď ğ

åIJÍäÿŁéİćçŽĎåőđçŐřäÿ■æĹŚäžňä¡£çŤÍäžĘ `setattr()` åĞ¡æŢřçśżèő¿ç¡őåśđæĂğåÄijïijŇ ä¡ăåŔŕèČ¡äÿ■æČşçŤÍè£ŹçğæŰźåijŔïijŇèĂŇæŸŕæČşçŻťæŐëæŻťæŰřåőć

```
class Structure:
    # Class variable that specifies expected fields
    _fields= []
    def __init__(self, *args):
        if len(args) != len(self._fields):
            raise TypeError('Expected {} arguments'.format(len(self.
↪_fields)))

        # Set the arguments (alternate)
        self.__dict__.update(zip(self._fields,args))
```

ař¡çőąè£ŹäźŞåŔŕäžëæ■čåÿÿåůëä¡IJïijŇä¡ĘæŸŕå¡ŞåőŽäźĽå■ŘçśżçŽĎæŮűåĂŹéŮőécŸåřśæİëäžĘãĂĆ å¡ŞäÿĂäÿłå■ŘçśżåőŽäźĽäžĘ `__slots__` æĹŰèĂĚéĂŽè£Ğproperty(æĹŰæŔŘè£řåŹÍ)æİëåŇĚècĚæ§Řäÿłå éĆčäźĹçŻťæŐëèő£éŮőåőđä¿Ńå■ŮåĚÿåřśäÿ■ètůä¡IJçŤÍäžĘãĂĆæĹŚäžňäÿŁéİćä¡£çŤÍ `setattr()` äijŽæŸ¿å¿ŮæŻťéĂŽçŤÍäžŽïijŇåŽääÿžåőČäź§éĂĆçŤÍäžŐå■ŘçśżæČĚåĘţãĂĆ

è£Źçğ■æŰźæşŢåŤŕäÿĂäÿ■åě¡çŽĎåIJŕæŰźåřśæŸŕåŕźæ§ŘäžŻIDEèĂŇèÍĂïijŇåIJÍæŸ¿çď žåÿőåŁľåĞ¡æŢ

```
>>> help(Stock)
Help on class Stock in module __main__:
class Stock(Structure)
...
| Methods inherited from Structure:
|
| __init__(self, *args, **kwargs)
|
...
>>>
```

åŔŕäžëåŔĆèĂČ9.16åřŔèŁĆæİëåijžåĹűåIJÍ `__init__()` æŰźæşŢäÿ■æŇĞåőŽåŔĆæŢřçŽĎçśżåđŃç■¿åŘ■ãĂĆ

10.12 8.12 åőŽäźĽæŐěåŘčæĹŰèĂĚæŁ¡èśąå§žçśż

éŮőécŸ

ä¡ăæČşåőŽäźĽäÿĂäÿłæŐěåŘčæĹŰæŁ¡èśąçśżïijŇåźűäÿŤéĂŽè£ĞæĽğèąŇçśżåđŃæčĂæ§ěæİěçąőä£Ìå■Ř

èğčåĘşæŰźæąĹ

ä¡£çŤÍ abc æÍąåİŮåŘŕäżěå¿Ĺè¡żæİ¿çŽĎåőŽäźĽæŁ¡èśąå§žçśżïijŽ

```
from abc import ABCMeta, abstractmethod

class IStream(metaclass=ABCMeta):
    @abstractmethod
    def read(self, maxbytes=-1):
        pass

    @abstractmethod
    def write(self, data):
        pass
```

æŁ¡èśąçśżçŽĎäÿĂäÿłçĽźçĆźæŸŕåőČäÿ■èČ¡çŻť'æŐěèćńåőđä¿ŃåŇŰïijŇæŕŤåęĆä¡ăæČşåČŔäÿŃéİćè£Ź

```
a = IStream() # TypeError: Can't instantiate abstract class
              # IStream with abstract methods read, write
```

æŁ¡èśąçśżçŽĎçŻőçŽĎåřśæŸŕèőľ'åĹńçŽĎçśżçżğæĽ'£åőČåźűåőđçŐřçĽ'źåőŽçŽĎæŁ¡èśąæŰźæşŢïijŽ

```
class SocketStream(IStream):
    def read(self, maxbytes=-1):
        pass

    def write(self, data):
        pass
```

æŁ¡èśąå§žçśżçŽĎäÿĂäÿłäÿżèęĄçŤÍéĂŤæŸŕåIJÍäżčçăĄäÿ■æčĂæ§ěæ§ŘäžŽçśżæŸŕåŘęäÿžçĽ'źåőŽçśżåđ

```
def serialize(obj, stream):
    if not isinstance(stream, IStream):
        raise TypeError('Expected an IStream')
    pass
```

éŹđ'äžĘçżğæĽ'£è£Źçğ■æŰźåijŔåđ'ŰïijŇè£ŸåŕŕäżěéĂŽè£ĞæşĺåĘŇæŰźåijŔæİèèőľ'æ§ŘäÿłçśżåőđçŐřæ

```
import io

# Register the built-in I/O classes as supporting our interface
IStream.register(io.IOBase)
```

```
# Open a normal file and type check
f = open('foo.txt')
isinstance(f, IStream) # Returns True
```

@abstractmethod è£ŸèČ¡æşĺèğčéİŹæĂĄæŰźæşŢãĂĄçśżæŰźæşŢåŠŇ properties ãĂĆ ä¡ääŔłéIJĂä£İèŕĄè£ŹäÿłæşĺèğčçťğéİääIJĺåĞ¡æŢřáőŽäźĽåĽ■å■şåŔŕïijŽ

```
class A(metaclass=ABCMeta):
    @property
    @abstractmethod
    def name(self):
        pass

    @name.setter
    @abstractmethod
    def name(self, value):
        pass

    @classmethod
    @abstractmethod
    def method1(cls):
        pass

    @staticmethod
    @abstractmethod
    def method2():
        pass
```

èőĺèőž

æăĞåĞĘåžŞäÿ■æIJL'å¿Ĺåď'ŽçŤĺåĹřæŁ¡èśąå§žçśżçŽĎåIJřæŰźãĂĆcollections æĺąåİŮåőŽäźL'äžĘå¿Ĺåď'Žèů§åőźåŹĺåŠŇè£■äžčåŹĺ(åžŔåĹŮãĂĄæŸäářĎãĂĄéŻĘåŘĹç■L')æIJL'åĚşçŽĎæŁ numbers åžŞåőŽäźL'äžĘèů§æŢřå■Ůářźèśą(æŢť'æŢřãĂĄæţőçĆźæŢřãĂĄæIJL'çŘĘæŢřç■L')æIJL'åĚşçŽĎå§ž åžŞåőŽäźL'äžĘå¿Ĺåď'Žèů§I/OæŞ■ä¡IJçŻÿåĚşçŽĎå§žçśżãĂĆ

ä¡ääŔŕäžěä¡£çŤĺécĎåőŽäźL'çŽĎæŁ¡èśąçśżæİěæL'ğèąŇæŻť'éĂŽçŤĺçŽĎçśżåđŃæčĂæ§ěïijŇä¿ŃåęĆïijŽ

```
import collections

# Check if x is a sequence
if isinstance(x, collections.Sequence):
...

# Check if x is iterable
if isinstance(x, collections.Iterable):
...

# Check if x has a size
if isinstance(x, collections.Sized):
```

```
...

# Check if x is a mapping
if isinstance(x, collections.Mapping):
```

åř¡çőąABCsåŘŕäżèèőľæĹŚäžňå¿ĹæŰźä¿£çŽĎåĄŽçśżåđŃæčĂæ§ěïijŇä¡ĘæŸŕæĹŚäžňåIJĺäžčçăĄäÿ■æI
åŻääÿžPythonçŽĎæIJňèť´ĺæŸŕäÿĂéŮĺĺåŁĺæĂĄçijŰçĺŃèŕ■èĺĂïijŇåĚűçŽőçŽĎåřśæŸŕçżŹä¡ăæŻť´åď´ŽçĄţæť´ża
åijžåĹűçśżåđŃæčĂæ§ěæĹŰèőľä¡ăäżčçăĄåŔŸå¿ŮæŻť´åď´■æİĆïijŇè£ŹæăůåĄŽæŮăåijĆäžŐèĹ■æIJňæśĆæIJ

10.13 8.13 åőđçŐřæŢřæ■őæĺąåđŃçŽĎçśżåđŃçžęæİ§

éŮőécŸ

ä¡ăæČşåőŽäźL´æ§ŘäžŻåIJĺåśđæĂğètŃåĂijäÿŁéİćæIJL´éŹŘåĹűçŽĎæŢřæ■őçżŞæđĎãĂĆ

èğčåĘşæŰźæąĹ

åIJĺè£ŹäÿłéŮőécŸäÿ■ïijŇä¡ăéIJĂèęĄåIJĺářźæ§ŘäžŻåőđä¿ŃåśđæĂğètŃåĂijæŮűè£ŻèąŇæčĂæ§ěãĂĆ
æL´Ăäżěä¡ăèęĄèĞłåőŽäźL´åśđæĂğètŃåĂijåĞ¡æŢřïijŇè£Źçğ■æČĚåĘţäÿŃæIJĂåě¡ä¡£çŤĺæŔŘè£řåŹĺãĂĆ

äÿŃéİćçŽĎäžčçăĄä¡£çŤĺæŔŘè£řåŹĺåőđçŐřäžĘäÿĂäÿłçşżçż§çśżåđŃåŠŇètŃåĂijéłŇèŕĄæąĘæđűïijŽ

```
# Base class. Uses a descriptor to set a value
class Descriptor:
    def __init__(self, name=None, **opts):
        self.name = name
        for key, value in opts.items():
            setattr(self, key, value)

    def __set__(self, instance, value):
        instance.__dict__[self.name] = value


# Descriptor for enforcing types
class Typed(Descriptor):
    expected_type = type(None)

    def __set__(self, instance, value):
        if not isinstance(value, self.expected_type):
            raise TypeError('expected ' + str(self.expected_type))
        super().__set__(instance, value)


# Descriptor for enforcing values
class Unsigned(Descriptor):
    def __set__(self, instance, value):
        if value < 0:
            raise ValueError('Expected >= 0')
```

```
        super().__set__(instance, value)


class MaxSized(Descriptor):
    def __init__(self, name=None, **opts):
        if 'size' not in opts:
            raise TypeError('missing size option')
        super().__init__(name, **opts)

    def __set__(self, instance, value):
        if len(value) >= self.size:
            raise ValueError('size must be < ' + str(self.size))
        super().__set__(instance, value)
```

è£ŹäžŻçśžåřśæŸŕäą¡äèęĄåĹŻåżžçŽĎæŢřæ■őæĺąåđŃæĹŰçśżåđŃçşżçż§çŽĎå§žçąĂæđĎåżžæĺąåİŮãĂĆäÿŃéİćåřśæŸŕæĹŚäžňåőđéŹĚåőŽäźL’çŽĎåŘĎçğ■äÿ■åŘŇçŽĎæŢřæ■őçśżåđŃïijŽ

```
class Integer(Typed):
    expected_type = int

class UnsignedInteger(Integer, Unsigned):
    pass

class Float(Typed):
    expected_type = float

class UnsignedFloat(Float, Unsigned):
    pass

class String(Typed):
    expected_type = str

class SizedString(String, MaxSized):
    pass
```

çĎűåŘŐä¡£çŤĺè£ŹäžŻèĞłåőŽäźL’æŢřæ■őçśżåđŃïijŇæĹŚäžňåőŽäźL’äÿĂäÿłçśżïijŽ

```
class Stock:
    # Specify constraints
    name = SizedString('name', size=8)
    shares = UnsignedInteger('shares')
    price = UnsignedFloat('price')

    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
        self.price = price
```

çĎűåŘŐæţŃèŕŢè£ŹäÿłçśżçŽĎåśđæĂğèţŃåĂijçžęæİ§ïijŇåŔŕåŘŚçŐřåŕźæ§ŘäžŽåśđæĂğçŽĎèţŃåĂijè£İ

```
>>> s.name
'ACME'
>>> s.shares = 75
>>> s.shares = -10
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 17, in __set__
        super().__set__(instance, value)
    File "example.py", line 23, in __set__
        raise ValueError('Expected >= 0')
ValueError: Expected >= 0
>>> s.price = 'a lot'
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 16, in __set__
        raise TypeError('expected ' + str(self.expected_type))
TypeError: expected <class 'float'>
>>> s.name = 'ABRACADABRA'
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 17, in __set__
        super().__set__(instance, value)
    File "example.py", line 35, in __set__
        raise ValueError('size must be < ' + str(self.size))
ValueError: size must be < 8
>>>
```

è£ŸæIJL'äÿĂäžŻæŁĂæIJŕåŔŕäżěçőĂåŇŰäÿŁéÍćçŽĎäžčçăĄïijŇåĚűäÿ■äÿĂçğ■æŸŕä¡£çŤĺçśżèčĚéěřåŹĺ

```
# Class decorator to apply constraints
def check_attributes(**kwargs):
    def decorate(cls):
        for key, value in kwargs.items():
            if isinstance(value, Descriptor):
                value.name = key
                setattr(cls, key, value)
            else:
                setattr(cls, key, value(key))
        return cls

    return decorate

# Example
@check_attributes(name=SizedString(size=8),
                  shares=UnsignedInteger,
                  price=UnsignedFloat)
class Stock:
    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
```

```
        self.price = price
```

åŔęåď Űäÿ Ăçğ■æŰźåijŔæŸŕä¡£çŤĺåĚČçśżïijŽ

```
# A metaclass that applies checking
class checkedmeta(type):
    def __new__(cls, clsname, bases, methods):
        # Attach attribute names to the descriptors
        for key, value in methods.items():
            if isinstance(value, Descriptor):
                value.name = key
        return type.__new__(cls, clsname, bases, methods)

# Example
class Stock2(metaclass=checkedmeta):
    name = SizedString(size=8)
    shares = UnsignedInteger()
    price = UnsignedFloat()

    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
        self.price = price
```

èőÍèőž

æIJňèŁĆä¡£çŤĺäžĘå¿Ĺåď ŽéńŸçžğæŁĂæIJŕïijŇåŇĚæŃňæŔŔè£řåŹĺãĂĄæůůåĚěçśżãĂĄsuper() çŽĎä¡£çŤĺãĂĄçśżèčĚéěřåŹĺåŠŇåĚČçśżãĂĆ äÿ■åŔŕèČ¡åIJÍè£ŹéĞŇäÿĂäÿĂèŕęçżĘåśŢåijĂæİěèőšïijŇä¡ĘæŸä¡ĘæŸŕïijŇæĹŚåIJÍè£ŹéĞŇè£ŸæŸŕèęĄæŔŔäÿĂäÿŃåĞăäÿłéIJĂèęĄæşÍæĎŔçŽĎçĆźãĂĆ

éęŰåĚĹïijŇåIJÍ Descriptor å§žçśżäÿ■ä¡ăäijŽçIJŃåĹřæIJL'äÿł __set__() æŰźæşŢïijŇå■ť'æšąæIJL'çŽÿåžŤçŽĎ __get__() æŰźæşŢãĂĆ åęĆæđIJäÿĂäÿłæŔŔè£řäżĚäżĚæŸŕäżŐåžŢåśĆåőđä¿Ńå■ŮåĚÿäÿ■èŐůåŔŰæ§ŘäÿłåśđæĂğåĂijçŽĎèŕÍïijŇéĆ __get__() æŰźæşŢãĂĆ

æL'ĂæIJL'æŔŔè£řåŹĺçśżéČ¡æŸŕå§žäžŐæůůåĚěçśżæİěåőđçŐřçŽĎãĂĆærŤåęĆ Unsigned åŠŇ MaxSized èęĄèů§åĚűäżŰçżğæL'£èĞł Typed çśżæůůåĚěãĂĆ è£ŹéĞŇåĹ'çŤĺåď Žçżğæ L'£æİěåőđçŐřçŽÿåžŤçŽĎåŁ§èČ¡ãĂĆ

æůůåĚěçśżçŽĎäÿĂäÿłærŤè¿ČéŽ¿çŘęğčçŽĎåIJřæŰźæŸŕïijŇèřČçŤĺ super() åĞ¡æŢřæŮűïijŇä¡ăăźűäÿ■ç§ěéĄŞçl'űçń§èęĄèřČçŤĺåŞłäÿłåĚůä¡ŞçśżãĂĆ ä¡ăéIJĂèęĄèů§åĚűäżŰçśżçżŞåŘĹåŘŐæL'■èČ¡æ■čçąőçŽĎä¡£çŤĺïijŇäź§åřśæŸŕå£ĚéążåŘĹä¡IJæL'■èČ¡äžğçŤ

ä¡£çŤĺçśżèčĚéěřåŹĺåŠŇåĚČçśżéĂŽåÿÿåŔŕäżěçőĂåŇŰäżčçăĄãĂĆäÿŁéİćäÿd'äÿłä¿Ńå■Řäÿ■ä¡ăäijŽåŔŠ

```
# Normal
class Point:
    x = Integer('x')
    y = Integer('y')
```

```
# Metaclass
class Point(metaclass=checkedmeta):
    x = Integer()
    y = Integer()
```

æĽĂæIJL'æŰźæşŢäÿ■ïijŇçśżèčĚéěřåŹĺæŰźæąĹåžŤèřěæŸřæIJĂçĄţæť'żåŠŇæIJĂéńŸæŸŐçŽĎãĂĆ éęŰåĚĹïijŇåőČåźűäÿ■ä¿İèţŰäżżä¡ŢåĚűäżŰæŰřçŽĎæŁĂæIJřïijŇæŕŤåęĆåĚČçśżãĂĆåĚűæňąïijŇèčĚéěřåŹĺå

æIJĂåŘŐïijŇèčĚéěřåŹĺè£ŸèČ¡ä¡IJäÿžæůůåĚěçśżçŽĎæŻ£äżčæŁĂæIJŕæİěåőđçŐřåŘŇæăůçŽĎæŢĹæđIJ

```
# Decorator for applying type checking
def Typed(expected_type, cls=None):
    if cls is None:
        return lambda cls: Typed(expected_type, cls)
    super_set = cls.__set__

    def __set__(self, instance, value):
        if not isinstance(value, expected_type):
            raise TypeError('expected ' + str(expected_type))
        super_set(self, instance, value)

    cls.__set__ = __set__
    return cls


# Decorator for unsigned values
def Unsigned(cls):
    super_set = cls.__set__

    def __set__(self, instance, value):
        if value < 0:
            raise ValueError('Expected >= 0')
        super_set(self, instance, value)

    cls.__set__ = __set__
    return cls


# Decorator for allowing sized values
def MaxSized(cls):
    super_init = cls.__init__

    def __init__(self, name=None, **opts):
        if 'size' not in opts:
            raise TypeError('missing size option')
        super_init(self, name, **opts)

    cls.__init__ = __init__

    super_set = cls.__set__
```

```
    def __set__(self, instance, value):
        if len(value) >= self.size:
            raise ValueError('size must be < ' + str(self.size))
        super_set(self, instance, value)

    cls.__set__ = __set__
    return cls


# Specialized descriptors
@Typed(int)
class Integer(Descriptor):
    pass


@Unsigned
class UnsignedInteger(Integer):
    pass


@Typed(float)
class Float(Descriptor):
    pass


@Unsigned
class UnsignedFloat(Float):
    pass


@Typed(str)
class String(Descriptor):
    pass


@MaxSized
class SizedString(String):
    pass
```

è£Źçğ■æŰźåijŔåőŽäźL’çŽĎçśżèů§äźŃåL’■çŽĎæŢĹæđIJäÿĂæăůïijŇèĂŇäÿŤæL’ğèąŇéĂ§åžęäijŽæŻť’å
èő¿ç¡őäÿĂäÿłçőĂå■ŢçŽĎçśżåđŇåśđæĂğçŽĎåĂijïijŇèčĚéěřåŹĺæŰźåijŔèęĄæŕŤäźŃåL’■çŽĎæůůåĚěçśżçŽĎ
çŐřåIJĺä¡ăåžŤèŕěåžĘáźÿèĞłåůśèŕżåőŇäžĘæIJňèŁĆåĚĺéČĺåĘĚåőźäžĘåŘğïij§^_^

10.14 8.14 åőđçŐřèĞłåőŽäźL’åőźåŹĺ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłèĞłåőŽäźL’çŽĎçśżæİěæĺąæŃ§åĘĚç¡őçŽĎåőźåŹĺçśżåŁ§èČ¡ïijŇæŕŤåęĆåĹŮèąĺåŠ

èğčåĘşæŰźæąĹ

collections åőŽäźL'äžEå¿Ĺåd'ŽæŁ¡èśąå§žçśżïijŇå¡Şä¡äæČşèĞłåőŽäźL'åőźåŹĺçśżçŽĎæŮűåĂŹåőČ
æŕŤåęĆä¡äæČşèől'ä¡äçŽĎçśżæŤŕæŇĄè£■äžčïijŇéĆčåřśèől'ä¡äçŽĎçśżçżğæL'£
collections.Iterable å■şåŔřïijŽ

```
import collections
class A(collections.Iterable):
    pass
```

äÿ■è£Ğä¡äéIJĂèęĄåőđçŐř collections.Iterable
æL'ĂæIJL'çŽĎæŁ¡èśąæŰźæşŢïijŇåŘęåĹŹäijŽæŁěéŤŹ:

```
>>> a = A()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: Can't instantiate abstract class A with abstract methods␣
 ↪__iter__
>>>
```

ä¡ääŔłèęĄåőđçŐř __iter__() æŰźæşŢåřśäÿ■äijŽæŁěéŤŹäžĘ(åŔĆèĂČ4.2åŠŇ4.7åřŔèŁĆ)ãĂĆ

ä¡ääŔŕäžěåĚĹèŕŢçİĂåŐżåőđä¿ŃåŇŰäÿĂäÿłåŕźèśąïijŇåIJĺéŤŹèŕŕæŔŘçđ'žäÿ■åŔŕäžěæL'¿åĹřéIJĂèęĄåő

```
>>> import collections
>>> collections.Sequence()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: Can't instantiate abstract class Sequence with abstract␣
 ↪methods \
__getitem__, __len__
>>>
```

äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎçđ'žä¿ŃïijŇçżğæL'£èĞłäÿŁéİćSequenceæŁ¡èśąçśżïijŇåźűäÿŤåőđçŐřåĚČ

```
class SortedItems(collections.Sequence):
    def __init__(self, initial=None):
        self._items = sorted(initial) if initial is not None else []

    # Required sequence methods
    def __getitem__(self, index):
        return self._items[index]

    def __len__(self):
        return len(self._items)

    # Method for adding an item in the right location
    def add(self, item):
        bisect.insort(self._items, item)
```

```
items = SortedItems([5, 1, 3])
print(list(items))
print(items[0], items[-1])
items.add(2)
print(list(items))
```

åŔŕäžěçIJŃåĹřïijŇSortedItemsèů§æŹőéĂŽçŽĎåžŔåĹŮæšąäżĂäźĹäÿd'æăůïijŇæŤŕæŇĄæL'ĂæIJL'åÿÿç

è£ŹéĞŇéİćä¡£çŤĺåĹřäžĘ `bisect` æĺąåİŮïijŇåőČæŸŕäÿĂäÿłåIJĺæŐŠåžŔåĹŮèąĺäÿ■æŔŠåĚěåĚČçt'ăçŽ

èőĺèőž

ä¡£çŤĺ `collections` äÿ■çŽĎæŁ¡èśąå§žçśżåŔŕäžěçąőä£İä¡äèĞłåőŽäźL'çŽĎåőźåŹĺåőđçŐřäžĘæL'ĂæI
ä¡ăçŽĎèĞłåőŽäźL'åőźåŹĺäijŽæżąèűşåd'ğéČĺåĹĘçśżåđŃæčĂæ§ěéIJĂèęĄïijŇåęĆäÿŃæL'Ăçd'žïijŽ

```
>>> items = SortedItems()
>>> import collections
>>> isinstance(items, collections.Iterable)
True
>>> isinstance(items, collections.Sequence)
True
>>> isinstance(items, collections.Container)
True
>>> isinstance(items, collections.Sized)
True
>>> isinstance(items, collections.Mapping)
False
>>>
```

`collections` äÿ■å¿Ĺåd'ŽæŁ¡èśąçśżäijŽäÿžäÿĂäžŽåÿÿèğĄåőźåŹĺæŞ■ä¡IJæŔŘä¿ŻéżŸèőd'çŽĎåőđçČ
è£ŹæăůäÿĂæİěä¡ăåŔłéIJĂèęĄåőđçŐřéĆčäžŽä¡ăæIJĂæĎ§åĚt'èűčçŽĎæŰźæşŢå■şåŔŕãĂĆåĄĞèő¿ä¡ăçŽĎçśż
`collections.MutableSequence` ïijŇåęĆäÿŃïijŽ

```
class Items(collections.MutableSequence):
    def __init__(self, initial=None):
        self._items = list(initial) if initial is not None else []

    # Required sequence methods
    def __getitem__(self, index):
        print('Getting:', index)
        return self._items[index]

    def __setitem__(self, index, value):
        print('Setting:', index, value)
        self._items[index] = value

    def __delitem__(self, index):
        print('Deleting:', index)
        del self._items[index]
```

```
    def insert(self, index, value):
        print('Inserting:', index, value)
        self._items.insert(index, value)

    def __len__(self):
        print('Len')
        return len(self._items)
```

åęĆædIJä¡ăåĹŻåżž `Items` çŽĎåőđä¿ŃïijŇä¡ăäijŽåŔŚçŐřåőČæŤŕæŇĄåĞäázŐæL'ĂæIJL'çŽĎæäÿå£Čå
äÿŃéİćæŸŕä¡£çŤĺæijŤçd'žïijŽ

```
>>> a = Items([1, 2, 3])
>>> len(a)
Len
3
>>> a.append(4)
Len
Inserting: 3 4
>>> a.append(2)
Len
Inserting: 4 2
>>> a.count(2)
Getting: 0
Getting: 1
Getting: 2
Getting: 3
Getting: 4
Getting: 5
2
>>> a.remove(3)
Getting: 0
Getting: 1
Getting: 2
Deleting: 2
>>>
```

æIJňářŔèŁĆåŘłæŸŕářźPythonæŁ¡èšąçšžåŁ§èČ¡çŽĎæŁŻçăŰåijŢçŐL'ãĂĆnumbers
æĺąåİŮæŔŘä¿ŻäžĘäÿĂäÿłçšżäijijçŽĎèů§æŢť'æŢřçšžåđŃçŻÿåĚşçŽĎæŁ¡èšąçšžåđŃéŻĘåŘĹãĂĆ
åŘŕäžěåŘĆèĂČ8.12ářŔèŁĆæİěædĎéĂăæŽť'åd'ŽèĞłåőŽäźL'æŁ¡èšąå§žçšžãĂĆ

10.15 8.15 åšđæĂğçŽĎäžčçŘĘèő£éŮő

éŮőécŸ

ä¡ăæČşářĘæ§Räÿłåőđä¿ŃçŽĎåšđæĂğèő£éŮőäžčçŘĘåĹřåĘĚéČĺåŔęäÿĂäÿłåőđä¿Ńäÿ■åŐżïijŇçŽőçŽĎå

èğčåĘşæŰźæąĹ

çőĂå■Ţæİěèŕť˙ïijŇäžčçŘĘæŸŕäÿĂçğ■çijŰçĺŃæĺąåijŔïijŇåőČåřĘæ§Řäÿłæş■ä¡IJè¡ňçğžçżŹåŔęåď˙ŰäÿĂæIJĂçőĂå■ŢçŽĎå¡ćåijŔåŔŕèČ¡æŸŕåČŔäÿŃéİćè£ŹæăůïijŽ

```
class A:
    def spam(self, x):
        pass

    def foo(self):
        pass


class B1:
    """çőĂå■ŢçŽĎäżčçŘĘ"""

    def __init__(self):
        self._a = A()

    def spam(self, x):
        # Delegate to the internal self._a instance
        return self._a.spam(x)

    def foo(self):
        # Delegate to the internal self._a instance
        return self._a.foo()

    def bar(self):
        pass
```

åęĆædIJäžĚäžĚåřśäÿď˙äÿłæŰźæşŢéIJĂèęĄäżčçŘĘïijŇéĆčäźĹåČŔè£ŹæăůåĘŹåřśèűşåď˙§äžĘãĂĆä¡ĘæŸéĆčäźĹä¡£çŤÍ `__getattr__()` æŰźæşŢæĹŰèőÿæĹŰæŻť˙åě¡äžŻïijŽ

```
class B2:
    """ä¡£çŤÍ__getattr__çŽĎäżčçŘĘïijŇäżčçŘĘæŰźæşŢærŤè¿Čåd'ŽæŮűåĂŹ"""

    def __init__(self):
        self._a = A()

    def bar(self):
        pass

    # Expose all of the methods defined on class A
    def __getattr__(self, name):
        ""
↪"è£ŹäÿłæŰźæşŢåIJĺèő£éŮőçŽĎattributeäÿ■å■ŸåIJĺçŽĎæŮűåĂŹèćńèřČçŤÍ
        the __getattr__() method is actually a fallback method
        that only gets called when an attribute is not found"""
        return getattr(self._a, name)
```

`__getattr__` æŰźæşŢæŸŕåIJĺèő£éŮőattributeäÿ■å■ŸåIJĺçŽĎæŮűåĂŹèćńèřČçŤÍïijŇä¡£çŤÍæijŤçd˙ži

```
b = B()
b.bar() # Calls B.bar() (exists on B)
b.spam(42) # Calls B.__getattr__('spam') and delegates to A.spam
```

åŔęåď Űäÿ Ăäÿłäżčç ŘĘä¿Ńå■ŘæŸŕåőđçŐřäżčçŘĘæĺąåijŔïijŇä¿ŃåęĆïijŽ

```
# A proxy class that wraps around another object, but
# exposes its public attributes
class Proxy:
    def __init__(self, obj):
        self._obj = obj

    # Delegate attribute lookup to internal obj
    def __getattr__(self, name):
        print('getattr:', name)
        return getattr(self._obj, name)

    # Delegate attribute assignment
    def __setattr__(self, name, value):
        if name.startswith('_'):
            super().__setattr__(name, value)
        else:
            print('setattr:', name, value)
            setattr(self._obj, name, value)

    # Delegate attribute deletion
    def __delattr__(self, name):
        if name.startswith('_'):
            super().__delattr__(name)
        else:
            print('delattr:', name)
            delattr(self._obj, name)
```

ä¡£çŤĺè£ŹäÿłäżčçŘĘçśżæŮűïijŇä¡ăåŔłéIJĂèęĄçŤĺåőČæĺěåŇĚèčĚäÿŃåĚűäżŰçśżå■şåŔŕïijŽ

```
class Spam:
    def __init__(self, x):
        self.x = x

    def bar(self, y):
        print('Spam.bar:', self.x, y)

# Create an instance
s = Spam(2)
# Create a proxy around it
p = Proxy(s)
# Access the proxy
print(p.x)  # Outputs 2
p.bar(3)  # Outputs "Spam.bar: 2 3"
p.x = 37  # Changes s.x to 37
```

éĂŽè£ĞèĞłåőŽäźL’åśđæĂğèő£éŮőæŰźæşŢïijŇä¡ääŔŕäžěçŤĺäÿ■åŘŇæŰźåijŔèĞłåőŽäźL’äžčçŘĘçśżèąŇ

èőĺèőž

äžčçŘĘçśżæIJL’æŮűåĂŹåŔŕäžěä¡IJäÿžçżğæL’£çŽĎæŻ£äžčæŰźæąĹãĂĆä¿ŃåęĆïijŇäÿĂäÿłçőĂå■ŢçŽĎ

```
class A:
    def spam(self, x):
        print('A.spam', x)
    def foo(self):
        print('A.foo')

class B(A):
    def spam(self, x):
        print('B.spam')
        super().spam(x)
    def bar(self):
        print('B.bar')
```

ä¡£çŤĺäžčçŘĘçŽĎèŕĭijŇåřśæŸŕäÿŃéİćè£ŹæăůïijŽ

```
class A:
    def spam(self, x):
        print('A.spam', x)
    def foo(self):
        print('A.foo')

class B:
    def __init__(self):
        self._a = A()
    def spam(self, x):
        print('B.spam', x)
        self._a.spam(x)
    def bar(self):
        print('B.bar')
    def __getattr__(self, name):
        return getattr(self._a, name)
```

å¡Şåőđç Őŕäžčç ŘĘæĺąåijŔæŮűïijŇè£ŸæIJL’äžŻçżĘèŁĆéIJĂèęĄæşĺæĎŔãĂĆ
éęŰåĚĹïijŇ__getattr__() åőđéŹĚæŸŕäÿĂäÿłåŘŐåď ĞæŰźæşŢïijŇåŔłæIJL’åIJĺåśđæĂğäÿ■å■ŸåIJĺæŮű
åŻăæ■ď’ïijŇåęĆæđIJäžčçŘĘçśżåőđä¿ŃæIJňèžńæIJL’è£ŹäÿłåśđæĂğçŽĎèŕĭijŇéĆčäźĹäÿ■äijŽèğęåŔŚè£Źäÿłą
åŔęåď ŰïijŇ__setattr__() åŠŇ __delattr__() éIJĂèęĄéćİåď ŰçŽĎé■ŤæşŢæİěåŇžåĹĘäžčçŘĘåőđ
_obj çŽĎåśđæĂğãĂĆ äÿĂäÿłéĂŽåÿÿçŽĎçžęåőŽæŸŕåŔłäžčçŘĘéĆčäźŻäÿ■äžëäÿŃåĹŠçž£
_ åijĂåď’t çŽĎåśđæĂğ(äžčçŘĘçśżåŔłæŽť éIJšèćńäžčçŘĘçśżçŽĎåĚňåĚśåśđæĂğ)ãĂĆ

è£ŸæIJL’äÿĂçĆźéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇ__getattr__()
åŕźäžŐåď ğéČĺåĹĘäžěåŔŇäÿŃåĹŠçž£(__)åijĂåğŃåŠŇçżŞår¿çŽĎåśđæĂğåźűäÿ■éĂĆçŤĺãĂĆ
æŕŤåęĆïijŇèĂČèŹŚåęĆäÿŃçŽĎçśżïijŽ

```
class ListLike:
    """__getattr__
↪åŕźäžŐåŔÑäÿŃåĹŠçž£åijÄåğŃåŠÑçżŞår¿çŽĎæŰźæşŢæŸŕäÿ■èČ¡çŤÍçŽĎïijÑéIJÄèęĄäÿÄäÿłäÿł
↪"""

    def __init__(self):
        self._items = []

    def __getattr__(self, name):
        return getattr(self._items, name)
```

åęĆæđIJæŸŕåĹŻåżžäÿĂäÿłListLikeåŕźèśąïijÑäijŽåŔŚçŐřåőČæŤŕæŇĄæŹőéĂŽçŽĎåĹŮèąÍæŰźæşŢïijÑä¡ĘæŸŕå■t’äÿ■æŤŕæŇĄlen()ãĂĄåĚČçt’äæ§ěæL’¿ç■L’ãĂĆä¿ŃåęĆïijŽ

```
>>> a = ListLike()
>>> a.append(2)
>>> a.insert(0, 1)
>>> a.sort()
>>> len(a)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: object of type 'ListLike' has no len()
>>> a[0]
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: 'ListLike' object does not support indexing
>>>
```

äÿžäžĘèől’åőČæŤŕæŇĄè£ŹäžŻæŰźæşŢïijÑä¡åå£ĚéążæL’ŃåŁÍçŽĎåőđçŐřè£ŹäžŻæŰźæşŢäžčçŘĘïijŽ

```
class ListLike:
    """__getattr__
↪åŕźäžŐåŔÑäÿŃåĹŠçž£åijÄåğŃåŠÑçżŞår¿çŽĎæŰźæşŢæŸŕäÿ■èČ¡çŤÍçŽĎïijÑéIJÄèęĄäÿÄäÿłäÿł
↪"""

    def __init__(self):
        self._items = []

    def __getattr__(self, name):
        return getattr(self._items, name)

    # Added special methods to support certain list operations
    def __len__(self):
        return len(self._items)

    def __getitem__(self, index):
        return self._items[index]

    def __setitem__(self, index, value):
        self._items[index] = value
```

```
    def __delitem__(self, index):
        del self._items[index]
```

11.8åřŔèŁĆè£ŸæIJL'äÿĂäÿłåIJÍè£IJçÍŃæŰźæşŢèřČçŤÍçŐřåćČäÿ■ä¡£çŤÍäžčçŘĘçŽĎä¿Ńå■ŘãĂĆ

10.16 8.16 åIJÍçśżäÿ■åőŽäźĽåďŽäÿłæđĎéĂäåŹÍ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłçśżïijŇéŹď'äžĘä¡£çŤÍ `__init__()` æŰźæşŢåď'ŰïijŇè£ŸæIJL'åĚűäžŰæŰźåijŔåŔŕäžěåĹİåğŃåŇŰåőČãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘåőđçŐřåď'ŽäÿłæđĎéĂäåŹÍïijŇä¡ăéIJĂèęĄä¡£çŤÍåĹřçśżæŰźæşŢãĂĆä¿ŃåęĆïijŽ

```
import time

class Date:
    """æŰźæşŢäÿĂïijŽä¡£çŤÍçśżæŰźæşŢ"""
    # Primary constructor
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

    # Alternate constructor
    @classmethod
    def today(cls):
        t = time.localtime()
        return cls(t.tm_year, t.tm_mon, t.tm_mday)
```

çŽť'æŐèèřČçŤÍçśżæŰźæşŢå■şåŔŕïijŇäÿŃéİćæŸŕä¡£çŤÍçď'žä¿ŃïijŽ

```
a = Date(2012, 12, 21) # Primary
b = Date.today() # Alternate
```

èőÍèőž

çśżæŰźæşŢçŽĎäÿĂäÿłäÿżèęĄçŤÍéĂŤåřśæŸŕåőŽäźĽåď'ŽäÿłæđĎéĂäåŹÍãĂĆåőČæŐěåŔŮäÿĂäÿł `class` ä¡IJäÿžçňňäÿĂäÿłåŔĆæŢř(cls)ãĂĆ ä¡ăåžŤèŕěæşÍæĎŔåĹřäžĘè£ŹäÿłçśżèćńçŤÍæİěåĹŻåžžåźűè£ŤåŽđa

```
class NewDate(Date):
    pass
```

```
c = Date.today() # Creates an instance of Date (cls=Date)
d = NewDate.today() # Creates an instance of NewDate (cls=NewDate)
```

10.17 8.17 åĹŻåżžäÿ■èřČçŤĺnitæŰźæşŢçŽĎåőđä¿Ń

éŮőécŸ

ä¡ăæČşåĹŻåżžäÿĂäÿłåőđä¿ŃïijŇä¡ĘæŸŕåÿŇæIJŻçżŢè£ĞæĽ'ğèąŇ `__init__()` æŰźæşŢãĂĆ

èğčåĘşæŰźæąĹ

åŘŕäžěéĂŽè£Ğ `__new__()` æŰźæşŢåĹŻåżžäÿĂäÿłæIJłåĹİåğŃåŇŰçŽĎåőđä¿ŃãĂĆä¿ŃåęĆèĂČèŹŚåę

```
class Date:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day
```

äÿŃéİćæijŤçd'žåęĆä¡Ţäÿ■èřČçŤĺ `__init__()` æŰźæşŢæİěåĹŻåżžè£ŹäÿłDateåőđä¿ŃïijŽ

```
>>> d = Date.__new__(Date)
>>> d
<__main__.Date object at 0x1006716d0>
>>> d.year
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: 'Date' object has no attribute 'year'
>>>
```

çżŞædIJåŘŕäžěçIJŃåĹřïijŇè£ŹäÿłDateåőđä¿ŃçŽĎåšđæĂğyearè£Ÿäÿ■å■ŸåIJĺïijŇæĽ'Ääžěä¡ăéIJÄèęĄæ

```
>>> data = {'year':2012, 'month':8, 'day':29}
>>> for key, value in data.items():
...     setattr(d, key, value)
...
>>> d.year
2012
>>> d.month
8
>>>
```

èőÍèőž

å¡ŞæĹŚäżňåIJÍåŔ■åžŔåĹŮåŕźèśąæĹŰèĂĚåőđçŐřæ§ŘäÿłçśżæŰźæşŢæđĎéĂäåĞ¡æŢřæŮűéIJĂèęĄçżŢè
`__init__()` æŰźæşŢæİěåĹŻåżžåŕźèśąãĂĆ ä¿ŃåęĆïijŇåŕźäžŐäÿŁéİćçŽĎDateæİěèőšïijŇæIJL'æŮűåĂŹä¡ä
`today()` ïijŽ

```
from time import localtime

class Date:
    def __init__(self, year, month, day):
        self.year = year
        self.month = month
        self.day = day

    @classmethod
    def today(cls):
        d = cls.__new__(cls)
        t = localtime()
        d.year = t.tm_year
        d.month = t.tm_mon
        d.day = t.tm_mday
        return d
```

åŘŇæăůïijŇåIJÍä¡ăåŔ■åžŔåĹŮåŇŰJSONæŢřæ■őæŮűäžğçŤ§äÿĂäÿłåęĆäÿŃçŽĎå■ŮåĚÿåŕźèśąïijŽ

```
data = { 'year': 2012, 'month': 8, 'day': 29 }
```

åęĆæđIJä¡ăæČşåřĘåőČè¡ňæ■ćæĹŘäÿĂäÿłDateçśżåđŃåőđä¿ŃïijŇåŔŕäżěä¡£çŤĺäÿŁéİćçŽĎæŁĂæIJŕãĂĆ

å¡Şä¡ăéĂŽè£Ğè£Źçğ■éİđåÿÿèğĎæŰźåijŔæİěåĹŻåżžåőđä¿ŃçŽĎæŮűåĂŹïijŇæIJĂåě¡äÿ■èęĄçŻť'æŐěåĆ
åŘęåĹŹçŽĎèŕİïijŇåęĆæđIJè£Źäÿłçśżä¡£çŤĺäžĘ `__slots__` ãĂĄproperties ãĂĄdescriptors æĹŰåĚűäżŰéńŸçžğæŁĂæIJŕçŽĎæŮűåĂŹäżčçăĄåřśäijŽåď'śæŢĹãĂĆ
èĂŇè£ŹæŮűåĂŹä¡£çŤĺ `setattr()` æŰźæşŢäijŽèőľ'ä¡ăçŽĎäżčçăĄåŔŸå¿ŮæŻť'åŁăéĂŽçŤĺãĂĆ

10.18 8.18 åĹľçŤĺMixinsæL'ľåśŢçśżåŁ§èČ¡

éŮőéćŸ

ä¡ăæIJL'å¿Ĺåď'ŽæIJL'çŤĺçŽĎæŰźæşŢïijŇæČşä¡£çŤĺåőČäżňæİěæL'ľåśŢåĚűäżŰçśżçŽĎåŁ§èČ¡ãĂĆä¡Ęa
åŻăæ■ď'ä¡ăäÿ■èČ¡çőĂå■ŢçŽĎåřĘè£ŹäžŻæŰźæşŢæŤ¿åĚěäÿĂäÿłå§žçśżïijŇçĎűåŘŐèćńåĚűäżŰçśżçżğæL'£ã

èğčåĘşæŰźæąĹ

éĂŽåÿÿå¡Şä¡ăæČşèĞłåőŽäźL'çśżçŽĎæŮűåĂŹäijŽçćřäÿŁè£ŹäžŻéŮőéćŸãĂĆåŔŕèČ¡æŸŕæ§ŘäÿłåžŞæŔ
ä¡ăåŔŕäżěåĹľçŤĺåőČäżňæİěæđĎéĂää¡äèĞłåůśçŽĎçśżãĂĆ

åĄĞèő¿ä¡ăæČşæL'ľåśŢæŸäåřĎåŕźèśąïijŇçżŹåőČäżňæůżåŁăæŮěå£ŮãĂĄåŤŕäÿĂæĂğèő¿ç¡őãĂĄçśżåđ

```
class LoggedMappingMixin:
    """
    Add logging to get/set/delete operations for debugging.
    """
    __slots__ = ()  #␣
↪æůůåĚěçśżéČ¡æšąæIJL’åőđä¿ŃåŔŸéĞŔïijŇåŽääÿžçŻt’æŐěåőđä¿ŃåŇŰæůůåĚěçśżæšąæIJL’äżż

    def __getitem__(self, key):
        print('Getting ' + str(key))
        return super().__getitem__(key)

    def __setitem__(self, key, value):
        print('Setting {} = {!r}'.format(key, value))
        return super().__setitem__(key, value)

    def __delitem__(self, key):
        print('Deleting ' + str(key))
        return super().__delitem__(key)


class SetOnceMappingMixin:
    '''
    Only allow a key to be set once.
    '''
    __slots__ = ()

    def __setitem__(self, key, value):
        if key in self:
            raise KeyError(str(key) + ' already set')
        return super().__setitem__(key, value)


class StringKeysMappingMixin:
    '''
    Restrict keys to strings only
    '''
    __slots__ = ()

    def __setitem__(self, key, value):
        if not isinstance(key, str):
            raise TypeError('keys must be strings')
        return super().__setitem__(key, value)
```

è£ŹäžŻçśżå■ŢçŃňä¡£çŤĺèţůæİěæšąæIJL’äżżä¡ŢæĎŔäźL’ïijŇäžŃåőđäÿŁåęĆæđIJä¡ääŐżåőđä¿ŃåŇŰäżżäåőČäžňæŸŕçŤĺæİěéĂŽè£ĞåđŽçżğæL’£æİěåŠŇåĚűäżŰæŸäářĎářźèśąæůůåĚěä¡£çŤĺçŽĎãĂĆä¿ŃåęĆïijŽ

```
class LoggedDict(LoggedMappingMixin, dict):
    pass

d = LoggedDict()
```

```
d['x'] = 23
print(d['x'])
del d['x']

from collections import defaultdict

class SetOnceDefaultDict(SetOnceMappingMixin, defaultdict):
    pass


d = SetOnceDefaultDict(list)
d['x'].append(2)
d['x'].append(3)
# d['x'] = 23  # KeyError: 'x already set'
```

è£Źäÿłä¿Ńå■Řäÿ■ïijŇåŔŕäžěçIJŃåĹřæůůåĚěçśżèů§åĚűäžŰåůšå■ŸåIJĺçŽĎçśż(æŕŤåęĆdictãĂĄdefaultd
çżŞåŘĹåŘŐåřśèČ¡åŔŚæŇěæ■čåÿÿåŁ§æŢĹäžĘãĂĆ

èőĺèőž

æůůåĚěçśżåIJĺæăĞåĞĘåžŞäÿ■å¿Ĺåď'ŽåIJřæŰźéČ¡åĞžçŐřè£ĞïijŇéĂŽåÿÿéČ¡æŸŕçŤĺæİěåČŔäÿŁéİćéĆ
åőČäžňäź§æŸŕåď'Žçżğæ L'£çŽĎäÿĂäÿłäÿżèęĄçŤĺéĂŤãĂĆæŕŤåęĆïijŇå¡Şä¡äçijŰåĘŹç¡ŚçżIJäžčçăĄæŮűåĂŹ
ä¡ăäijŽçżŘåÿÿä¡£çŤĺ `socketserver` æĺąåİŮäÿ■çŽĎ `ThreadingMixIn`
æİěçżŹåĚűäžŰç¡ŚçżIJçŽÿåĚşçśżåćđåŁååď'Žçž£çĺŇæŤŕæŇĄãĂĆ
ä¿ŇåęĆïijŇäÿŇéİćæŸŕäÿĂäÿłåď'Žçž£çĺŇçŽĎXML-RPCæIJ■åŁąïijŽ

```
from xmlrpc.server import SimpleXMLRPCServer
from socketserver import ThreadingMixIn
class ThreadedXMLRPCServer(ThreadingMixIn, SimpleXMLRPCServer):
    pass
```

åŘŇæŮűåIJĺäÿĂäžŻåď'ğåđŇåžŞåŠŇæąĘæđűäÿ■äź§äijŽåŔŚçŐřæůůåĚěçśżçŽĎä¡£çŤĺïijŇçŤĺéĂŤåŘŇæ

åŕźäžŐæůůåĚěçśżïijŇæIJL'åĞăçĆźéIJĂèęĄèőŕä¡ŔãĂĆéęŰåĚĹæŸŕïijŇæůůåĚěçśżäÿ■èČ¡çŻť'æŐěèćńåő
åĚűæňąïijŇæůůåĚěçśżæšąæIJL'èĞłåůśçŽĎçŁűæĂĄä£ąæĄŕïijŇäź§åřśæŸŕèŕť'åőČäžňåźűæšąæIJL'åőŽäźL'
`__init__()` æŰźæşŢïijŇåźűäÿŤæšąæIJL'åőđä¿ŃåśđæĂğãĂĆ
è£Źäź§æŸŕäÿžäžĂäźĹæĹŚäžňåIJĺäÿŁéİćæŸŐçąőåőŽäźL'äžĘ `__slots__ = ()` ãĂĆ

è£ŸæIJL'äÿĂçğ■åőđçŐřæůůåĚěçśżçŽĎæŰźåijŔåřśæŸŕä¡£çŤĺçśżèčĚéěřåŹĺïijŇąęĆäÿŃæL'Ăçđ'žïijŽ

```
def LoggedMapping(cls):
    """çňňäžŇçğ■æŰźåijŔïijžä¡£çŤĺçśżèčĚéěřåŹĺ"""
    cls_getitem = cls.__getitem__
    cls_setitem = cls.__setitem__
    cls_delitem = cls.__delitem__

    def __getitem__(self, key):
        print('Getting ' + str(key))
        return cls_getitem(self, key)
```

```
    def __setitem__(self, key, value):
        print('Setting {} = {!r}'.format(key, value))
        return cls_setitem(self, key, value)

    def __delitem__(self, key):
        print('Deleting ' + str(key))
        return cls_delitem(self, key)

    cls.__getitem__ = __getitem__
    cls.__setitem__ = __setitem__
    cls.__delitem__ = __delitem__
    return cls


@LoggedMapping
class LoggedDict(dict):
    pass
```

è£Źäÿ łæŢĹæđIJèů§äźŃåL’■çŽĎæŸŕäÿĂæăůçŽĎïijŇèĂŇäÿŤäÿ■åĘ■éIJĂèęĄä¡£çŤĺåď’Žçżğæ L’£äžĘãĂ
åŔĆèĂČ8.13åřŔèŁĆæ§ěçIJŃæŻt’åď’ŽæůůåĚěçśżåŠŇçśżèčĚéěřåŹĺçŽĎä¿Ńå■ŘãĂĆ

10.19 8.19 åőđçŐřçŁűæĂĄåŕźèśąæĹŰèĂĚçŁűæĂĄælJž

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłçŁűæĂĄæIJžæĹŰèĂĚæŸŕåIJĺäÿ■åŘŇçŁűæĂĄäÿŃæL’ğèąŇæŞ■ä¡IJçŽĎåŕźèśąïijĺ

èğčåĘşæŰźæąĹ

åIJĺå¿Ĺåď’ŽçĺŃåžŔäÿ■ïijŇæIJL’äžŻåŕźèśąäijŽæăźæ■őçŁűæĂĄçŽĎäÿ■åŘŇæİěæL’ğèąŇäÿ■åŘŇçŽĎæŞ

```
class Connection:
    """æŹőéĂŽæŰźæąĹïijŇåě¡åd’ŽäÿłåĹd’æŰ■èŕ■åŔěïijŇæŢĹçŐĞä¡ŐäÿŃ~~"""

    def __init__(self):
        self.state = 'CLOSED'

    def read(self):
        if self.state != 'OPEN':
            raise RuntimeError('Not open')
        print('reading')

    def write(self, data):
        if self.state != 'OPEN':
            raise RuntimeError('Not open')
        print('writing')

    def open(self):
```

```
        if self.state == 'OPEN':
            raise RuntimeError('Already open')
        self.state = 'OPEN'

    def close(self):
        if self.state == 'CLOSED':
            raise RuntimeError('Already closed')
        self.state = 'CLOSED'
```

è£Źæăůå ĘŹæIJL’å¿Ĺåď’ŽçijžçĆźïijŇéęŰůĚĹæŸŕäžčçăĄåď’łåď’■æİĆäžĘïijŇåě¡åď’ŽçŽĎæİąäżűůĺĹď’æŰ■åŻăäÿžäÿĂäžŻåÿÿèğĄçŽĎæŞ■ä¡IJærŤåęĆread()ãĂĄwrite()æŕŔæñąæL’ğèąŇůL’■éČ¡éIJĂèęĄæL’ğèąŇæčĂæş

äÿĂäÿłæŻť’åě¡çŽĎůĹđæşŢæŸŕäÿžæŕŔäÿłçŁűæĂĄůőŽäźL’äÿĂäÿłůŕźèśąïijŽ

```
class Connection1:
    """æŰřæŰźæąĹâĂŤâĂŤůŕźæŕŔäÿłçŁűæĂĄůőŽäźL’äÿĂäÿłçśż"""

    def __init__(self):
        self.new_state(ClosedConnectionState)

    def new_state(self, newstate):
        self._state = newstate
        # Delegate to the state class

    def read(self):
        return self._state.read(self)

    def write(self, data):
        return self._state.write(self, data)

    def open(self):
        return self._state.open(self)

    def close(self):
        return self._state.close(self)


# Connection state base class
class ConnectionState:
    @staticmethod
    def read(conn):
        raise NotImplementedError()

    @staticmethod
    def write(conn, data):
        raise NotImplementedError()

    @staticmethod
    def open(conn):
        raise NotImplementedError()
```

```
    @staticmethod
    def close(conn):
        raise NotImplementedError()


# Implementation of different states
class ClosedConnectionState(ConnectionState):
    @staticmethod
    def read(conn):
        raise RuntimeError('Not open')

    @staticmethod
    def write(conn, data):
        raise RuntimeError('Not open')

    @staticmethod
    def open(conn):
        conn.new_state(OpenConnectionState)

    @staticmethod
    def close(conn):
        raise RuntimeError('Already closed')


class OpenConnectionState(ConnectionState):
    @staticmethod
    def read(conn):
        print('reading')

    @staticmethod
    def write(conn, data):
        print('writing')

    @staticmethod
    def open(conn):
        raise RuntimeError('Already open')

    @staticmethod
    def close(conn):
        conn.new_state(ClosedConnectionState)
```

äÿŃéİćæŸŕä¡£çŤĺæijŤçďžïijŽ

```
>>> c = Connection()
>>> c._state
<class '__main__.ClosedConnectionState'>
>>> c.read()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example.py", line 10, in read
```

```
        return self._state.read(self)
    File "example.py", line 43, in read
        raise RuntimeError('Not open')
RuntimeError: Not open
>>> c.open()
>>> c._state
<class '__main__.OpenConnectionState'>
>>> c.read()
reading
>>> c.write('hello')
writing
>>> c.close()
>>> c._state
<class '__main__.ClosedConnectionState'>
>>>
```

èőÍèőž

åęĆæđIJäżčçăĄäÿ■åĞžçŐřåď'łåď'ŽçŽĎæİąäżűåĹď'æŰ■èŕ■åŔěçŽĎèŕİïijŇäżčçăĄåřśäijŽåŔŸå¿ŮéŽ¿äžě
è£ŹéĞŇçŽĎèğčåĘşæŰźæąĹæŸŕářĘæŕŔäÿłçŁűæĂĄæŁ¡åŔŰåĞžæİěåőŽäźĽ'æĹŘäÿĂäÿłçśžãĂĆ

è£ŹéĞŇçIJŃäÿŁåŐżæIJL'çĆźåěĞæĂłïijŇæŕŔäÿłçŁűæĂĄåŕźèśąéČ¡åŔłæIJL'éİŹæĂĄæŰźæşŢïijŇåźűæš
åőđéŹĚäÿŁïijŇæL'ĂæIJL'çŁűæĂĄä£ąæĄŕéČ¡åŔłå■ŸåĆĺåIJĺ Connection
åőđä¿Ńäÿ■ãĂĆ åIJĺå§žçśżäÿ■åőŽäźĽ'çŽĎ NotImplementedError
æŸŕäÿžäžĘçąőä£Ìå■ŘçśżåőđçŐřäžĘçŻÿåžŢçŽĎæŰźæşŢãĂĆ è£ŹéĞŇä¡äæĹŰèőÿè£ŸæČşä¡£çŢĺ8.12åřŔèŁĆ

èő¿èőąæĺąåijŔäÿ■æIJL'äÿĂçğ■æĺąåijŔåŔńçŁűæĂĄæĺąåijŔïijŇè£ŹäÿĂåřŔèŁĆçőŮæŸŕäÿĂäÿłåĹİæ■ěå

10.20 8.20 éĂŽè£Ğå■ŮçňęäÿšèřČçŢĺåŕźèśąæŰźæşŢ

éŮőécŸ

ä¡äæIJL'äÿĂäÿłå■Ůçňęäÿšå¡ćåijŔçŽĎæŰźæşŢåŘ■çğřïijŇæČşéĂŽè£ĞåőČèřČçŢĺæ§ŔäÿłåŕźèśąçŽĎåŕźå

èğčåĘşæŰźæąĹ

æIJĂçőĂå■ŢçŽĎæČĚåĘţïijŇåŔŕäżěä¡£çŢĺ `getattr()` ïijŽ

```
import math

class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __repr__(self):
        return 'Point({!r:},{!r:})'.format(self.x, self.y)
```

```
    def distance(self, x, y):
        return math.hypot(self.x - x, self.y - y)


p = Point(2, 3)
d = getattr(p, 'distance')(0, 0)  # Calls p.distance(0, 0)
```

åŔęåď Űäÿ Ăçğ■æŰźæşŢæŸŕä¡£çŤĺ `operator.methodcaller()` ïijŇä¿ŃåęĆïijŽ

```
import operator
operator.methodcaller('distance', 0, 0)(p)
```

å¡Şä¡äéIJĂèęĄéĂŽè£ĞçŻÿåŘŇçŽĎåŔĆæŢřåď ŽæňąèřČçŤĺæ§ŘäÿłæŰźæşŢæŮűïijŇä¡£çŤĺ `operator.methodcaller` åŕśå¿ĹæŰźä¿£äžĘãĂĆ æŕŤåęĆä¡äéIJĂèęĄæŐŠåžŔäÿĂçşżåĹŮçŽĎçĆźïijŇåř

```
points = [
    Point(1, 2),
    Point(3, 0),
    Point(10, -3),
    Point(-5, -7),
    Point(-1, 8),
    Point(3, 2)
]
# Sort by distance from origin (0, 0)
points.sort(key=operator.methodcaller('distance', 0, 0))
```

èőĺèőž

èřČçŤĺäÿĂäÿłæŰźæşŢåőđéŹĚäÿŁæŸŕäÿď éČĺçŃňçńŃæŞ■ä¡IJïijŇçňňäÿĂæ■ěæŸŕæ§ěæĽ¿åśđæĂğïijŇç
åŻăæ■ď ïijŇäÿžäžĘèřČçŤĺæ§ŘäÿłæŰźæşŢïijŇä¡ăåŔŕäżěéęŰåĚĹéĂŽè£Ğ `getattr()`
æİěæ§ěæĽ¿åĹřè£ŹäÿłåśđæĂğïijŇçĎűåŘŐåĘ■åŐżäżěåĞ¡æŢřæŰźåijŔèřČçŤĺåőČå■şåŔŕãĂĆ

`operator.methodcaller()` åĹŻåżžäÿĂäÿłåŔŕèřČçŤĺåŕźèśąïijŇåźűåŘŇæŮűæŔŘä¿ŻæĽĂæIJĽåź
çĎűåŘŐèřČçŤĺçŽĎæŮűåĂŹåŔłéIJĂèęĄåřĘåőđä¿ŃåŕźèśąäijäéĂŠçżŹåőČå■şåŔŕïijŇæŕŤåęĆïijŽ

```
>>> p = Point(3, 4)
>>> d = operator.methodcaller('distance', 0, 0)
>>> d(p)
5.0
>>>
```

éĂŽè£ĞæŰźæşŢåŘ■çğřå■ŮçňęäÿšæİěèřČçŤĺæŰźæşŢéĂŽåÿÿåĞžçŐřåIJĺéIJĂèęĄæĺąæŃ§
`case` èŕ■åŔěæĹŰåőđçŐřèő£éŮőèĂĚæĺąåijŔçŽĎæŮűåĂŹãĂĆ
åŔĆèĂČäÿŃäÿĂåřŔèŁĆèŐůåŔŰæŻť åď ŽéńŸçžğä¿Ńå■ŘãĂĆ

10.21 8.21 åőđçŐřèő£éŮőèĂĚæÍąåijŔ

éŮőécŸ

ä¡ăèęĄåď'ĎçŘĘçŤšåď'ğéĞŔäÿ■åŘŇçśżåđŃçŽĎåŕźèšąçżĎæĹŘçŽĎåď'■æİĆæŢřæ■őçżŞæđĎïijŇæŕŔäÿĂ
æŕŤåęĆïijŇéĄ■åŐĘäÿĂäÿłæăŠå¡ćçżŞæđĎïijŇçĎűåŘŐæăźæ■őæŕŔäÿłèŁĆçĆźçŽĎçŻÿåžŤçŁűæĂĄæĽğèąŇ

èğčåĘşæŰźæąĹ

è£ŹéĞŇéĄĞåĹřçŽĎéŮőécŸåIJĺçijŰçĺŃécĘå§§äÿ■æŸŕå¿ĹæŹőéĄ■çŽĎïijŇæIJL'æŮűåĂŹäijŽæđĎåżžä
åĄĞèő¿ä¡ăèęĄåĘŹäÿĂäÿłèąĺçď'žæŢřå■ęèąĺè¿¿åijŔçŽĎçĺŃåžŔïijŇéĆčäźĹä¡ăåŔŕèČ¡éIJĂèęĄåőŽäźL'åęĆäÿŃ

```
class Node:
    pass

class UnaryOperator(Node):
    def __init__(self, operand):
        self.operand = operand

class BinaryOperator(Node):
    def __init__(self, left, right):
        self.left = left
        self.right = right

class Add(BinaryOperator):
    pass

class Sub(BinaryOperator):
    pass

class Mul(BinaryOperator):
    pass

class Div(BinaryOperator):
    pass

class Negate(UnaryOperator):
    pass

class Number(Node):
    def __init__(self, value):
        self.value = value
```

çĎűåŘŐåĹl'çŤĺè£ŹäžŽçśżæđĎåżžåţŇåěŮæŢřæ■őçżŞæđĎïijŇåęĆäÿŃæL'Ăçď'žïijŽ

```
# Representation of 1 + 2 * (3 - 4) / 5
t1 = Sub(Number(3), Number(4))
t2 = Mul(Number(2), t1)
```

```
t3 = Div(t2, Number(5))
t4 = Add(Number(1), t3)
```

è£ŹæăůåĄŽçŽĎéŮőécŸæŸŕåŕźäžŐæŕŔäÿłèąĺè¿¿åijŔïijŇæŕŔæňąéČ¡èęĄéĞ■æŰřåőŽäźĽäÿĂéĄ■ïijŇæ
è£ŹéĞŇæĹŚäżňä¡£çŤĺèő£éŮőèĂĚæĺąåijŔåŔŕäžěè¿¿åĹřè£ŹæăůçŽĎçŻőçŽĎïijŽ

```
class NodeVisitor:
    def visit(self, node):
        methname = 'visit_' + type(node).__name__
        meth = getattr(self, methname, None)
        if meth is None:
            meth = self.generic_visit
        return meth(node)

    def generic_visit(self, node):
        raise RuntimeError('No {} method'.format('visit_' +␣
↪type(node).__name__))
```

äÿžäžĘä¡£çŤĺè£ŹäÿłçśżïijŇåŔŕäžěåőŽäźĽäÿĂäÿłçśżçżğæĽ£åőČåźűäÿŤåőđçŐřåŘĎçğ■
visit_Name() æŰźæşŢïijŇåĚűäÿ■NameæŸŕnodeçśżåđŃãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăæČşæśĆèąĺè¿¿åijŔçŽĎåĂijïijŇåŔŕäžěè£ŹæăůåĘŹïijŽ

```
class Evaluator(NodeVisitor):
    def visit_Number(self, node):
        return node.value

    def visit_Add(self, node):
        return self.visit(node.left) + self.visit(node.right)

    def visit_Sub(self, node):
        return self.visit(node.left) - self.visit(node.right)

    def visit_Mul(self, node):
        return self.visit(node.left) * self.visit(node.right)

    def visit_Div(self, node):
        return self.visit(node.left) / self.visit(node.right)

    def visit_Negate(self, node):
        return -node.operand
```

ä¡£çŤĺçd'žä¿ŃïijŽ

```
>>> e = Evaluator()
>>> e.visit(t4)
0.6
>>>
```

ä¡IJäÿžäÿĂäÿłäÿ■åŘŇçŽĎä¿Ńå■ŘïijŇäÿŃéİćåőŽäźĽäÿĂäÿłçśżåIJĺäÿĂäÿłæăĹäÿŁéİćářĘäÿĂäÿłèąĺè¿¿å

```
class StackCode(NodeVisitor):
    def generate_code(self, node):
        self.instructions = []
        self.visit(node)
        return self.instructions

    def visit_Number(self, node):
        self.instructions.append(('PUSH', node.value))

    def binop(self, node, instruction):
        self.visit(node.left)
        self.visit(node.right)
        self.instructions.append((instruction,))

    def visit_Add(self, node):
        self.binop(node, 'ADD')

    def visit_Sub(self, node):
        self.binop(node, 'SUB')

    def visit_Mul(self, node):
        self.binop(node, 'MUL')

    def visit_Div(self, node):
        self.binop(node, 'DIV')

    def unaryop(self, node, instruction):
        self.visit(node.operand)
        self.instructions.append((instruction,))

    def visit_Negate(self, node):
        self.unaryop(node, 'NEG')
```

ä¡£çŤĺçď žä¿ŃïijŽ

```
>>> s = StackCode()
>>> s.generate_code(t4)
[('PUSH', 1), ('PUSH', 2), ('PUSH', 3), ('PUSH', 4), ('SUB',),
('MUL',), ('PUSH', 5), ('DIV',), ('ADD',)]
>>>
```

èőĺèőž

åĹŽåijĂåğŃçŽĎæŮűåĂŹä¡ăåŔŕèČ¡äijŽåĘŹåď ğéĞŔçŽĎif/elseèŕ■åŔěæİěåőđçŐřïijŇè£ŹéĞŇèő£éŮőèĂĚæĺąåijŔçŽĎåě¡åď ĎåřśæŸŕéĂŽè£Ğ `getattr()` æİěèŐůåŔŰçŻÿåžŢçŽĎæŰźæşŢïijŇåźűåĹl’çŤĺéĂŠå¡ŠæİěéĄ■åŐĘæL’ĂæIJL’çŽĎèŁĆçĆźïijŽ

```
def binop(self, node, instruction):
    self.visit(node.left)
```

```
    self.visit(node.right)
    self.instructions.append((instruction,))
```

è£ŸæIJL'äÿĂçĆźéIJĂèęĄæŇĞåĞžçŽĎæŸŕïijŇè£Źçğ■æŁĂæIJŕäź§æŸŕåőđçŐřåĔűäżŰèŕ■èĺĂäÿ■switch
æŕŤåęĆïijŇåęĆæđIJä¡ăæ■čåIJĺåĘŹäÿĂäÿłHTTPæąĘæđűïijŇä¡ăåŔŕèČ¡äijŽåĘŹè£Źæăůäÿ ÄäÿłèŕůæśĆåĹĘåŔ

```
class HTTPHandler:
    def handle(self, request):
        methname = 'do_' + request.request_method
        getattr(self, methname)(request)
    def do_GET(self, request):
        pass
    def do_POST(self, request):
        pass
    def do_HEAD(self, request):
        pass
```

èő£éŮőèĂĔæĺąåijŔäÿĂäÿłçijžçĆźåřśæŸŕåőČäÿěéĞ■ä¿İèţŰéĂŠå¡ŠïijŇåęĆæđIJæŢřæ■őçżŞæđĎåţŇåěŮ
æIJL'æŮűåĂŹäijŽèűĚè£ĞPythonçŽĎéĂŠå¡ŠæůśåžęéŹŘåĹű(åŔĆèĂČ `sys.getrecursionlimit()`)ãĂĆ

åŔŕäżěåŔĆçĚğ8.22åřŔèŁĆïijŇåĹl'çŤĺçŤ§æĹŘåŹĺæĹŰè£■äżčåŹĺæİěåőđçŐřéİđéĂŠå¡ŠéĄ■åŐĘçőŮæşŢ

åIJĺèů§èğčæđŘåŠŇçijŰèŕŚçŻÿåĚşçŽĎçijŰçĺŃäÿ■ä¡£çŤĺèő£éŮőèĂĔæĺąåijŔæŸŕéİđåÿÿåÿÿèğĄçŽĎãĂĆ
PythonæIJňèžńçŽĎ `ast` æĺąåİŮåĂijçŽĎåĚşæşĺäÿŃïijŇåŔŕäżěåŐżçIJŃçIJŃæžŘçăĄãĂĆ
9.24åřŔèŁĆæijŤçd'žäžĘäÿĂäÿłåĹl'çŤĺ `ast` æĺąåİŮæİěåd'ĎçŘĘPythonæžŘäżčçăĄçŽĎä¿Ńå■ŘãĂĆ

10.22 8.22 äÿ■çŤĺéĂŠå¡ŠåőđçŐřèő£éŮőèĂĔæĺąåijŔ

éŮőécŸ

ä¡ăä¡£çŤĺèő£éŮőèĂĔæĺąåijŔéĄ■åŐĘäÿĂäÿłå¿ĹæůśçŽĎåţŇåěŮæăŠå¡ćæŢřæ■őçżŞæđĎïijŇåźűäÿŤåŻäă
ä¡ăæČşæűĹéŹd'éĂŠå¡ŠïijŇåźűåŘŇæŮűä£ÍæŇĄèő£éŮőèĂĔçijŰçĺŃæĺąåijŔãĂĆ

èğčåĘşæŰźæąĹ

éĂŽè£Ğåůğåę́ŹçŽĎä¡£çŤĺçŤ§æĹŘåŹĺåŕäżěåIJĺæăŚéĄ■åŐĘæĹŰæŘIJçt'ćçőŮæşŢäÿ■æűĹéŹd'éĂŠå¡Š
åIJĺ8.21åřŔèŁĆäÿ■ïijŇæĹŚäžňçžŹåĞžäžĘäÿĂäÿłèő£éŮőèĂĔçśžãĂĆ
äÿŃéİćæĹŚäžňåĹl'çŤĺäÿĂäÿłæăĹåŠŇçŤ§æĹŘåŹĺéĞ■æŰřåőđçŐřè£Źäÿłçśžïijž

```
import types

class Node:
    pass

class NodeVisitor:
    def visit(self, node):
        stack = [node]
```

```
        last_result = None
        while stack:
            try:
                last = stack[-1]
                if isinstance(last, types.GeneratorType):
                    stack.append(last.send(last_result))
                    last_result = None
                elif isinstance(last, Node):
                    stack.append(self._visit(stack.pop()))
                else:
                    last_result = stack.pop()
            except StopIteration:
                stack.pop()

        return last_result

    def _visit(self, node):
        methname = 'visit_' + type(node).__name__
        meth = getattr(self, methname, None)
        if meth is None:
            meth = self.generic_visit
        return meth(node)

    def generic_visit(self, node):
        raise RuntimeError('No {} method'.format('visit_' +␣
↪type(node).__name__))
```

åęĆædIJä¡ăä¡£çŤĺè£Źäÿłçśżïij Ñäź§èČ¡è¿¿åĹřçŻÿåŘŇçŽĎæŢĹædIJãĂĆäžŃåőđäÿŁä¡ăåőŇåĚĺåŔřäżěåřĿ
èĂČèŹŚåęĆäÿŃäżčçăĄïijŇéĄ■åŐĘäÿĂäÿłèąĺè¿¿åijŔçŽĎæăŚïijŽ

```
class UnaryOperator(Node):
    def __init__(self, operand):
        self.operand = operand

class BinaryOperator(Node):
    def __init__(self, left, right):
        self.left = left
        self.right = right

class Add(BinaryOperator):
    pass

class Sub(BinaryOperator):
    pass

class Mul(BinaryOperator):
    pass

class Div(BinaryOperator):
    pass
```

```
class Negate(UnaryOperator):
    pass

class Number(Node):
    def __init__(self, value):
        self.value = value

# A sample visitor class that evaluates expressions
class Evaluator(NodeVisitor):
    def visit_Number(self, node):
        return node.value

    def visit_Add(self, node):
        return self.visit(node.left) + self.visit(node.right)

    def visit_Sub(self, node):
        return self.visit(node.left) - self.visit(node.right)

    def visit_Mul(self, node):
        return self.visit(node.left) * self.visit(node.right)

    def visit_Div(self, node):
        return self.visit(node.left) / self.visit(node.right)

    def visit_Negate(self, node):
        return -self.visit(node.operand)

if __name__ == '__main__':
    # 1 + 2*(3-4) / 5
    t1 = Sub(Number(3), Number(4))
    t2 = Mul(Number(2), t1)
    t3 = Div(t2, Number(5))
    t4 = Add(Number(1), t3)
    # Evaluate it
    e = Evaluator()
    print(e.visit(t4))  # Outputs 0.6
```

åęĆædIJåţŇåěŮåśĆæňąåd'łæůśéĆčäźĹäýŁè£řçŽĎEvaluatorářšäijŽåd'śæŢĹïijŽ

```
>>> a = Number(0)
>>> for n in range(1, 100000):
...     a = Add(a, Number(n))
...
>>> e = Evaluator()
>>> e.visit(a)
Traceback (most recent call last):
...
  File "visitor.py", line 29, in _visit
    return meth(node)
```

```
    File "visitor.py", line 67, in visit_Add
return self.visit(node.left) + self.visit(node.right)
RuntimeError: maximum recursion depth exceeded
>>>
```

çŐřåIJÍæĹŚäžňçÍ■å¿őä£őæŤźäÿŃäÿŁéİćçŽĎEvaluatorïijŽ

```
class Evaluator(NodeVisitor):
    def visit_Number(self, node):
        return node.value

    def visit_Add(self, node):
        yield (yield node.left) + (yield node.right)

    def visit_Sub(self, node):
        yield (yield node.left) - (yield node.right)

    def visit_Mul(self, node):
        yield (yield node.left) * (yield node.right)

    def visit_Div(self, node):
        yield (yield node.left) / (yield node.right)

    def visit_Negate(self, node):
        yield - (yield node.operand)
```

åĘ■æňąè£ŘèąŇïijŇářśäÿ■äijŽæŁěéŤŹäžĘïijŽ

```
>>> a = Number(0)
>>> for n in range(1,100000):
...     a = Add(a, Number(n))
...
>>> e = Evaluator()
>>> e.visit(a)
4999950000
>>>
```

åęĆæđIJä¡ăè£ŸæČşæůżåŁăåĚűäžŰèĞłåőŽäźĽéĂżè¿Śäź§æšąéŮőécŸïijŽ

```
class Evaluator(NodeVisitor):
    ...
    def visit_Add(self, node):
        print('Add:', node)
        lhs = yield node.left
        print('left=', lhs)
        rhs = yield node.right
        print('right=', rhs)
        yield lhs + rhs
    ...
```

äÿŃéİćæŸŕçőĂå■ŢçŽĎæţŃèŕŢïijŽ

```
>>> e = Evaluator()
>>> e.visit(t4)
Add: <__main__.Add object at 0x1006a8d90>
left= 1
right= -0.4
0.6
>>>
```

èőÍèőž

è£ŹäÿĂåřŔèŁĆæĹŚäžňæijŢçd'žäžĘçŢ§æĹŘåŹĺåŠŇå■ŔçÍŃåIJÍçÍŃåžŔæŐğåĹűæţĄæŰźéİćçŽĎåijžåd'ğ
éĄ£åĚ■éĂŠå¡ŠçŽĎäÿĂäÿłéĂŽåÿÿæŰźæşŢæŸŕä¡£çŢĺäÿĂäÿłæăĹæĹŰéŸ§åĹŮçŽĎæŢřæ■őçżŞæđĎãĂĆ
ä¿ŃåęĆïijŇæůšåžęäijŸåĚĹçŽĎéĄ■åŐĘçőŮæşŢïijŇçňňäÿĂæňąçćřåĹřäÿĂäÿłèŁĆçĆźæŮűåřĘåĚűåŐŃåĚěæă
æŰźæşŢçŽĎæäÿå£ČæĂİèůŕåřśæŸŕè£Źæăůã ĂĆ

åŔęåd'ŰäÿĂäÿłéIJĂèęĄçŘĘèğčçŽĎåřśæŸŕçŢ§æĹŘåŹĺäÿ■yieldèŕ■åŔěãĂĆå¡Şçćřå Ĺřyieldèŕ■åŔěæŮűïij
äÿŁéİćçŽĎä¿Ńå■Řä¡£çŢĺè£ŹäÿłæŁĂæIJŕæİěäžčæŻ£äžĘéĂŠå¡ŠãĂĆä¿ŃåęĆïijŇäźŃåL'■æĹŚäžňæŸŕè£Źæăů

```
value = self.visit(node.left)
```

çŐřåIJÍæ■ćæĹŘyieldèŕ■åŔëijŽ

```
value = yield node.left
```

åőČäijŽåřĘ `node.left` è£ŤåŻđçżŹ `visit()` æŰźæşŢïijŇçĎűåŘŐ `visit()` æŰźæşŢèřČçŢÍéĆčäÿłèŁĆçĆźçŻÿåžŢçŽĎ `visit_Name()` æŰźæşŢãĂĆ yield-æŽĆæŮűåřĘçÍŃåžŔæŐğåĹűåŹÍèőľ'åĞžçżŹèřČçŢÍèĂĚïijŇå¡ŞæL'ğèąŇåőŇåŘŐïijŇçżŞæđIJäijŽèţŃåĂijçżŹv

çIJŃåőŇè£ŹäÿĂåřŔèŁĆïijŇä¡ääź§èőÿæČşåŐżåŕżæL'¿åĚűåőČæšąæIJL'yieldèŕ■åŔęçŽĎæŰźæąĹãĂĆä¡E
ä¿ŃåęĆïijŇäÿžäžĘæűĹéŹđ'éĂŠå¡ŠïijŇä¡ää£ĚéążèęĄçżt'æŁđ'äÿĂäÿłæăĹçżŞæđĎïijŇåęĆæđIJäÿ■ä¡£çŢÍçŢ§a
åőđéŹĚäÿŁïijŇä¡£çŢÍyieldèŕ■åŔěåŔŕäžěèőľ'ä¡ääĘŹåĞžéİđåÿÿæijĆäžőçŽĎäżčçăĄïijŇåőČæűĹéŹđ'äžĘéĂŠå¡

10.23 8.23 å¿łçŐŕåijŢçŢÍæŢřæ■őçżŞæđĎçŽĎåĘĚå■Ÿçőąçŕę

éŮőécŸ

ä¡ăçŽĎçÍŃåžŔåĹŻåżžäžĘå¿Ĺåđ'Žå¿łçŐŕåijŢçŢÍæŢřæ■őçżŞæđĎ(æŕŤåęĆæăŚãĂĄåŻ¿ãĂĄèğĆår§èĂĚæ

èğčåĘşæŰźæąĹ

äÿĂäÿłçőĂå■ŢçŽĎå¿łçŐŕåijŢçŢÍæŢřæ■őçżŞæđĎä¿Ńå■Řåřśæ Ÿŕäÿ Ăäÿłæăŝå¡ćçżŞæđĎïijŇåŔŇäžšèŁĆ
è£Źçğ■æČĚåĘţäÿŃïijŇåŔŕäżěèĂČèŹŚä¡£çŢÍ `weakref` åžŞäÿ■çŽĎåijśåijŢçŢÍãĂĆä¿ŃåęĆïijŽ

```
import weakref

class Node:
```

```
    def __init__(self, value):
        self.value = value
        self._parent = None
        self.children = []

    def __repr__(self):
        return 'Node({!r:})'.format(self.value)

    # property that manages the parent as a weak-reference
    @property
    def parent(self):
        return None if self._parent is None else self._parent()

    @parent.setter
    def parent(self, node):
        self._parent = weakref.ref(node)

    def add_child(self, child):
        self.children.append(child)
        child.parent = self
```

è£Źçğ■æŸŕæČşæŰźåijŔåĚĄèőÿparentéİŹéżŸçżĹæ■ćãĂĆä¿ŃåęĆïijŽ

```
>>> root = Node('parent')
>>> c1 = Node('child')
>>> root.add_child(c1)
>>> print(c1.parent)
Node('parent')
>>> del root
>>> print(c1.parent)
None
>>>
```

èőÍèőž

å¿łçŐŕåijŢçŤÍçŽĎæŢřæ■őçżŞæđĎåIJÍPythonäÿ■æŸŕäÿĂäÿłå¿ĹæčŸæĽŃçŽĎéŮőécŸïijŇåŻääÿžæ■čåÿä¿ŃåęĆèĂČèŹŚåęĆäÿŃäżčçăĄïijŽ

```
# Class just to illustrate when deletion occurs
class Data:
    def __del__(self):
        print('Data.__del__')

# Node class involving a cycle
class Node:
    def __init__(self):
        self.data = Data()
        self.parent = None
        self.children = []
```

```
    def add_child(self, child):
        self.children.append(child)
        child.parent = self
```

äÿŃéİćæĹŚäżñä¡£çŤĺè£ŹäÿłäżčçăĄæİěåĄŽäÿĂäžŽåđČåIJ¿åŻđæŤűèŕŢéłŃïijŽ

```
>>> a = Data()
>>> del a # Immediately deleted
Data.__del__
>>> a = Node()
>>> del a # Immediately deleted
Data.__del__
>>> a = Node()
>>> a.add_child(Node())
>>> del a # Not deleted (no message)
>>>
```

åŘŕäżěçIJŃåĹřïijŇæIJĂåŘŐäÿĂäÿłçŽĎåĹăéŹď'æŮűæL'Şå■řèŕ■åŘěæšąæIJL'åĞžçŐřãĂĆåŐ§åŻăæŸŕPy
å¡ŞäÿĂäÿłåŕźèśąçŽĎåijŢçŤĺæŢřåŔŸæĹŘ0çŽĎæŮűåĂŹæL'■äijŽçńŇå■şåĹăéŹď'æŐL'ãĂĆèĂŇåŕźäžŐå¿łçŐŕ
åŻăæ■d'ïijŇåIJĺäÿŁéİć㿊Ňå■Řäÿ■æIJĂåŘŐéČĺåĹĘïijŇçĹűèŁĆçĆźåŠŇå■l'å■ŘèŁĆçĆźäžŠçŻÿæŃěæIJL'åŕźa

PythonæIJL'åŘęåď'ŰçŽĎåđČåIJ¿åŻđæŤűåŹĺæİěäÿŞéŮĺéŠĹåŕźå¿łçŐŕåijŢçŤĺçŽĎïijŇä¡ĘæŸŕä¡ăæřÿè£IJ
åŘęåď'Űä¡ăè£ŸåŘŕäżěæL'ŃåŁĺçŽĎèğęåŘŠåőČïijŇä¡ĘæŸŕäżčçăĄçIJŃäÿŁåŐżå¿ĹæŇńïijŽ

```
>>> import gc
>>> gc.collect() # Force collection
Data.__del__
Data.__del__
>>>
```

åęĆæđIJå¿łçŐŕåijŢçŤĺçŽĎåŕźèśąèĞłåůśè£ŸåőŽäźL'äžĘèĞłåůśçŽĎ
__del__() æŰźæşŢïijŇéĆčäźĹäijŽèőľ'æČĚåĘţåŘŸå¿ŮæŻt'çş§çşŢãĂĆ
åĄĞèő¿ä¡ăåČŔäÿŃéİć㣹ŹæăůçżŹNodeåőŽäźL'èĞłåůśçŽĎ __del__() æŰźæşŢïijŽ

```
# Node class involving a cycle
class Node:
    def __init__(self):
        self.data = Data()
        self.parent = None
        self.children = []

    def add_child(self, child):
        self.children.append(child)
        child.parent = self

    # NEVER DEFINE LIKE THIS.
    # Only here to illustrate pathological behavior
    def __del__(self):
        del self.data
        del.parent
```

```
        del.children
```

è£Źçğ■æČĚåĘţäÿŃïijŇåđČåIJ¿åŻđæŤűæřÿè£IJéČ¡äÿ■äijŽåŐżåŻđæŤűè£ŹäÿłářźèśąçŽĎïijŇè£ŸäijŽářij
åęĆæđIJä¡ăèřŢçİĂåŐżè£ŘèąŇåőČäijŽåŔŚçŐřïijŇData.__del__
æűĹæĄŕæřÿè£IJäÿ■äijŽåĞžçŐřäžĘ,çŤŽèĞşåIJĺä¡ăåijžåĹűåĘĚå■ŸåŻđæŤűæŮűïijŽ

```
>>> a = Node()
>>> a.add_child(Node())
>>> del a # No message (not collected)
>>> import gc
>>> gc.collect() # No message (not collected)
>>>
```

åijśåijŢçŤĺæűĹéŹd'äžĘåijŢçŤĺå¿łçŐŕçŽĎè£ŹäÿłéŮőécŸïijŇæIJňèt'ĺæİèèőšïijŇåijśåijŢçŤĺářśæŸŕäÿĂäÿ
ä¡ăåŔŕäžěéĂŽè£Ğ weakref æİěåĹŽåżžåijśåijŢçŤĺãĂĆä¿ŃåęĆïijŽ

```
>>> import weakref
>>> a = Node()
>>> a_ref = weakref.ref(a)
>>> a_ref
<weakref at 0x100581f70; to 'Node' at 0x1005c5410>
>>>
```

äÿžäžĘèő£éŮőåijśåijŢçŤĺæL'ĂåijŢçŤĺçŽĎåŕźèśąïijŇä¡ăåŔŕäžěåČŔåĞ¡æŢřäÿĂæăůåŐżèřČçŤĺåőČå■şåŔ
çŤśäžŐåŐ§ågŃåŕźèśąçŽĎåijŢçŤĺèőąæŢřæšąæIJL'åćđåŁäïijŇéĆčäźĹåřśåŔŕäžěåŐżåĹăéŹd'åőČäžĘãĂĆä¿Ńåę

```
>>> print(a_ref())
<__main__.Node object at 0x1005c5410>
>>> del a
Data.__del__
>>> print(a_ref())
None
>>>
```

éĂŽè£Ğè£ŹéĞŇæijŤçd'žçŽĎåijśåijŢçŤĺæŁĂæIJŕïijŇä¡ăäijŽåŔŚçŐřäÿ■åĘ■æIJL'å¿łçŐŕåijŢçŤĺéŮőécŸ
ä¡ăè£ŸèČ¡åŔĆèĂČ8.25åřŔèŁĆåĚşäžŐåijśåijŢçŤĺçŽĎåŔęåd'ŰäÿĂäÿłä¿Ńå■ŘãĂĆ

10.24 8.24 èőľçśżæŤŕæŇĄæŕŤè¿ČæŞ■ä¡IJ

éŮőécŸ

ä¡ăæČşèőľæ§ŘäÿłçśżçŽĎåőđä¿ŃæŤŕæŇĄæăĞåĞĘçŽĎæŕŤè¿Čè£ŘçőŮ(æŕŤåęĆ>=,!=,<=,<ç■L')ïijŇä¡E

èğčåĘşæŰźæąĹ

PythonçśżåŕźæŕŔäÿłæŕŤè¿ČæŞ■ä¡IJéČ¡éIJĂèęĄåőđçŐřäÿĂäÿłçL'źæőŁæŰźæşŢæİěæŤŕæŇĄãĂĆ
ä¿ŃåęĆäÿžäžĘæŤŕæŇĄ>=æŞ■ä¡IJçňęïijŇä¡ăéIJĂèęĄåőŽäźL'äÿĂäÿł __ge__()
æŰźæşŢãĂĆ år¡çőąåőŽäźL'äÿĂäÿłæŰźæşŢæšąäżĂäźĹéŮőécŸïijŇä¡ĘåęĆæđIJèęĄä¡ăåőđçŐřæL'ĂæIJL'åŔŕè

èčĚéěřåŹÍ `functools.total_ordering` åřśæŸŕçŤÍæİěçőĂåŇŰè£Źäÿłåď'ĎçŘĘçŽĎãĂĆ ä¡£çŤÍåőČæİěèčĚéěřäÿĂäÿłæİěïijŇä¡ăåŔłéIJĂåőŽäźL'äÿĂäÿł `__eq__()` æŰźæşŢïijŇ åď'ŰåŁăåĚűäżŰæŰźæşŢ(__lt__, __le__, __gt__, or __ge__)äÿ■çŽĎäÿĂäÿłå■şåŘŕãĂĆ çĎűåŘŐèčĚéěřåŹĺäijŽèĞłåŁĺäÿžä¡ăåąńåĚĚåĚűåőČæŕŤè¿ČæŰźæşŢãĂĆ

ä¡IJäÿžä¿Ńå■ŘïijŇæĹŚäżňædĎåżžäÿĂäžŽæĹ£å■ŘïijŇçĎűåŘŐçżŹåőČäżňáćđåŁääÿĂäžŽæĹ£éŮt'ïijŇæ

```
from functools import total_ordering

class Room:
    def __init__(self, name, length, width):
        self.name = name
        self.length = length
        self.width = width
        self.square_feet = self.length * self.width

@total_ordering
class House:
    def __init__(self, name, style):
        self.name = name
        self.style = style
        self.rooms = list()

    @property
    def living_space_footage(self):
        return sum(r.square_feet for r in self.rooms)

    def add_room(self, room):
        self.rooms.append(room)

    def __str__(self):
        return '{}: {} square foot {}'.format(self.name,
                self.living_space_footage,
                self.style)

    def __eq__(self, other):
        return self.living_space_footage == other.living_space_
↪footage

    def __lt__(self, other):
        return self.living_space_footage < other.living_space_
↪footage
```

è£ŹéĞŇæĹŚäżňåŘłæŸŕçżŹHouseçśżåőŽäźL'äžĘäÿď'äÿłæŰźæşŢïijŽ`__eq__()` åŠŇ `__lt__()` ïijŇåőČåřśèČ¡æŤŕæŇĄæL'ĂæIJL'çŽĎæŕŤè¿ČæŞ■ä¡IJïijŽ

```
# Build a few houses, and add rooms to them
h1 = House('h1', 'Cape')
h1.add_room(Room('Master Bedroom', 14, 21))
h1.add_room(Room('Living Room', 18, 20))
h1.add_room(Room('Kitchen', 12, 16))
```

```
h1.add_room(Room('Office', 12, 12))
h2 = House('h2', 'Ranch')
h2.add_room(Room('Master Bedroom', 14, 21))
h2.add_room(Room('Living Room', 18, 20))
h2.add_room(Room('Kitchen', 12, 16))
h3 = House('h3', 'Split')
h3.add_room(Room('Master Bedroom', 14, 21))
h3.add_room(Room('Living Room', 18, 20))
h3.add_room(Room('Office', 12, 16))
h3.add_room(Room('Kitchen', 15, 17))
houses = [h1, h2, h3]
print('Is h1 bigger than h2?', h1 > h2) # prints True
print('Is h2 smaller than h3?', h2 < h3) # prints True
print('Is h2 greater than or equal to h1?', h2 >= h1) # Prints False
print('Which one is biggest?', max(houses)) # Prints 'h3: 1101-
↪square-foot Split'
print('Which is smallest?', min(houses)) # Prints 'h2: 846-square-
↪foot Ranch'
```

èőÍèőž

åĚűåőđ `total_ordering` èčĚéěřåŹÍäź§æšąéĆčäźĹçěđçğŸãĂĆ åőČåřśæŸŕåőŽäźĽäžĘäÿĂäÿłäżŐæŕŔäÿłæŕŤè¿ČæŤŕæŇĄæŰźæşŢåĹřæĽĂæIJĽéIJĂèęĄåőŽäźĽçŽĎåĚűäżŰ æŕŤåęĆä¡ăåőŽäźĽäžĘ `__le__()` æŰźæşŢïijŇéĆčäźĹåőČåřśèćńçŤÍæÌěæđĎåžžæĽĂæIJĽåĚűäżŰçŽĎéIJĂè åőđéŹĚäÿŁåřśæŸŕåIJÍçśžéĞŇéÌćåČŔäÿŇéÌćè£ŹæăůåőŽäźĽäžĘäÿĂäžŻçĽźæőŁæŰźæşŢïijŽ

```
class House:
    def __eq__(self, other):
        pass
    def __lt__(self, other):
        pass
    # Methods created by @total_ordering
    __le__ = lambda self, other: self < other or self == other
    __gt__ = lambda self, other: not (self < other or self == other)
    __ge__ = lambda self, other: not (self < other)
    __ne__ = lambda self, other: not self == other
```

å¡ŞçĎűïijŇä¡äèĞłåůśåŐżåĘŹäź§å¿ĹåőźæŸŞïijŇä¡ĘæŸŕä¡£çŤÍ `@total_ordering` åŘŕäžěçőĂåŇŰäžčçăĄïijŇä¡ŢäźŘèĂŇäÿ■äÿžåŚćãĂĆ

10.25 8.25 åĹŻåžžçijŞå■Ÿåőđä¿Ń

éŮőécŸ

åIJĺåĹŻåžžäÿĂäÿłçśžçŽĎåŕźèśąæŮűïijŇåęĆæđIJäźŃåL’■ä¡£çŤĺåŘŇæăůåŔĆæŢřåĹŻåžžè£Ğè£Źäÿłåŕźèś ä¡ăæČşè£ŤåŻđåőČçŽĎçijŞå■ŸåijŢçŤÍãĂĆ

èğčåĘşæŰźæąĹ

è£Źçğ■éĂŽåÿÿæŸŕåŽääÿžä¡ăåÿŇæIJŻçŻÿåŘŇåŔĆæŢřåĹŻåžžçŽĎåŕźèśąæŮűå■Ţä¿ŃçŽĎãĂĆ åIJĺå¿Ĺåď'ŽåžŞäÿ■éČ¡æIJL'åőđéŹĚçŽĎä¿Ńå■ŘïijŇærŤåęĆ `logging` æĺąåİŮïijŇä¡£çŤĺçŻÿåŘŇçŽĎåŘ■çğřåĹŻåžžçŽĎ `logger` åőđä¿ŃærÿèIJåŔłæIJL'äÿĂäÿłãĂĆä¿ŃåęĆïijŽ

```
>>> import logging
>>> a = logging.getLogger('foo')
>>> b = logging.getLogger('bar')
>>> a is b
False
>>> c = logging.getLogger('foo')
>>> a is c
True
>>>
```

äÿžäžĘè¿¿åĹřè£ŹæăůçŽĎæŢĹæđIJïijŇä¡äéIJĂèęĄä¡£çŤĺäÿĂäÿłåŠŇçśżæIJňèžńåĹĘåijĂçŽĎåůěåŐĆåĞ

```
# The class in question
class Spam:
    def __init__(self, name):
        self.name = name

# Caching support
import weakref
_spam_cache = weakref.WeakValueDictionary()
def get_spam(name):
    if name not in _spam_cache:
        s = Spam(name)
        _spam_cache[name] = s
    else:
        s = _spam_cache[name]
    return s
```

çĎűåŘŐåĄŽäÿĂäÿłæţŃèŕŢïijŇä¡ääijŽåŘŠçŐřèů§äźŃåL'■éĆčäÿłæŮěå£ŮåŕźèśąçŽĎåĹŻåžžèąŇäÿžæŸŕ

```
>>> a = get_spam('foo')
>>> b = get_spam('bar')
>>> a is b
False
>>> c = get_spam('foo')
>>> a is c
True
>>>
```

èőÍèőž

çijŰåĘŹäÿĂäÿłåůěåŐĆåĞ¡æŢřæİěä£őæŤźæŹőéĂŽçŽĎåőđä¿ŃåĹŻåžžèąŇäÿžéĂŽåÿÿæŸŕäÿĂäÿłærŤè¿ ä¡ĘæŸŕæĹŠäžňè£ŸèČ¡åŘęæL'¿åĹřæŻť'äijŸéŻĚçŽĎèğčåĘşæŰźæąĹåŚćïij§

ä¿ŃåęĆïijŇä¡ăåŔŕèČ¡äijŽèĂČèŹŚéĞ■æŰřåőŽäźĽçśżçŽĎ `__new__()` æŰźæşŢïijŇåřśåČŔäÿŃéİćè£ŹæăůïijŽ

```
# Note: This code doesn't quite work
import weakref

class Spam:
    _spam_cache = weakref.WeakValueDictionary()
    def __new__(cls, name):
        if name in cls._spam_cache:
            return cls._spam_cache[name]
        else:
            self = super().__new__(cls)
            cls._spam_cache[name] = self
            return self
    def __init__(self, name):
        print('Initializing Spam')
        self.name = name
```

åĹİçIJŃètůæİěåě¡åČŔåŔŕäżěè¿¿åĹřécĎæIJ§æŢĹæđIJïijŇä¡ĘæŸŕéŮőécŸæŸŕ `__init__()` æŕŔæñąéČ¡äijŽècńěřČçŤĺïijŇäÿ■çőąè£Źäÿłåőđä¿ŃæŸŕåŘęècńçijŞå■ŸäžĘãĂĆä¿ŃåęĆïijŽ

```
>>> s = Spam('Dave')
Initializing Spam
>>> t = Spam('Dave')
Initializing Spam
>>> s is t
True
>>>
```

è£ŹäÿłæĹŰèőÿäÿ■æŸŕä¡ăæČşèęĄçŽĎæŢĹæđIJïijŇåŻăæ■d'è£Źçğ■æŰźæşŢåźűäÿ■åŔŕåŘŰãĂĆ

äÿŁéİćæĹŚäżñä¡£çŤĺåĹřäžĘåijśåijŢçŤĺèőąæŢřïijŇårźäžŐåđČåIJ¿åŻđæŤűæİèèőšæŸŕå¿ĹæIJL'åÿőåŁl'çŽ å¡ŞæĹŚäżñä£İæŇĄåőđä¿ŃçijŞå■ŸæŮűïijŇä¡ăåŔŕèČ¡åŔłæČşåIJÍçÍŃåžŔäÿ■ä¡£çŤĺåĹřåőČäžñæŮűæL'■ä£İå■ äÿĂäÿł `WeakValueDictionary` åőđä¿ŃåŔłäijŽä£İå■ŸéĆčäžŻåIJĺåĚűåőČåIJřæŰźè£ŸåIJÍècńä¡£çŤĺçŽĎå åŘęåĹŹçŽĎèŕİïijŇåŔłèęĄåőđä¿Ńäÿ■åĘ■èćńä¡£çŤĺäžĘïijŇåőČårśäżŐå■ŮåĚÿäÿ■èćńçğżéŹd'äžĘãĂĆèğĆår§ă

```
>>> a = get_spam('foo')
>>> b = get_spam('bar')
>>> c = get_spam('foo')
>>> list(_spam_cache)
['foo', 'bar']
>>> del a
>>> del c
>>> list(_spam_cache)
['bar']
>>> del b
>>> list(_spam_cache)
[]
>>>
```

ařźäžŐåd'ğéČĺåĹĘçÍŃåžŔèĂŇåůšïijŇè£ŹéĞŇäžčçăĄåůšçżŘåd'§çŤĺäžĘãĂĆäÿ■è£Ğè£ŸæŸŕæIJL'äÿĂä

éęŰåĚĹæŸŕè£ŹéĞŇä¡£çŤĺåĹřäžĘäÿĂäÿłåĚĺåśĂåŔŸéĞŔïijŇåźűäÿŤåůěåŐĆåĞ¡æŢřèů§çśżæŤ¿åIJĺäÿĂ

```
import weakref

class CachedSpamManager:
    def __init__(self):
        self._cache = weakref.WeakValueDictionary()

    def get_spam(self, name):
        if name not in self._cache:
            s = Spam(name)
            self._cache[name] = s
        else:
            s = self._cache[name]
        return s

    def clear(self):
            self._cache.clear()

class Spam:
    manager = CachedSpamManager()
    def __init__(self, name):
        self.name = name

    def get_spam(name):
        return Spam.manager.get_spam(name)
```

è£ŹæăůçŽĎèŕÍäžčçăĄæŻť æÿĚæŹřïijŇåźűäÿŤäź§æŻť çĄţæť żïijŇæĹŚäžňåŔŕäžěåćđåŁăæŻť åď ŽçŽĎçij

è£ŸæIJL’äÿĂçĆźářśæŸŕïijŇæĹŚäžňæŻť éIJšäžĘçśżçŽĎåőđä¿ŃåŇŰçżŹçŤĺæĹůïijŇçŤĺæĹůå¿Ĺåőźæ ŸŞ

```
>>> a = Spam('foo')
>>> b = Spam('foo')
>>> a is b
False
>>>
```

æIJL’åĞăçğ■æŰźåijŔåŔŕäžěéŸšæ■ćçŤĺæĹůè£ŹæăůåĄŽïijŇçňňäÿĂäÿłæŸŕårĘçśżçŽĎåŘ■å■Ůä£őæŤźäÿ çňňäžŇçğ■ářśæŸŕèőľè£ŹäÿłçśżçŽĎ `__init__()` æŰźæşŢæŁŻåĞžäÿĂäÿłåijĆåÿÿïijŇèőľåőČäÿ■èČ¡èćńåĹ

```
class Spam:
    def __init__(self, *args, **kwargs):
        raise RuntimeError("Can't instantiate directly")

    # Alternate constructor
    @classmethod
    def _new(cls, name):
        self = cls.__new__(cls)
        self.name = name
```

çĎűåŘŐä£őæŤźçijŞå■ŸçőąçŘĘåŹĺäžčçăĄïijŇä¡£çŤĺ `Spam._new()` æĺěåĹŻåžžåőđä¿ŃïijŇèĂŇäÿ■æŸŕçŻť æŐěěřČçŤĺ `Spam()` æđĎéĂäåĞ¡æŢřïijŽ

```
# ------------------------æIJÄåŘŐçŽĎä£őæ■čæŰźæąĹ----------------------
↪---
class CachedSpamManager2:
    def __init__(self):
        self._cache = weakref.WeakValueDictionary()

    def get_spam(self, name):
        if name not in self._cache:
            temp = Spam3._new(name)  # Modified creation
            self._cache[name] = temp
        else:
            temp = self._cache[name]
        return temp

    def clear(self):
            self._cache.clear()

class Spam3:
    def __init__(self, *args, **kwargs):
        raise RuntimeError("Can't instantiate directly")

    # Alternate constructor
    @classmethod
    def _new(cls, name):
        self = cls.__new__(cls)
        self.name = name
        return self
```

æIJÄåŘŐè£ŹæäůçŽĎæŰźæąĹåřśåůšçżŔèűşåď§åě¡äžĘãÄĆ
çijŞå■ŸåŠŇåĚűäžŰæđĎéĂäæĺąåijŔè£ŸåŔŕäžěä¡£çŤĺ9.13åřŔèŁĆäÿ■çŽĎåĚČçśżåőđçŐřçŽĎæŻť äijŸéŻĚäÿ

11 çňňäźİçńäïijŽåĚČçijŰçÍŃ

è¡ŕäžűåijÄåŔŚéćĘå§§äÿ■æIJÄçżŔåĚÿçŽĎåŔčåď'ť çęĚåřśæŸŕâĂIJdonâĂŹt repeat yourselfâĂİãĂĆ äź§åřśæŸŕèŕť'ïijŇäžżä¡ŢæŮűåĂŹå¡Şä¡ăçŽĎçÍŃåžŔäÿ■å■ŸåIJĺéńŸåžęéĞ■åď'■(æĹŰèĂĚæŸŕéĂ
åIJÍPythonå¡Şäÿ■ïijŇéĂŽåÿÿéČ¡åŔŕäžěéĂŽè£ĞåĚČçijŰçÍŃæİěèğčåĘşè£ŹçśżéŮőécŸãĂĆ
çőÄèĂŇèÍĂäźŃïijŇåĚČçijŰçÍŃåřśæŸŕåĚşäžŐåĹŻåžżæŞ■ä¡IJæžŘäžčçăĄ(æŕŤåęĆä£őæŤźãĂĄçŢ§æĹŘæĹŰ
äÿżèęĄæŁĂæIJŕæŸŕä¡£çŤĺèčĚéěřåŹĺãĂĄçśżèčĚéěřåŹĺåŠŇåĚČçśżãĂĆäÿ■è£Ğè£ŸæIJL'äÿĂäžŻåĚűäžŰæŁĂ
åŇĚæŃńç■¿åŘ■åŕźèśąãĂĄä¡£çŤĺexec() æL'ğèąŇäžčçăĄäżěåŔŁåŕźåĘĚéČĺåĞ¡æŢŕåŠŇçśżçŽĎåŔ■åřĎæŁ
æIJńçńäçŽĎäÿżèęĄçŽőçŽĎæŸŕåŘŚåď'ğåőűäžŃçż■è£ŹäžŻåĚČçijŰçÍŃæŁĂæIJŕïijŇåźűäÿŤçżŹåĞžåőđä¿Ńæ

Contents:

11.1 9.1 ålJÍåĞ¡æŢřäÿŁæůżåŁääÿĂäÿłåŇĚèčĚåŹÍ

éŮőécŸ

ä¡ăæČşåIJÍåĞ¡æŢřäÿŁæůżåŁääÿĂäÿłåŇĚèčĚåŹÍïijŇåćđåŁäécÍåd'ŰçŽĎæŞ■ä¡IJåd'ĎçŘĘ(æŕŤåęĆæŮěå‡

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşä¡£çŤÍécÍåd'ŰçŽĎäžčçăĄåŇĚèčĚäÿĂäÿłåĞ¡æŢřijŇåŕŕäžěåőŽäźL'äÿĂäÿłèčĚéěřåŹÍåĞ

```
import time
from functools import wraps

def timethis(func):
    '''
    Decorator that reports the execution time.
    '''
    @wraps(func)
    def wrapper(*args, **kwargs):
        start = time.time()
        result = func(*args, **kwargs)
        end = time.time()
        print(func.__name__, end-start)
        return result
    return wrapper
```

äÿŇéÍćæŸŕä¡£çŤÍèčĚéěřåŹÍçŽĎä¿Ňå■ŘïijŽ

```
>>> @timethis
... def countdown(n):
...     '''
...     Counts down
...     '''
...     while n > 0:
...         n -= 1
...
>>> countdown(100000)
countdown 0.008917808532714844
>>> countdown(10000000)
countdown 0.87188299392912
>>>
```

èőÍèőž

äÿĂäÿłèčĚéěřåŹÍåřśæŸŕäÿĂäÿłåĞ¡æŢřïijŇåőČæŐěåŔŮäÿĂäÿłåĞ¡æŢřä¡IJäÿžåŔĆæŢřåźűè£ŤåŻđäÿĂäÿ
å¡Şä¡ăåČŔäÿŇéÍćè£ŹæăůåĘŹïijŽ

```
@timethis
def countdown(n):
    pass
```

èů§åČŔäÿŃéİćè£ŹæăůåĘŹåĚűåőđæŢĹæđIJæŸŕäÿĂæăůçŽĎïijŽ

```
def countdown(n):
    pass
countdown = timethis(countdown)
```

éąžä¿£èŕťäÿĂäÿŃïijŇåĘĚç¡őçŽĎèčĚéěřåŹĺæŕŤåęĆ `@staticmethod`, `@classmethod`,`@property` åŐ§çŘĘäź§æŸŕäÿĂæăůçŽĎãĂĆ ä¿ŃåęĆïijŇäÿŃéİćè£ŹäÿďäÿłäžčçăĄçL'ĞæőţæŸŕç■L'äžůçŽĎïijŽ

```
class A:
    @classmethod
    def method(cls):
        pass

class B:
    # Equivalent definition of a class method
    def method(cls):
        pass
    method = classmethod(method)
```

åIJĺäÿŁéİćçŽĎ `wrapper()` åĞ¡æŢřäÿ■ïijŇ èčĚéěřåŹĺåĘĚéČĺåőŽäźL'äžĘäÿĂäÿłä¡£çŢĺ `*args` åŠŇ `**kwargs` æİěæŐěåŔŮäżżæĎŔåŔĆæŢřçŽĎåĞ¡æŢřãĂĆ åIJĺè£ŹäÿłåĞ¡æŢréĞŇéİćèřČçŢĺäžĘåŐ§ågŃåĞ¡æŢřáźůårĘåĚűçżŞædIJè£ŤåŻđïijŇäÿ■è£Ğä¡ăè£ŸåŔŕäžěæůżçĎűåŘŐè£ŹäÿłæŰřçŽĎåĞ¡æŢřåŇĚèčĚåŹĺèćńä¡IJäÿžçżŞædIJè£ŤåŻđæİěäžčæŻ£åŐ§ågŃåĞ¡æŢřãĂĆ

éIJĂèęĄåijžěřČçŽĎæŸŕèčĚéěřåŹĺáźůäÿ■äijŽä£őæŤźåŐ§ågŃåĞ¡æŢřçŽĎåŔĆæŢřç■¿åŘ■äžěåŔŁè£ŤåŻ ä¡£çŢĺ `*args` åŠŇ `**kwargs` çŻőçŽĎåřśæŸŕçąőä£İäżżä¡ŢåŔĆæŢřéČ¡èČ¡éĂĆçŢĺãĂĆ èĂŇè£ŤåŻđçżŞædIJåĂijå§žæIJňéČ¡æŸŕèřČçŢĺåŐ§ågŃåĞ¡æŢř `func(*args, **kwargs)` çŽĎè£ŤåŻđçżŞædIJïijŇåĚűäÿ■funcåřśæŸŕåŐ§ågŃåĞ¡æŢřãĂĆ

åĹŽåijĂåğŃå■ęäźăèčĚéěřåŹĺçŽĎæŮűåĂŹïijŇäijŽä¡£çŢĺäÿĂäžŻçőĂå■ŢçŽĎä¿Ńå■ŘæİèèŕťæŸŐïijŇæŕ äÿ■è£ĞåőđéŹĚåIJžæŹŕä¡£çŢĺæŮűïijŇè£ŸæŸŕæIJL'äÿĂäžŻçżĘèŁĆéŮőécŸèęĄæşĺæĎŔçŽĎãĂĆ æŕŤåęĆäÿĹéİćä¡£çŢĺ `@wraps(func)` æşĺèğčæŸŕå¿ĹéĞ■èęĄçŽĎïijŇ åőČèČ¡ä£İçŢŹåŐ§ågŃåĞ¡æŢřçŽĎåĚČæŢřæ■ő(äÿŃäÿĂåřŔèŁĆäijŽèőšåĹř)ïijŇæŰřæL'ŃçżŘåÿÿäijŽå£¡çŢĕè æŐěäÿŃæİěçŽĎåĞăäÿłåřŔèŁĆæĹŚäžňäijŽæŻť'åŁăæůśåĚěçŽĎèőšèğčèčĚéěřåŹĺåĞ¡æŢřçŽĎçżĘèŁĆéŮőécŸ

11.2 9.2 åĹŽåžžèčĚéěřåŹĺæŮűä£İçŢŹåĞ¡æŢřåĚČä£ąæĄŕ

éŮőécŸ

ä¡ăåĘŹäžĘäÿĂäÿłèčĚéěřåŹĺä¡IJçŤĺåIJĺæ§ŘäÿłåĞ¡æŢřäŸŁïijŇä¡ĘæŸŕè£ŹäÿłåĞ¡æŢřçŽĎéĞ■èęĄçŽĎåĚ

èğčåĘşæŰźæąĹ

äżżä¡ŢæŮűåĂŹä¡ăåőŽäźL'èčĚéěřåŹÍçŽĎæŮűåĂŹïijŇéČ¡åžŤèŕěä¡£çŤÍ `functools` åžŞäÿ■çŽĎ `@wraps` èčĚéěřåŹÍæİěæşÍèğčåžŢåśĆåŇĚèčĚåĞ¡æŢřãĂĆä¿ŇåęĆïijŽ

```
import time
from functools import wraps
def timethis(func):
    '''
    Decorator that reports the execution time.
    '''
    @wraps(func)
    def wrapper(*args, **kwargs):
        start = time.time()
        result = func(*args, **kwargs)
        end = time.time()
        print(func.__name__, end-start)
        return result
    return wrapper
```

äÿŃéİćæĹŚäżñä¡£çŤÍè£ŹäÿłèćńåŇĚèčĚåŘŐçŽĎåĞ¡æŢřåźűæčĂæ§ěåőČçŽĎåĚČä£ąæĄŕïijŽ

```
>>> @timethis
... def countdown(n):
...     '''
...     Counts down
...     '''
...     while n > 0:
...         n -= 1
...
>>> countdown(100000)
countdown 0.008917808532714844
>>> countdown.__name__
'countdown'
>>> countdown.__doc__
'\n\tCounts down\n\t'
>>> countdown.__annotations__
{'n': <class 'int'>}
>>>
```

èőÍèőž

åIJÍçijŰåĘŹèčĚéěřåŹÍçŽĎæŮűåĂŹåd'■åĹűåĚČä£ąæĄŕæŸŕäÿĂäÿłéİđåÿÿéĞ■èęĄçŽĎéČÍåĹĘãĂĆåęĆæ `@wraps` ïijŇ éĆčäźĹä¡ääijŽåŔŞçŐřèćńèčĚéěřåĞ¡æŢřäÿćåd'šäžĘæL'ĂæIJL'æIJL'çŤÍçŽĎä£ąæĄŕãĂĆæŕŤåęĆ `@wraps` åŘŐçŽĎæŢĹæđIJæŸŕäÿŃéİćè£ŹæăůçŽĎïijŽ

```
>>> countdown.__name__
'wrapper'
>>> countdown.__doc__
```

```
>>> countdown.__annotations__
{}
>>>
```

@wraps æIJL'äÿĂäÿłéĞ■èęĄçL'źå¿ĄæŸŕåőČèČ¡èől'ä¡ăéĂŽè£ĞåšđæĂğ __wrapped__ çŻť'æŐěèő£éŮőèćńåŇĚèčĚåĞ¡æŢřãĂĆä¿ŃåęĆ:

```
>>> countdown.__wrapped__(100000)
>>>
```

__wrapped__ åšđæĂğè£ŸèČ¡èől'èćńèčĚéěřåĞ¡æŢřæ■čçąőæŽť'éIJšåžŢåśĆçŽĎåŔĆæŢřç■¿åŘ■ä£ąæĄ

```
>>> from inspect import signature
>>> print(signature(countdown))
(n:int)
>>>
```

äÿĂäÿłå¿ĹæŹőéĄ■çŽĎéŮőécŸæŸŕæĂŐæăůèől'èčĚéěřåŹĺåŐżçŻť'æŐěåđ'■åĹűåŐ§ågŃåĞ¡æŢřçŽĎåŔĆ åęĆæđIJæČşèĞłåůśæĽŃåŁĺåőđçŐřçŽĎèŕÍéIJĂèęĄåĄŽåđ'ğéĞŔçŽĎåůěä¡IJïijŇæIJĂåě¡åřśçőĂå■ŢçŽĎä¡£çŤ @wraps èčĚéěřåŹĺãĂĆ éĂŽè£ĞåžŢåśĆçŽĎ __wrapped__ åšđæĂğèő£éŮőåĹřåĞ¡æŢřç■¿åŘ■ä£ąæĄfãĂĆæŻť'åđ'ŽåĚşäžŐç■¿åŘ■çŽĎåĘĚåőźåŔŕäżěåŔĆèĂČ9.16åřŔèŁ

11.3 9.3 èğčéŹđ'äÿĂäÿłècĚéěřåŹĺ

éŮőécŸ

äÿĂäÿłècĚéěřåŹĺåůšçżŔä¡IJçŤĺåIJĺäÿĂäÿłåĞ¡æŢřäŸŁïijŇä¡ăæČşæŠđ'éŤĂåőČïijŇçŻť'æŐèèő£éŮőåŐ§åğ

èğčåĘşæŰźæąĹ

åĄĞèő¿èčĚéěřåŹĺæŸŕéĂŽè£Ğ @wraps (åŔĆèĂČ9.2åřŔèŁĆ)æİěåőđçŐřçŽĎïijŇéĆčäźĹä¡ăåŔŕäżěéĂŽè __wrapped__ åšđæĂğæİèèő£éŮőåŐ§åğŃåĞ¡æŢřïijŽ

```
>>> @somedecorator
>>> def add(x, y):
...     return x + y
...
>>> orig_add = add.__wrapped__
>>> orig_add(3, 4)
7
>>>
```

èőĺèőž

çŻť'æŐèèő£éŮőæIJłåŇĚèčĚçŽĎåŐ§åğŃåĞ¡æŢřåIJĺèřČèŕŢãĂĄåĘĚçIJĄåŠŇåĚűäżŰåĞ¡æŢřæŞ■ä¡IJæŮ ä¡ĘæŸŕæĹŚäžňè£ŹéĞŇçŽĎæŰźæąĹäżĚäżĚéĂĆçŤĺäžŐåIJĺåŇĚèčĚåŹĺäÿ■æ■čçąőä¡£çŤĺäžĘ

@wraps æĹŰèĂĚçŻťæŐèèő¿ç¡őäžĘ __wrapped__ åšđæĂğçŽĎæČĚåĘţãĂĆ

åęĆæđIJæIJL'åđ'Žäÿ łåŇĚèčĚåŹĺïijŇéĆčäźĹèő£éŮő __wrapped__ åšđæĂğçŽĎèąŇäÿžæŸŕäÿ■åŔŕéćĎç§ěçŽĎïijŇåžŤèŕěéĄ£åĚ■è£Źæăůå ĄŽãĂĆ åIJĺPython3.3äÿ■ïijŇåőČäijŽçŢèè£ĞæL'ĂæIJL'çŽĎåŇĚèčĚåśĆïijŇæŕŤåęĆïijŇåĄĞåęĆä¡ăæIJL'åęĆäÿŃçŽĎ

```
from functools import wraps

def decorator1(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        print('Decorator 1')
        return func(*args, **kwargs)
    return wrapper

def decorator2(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        print('Decorator 2')
        return func(*args, **kwargs)
    return wrapper

@decorator1
@decorator2
def add(x, y):
    return x + y
```

äÿŃéİćæĹŚäžňåIJĺPython3.3äÿŃæţŃèŕŢïijŽ

```
>>> add(2, 3)
Decorator 1
Decorator 2
5
>>> add.__wrapped__(2, 3)
5
>>>
```

äÿŃéİćæĹŚäžňåIJĺPython3.4äÿŃæţŃèŕŢïijŽ

```
>>> add(2, 3)
Decorator 1
Decorator 2
5
>>> add.__wrapped__(2, 3)
Decorator 2
5
>>>
```

æIJĂåŘŐèęĄèŕť'çŽĎæŸŕïijŇåźűäÿ■æŸŕæL'ĂæIJL'çŽĎèčĚéěřåŹĺéČ¡ä¡£çŢĺäžĘ @wraps ïijŇåŻăæ■d'è£ŹéĞŇçŽĎæŰźæąĹåźűäÿ■åĚĺéČĺéĂĆçŢĺãĂĆ çL'źåĹńçŽĎïijŇåĘĚç¡őçŽĎèčĚéěřåŹĺ @staticmethod åŠŇ @classmethod

åřśæšąæIJL'éĄţå¿łè£Źäÿłçžęåőž (åőČäžňæŁŁåŐ§åğŃåĞ¡æŢřå■ŸåĆĺåIJĺåśđæĂğ __func__ äÿ■)ãĂĆ

11.4 9.4 åőŽäźL'äÿĂäÿłåÿęåŔĆæŢřçŽĎèčĚéěřåŹÍ

éŮőécŸ

ä¡ăæČşåőŽäźL'äÿĂäÿłåŔŕäžěæŐěåŔŮåŔĆæŢřçŽĎèčĚéěřåŹÍ

èğčåĘşæŰźæąĹ

æĹŚäžňçŤĺäÿĂäÿłä¿Ńå■Řèŕęçżę́ÿŘè£řäÿŃæŐěåŔŮåŔĆæŢřçŽĎåď'ĎçŘĘè£ĞçÍŃãĂĆ åĄĞèő¿ä¡ăæČşåĘŹäÿĂäÿłèčĚéěřåŹÍïijŇçżŹåĞ¡æŢřæůžåŁăæŮěå£ŮåŁ§èČ¡ïijŇåŘŇæŮűåĚąèőÿçŤĺæĹůæŇ äÿŃéIćæŸŕè£ŹäÿłèčĚéěřåŹÍçŽĎåőŽäźL'åŠŇä¡£çŤĺçď'žä¿Ńïij Ž

```
from functools import wraps
import logging

def logged(level, name=None, message=None):
    """
    Add logging to a function. level is the logging
    level, name is the logger name, and message is the
    log message. If name and message aren't specified,
    they default to the function's module and name.
    """
    def decorate(func):
        logname = name if name else func.__module__
        log = logging.getLogger(logname)
        logmsg = message if message else func.__name__

        @wraps(func)
        def wrapper(*args, **kwargs):
            log.log(level, logmsg)
            return func(*args, **kwargs)
        return wrapper
    return decorate

# Example use
@logged(logging.DEBUG)
def add(x, y):
    return x + y

@logged(logging.CRITICAL, 'example')
def spam():
    print('Spam!')
```

åĹİçIJŃèţůæİěïijŇè£Źçğ■åőđçŐřçIJŃäÿŁåŐżå¿Ĺåď'■æİĆïijŇä¡ĘæŸŕæäÿå£ČæĂİæČşå¿ĹçőĂå■ŢãĂĆ æIJĂåď'ŰåśĆçŽĎåĞ¡æŢř `logged()` æŐěåŔŮåŔĆæŢřåźűåřĘåőČäžňä¡IJçŤĺåIJĺåĘĚéČĺçŽĎèčĚéěřåŹĺåĞ¡æ

åĘĚåśĆçŽĎåĞ¡æŢř `decorate()` æŐěåŔŮäÿĂäÿłåĞ¡æŢřä¡IJäÿžåŔĆæŢřïijŇçĎűåŘŐåIJĺåĞ¡æŢřäÿŁéİćæŤ
è£ŹéĞŇçŽĎåĔşéŤőçĆźæŸŕåŇĚèčĚåŹĺæŸŕåŔŕäžěä¡£çŤĺäijåéĂŠçżŹ `logged()`
çŽĎåŔĆæŢřçŽĎãĂĆ

èőÍèőž

åőŽäźĽäÿĂäÿłæŐěåŔŮåŔĆæŢřçŽĎåŇĚèčĚåŹĺçIJŃäÿŁåŐżæŕŤè¿Čåd'■æİĆäÿžèęĄæŸŕåŻääÿžåžŢåśĆ

```
@decorator(x, y, z)
def func(a, b):
    pass
```

èčĚéěřåŹĺåd'ĎçŘĘè£ĞçÍŃèů§äÿŃéİćçŽĎèřČçŤĺæŸŕç■L'æŢĹçŽĎ;

```
def func(a, b):
    pass
func = decorator(x, y, z)(func)
```

`decorator(x, y, z)` çŽĎè£ŤåŻđçżŞæđIJå£ĚéążæŸŕäÿĂäÿłåŔŕèřČçŤĺårźèśąïijŇåőČæŐěåŮäÿĂ
åŘŕäžěåŔĆèĂČ9.7åřŔèŁĆäÿ■åŔęåd'ŰäÿĂäÿłåŘŕæŐěåŔŮåŔĆæŢřçŽĎåŇĚèčĚåŹĺä¿Ńå■ŘãĂĆ

11.5 9.5 åŘŕèĞłåőŽäźĽåśđæĂğçŽĎèčĚéěřåŹĺ

éŮőécŸ

ä¡ăæČşåĘŹäÿĂäÿłèčĚéěřåŹĺæİěåŇĚèčĚäÿĂäÿłåĞ¡æŢřïijŇåźűäÿŤåĚĄèőÿçŤĺæĹůæŔŘä¿ŻåŔĆæŢřåIJĺè

èğčåĘşæŰźæąĹ

åijŢåĚěäÿĂäÿłèő£éŮőåĞ¡æŢřïijŇä¡£çŤĺ `nonlocal` æİěä£őæŤźåĘĚéČĺåŔŸéĞŔãĂĆ
çĎűåŘŐè£Źäÿłèő£éŮőåĞ¡æŢřèćńä¡IJäÿžäÿĂäÿłåśđæĂğètŃåĂijçżŹåŇĚèčĚåĞ¡æŢřãĂĆ

```
from functools import wraps, partial
import logging
# Utility decorator to attach a function as an attribute of obj
def attach_wrapper(obj, func=None):
    if func is None:
        return partial(attach_wrapper, obj)
    setattr(obj, func.__name__, func)
    return func

def logged(level, name=None, message=None):
    '''
    Add logging to a function. level is the logging
    level, name is the logger name, and message is the
    log message. If name and message aren't specified,
    they default to the function's module and name.
```

```
    '''
    def decorate(func):
        logname = name if name else func.__module__
        log = logging.getLogger(logname)
        logmsg = message if message else func.__name__

        @wraps(func)
        def wrapper(*args, **kwargs):
            log.log(level, logmsg)
            return func(*args, **kwargs)

        # Attach setter functions
        @attach_wrapper(wrapper)
        def set_level(newlevel):
            nonlocal level
            level = newlevel

        @attach_wrapper(wrapper)
        def set_message(newmsg):
            nonlocal logmsg
            logmsg = newmsg

        return wrapper

    return decorate

# Example use
@logged(logging.DEBUG)
def add(x, y):
    return x + y

@logged(logging.CRITICAL, 'example')
def spam():
    print('Spam!')
```

äÿŃéÍćæŸŕäžđ'äžŠçŐŕåćČäÿŃçŽĎä¡£çŤĺä¿Ńå■ŘïijŽ

```
>>> import logging
>>> logging.basicConfig(level=logging.DEBUG)
>>> add(2, 3)
DEBUG:__main__:add
5
>>> # Change the log message
>>> add.set_message('Add called')
>>> add(2, 3)
DEBUG:__main__:Add called
5
>>> # Change the log level
>>> add.set_level(logging.WARNING)
>>> add(2, 3)
```

```
WARNING:__main__:Add called
5
>>>
```

èőÍèőž

è£ŹäÿĂåřŔèŁĆçŽĎåĚşéŤőçĆźåIJÍäžŐèő£éŮőåĞ¡æŢř(åęĆ `set_message()` åŠŇ `set_level()`)ïijŇåőČäžňèćńä¡IJäÿžåśđæĂğètŅçżŹåŇĚèčĚåŹÍãĂĆ æŕŔäÿłèő£éŮőåĞ¡æŢřåĚĄèőÿä¡£çŤÍ `nonlocal` æÌěä£őæŤźåĞ¡æŢřåĘĚéČÍçŽĎåŔŸéĞŔãĂĆ

è£ŸæIJL'äÿĂäÿłäżd'äžžåŘČæČŁçŽĎåIJřæŰźæŸŕèő£éŮőåĞ¡æŢřäijŽåIJÍåd'ŽåśĆèčĚéěřåŹÍéŮt'äijăæŠ■ `@functools.wraps` æşÍèğč)ãĂĆ ä¿ŃåęĆïijŇåĄĞèő¿ä¡ååijŢåĚěåŔęåd'ŰäÿĂäÿłèčĚéěřåŹÍïijŇæŕŤåęĆ9.2å `@timethis` ïijŇåČŔäÿŃéÍćè£Źæăůïijž

```
@timethis
@logged(logging.DEBUG)
def countdown(n):
    while n > 0:
        n -= 1
```

ä¡ăäijŽåŔŞçŐřèő£éŮőåĞ¡æŢřä¿ÌæŮğæIJL'æŢĹïijž

```
>>> countdown(10000000)
DEBUG:__main__:countdown
countdown 0.8198461532592773
>>> countdown.set_level(logging.WARNING)
>>> countdown.set_message("Counting down to zero")
>>> countdown(10000000)
WARNING:__main__:Counting down to zero
countdown 0.8225970268249512
>>>
```

ä¡ăè£ŸäijŽåŔŞçŐřå■şä¡£èčĚéěřåŹÍåČŔäÿŃéÍćè£ŹæăůäžěçŻÿåŔ■çŽĎæŰźåŘŚæŐŠæŤ¿ïijŇæŢĹæđIJäź

```
@logged(logging.DEBUG)
@timethis
def countdown(n):
    while n > 0:
        n -= 1
```

è£ŸèČ¡éĂŽè£Ğä¡£çŤÍlambdaèąÍè¿¿åijŔäžčçăĄæÌěèőľ'èő£éŮőåĞ¡æŢřçŽĎè£ŤåŻđäÿ■åŘŇçŽĎèő¿åőŽåź

```
@attach_wrapper(wrapper)
def get_level():
    return level

# Alternative
wrapper.get_level = lambda: level
```

äÿĂäÿłæŕŤè¿ČéŽ¿çŘĘèğčçŽĎåIJřæŰźåřśæŸŕåŕźäžŐèő£éŮőåĞ¡æŢřçŽĎéęŰæňąä¡£çŤĺãĂĆä¿ŃåęĆïijŇ

```
@wraps(func)
def wrapper(*args, **kwargs):
    wrapper.log.log(wrapper.level, wrapper.logmsg)
    return func(*args, **kwargs)

# Attach adjustable attributes
wrapper.level = level
wrapper.logmsg = logmsg
wrapper.log = log
```

è£ŹäÿłæŰźæşŢäź§åŘŕèČ¡æ■čåÿÿåůëä¡IJïijŇä¡EąL'■æŔŘæŸŕåőČå£ĚéążæŸŕæIJĂåď'ŰåśÇçŽĎèčĚéěřåŹ
åęĆæđIJåőČçŽĎäÿŁéİćè£ŸæIJL'åŘęåď'ŰçŽĎèčĚéěřåŹÍ(æŕŤåęĆäÿŁéİćæŔŘåĹřçŽĎ
`@timethis` ä¿Ńå■Ř)ïijŇéĆčäźĹåőČäijŽéŽŘèŮŔåžŢåśĆåśđæĂğïijŇä¡£å¿Ůä£őæŤźåőČäžñæšąæIJL'äżżä¡Ţ
èĂŇéĂŽè£Ğä¡£çŤĺèő£éŮőåĞ¡æŢřåŕśèČ¡éĄ£åĚ■è£ŹæăůçŽĎåśĂéŹŘæĂğãĂĆ

æIJĂåŘŐæŔŘäÿĂçĆźïijŇè£ŹäÿĂåřŔèŁĆçŽĎæŰźæąĹäź§åŘŕäżěä¡IJäÿž9.9åřŔèŁĆäÿ■èčĚéěřåŹĺçśżçŽĺ

11.6 9.6 åÿęåŘŕéĂĽåŘĆæŢřçŽĎèčĚéěřåŹÍ

éŮőécŸ

ä¡ăæČşåĘŹäÿĂäÿłèčĚéěřåŹĺïijŇæŮćåŘŕäżěäÿ■äijăåŘĆæŢřçżŹåőČïijŇæŕŤåęĆ
`@decorator` ïijŇ äź§åŘŕäżěäijăéĂŠåŘŕéĂĽåŘĆæŢřçżŹåőČïijŇæŕŤåęĆ
`@decorator(x,y,z)` ãĂĆ

èğčåĘşæŰźæąĹ

äÿŃéİćæŸŕ9.5åřŔèŁĆäÿ■æŮěå£ŮèčĚéěřåŹĺçŽĎäÿĂäÿłä£őæŤźçĽ'ĹæIJñïijŽ

```
from functools import wraps, partial
import logging

def logged(func=None, *, level=logging.DEBUG, name=None,␣
↪message=None):
    if func is None:
        return partial(logged, level=level, name=name,␣
↪message=message)

    logname = name if name else func.__module__
    log = logging.getLogger(logname)
    logmsg = message if message else func.__name__

    @wraps(func)
    def wrapper(*args, **kwargs):
        log.log(level, logmsg)
        return func(*args, **kwargs)
```

```
    return wrapper

# Example use
@logged
def add(x, y):
    return x + y

@logged(level=logging.CRITICAL, name='example')
def spam():
    print('Spam!')
```

åŘŕäžěçIJŃåĹřïijŇ`@logged`èčĚéěřåŹĺåŘŕäžěåŘŇæŮűäÿ■åÿęåŘĆæŢřæĹŰåÿęåŘĆæŢřãĂĆ

èőĺèőž

è£ŹéĞŇæŔŘåĹřçŽĎè£ŹäÿłéŮőécŸåřśæŸŕéĂŽåÿÿæL’Ăèŕť’çŽĎçijŰçĺŃäÿĂèĞť’æĂğéŮőécŸãĂĆ
å¡ŞæĹŚäžňä¡£çŤĺèčĚéěřåŹĺçŽĎæŮűåĂŹïijŇåď’ğéČĺåĹĘçĺŃåžŔåŚŸäźăæČŕäžĘèęĄäźĹäÿ■çżŹåőČäžňäijåéĂ
åĚűåőđäżŐæŁĂæIJŕäÿŁæİěèőšïijŇæĹŚäžňåŔŕäžěåőŽäźL’äÿĂäÿłæL’ĂæIJL’åŔĆæŢřéČ¡æŸŕåŔŕéĂL’çŽĎèčĚ

```
@logged()
def add(x, y):
    return x+y
```

ä¡ĘæŸŕïijŇè£Źçğ■åĘŹæşŢåźűäÿ■çňęåŘĹæĹŚäžňçŽĎäźăæČŕïijŇæIJL’æŮűåĂŹçĺŃåžŔåŚŸå£Ÿèőřå£ăä
è£ŹéĞŇæĹŚäžňåŘŚä¡ăåśŢçď’žäžĘåęĆä¡Ţäžěäÿ ĂèĞť’çŽĎçijŰçĺŃéčŐæăijæİěåŘŇæŮűæżąèűşæšąæIJL’æŃňå

äÿžäžĘçŘĘèğčäžčçăĄæŸŕåęĆä¡Ţåůěä¡IJçŽĎïijŇä¡ăéIJĂèęĄéİđåÿÿçE§æĆL’èčĚéěřåŹĺæŸŕåęĆä¡Ţä¡IJçĚ
åŕźäžŐäÿĂäÿłåČŘäÿŃéİćè£ŹæăůçŽĎçőĂå■ŢèčĚéěřåŹĺïijŽ

```
# Example use
@logged
def add(x, y):
    return x + y
```

è£ŹäÿłèřČçŤĺåžŔåĹŮèů§äÿŃéİćç■L’äžůïijŽ

```
def add(x, y):
    return x + y

add = logged(add)
```

è£ŹæŮűåĂŹïijŇèćńèčĚéěřåĞ¡æŢřäijŽèćńå¡ŞåĄŽçňňäÿĂäÿłåŔĆæŢřçŻť’æŐëäijăéĂŠçżŹ
`logged`èčĚéěřåŹĺãĂĆ åŽăæ■d’ïijŇ`logged()` äÿ■çŽĎçňňäÿĂäÿłåŔĆæŢřářśæŸŕèćńåŇĚèčĚåĞ¡æŢřæIJňè

èĂŇářźäžŐäÿĂäÿłäÿŃéİćè£ŹæăůæIJL’åŔĆæŢřçŽĎèčĚéěřåŹĺïijŽ

```
@logged(level=logging.CRITICAL, name='example')
def spam():
    print('Spam!')
```

èřČçŤÍåžŔåĹŮèů§äÿŃéİćç■L’äžůïijŽ

```
def spam():
    print('Spam!')
spam = logged(level=logging.CRITICAL, name='example')(spam)
```

åĹİåğŃèřČçŤÍ `logged()` åĞ¡æŢřæŮűïijŃèćńåŇĚèčĚåĞ¡æŢřåźűæšąæIJL’äijăéĂŠè£ŻæİěãĂĆ åŻăæ■ď’åIJÍèčĚéěřåŹÍåĘĚïijŇåőČå£Ěéążæ˙ŸŕåŔŕéĂL’çŽĎãĂĆè£ŹäÿłåŔ■è£ĞæİěäijŽè£ńä¡£åĚűäžŮåŔĆæŢřă åźűäÿŤïijŇä¡Ęè£ŹäžŻåŔĆæŢřèćńäijăéĂŠè£ŻæİěåŘŐïijŃèčĚéěřåŹÍèęĄè£ŤåŻđäÿĂäÿłæŐěåŔŮäÿĂäÿłåĞ¡æŢ äÿžäžĘè£ŹæăůåĄŽïijŇæĹŚäžňä¡£çŤÍäžĘäÿĂäÿłæŁĂåůğïijŇářśæŸŕåĹ’çŤÍ `functools.partial` ãĂĆ åőČäijŽè£ŤåŻđäÿĂäÿłæIJłåőŇåĚÍåĹİåğŃåŇŰçŽĎèĞłèžńïijŃéŹď’äžĘèćńåŇĚèčĚåĞ¡æŢřåď’ åŔŕäžěåŔĆèĂČ7.8ářŔèŁĆèŐůåŔŰæŻť’åď’Ž `partial()` æŰźæşŢçŽĎç§èèŕĘãĂĆ

11.7 9.7 åĹ’çŤÍèčĚéěřåŹÍåijžåĹűåĞ¡æŢřäŸŁçŽĎçśžåđŃæčĂæ§ě

éŮőécŸ

ä¡IJäÿžæ§Řçğ■çijŰçÍŃèğĎçžęïijŃä¡ăæČşåIJÍåŕźåĞ¡æŢřåŔĆæŢřè£ŻèąŇåijžåĹűçśžåđŃæčĂæ§ěãĂĆ

èğčåĘşæŰźæąĹ

åIJÍæijŤçď’žåőđéŹĚäžčçăĄåL’■ïijŇåĚĹèŕť’æŸŐæĹŚäžňçŽĎçŻőæăĞïijŽèČ¡åŕźåĞ¡æŢřåŔĆæŢřçśžåđŃè

```
>>> @typeassert(int, int)
... def add(x, y):
...     return x + y
...
>>>
>>> add(2, 3)
5
>>> add(2, 'hello')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "contract.py", line 33, in wrapper
TypeError: Argument y must be <class 'int'>
>>>
```

äÿŃéİćæŸŕä¡£çŤÍèčĚéěřåŹÍæŁĂæIJŕæİěåőđçŐř `@typeassert` ïijŽ

```
from inspect import signature
from functools import wraps

def typeassert(*ty_args, **ty_kwargs):
    def decorate(func):
        # If in optimized mode, disable type checking
        if not __debug__:
            return func
```

```
        # Map function argument names to supplied types
        sig = signature(func)
        bound_types = sig.bind_partial(*ty_args, **ty_kwargs).
↪arguments

        @wraps(func)
        def wrapper(*args, **kwargs):
            bound_values = sig.bind(*args, **kwargs)
            # Enforce type assertions across supplied arguments
            for name, value in bound_values.arguments.items():
                if name in bound_types:
                    if not isinstance(value, bound_types[name]):
                        raise TypeError(
                            'Argument {} must be {}'.format(name,␣
↪bound_types[name])
                            )
            return func(*args, **kwargs)
        return wrapper
    return decorate
```

åŔŕäžěçIJŃåĞžè£ŹäýłèčĚéěřåŹĺéİđåýÿçĄţæť żïijŇæŮćåŔŕäžěæŇĞåőŽæĽĂæIJĽåŔĆæŢřçśžåđŃïijŇäź
åźűäýŤåŔŕäžěéĂŽè£Ğä¡■ç¡őæĹŰåĚşéŤőå■ŮæİěæŇĞåőŽåŔĆæŢřçśžåđŃãĂĆäýŃéĺćæŸŕä¡£çŤĺçď žä¿ŃïijŽ

```
>>> @typeassert(int, z=int)
... def spam(x, y, z=42):
...     print(x, y, z)
...
>>> spam(1, 2, 3)
1 2 3
>>> spam(1, 'hello', 3)
1 hello 3
>>> spam(1, 'hello', 'world')
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "contract.py", line 33, in wrapper
TypeError: Argument z must be <class 'int'>
>>>
```

èőĺèőž

è£ŹèŁĆæŸŕéńŸçžğècĚéěřåŹĺçď žä¿ŃïijŇåijŢåĚěäžĘå¿Ĺåď ŽéĞ■èęĄçŽĎæęĆå£ţãĂĆ

éęŰåĚĹïijŇècĚéěřåŹĺåŔłäijŽåIJĺåĞ¡æŢřåőŽäźĽæŮűècńèřČçŤĺäÿĂæňąãĂĆ
æIJĽæŮűåĂŹä¡ääŐżæŐĽèčĚéěřåŹĺçŽĎåŁ§èČ¡ïijŇéĆčäźĹä¡ääŔłéIJĂèęĄçőĂå■ŢçŽĎè£ŤåŽđècńèčĚéěřåĞ¡
äÿŃéİćçŽĎäžčçăĄäÿ■ïijŇåęĆæđIJåĚĺåśĂåŔŸéĞŘãĂĂ__debug__
èćńèő¿ç¡őæĹŘäžĘFalse(å¡Şä¡ää¡£çŤĺ-OæĹŰ-OOåŔĆæŢřçŽĎäijŸåŇŰæĺąåijŔæĽğèąŇçĺŃåžŔæŮű)ïijŇ
éĆčäźĹåřśçŻťæŐèè£ŤåŽđæIJłä£őæŤźè£ĞçŽĎåĞ¡æŢřæIJňèžńïijŽ

```
def decorate(func):
    # If in optimized mode, disable type checking
    if not __debug__:
        return func
```

åĚűæňąïijŇè£ŹéĞŇè£ŸåŕźèćńåŇĚèčĚåĞ¡æŢřçŽĎåŔĆæŢřç■¿åŘ■è£ŻèąŇäžĘæčĂæ§ěïijŇæĹŚäžňä¡£çŤ
inspect.signature() åĞ¡æŢřãĂĆ çőĂå■ŢæİěëőšïijŇåőČè£ŘèąŇä¡ăæŔŘåŔŰäÿĂäÿłåŔŕěřČçŤĺåŕźèšąę

```
>>> from inspect import signature
>>> def spam(x, y, z=42):
...     pass
...
>>> sig = signature(spam)
>>> print(sig)
(x, y, z=42)
>>> sig.parameters
mappingproxy(OrderedDict([('x', <Parameter at 0x10077a050 'x'>),
('y', <Parameter at 0x10077a158 'y'>), ('z', <Parameter at␣
↪0x10077a1b0 'z'>)]))
>>> sig.parameters['z'].name
'z'
>>> sig.parameters['z'].default
42
>>> sig.parameters['z'].kind
<_ParameterKind: 'POSITIONAL_OR_KEYWORD'>
>>>
```

èčĚéěřåŹĺçŽĎåijĂåğŃéČĺåĹĘïijŇæĹŚäžňä¡£çŤĺäžĘ bind_partial()
æŰźæşŢæİěæĽğèąŇäžŐæŇĞåőŽçśžåđŃåĹřåŘ■çğřçŽĎéČĺåĹĘçżŚåőŽãĂĆ
äÿŃéİćæŸŕä¿Ńå■ŘæijŤçď žïijŽ

```
>>> bound_types = sig.bind_partial(int,z=int)
>>> bound_types
<inspect.BoundArguments object at 0x10069bb50>
>>> bound_types.arguments
OrderedDict([('x', <class 'int'>), ('z', <class 'int'>)])
>>>
```

åIJĺè£ŹäÿłéČĺåĹĘçżŚåőŽäÿ■ïijŇä¡ăåŔŕäžěæşĺæĎŔåĹřçijžåď śçŽĎåŔĆæŢřèćńå£¡çŢěäžĘ(æŕŤåęĆåźűæš
äÿ■è£ĞæIJĂéĞ■èęĄçŽĎæŸŕåĹŻåžžäžĘäÿĂäÿłæIJĽåžŔå■ŮåĚÿ bound_types.
arguments ãĂĆ è£ŹäÿłåŮåĚÿäijŽåřĘåŔĆæŢřåŘ■äžěåĞ¡æŢřç■¿åŘ■äÿ■çŽÿåŘŇéąžåžŔæŸăåřĎåĹřæŇĞå
åIJĺæĹŚäžňçŽĎèčĚéěřåŹĺä¿Ńå■Řäÿ■ïijŇè£ŹäÿłæŸăåřĎåŇĚåŘńäžĘæĹŚäžňèęĄåijžåĹűæŇĞåőŽçŽĎçśžåđŇ

åIJĺèčĚéěřåŹĺåĹŻåžžçŽĎåőđéŹĚåŇĚèčĚåĞ¡æŢřäÿ■ä¡£çŤĺåŕžĘ
sig.bind() æŰźæşŢãĂĆ bind() èů§ bind_partial()
çśžäijijïijŇä¡ĘæŸŕåőČäÿ■åĚĄèőÿå£¡çŢěäžžä¡ŢåŔĆæŢřãĂĆåŻăæ■d’æIJĽäžĘäÿŃéİćçŽĎçżŞæđIJïijŽ

```
>>> bound_values = sig.bind(1, 2, 3)
>>> bound_values.arguments
OrderedDict([('x', 1), ('y', 2), ('z', 3)])
>>>
```

ä¡£çŤĺè£ŹäÿłæŸăåřĎæĹŚäżňåŔŕäżěå¿Ĺè¡żæİ¿çŽĎåőđçŐřæĹŚäżňçŽĎåijžåĹűçśżåđŃæčĂæ§ěïijŽ

```
>>> for name, value in bound_values.arguments.items():
...     if name in bound_types.arguments:
...         if not isinstance(value, bound_types.arguments[name]):
...             raise TypeError()
...
>>>
```

äÿ■è£Ğè£ŹäÿłæŰźæąĹè£ŸæIJĽçĆźåřŔçŚŢçŰţïijŇåőČåŕźäžŐæIJĽéżŸèőďåĂijçŽĎåŔĆæŢřåźűäÿ■éĂ
æŕŤåęĆäÿŃéİćçŽĎäżčçăĄåŔŕäżěæ■čåÿÿåůěä¡IJïijŇåř¡çőąitemsçŽĎçśżåđŃæŸŕéŤŹèŕŕçŽĎïijŽ

```
>>> @typeassert(int, list)
... def bar(x, items=None):
...     if items is None:
...         items = []
...     items.append(x)
...     return items
>>> bar(2)
[2]
>>> bar(2,3)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "contract.py", line 33, in wrapper
TypeError: Argument items must be <class 'list'>
>>> bar(4, [1, 2, 3])
[1, 2, 3, 4]
>>>
```

æIJĂåŘŐäÿĂçĆźæŸŕåĚşäžŐéĂĆçŤĺèčĚéěřåŹĺåŔĆæŢřåŠŇåĞ¡æŢřæşĺèğčäźŃéŮťçŽĎäžĽèőžãĂĆ
ä¿ŃåęĆïijŇäÿžäżĂäźĹäÿ■åČŔäÿŃéİćè£ŹæăůåĘŹäÿĂäÿłèčĚéěřåŹĺæİěæ§ěæĽ¿åĞ¡æŢřäÿ■çŽĎæşĺèğčåŚćïij§

```
@typeassert
def spam(x:int, y, z:int = 42):
    print(x,y,z)
```

äÿĂäÿłåŔŕèČ¡çŽĎåŐ§åŻăæŸŕåęĆæđIJä¡£çŤĺäžĘåĞ¡æŢřåŔĆæŢřæşĺèğčïijŇéĆčäźĹåřśèćńéŹŘåĹűäžĘãĂ
åęĆæđIJæşĺèğčèćńçŤĺæİěåĄŽçśżåđŃæčĂæ§ěåřśäÿ■èČ¡åĄŽåĚűäżŰäžŃæČĚäžĘãĂĆèĂŇäÿŤ
@typeassert äÿ■èČ¡åĘ■çŤĺäžŐä¡£çŤĺæşĺèğčåĄŽåĚűäżŰäžŃæČĚçŽĎåĞ¡æŢřäžĘãĂĆ
èĂŇä¡£çŤĺäÿŁéİćçŽĎèčĚéěřåŹĺåŔĆæŢřçĄţæťżæĂğåďğåďŽäžĘïijŇäź§æŻťåŁäéĂŽçŤĺãĂĆ

åŔŕäżěåIJĺPEP 362äżěåŔŁ `inspect` æĺąåİŮäÿ■æĽ¿åĹřæŻťåďŽåĚşäžŐåĞ¡æŢřåŔĆæŢřåŕźèśąçŽĎä£ą

11.8 9.8 åřĘèčĚéěřåŹĺåőŽäźĽäÿžçśżçŽĎäÿĂéČĺåĹĘ

éŮőéćŸ

ä¡ăæČşåIJĺçśżäÿ■åőŽäźĽèčĚéěřåŹĺïijŇåźűåřĘåĚűä¡IJçŤĺåIJĺåĚűäżŰåĞ¡æŢřæĹŰæŰźæşŢäÿŁãĂĆ

èğčåĘşæŰźæąĹ

åIJÍçśżéĞŇéİćåőŽäźĽ'èčĚéěřåŹÍå¿ĹçőĂå■ŢïijŇä¡ĘæŸŕä¡ăéęŰåĚĹèęĄçąőèőď'åőČçŽĎä¡£çŤÍæŰźåijŔãÁ
äÿŃéİćæĹŚäżñçŤÍä¿Ńå■ŘæİěéŸŘè£řåőČäżñçŽĎäÿ■åŘŇïijŽ

```
from functools import wraps

class A:
    # Decorator as an instance method
    def decorator1(self, func):
        @wraps(func)
        def wrapper(*args, **kwargs):
            print('Decorator 1')
            return func(*args, **kwargs)
        return wrapper

    # Decorator as a class method
    @classmethod
    def decorator2(cls, func):
        @wraps(func)
        def wrapper(*args, **kwargs):
            print('Decorator 2')
            return func(*args, **kwargs)
        return wrapper
```

äÿŃéİćæŸŕäÿĂä¡£çŤÍä¿Ńå■ŘïijŽ

```
# As an instance method
a = A()
@a.decorator1
def spam():
    pass
# As a class method
@A.decorator2
def grok():
    pass
```

äżŤçżĘèğĆåŕ§åŔŕäżěåŔŚçŐřäÿĂäÿłæŸŕåőđä¿ŃèřČçŤÍïijŇäÿĂäÿłæŸŕçśżèřČçŤÍãĂĆ

èőÍèőž

åIJÍçśżäÿ■åőŽäźĽ'èčĚéěřåŹÍåĹİçIJŃäÿŁåŐżåě¡åČŔå¿ĹåěĞæĂłïijŇä¡ĘæŸŕåIJÍæăĞåĞĘåžŞäÿ■æIJL'å¿Ĺ
çĽźåĹńçŽĎïijŇ`@property` èčĚéěřåŹÍåőđéŹĚäÿŁæŸŕäÿĂäÿłçśżïijŇåőČéĞŇéİćåőŽäźĽ'äžĘäÿĽ'äÿłæŰźæşŢ
`getter()`, `setter()`, `deleter()`,æŕŔäÿĂäÿłæŰźæşŢéČ¡æŸŕäÿĂäÿłèčĚéěřåŹÍãĂĆä¿ŃåęĆïijŽ

```
class Person:
    # Create a property instance
    first_name = property()

    # Apply decorator methods
```

```
    @first_name.getter
    def first_name(self):
        return self._first_name

    @first_name.setter
    def first_name(self, value):
        if not isinstance(value, str):
            raise TypeError('Expected a string')
        self._first_name = value
```

åőČäÿžäżĂäźĹèęĄè£ŹäźĹåőŽäźL'çŽĎäÿżèęĄåŐ§åŻäæŸŕåŘĎçğ■äÿ■åŘŇçŽĎèčĚéěřåŹĺæŰźæşŢäijŽåI
`property` åőđä¿ŃäÿŁæŞ■ä¡IJåőČçŽĎçŁűæĂĄãĂĆ åŻăæ■d'ïijŇäżżä¡ŢæŮűåĂŹåŘłèęĄä¡ăçćřåĽřéIJĂèęĄå

åIJĺçśżäÿ■åőŽäźL'èčĚéěřåŹĺæIJL'äÿłéŽ¿çŘĘèğčçŽĎåIJřæŰźåřśæŸŕåŕźäžŐéćÍåd'ŰåŔĆæŢř
`self` æĹŰ `cls` çŽĎæ■čçąőä¡£çŤĺãĂĆ åř¡çőąæIJĂåd'ŰåśĆçŽĎèčĚéěřåŹĺåĞ¡æŢřæŕŤåęĆ
`decorator1()` æĹŰ `decorator2()` éIJĂèęĄæŔŘä¿ŻäÿĂäÿł `self`
æĹŰ `cls` åŔĆæŢřïijŇ ä¡ĘæŸŕåIJĺäÿd'äÿłèčĚéěřåŹĺåĘĚéČĺèćńåĹŻåżžçŽĎ
`wrapper()` åĞ¡æŢřåźűäÿ■éIJĂèęĄåŇĚåŘńè£Źäÿł `self` åŔĆæŢřãĂĆ
ä¡ăåŤŕäÿĂéIJĂèęĄè£ŹäÿłåŔĆæŢřæŸŕåIJĺä¡äçąőåőđèęĄèő£éŮőåŇĚèčĚåŹĺäÿ■è£Źäÿłåőđä¿ŃçŽĎæ§ŘäżŽéČ

åŕźäžŐçśżéĞŇéİćåőŽäźL'çŽĎåŇĚèčĚåŹĺè£ŸæIJL'äÿĂçĆźæŕŤè¿ČéŽ¿çŘĘèğčïijŇåŕśæŸŕåIJĺæűL'åŘŁåĺ
ä¿ŃåęĆïijŇåĄĞèő¿ä¡ăæČşèőľ'åIJĺAäÿ■åőŽäźL'çŽĎèčĚéěřåŹĺä¡IJçŤĺåIJĺå■ŘçśżBäÿ■ãĂĆä¡ăéIJĂèęĄåČŔäÿŇ

```
class B(A):
    @A.decorator2
    def bar(self):
        pass
```

äź§åŕśæŸŕèŕt'ïijŇèčĚéěřåŹĺèęĄèćńåőŽäźL'æĹŘçśżæŰźæşŢåźűäÿŤä¡ăå£ĚéążæŸ¿åijŔçŽĎä¡£çŤĺçĹűçśż
ä¡ăäÿ■èČ¡ä¡£çŤĺ `@B.decorator2` ïijŇåŽăäÿžåIJĺæŰźæşŢåőŽäźL'æŮűïijŇè£ŹäÿłçśżBè£ŸæšąæIJL'èćńåĹŽ

11.9 9.9 åřĘèčĚéěřåŹĺåőŽäźL'äÿžçśż

éŮőécŸ

ä¡ăæČşä¡£çŤĺäÿĂäÿłèčĚéěřåŹĺåŐżåŇĚèčĚåĞ¡æŢřïijŇä¡EæŸŕåÿŇæIJŻè£ŤåŽđäÿĂäÿłåŔŕèřČçŤĺçŽĎåő
ä¡ăéIJĂèęĄèőľä¡äçŽĎèčĚéěřåŹĺåŔŕäżěåŘŇæŮűåůěä¡IJåIJĺçśżåőŽäźL'çŽĎåĘĚéČĺåŠŇåd'ŰéČĺãĂĆ

èğčåĘşæŰźæąĹ

äÿžäžĘåřĘèčĚéěřåŹĺåőŽäźL'æĹŘäÿĂäÿłåőđä¿ŃïijŇä¡ăéIJĂèęĄçąőä£ÍåőČåőđçŐřäžĘ
`__call__()` åŠŇ `__get__()` æŰźæşŢãĂĆ ä¿ŃåęĆïijŇäÿŃéİćçŽĎäżčçăĄåőŽäźL'äžĘäÿĂäÿłçśżïijŇåőČå

```
import types
from functools import wraps

class Profiled:
    def __init__(self, func):
```

```
        wraps(func)(self)
        self.ncalls = 0

    def __call__(self, *args, **kwargs):
        self.ncalls += 1
        return self.__wrapped__(*args, **kwargs)

    def __get__(self, instance, cls):
        if instance is None:
            return self
        else:
            return types.MethodType(self, instance)
```

ä¡ääŔŕäžěářĘåőČå¡ŞåĄŽäÿĂäÿłæŹőéĂŽçŽĎèčĚéěřåŹĺæİěä¡£çŤĺïijŇåIJĺçśžéĞŇéİćæĹŰåď˙ŰéİćéČ¡åŔŕ

```
@Profiled
def add(x, y):
    return x + y

class Spam:
    @Profiled
    def bar(self, x):
        print(self, x)
```

åIJĺäžď˙äžŠçŐŕåćČäÿ■çŽĎä¡£çŤĺçď˙žä¿ŃïijŽ

```
>>> add(2, 3)
5
>>> add(4, 5)
9
>>> add.ncalls
2
>>> s = Spam()
>>> s.bar(1)
<__main__.Spam object at 0x10069e9d0> 1
>>> s.bar(2)
<__main__.Spam object at 0x10069e9d0> 2
>>> s.bar(3)
<__main__.Spam object at 0x10069e9d0> 3
>>> Spam.bar.ncalls
3
```

èőĺèőž

åřĘèčĚéěřåŹĺåőŽäźL˙æĹŘçśžéĂŽåÿÿæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆä¡ĘæŸŕè£ŹéĞŇè£ŸæŸŕæIJL˙äÿĂäžŻçżĘ

éęŰåĚĹïijŇä¡£çŤĺ `functools.wraps()` åĞ¡æŢřçŽĎä¡IJçŤĺèů§äźŇåL˙■è£ŸæŸŕäÿĂæăůïijŇåřĘèćńå

åĚűæňąïijŇéĂŽåÿÿå¿Ĺåőźæ˙ŸŞäijŽå£¡èğĘäÿŁéİćçŽĎ `__get__()` æŰźæşŢãĂĆåęĆæđIJä¡åå£¡çŢěåőČïijŇä£ÍæŇĄåĚűäžŰäžčçăĄäÿ■åŔŸåĘ■æňąè£ŘèąŇïijŇ

ä¡ääijŽåŔŚçŐřå¡Şä¡ääŐżèřČçŤĺèćńèčĚéěřåőđä¿ŃæŰźæşŢæŮűåĞžçŐřå¿ĹåěĞæĂłçŽĎéŮőécŸãĂĆä¿ŃåęĆï

```
>>> s = Spam()
>>> s.bar(3)
Traceback (most recent call last):
...
TypeError: bar() missing 1 required positional argument: 'x'
```

åĞžéŤŹåŐ§åŻäæŸřå¡ŞæŰźæşŢåĞ¡æŢřåIJĺäÿĂäÿłçśżäÿ■èćńæ§ěæL’¿æŮűïijŇåőČäżňçŽĎ __get__() æŰźæşŢä¿İæ■őæŔŔè£řåŹĺå■ŔèőőèćńèřČçŤĺïijŇ åIJĺ8.9åřŔèŁĆåůšçżŔèőšè£řè£ĞæŔŔè£řåŹĺå■Ŕèőőäž ĘãĂĆåIJĺè£ŹéĞŇïijŇ__get__() çŽĎçŽőçŽĎæŸřåĽŻåżžäÿĂäÿłçżŚåőŽæŰźæşŢåŕźèšą (æIJĂçżĹäijŽçżŹè£ŹäÿłæŰźæşŢäijåéĂŠselfåŔĆæŢř)ã

```
>>> s = Spam()
>>> def grok(self, x):
...     pass
...
>>> grok.__get__(s, Spam)
<bound method Spam.grok of <__main__.Spam object at 0x100671e90>>
>>>
```

__get__() æŰźæşŢæŸřäÿžäžĘçąőä£İçżŚåőŽæŰźæşŢåŕźèšąèČ¡èćńæ■čçąőçŽĎåĽŻåżžãĂĆ type.MethodType() æĽŃåŁĺåĽŻåżžäÿĂäÿłçżŚåőŽæŰźæşŢæİěä¡£çŤĺãĂĆåŔłæIJL’å¡Şåőđä¿ŃèćńäĄ£çŤ åęĆæđIJè£ŹäÿłæŰźæşŢæŸřåIJĺçśżäÿŁéİćæİěèő£éŮőïijŇ éĆčäźĹ __get__() äÿ■çŽĎin-stanceåŔĆæŢřäijŽèćńèő¿ç¡őæĽŘNoneåźűçŻt’æŐěè£ŤåŻđ Profiled åőđä¿ŃæIJňèžńãĂĆ è£ŹæåůçŽĎèřİæĹŚäžňåřšåŔřäżěæŔŘåŔŰåőČçŽĎ ncalls åšđæĂğäžĘãĂĆ

åęĆæđIJä¡ăæČşéĄ£åĚ■äÿĂäžŻæůůäźśïijŇäź§åŔřäżěèĂČèŹŚåŔęåd’ŰäÿĂäÿłä¡£çŤĺéŮ■åŇĚåŠŇ nonlocal åŔŸéĞŘåőđçŐřçŽĎèčĚéěřåŹĺïijŇè£ŹäÿłåIJĺ9.5åřŔèŁĆæIJL’èőšåĹřãĂĆä¿ŃåęĆïijŽ

```
import types
from functools import wraps

def profiled(func):
    ncalls = 0
    @wraps(func)
    def wrapper(*args, **kwargs):
        nonlocal ncalls
        ncalls += 1
        return func(*args, **kwargs)
    wrapper.ncalls = lambda: ncalls
    return wrapper

# Example
@profiled
def add(x, y):
    return x + y
```

è£ŹäÿłæŰźåijŔèů§äźŃåL’■çŽĎæŢĹæđIJåĞääźŐäÿĂæăůïijŇéŹd’äžĘåŕźäžŐ ncalls çŽĎèő£éŮőçŐřåIJĺæŸřéĂŽè£ĞäÿĂäÿłèćńçżŚåőŽäÿžåšđæĂğçŽĎåĞ¡æŢřæİěåőđçŐřïijŇä¿ŃåęĆïijŽ

```
>>> add(2, 3)
5
>>> add(4, 5)
9
>>> add.ncalls()
2
>>>
```

11.10 9.10 äÿžçśżåŠŇéİŹæĂĄæŰźæşŢæŔŘä¿ŻèčĚéěřåŹÍ

éŮőécŸ

ä¡ăæČşçżŹçśżæĹŰéİŹæĂĄæŰźæşŢæŔŘä¿ŻèčĚéěřåŹÍãĂĆ

èğčåĘşæŰźæąĹ

çżŹçśżæĹŰéİŹæĂĄæŰźæşŢæŔŘä¿ŻèčĚéěřåŹÍæŸřå¿ĹçőĂå■ŢçŽĎïijŇäÿ■è£ĞèęAçąőä£İèčĚéěřåŹÍåIJ
`@classmethod` æĹŰ `@staticmethod` äźŃåL’■ãĂĆä¿ŃåęĆïijŽ

```
import time
from functools import wraps

# A simple decorator
def timethis(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        start = time.time()
        r = func(*args, **kwargs)
        end = time.time()
        print(end-start)
        return r
    return wrapper

# Class illustrating application of the decorator to different␣
↪kinds of methods
class Spam:
    @timethis
    def instance_method(self, n):
        print(self, n)
        while n > 0:
            n -= 1

    @classmethod
    @timethis
    def class_method(cls, n):
        print(cls, n)
        while n > 0:
```

```
            n -= 1

    @staticmethod
    @timethis
    def static_method(n):
        print(n)
        while n > 0:
            n -= 1
```

èčĚéěřåŘŐçŽĎçśżåŠŇéİŹæĂĄæŰźæşŢåŔŕæ■čåÿÿåůěä¡IJïijŇåŔłäÿ■è£Ğåćđå£ăäžĘéćİåđ’ŰçŽĎèőąæŮ

```
>>> s = Spam()
>>> s.instance_method(1000000)
<__main__.Spam object at 0x1006a6050> 1000000
0.11817407608032227
>>> Spam.class_method(1000000)
<class '__main__.Spam'> 1000000
0.11334395408630371
>>> Spam.static_method(1000000)
1000000
0.11740279197692871
>>>
```

èőÍèőž

åęĆæđIJä¡ăæŁŁèčĚéěřåŹÍçŽĎéąžåžŔåĘŹéŤŹäžĘåřśäijŽåĞžéŤŹãĂĆä¿ŃåęĆïijŇåĄĞèő¿ä¡ăåČŔäÿŃéİć

```
class Spam:
    @timethis
    @staticmethod
    def static_method(n):
        print(n)
        while n > 0:
            n -= 1
```

éĆčäźĹä¡ăèřČçŤÍè£ŹäÿłéİŹæĂĄæŰźæşŢæŮűůåřśäijŽæŁěéŤŹïijŽ

```
>>> Spam.static_method(1000000)
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
File "timethis.py", line 6, in wrapper
start = time.time()
TypeError: 'staticmethod' object is not callable
>>>
```

éŮőécŸåIJÍäžŐ `@classmethod` åŠŇ `@staticmethod` åőđéŹĚäÿŁåźűäÿ■äijŽåĹŽåżžåŔŕçŻt’æŐěřČçŤÍçŽĎåŕźèśąïijŇ
èĂŇæŸŕåĹŽåżžçL’źæőŁçŽĎæŔŔè£ŕåŹÍåŕźèśą(åŔĆèĂČ8.9åřŔèŁĆ)ãĂĆåŻăæ■d’å¡Şä¡ăèŕŢçİĂåIJÍåĚűäžŰèč
çąőä£Ìè£Źçğ■èčĚéěřåŹÍåĞžçŐřåIJÍèčĚéěřåŹÍéŞ¿äÿ■çŽĎçňňäÿĂäÿłä¡■ç¡őåŔŕäżěä£őåd’■è£ŹäÿłéŮőécŸãĂĆ

å¡ŞæĹŚäžňåIJÍæŁ¡èśąå§žçśżäÿ■åőŽäźL’çśżæŰźæşŢåŠŇéİŹæĂĄæŰźæşŢ(åŔĆèĂČ8.12åřŔèŁĆ)æŮűïijŇ
ä¿ŇåęĆïijŇåęĆæđIJä¡äæČşåőŽäźL’äÿĂäÿłæŁ¡èśąçśżæŰźæşŢïijŇåŔřäžěä¡£çŤĺçśżäijijäÿŇéİćçŽĎäžčçăĄïijŽ

```
from abc import ABCMeta, abstractmethod
class A(metaclass=ABCMeta):
    @classmethod
    @abstractmethod
    def method(cls):
        pass
```

åIJÍè£Źæőţäžčçă Ąäÿ■ïijŇ@classmethod èů§ @abstractmethod äÿd’èĂĚçŽĎéąžåžŘæŸŕæIJL’èőšçl’űçŽĎïijŇåęĆæđIJä¡ăèřČæ■ćåőČäžňçŽĎéąžåžŘåřśäijŽåĞžéŤŹãĂĆ

11.11 9.11 èčĚéěřåŹĺäÿžèćńåŇĚèčĚåĞ¡æŢřåćđåŁååŔĆæŢř

éŮőécŸ

ä¡ăæČşåIJÍèčĚéěřåŹĺäÿ■çżŹèćńåŇĚèčĚåĞ¡æŢřåćđåŁăécİåd’ŰçŽĎåŔĆæŢřïijŇä¡ĘæŸŕäÿ■èČ¡å¡śåŞ■è£Ź

èğčåĘşæŰźæąĹ

åŘŕäžěä¡£çŤĺåĚşéŤőå■ŮåŔĆæŢřæİěçżŹèćńåŇĚèčĚåĞ¡æŢřåćđåŁăécİåd’ŰåŔĆæŢřãĂĆèĂČèŹŚäÿŇéİćç

```
from functools import wraps

def optional_debug(func):
    @wraps(func)
    def wrapper(*args, debug=False, **kwargs):
        if debug:
            print('Calling', func.__name__)
        return func(*args, **kwargs)

    return wrapper
```

```
>>> @optional_debug
... def spam(a,b,c):
...     print(a,b,c)
...
>>> spam(1,2,3)
1 2 3
>>> spam(1,2,3, debug=True)
Calling spam
1 2 3
>>>
```

èőÍèőž

éĂŽè£ĞèčĚéěřåŹÍæİěçżŹèćńåŇĚèčĚåĞ¡æŢřåćđåŁååŔĆæŢřçŽĎåĄŽæşŢåźűäÿ■åÿÿèğĄãĂĆ
år¡çőąåęĆæ■ď’ïijŇæIJL’æŮűåĂŹåőČåŔŕäžěéĄ£åĚ■äÿĂäžŻéĞ■åď’■äžčçăĄãĂĆä¿ŃåęĆïijŇåęĆæđIJä¡ăæIJL’

```
def a(x, debug=False):
    if debug:
        print('Calling a')

def b(x, y, z, debug=False):
    if debug:
        print('Calling b')

def c(x, y, debug=False):
    if debug:
        print('Calling c')
```

éĆčäźĹä¡ăåŔŕäžěåřĘåĚűéĞ■æđĎæĹŘè£Źæăůïijž

```
from functools import wraps
import inspect

def optional_debug(func):
    if 'debug' in inspect.getargspec(func).args:
        raise TypeError('debug argument already defined')

    @wraps(func)
    def wrapper(*args, debug=False, **kwargs):
        if debug:
            print('Calling', func.__name__)
        return func(*args, **kwargs)
    return wrapper

@optional_debug
def a(x):
    pass

@optional_debug
def b(x, y, z):
    pass

@optional_debug
def c(x, y):
    pass
```

è£Źçğ■åőđçŐřæŰźæąĹäźŃæL’ĂäžěèąŇå¿ŮéĂŽïijŇåIJÍäžŐåijžåĹűåĚşéŤőå■ŮåŔĆæŢřå¿Ĺåőźæ
`*args` åŠŇ `**kwargs` åŔĆæŢřçŽĎåĞ¡æŢřäÿ■ãĂĆ éĂŽè£Ğä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŔĆæŢřïijŇåőČèćńä
åźűäÿŤæŐěäÿŃæİěäžĚäžĚä¡£çŤĺåL’l’ä¡ŹçŽĎä¡■ç¡őåŠŇåĚşéŤőå■ŮåŔĆæŢřåŐżèřČçŤĺè£ŹäÿłåĞ¡æŢřæŮűïij
äź§åřśæŸŕèŕť’ïijŇåőČåźűäÿ■äijŽèćńçžşåĚěåĹř `**kwargs` äÿ■åŐžãĂĆ

è£ŸæIJL’äÿĂäÿłéŽ¿çĆźåřśæŸŕåęĆä¡ŢåŐžåď’ĎçŘĘèćńæůžåŁăçŽĎåŔĆæŢřäÿŐèćńåŇĚèčĚåĞ¡æŢřåŔĆ

ä¿ÑåęĆïijÑåęĆæđIJèčĚéěřåŹÍ `@optional_debug` ä¡IJçŤĺåIJĺäÿĂäÿłåůšçżŔæŃěæIJL'äÿĂäÿł `debug` åŘĆæŢřçŽĎåĞ¡æŢřäÿŁæŮűäijŽæIJL'éŮőécŸãĂĆ è£ŹéĞŇæĹŚäžňåćđåŁääžĘäÿĂæ■ěåŘ■å■ŮæčĂæ

äÿŁéİćçŽĎæŰźæąĹè£ŸåŔŕäżěæŻť'åőÑç¿ŐäÿĂçĆźïijÑåŽääÿžçš¿æŸŐçŽĎçÍÑåžŔåŚŸåžŤèŕěåŔŚçŐřäž

```
>>> @optional_debug
... def add(x,y):
...     return x+y
...
>>> import inspect
>>> print(inspect.signature(add))
(x, y)
>>>
```

éĂŽè£ĞåęĆäÿÑçŽĎä£őæŤźïijÑåŔŕäżěèğčåĘşè£ŹäÿłéŮőécŸïijŽ

```
from functools import wraps
import inspect

def optional_debug(func):
    if 'debug' in inspect.getargspec(func).args:
        raise TypeError('debug argument already defined')

    @wraps(func)
    def wrapper(*args, debug=False, **kwargs):
        if debug:
            print('Calling', func.__name__)
        return func(*args, **kwargs)

    sig = inspect.signature(func)
    parms = list(sig.parameters.values())
    parms.append(inspect.Parameter('debug',
                inspect.Parameter.KEYWORD_ONLY,
                default=False))
    wrapper.__signature__ = sig.replace(parameters=parms)
    return wrapper
```

éĂŽè£Ğè£ŹæăůçŽĎä£őæŤźïijÑåŇĚèčĚåŘŐçŽĎåĞ¡æŢřç■¿åŘ■åřśèČ¡æ■čçąőçŽĎæŸ¿çd'ž `debug` åŘĆæŢřçŽĎå■ŸåIJĺäžĘãĂĆä¿ÑåęĆïijŽ

```
>>> @optional_debug
... def add(x,y):
...     return x+y
...
>>> print(inspect.signature(add))
(x, y, *, debug=False)
>>> add(2,3)
5
>>>
```

åŘĆèĂČ9.16åřŔèŁĆèŐůåŘŰæŻť'åd'ŽåĚşäžŐåĞ¡æŢřç■¿åŘ■çŽĎä£ąæĄŕãĂĆ

11.12 9.12 ä¡£çŤÍèčĚéěřåŹÍæĽľåĚĚçśżçŽĎåŁ§èČ¡

éŮőécŸ

ä¡ăæČşéĂŽè£ĞåŔ■çIJĄæĹŰèĂĚéĞ■åĘŹçśżåőŽäźĽ'çŽĎæ§ŘéČÍåĹĘæİěä£őæŤźåőČçŽĎèąŇäÿžïijŇä¡E

èğčåĘşæŰźæąĹ

è£Źçğ■æČĚåĘţåŔŕèČ¡æŸŕçśżèčĚéěřåŹÍæIJĂåě¡çŽĎä¡£çŤÍåIJžæŹŕäžĘãĂĆä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂä
__getattribute__ çŽĎçśżèčĚéěřåŹÍïijŇ åŔŕäżěæĽ'Şå■řæŮěå£ŮïijŽ

```
def log_getattribute(cls):
    # Get the original implementation
    orig_getattribute = cls.__getattribute__

    # Make a new definition
    def new_getattribute(self, name):
        print('getting:', name)
        return orig_getattribute(self, name)

    # Attach to the class and return
    cls.__getattribute__ = new_getattribute
    return cls

# Example use
@log_getattribute
class A:
    def __init__(self,x):
        self.x = x
    def spam(self):
        pass
```

äÿŃéİćæŸŕä¡£çŤÍæŢĹæđIJïijŽ

```
>>> a = A(42)
>>> a.x
getting: x
42
>>> a.spam()
getting: spam
>>>
```

èőÍèőž

çśżèčĚéěřåŹÍéĂŽåÿÿåŔŕäżěä¡IJäÿžåĚűäżŰéńŸçžğæŁĂæIJŕæŕŤåęĆæůůåĚěæĹŰåĚČçśżçŽĎäÿĂçğ■éİđă
ærŤåęĆïijŇäÿŁéİćçd'žä¿Ńäÿ■çŽĎåŔęåd'ŰäÿĂçğ■åőđçŐřä¡£çŤÍåĹřçžğæĽ£ïijŽ

```
class LoggedGetattribute:
    def __getattribute__(self, name):
        print('getting:', name)
        return super().__getattribute__(name)

# Example:
class A(LoggedGetattribute):
    def __init__(self,x):
        self.x = x
    def spam(self):
        pass
```

è£Źçğ■æŰźæąĹäź§èąŇå¿ŮéĂŽïijŇä¡ĘæŸŕäÿžäžĘåŐżçŘĘèğčåőČïijŇä¡ăåřśå£Ěéążç§ěéĄŞæŰźæşŢèřČ
äžěåŔŁåĚŰåőČ8.7åřŔèŁĆäżŃçż■çŽĎçżğæL’£ç§ěèřĘãĂĆ æ§Řçğ■çĺŃåžęäÿŁæİěèőšïijŇçśžèčĚéěřåŹĺæŰźæą
åŽääÿžäžŰåźűäÿ■ä¿İèţŰ `super()` åĞ¡æŢřãĂĆ

åęĆæđIJä¡ăçşżæČşåIJĺäÿĂäÿłçśżäÿŁéİć䡣çŤĺåď’ŽäÿłçśżèčĚéěřåŹĺïijŇéĆčäźĹåřśéIJĂèęĄæşĺæĎŔäÿŃé
ä¿ŃåęĆïijŇäÿĂäÿłèčĚéěřåŹĺAäijŽåřĘåĚűèčĚéěřçŽĎæŰźæşŢåőŇæŢt’æŻ£æ■ćæĹŘåŔęäÿĂçğ■åőđçŐřïijŇ
èĂŇåŔęäÿĂäÿłèčĚéěřåŹĺBåŔłæŸŕçőĂå■ŢçŽĎåIJĺåĚűèčĚéěřçŽĎæŰźæşŢäÿ■æůžåŁăçĆźéćĺåď’ŰéĂżè¿ŚãĂ
éĆčäźĹè£ŹæŮűåĂŹèčĚéěřåŹĺAåřśéIJĂèęĄæŤ¿åIJĺèčĚéěřåŹĺBçŽĎåL’■éİćãĂĆ

ä¡ăè£ŸåŔŕäžěåŻđéą¿äÿĂäÿŃ8.13åřŔèŁĆåŔęåď’ŰäÿĂäÿłåĚşäžŐçśżèčĚéěřåŹĺçŽĎæIJL’çŤĺçŽĎä¿Ńå■Ř

11.13 9.13 ä¡£çŤĺåĚČçśżæŐğåĹűåőđä¿ŃçŽĎåĹŻäžž

éŮőécŸ

ä¡ăæČşéĂŽè£ĞæŤźåŔŸåőđä¿ŃåĹŻäžžæŰźåijŔæİèåőđçŐřå■Ţä¿ŃãĂĄçijŞå■ŸæĹŰåĚűäžŰçśżäijijçŽĎ

èğčåĘşæŰźæąĹ

PythonçĺŃåžŔåŚŸéČ¡ç§ěéĄŞïijŇåęĆæđIJä¡ăåőŽäźL’äžĘäÿĂäÿłçśżïijŇåřśèČ¡åČŔåĞ¡æŢřäÿĂæăůçŽĎè

```
class Spam:
    def __init__(self, name):
        self.name = name

a = Spam('Guido')
b = Spam('Diana')
```

åęĆæđIJä¡ăæČşèĞłåőŽäźL’è£Źäÿłæ■ěélď’ïijŇä¡ăåŔŕäžěåőŽäźL’äÿĂäÿłåĚČçśżåźűèĞłåůśåőđçŐř
`__call__()` æŰźæşŢãĂĆ

äÿžäžĘæijŤçď’žïijŇåĄĞèő¿ä¡ăäÿ■æČşäżżä¡ŢäžžåĺŽäžžè£ŹäÿłçśżçŽĎåőđä¿ŃïijŽ

```
class NoInstances(type):
    def __call__(self, *args, **kwargs):
        raise TypeError("Can't instantiate directly")
```

```
# Example
class Spam(metaclass=NoInstances):
    @staticmethod
    def grok(x):
        print('Spam.grok')
```

è£ŹæăůçŽĎèŕÍïijŇçŤĺæĹůåŔłèČ¡èřČçŤĺè£ŹäÿłçśżçŽĎéİŹæĂĄæŰźæşŢïijŇèĂŇäÿ■èČ¡ä¡£çŤĺéĂŽåÿÿç

```
>>> Spam.grok(42)
Spam.grok
>>> s = Spam()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "example1.py", line 7, in __call__
        raise TypeError("Can't instantiate directly")
TypeError: Can't instantiate directly
>>>
```

çŐřåIJÍïijŇåĄĞåęĆä¡ăæČşåőđçŐřå■Ţä¿ŃæĺąåijŔïijĹåŔłèČ¡åĹŻåžžåŤŕäÿĂåőđä¿ŃçŽĎçśżïijL’ïijŇåőđçĆ

```
class Singleton(type):
    def __init__(self, *args, **kwargs):
        self.__instance = None
        super().__init__(*args, **kwargs)

    def __call__(self, *args, **kwargs):
        if self.__instance is None:
            self.__instance = super().__call__(*args, **kwargs)
            return self.__instance
        else:
            return self.__instance

# Example
class Spam(metaclass=Singleton):
    def __init__(self):
        print('Creating Spam')
```

éĆčäźĹSpamçśżåřśåŔłèČ¡åĹŻåžžåŤŕäÿĂçŽĎåőđä¿ŃäžĘïijŇæijŤçd’žåęĆäÿŃïijŽ

```
>>> a = Spam()
Creating Spam
>>> b = Spam()
>>> a is b
True
>>> c = Spam()
>>> a is c
True
>>>
```

æIJĂåŘŐïijŇåĄĞèő¿ä¡ăæČşåĹŻåžž8.25åřŔèŁĆäÿ■éĆčæăůçŽĎçijŞå■Ÿåőđä¿ŃãĂĆäÿŃéİćæĹŚäžňåŔŕä

```
import weakref

class Cached(type):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.__cache = weakref.WeakValueDictionary()

    def __call__(self, *args):
        if args in self.__cache:
            return self.__cache[args]
        else:
            obj = super().__call__(*args)
            self.__cache[args] = obj
            return obj

# Example
class Spam(metaclass=Cached):
    def __init__(self, name):
        print('Creating Spam({!r})'.format(name))
        self.name = name
```

çĎűåŘŐæĹŚäź§æİěæţŃèŕŢäÿĂäÿŃïijŽ

```
>>> a = Spam('Guido')
Creating Spam('Guido')
>>> b = Spam('Diana')
Creating Spam('Diana')
>>> c = Spam('Guido') # Cached
>>> a is b
False
>>> a is c # Cached value returned
True
>>>
```

èőÍèőž

åĹľçŤĺåĚČçśżåőđçŐřåď'Žçğ■åőđä¿ŃåĹŻåżžæÍąåijŔéĂŽåÿÿèęĄæŕŤäÿ■ä¡£çŤĺåĚČçśżçŽĎæŰźåijŔäijŸ

åĄĞèő¿ä¡ääÿ■ä¡£çŤĺåĚČçśżïijŇä¡ăåŔŕèČ¡éIJĂèęĄåřĘçśżéŽŘèŮŔåIJÍæ§ŘäžŻåůěåŐĆåĞ¡æŢřåŘŐéİćãĂæŕŤåęĆäÿžäžĘåőđçŐřäÿĂäÿłå■Ţä¿ŃïijŇä¡ää¡ăåŔŕèČ¡äijŽåČŔäÿŃéİćè£ŹæăůåĘŹïijŽ

```
class _Spam:
    def __init__(self):
        print('Creating Spam')

_spam_instance = None

def Spam():
    global _spam_instance
```

```
    if _spam_instance is not None:
        return _spam_instance
    else:
        _spam_instance = _Spam()
        return _spam_instance
```

åř¡çőąä¡£çŤĺåĚČçśżåŔŕèČ¡äijŽæűĽ’åŔŁåĹřæŕŤè¿ČéńŸçžğçĆźçŽĎæŁĂæIJŕïijŇä¡ĘæŸŕåőČçŽĎäżčçăĄ

æŻť’åď’ŽåĚşäžŐåĹŻåżžçijŞå■Ÿåőđä¿ŃãĂĄåijśåijŢçŤĺç■Ľ’åĘĚåőźïijŇèŕůåŔĆèĂČ8.25åřŔèŁĆãĂĆ

11.14 9.14 æ■ŢèŐůçśżçŽĎåśđæĂğåőŽäźĽéąžåžŔ

éŮőécŸ

ä¡ăæČşèĞłåŁĺèőřå¡ŢäÿĂäÿłçśżäÿ■åśđæĂğåŠŇæŰźæşŢåőŽäźĽ’çŽĎéąžåžŔïijŇ çĎűåŘŐåŔŕäżčåĹľ’çŤĺåőČæİěåĄŽå¿Ĺåď’ŽæŞ■ä¡IJïijĹæŕŤåęĆåžŔåĹŮåŇŰãĂĄæŸäářĎåĹřæŢřæ■őåžŞç■Ľ’ç

èğčåĘşæŰźæąĹ

åĹľ’çŤĺåĚČçśżåŔŕäżčå¿Ĺåőźæ ŸŞçŽĎæ■ŢèŐůçśżçŽĎåőŽäźĽ’ä£ąæĄŕãĂĆäÿŇéİćæŸŕäÿĂäÿłä¿Ňå■ŘïijŇ

```
from collections import OrderedDict

# A set of descriptors for various types
class Typed:
    _expected_type = type(None)
    def __init__(self, name=None):
        self._name = name

    def __set__(self, instance, value):
        if not isinstance(value, self._expected_type):
            raise TypeError('Expected ' + str(self._expected_type))
        instance.__dict__[self._name] = value

class Integer(Typed):
    _expected_type = int

class Float(Typed):
    _expected_type = float

class String(Typed):
    _expected_type = str

# Metaclass that uses an OrderedDict for class body
class OrderedMeta(type):
    def __new__(cls, clsname, bases, clsdict):
        d = dict(clsdict)
```

```
        order = []
        for name, value in clsdict.items():
            if isinstance(value, Typed):
                value._name = name
                order.append(name)
        d['_order'] = order
        return type.__new__(cls, clsname, bases, d)

    @classmethod
    def __prepare__(cls, clsname, bases):
        return OrderedDict()
```

åIJÍè£ŹäýłåĚČçśżäý■ïijŇæL’ğèąŇçśżäýżä¡ŞæŮűæŔŕè£řåŹÍçŽĎåőŽäźL’éąžåžŕäijŽèćńäýĂäýł
`OrderedDict``æ■ŢèŐůåĹřïijŇ çŤ$æĹŘçŽĎæIJL’åžŕåŘ■çğřäżŐå■ŮåĚÿäý■æŔŘåŕŰåĞžæİě
``_order äý■ãĂĆè£ŹæăůçŽĎèŕÍçśżäý■çŽĎæŰźæşŢåŕřäžěéĂŽè£Ğåď’Žçğ■æŰźåijŕæİěä¡£çŤĺåőČãĂĆ
ä¿ŇåęĆïijŇäýŇéÍćæŸřäýĂäýłçőĂå■ŢçŽĎçśżïijŇä¡£çŤÍè£ŹäýłæŐŠåžŕå■ŮåĚÿæİěåőđçŐřårĘäýĂäýłçśżåőđă

```
class Structure(metaclass=OrderedMeta):
    def as_csv(self):
        return ','.join(str(getattr(self,name)) for name in self._
 ↪order)

# Example use
class Stock(Structure):
    name = String()
    shares = Integer()
    price = Float()

    def __init__(self, name, shares, price):
        self.name = name
        self.shares = shares
        self.price = price
```

æĹŚäžňåIJÍäžd’äžŠåijŕçŐŕåćČäý■æţŃèŕŢäýĂäýŃè£ŹäýłStockçśżïijŽ

```
>>> s = Stock('GOOG',100,490.1)
>>> s.name
'GOOG'
>>> s.as_csv()
'GOOG,100,490.1'
>>> t = Stock('AAPL','a lot', 610.23)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "dupmethod.py", line 34, in __init__
TypeError: shares expects <class 'int'>
>>>
```

èőÍèőž

æIJňèŁĆäÿĂäÿłåĚşéŤőçĆźåřśæŸŕOrderedMetaåĚČçśżäÿ■åőŽäźL’çŽĎ ‘‘ __prepare__()‘‘ æŰźæşŢãĂĆ è£ŹäÿłæŰźæşŢäijŽåIJĺåijĂåğŃåőŽäźL’çśżåŠŇåőČçŽĎçĹűçśżçŽĎæŮűåĂŹèćńæL’ğèæĹŚäžňè£ŹéĞŇéĂŽè£Ğè£ŤåŻđäžĘäÿĂäÿłOrderedDictèĂŇäÿ■æŸŕäÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿïijŇåŔŕäžěå

åęĆæđIJä¡ăæČşæđĎéĂăèĞłåůśçŽĎçśżå■ŮåĚÿåŕźèśąïijŇåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎæL’l’åśŢè£ŹäÿłåŁ§èČ¡ă

```
from collections import OrderedDict

class NoDupOrderedDict(OrderedDict):
    def __init__(self, clsname):
        self.clsname = clsname
        super().__init__()
    def __setitem__(self, name, value):
        if name in self:
            raise TypeError('{} already defined in {}'.format(name,␣
↪self.clsname))
        super().__setitem__(name, value)

class OrderedMeta(type):
    def __new__(cls, clsname, bases, clsdict):
        d = dict(clsdict)
        d['_order'] = [name for name in clsdict if name[0] != '_']
        return type.__new__(cls, clsname, bases, d)

    @classmethod
    def __prepare__(cls, clsname, bases):
        return NoDupOrderedDict(clsname)
```

äÿŃéÍćæĹŚäžňæţŃèŕŢéĞ■åď■çŽĎåőŽäźL’äijŽåĞžçŐřäżĂäźĹæČĚåĘţïijŽ

```
>>> class A(metaclass=OrderedMeta):
...     def spam(self):
...         pass
...     def spam(self):
...         pass
...
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "<stdin>", line 4, in A
    File "dupmethod2.py", line 25, in __setitem__
        (name, self.clsname))
TypeError: spam already defined in A
>>>
```

æIJĂåŘŐè£ŸæIJL’äÿĂçĆźå¿ĹéĞ■èęĄïijŇåřśæŸŕåIJĺ __new__() æŰźæşŢäÿ■åŕźäžŐåĚČçśżäÿ■èćńä£őæŤźå■ŮåĚÿçŽĎåď ĎçŘĘãĂĆ åř¡çőąçśżä¡£çŤĺäžĘåŔęåď ŰäÿĂäÿłå■ŮåĚÿæİěåőŽäźL’ïijŇåIJĺæđĎéĂăæIJĂçźĹçŽĎ class åŕźèśąçŽĎæŮűåĂŹïijŇ æĹŚäžňäż■çĎűéIJĂèęĄåřĘè£Źäÿłå■ŮåĚÿè¡ňæ■ćäÿžäÿĂäÿłæ■čçąőçŽĎ dict åőđä¿ŃãĂĆ éĂŽè£Ğèŕ■åŘě d = dict(clsdict)

æÌěåőŇæĹŘè£ŹäÿłæŢĹæđIJãĂĆ

åŕźäžŐå¿Ĺåď˙ŽåžŤçŤĺçĺŃåžŔèĂŇåůšïijŇèČ¡åď˙§æ■ŢèŐůçśżåőŽäźL˙çŽĎéąžåžŔæŸŕäÿĂäÿłçIJŃäijijäÿ■ä¿ŃåęĆïijŇåIJĺåŕźèśąåĚşçşżæŸăåřĎäÿ■ïijŇæĹŚäžňéĂŽåÿÿäijŽçIJŃåĹřäÿŃéÌćè£Źçğ■æŰźåijŔåőŽäźL˙çŽĎç

```
class Stock(Model):
    name = String()
    shares = Integer()
    price = Float()
```

åIJÍæąĘæđűåžŢåśĆïijŇæĹŚäžňå£Ěéążæ■ŢèŐůåőŽäźL˙çŽĎéąžåžŔæÌěåřĘåŕźèśąæŸăåřĎåĹřåĚČçżĎæĹÚ as_csv() çŽĎåŁ§èČ¡ïijL˙ãĂĆ è£ŹèŁĆæijŤçd˙žçŽĎæŁĂæIJŕéÌđåÿÿçőĂå■ŢïijŇåźűäÿŤéĂŽåÿÿäijŽæŕŤåĚű

11.15 9.15 åőŽäźL˙æIJL˙åŔŕéĂL˙åŔĆæŢřçŽĎåĚČçśż

éŮőécŸ

ä¡ăæČşåőŽäźL˙äÿĂäÿłåĚČçśżïijŇåĚĄèőÿçśżåőŽäźL˙æŮűæŔŘä¿ŻåŔŕéĂL˙åŔĆæŢřïijŇè£ŹæăůåŔŕäžěæĆ

èğčåĘşæŰźæąĹ

åIJÍåőŽäźL˙çśżçŽĎæŮűåĂŹïijŇPythonåĚĄèőÿæĹŚäžňä¡£çŤÍ “metaclass‘‘åĚşéŤőå■ŮåŔĆæŢřæÌěæŇĞåőŽçL˙źåőŽçŽĎåĚČçśżãĂĆ ä¿ŃåęĆä¡£çŤÍæŁ¡èśąå§žçśżïijŽ

```
from abc import ABCMeta, abstractmethod
class IStream(metaclass=ABCMeta):
    @abstractmethod
    def read(self, maxsize=None):
        pass

    @abstractmethod
    def write(self, data):
        pass
```

çĎűèĂŇïijŇåIJÍèĞłåőŽäźL˙åĚČçśżäÿ■æĹŚäžňè£ŸåŕźäžěæŔŘä¿ŻåĚűäžŰçŽĎåĚşéŤőå■ŮåŔĆæŢřïijŇå

```
class Spam(metaclass=MyMeta, debug=True, synchronize=True):
    pass
```

äÿžäžĘä¡£åĚČçśżæŤŕæŇĄè£ŹäžŻåĚşéŤőå■ŮåŔĆæŢřïijŇä¡ăå£Ěéążçąőä£ÌåIJÍ __prepare__() , __new__() åŠŇ __init__() æŰźæşŢäÿ■éČ¡ä¡£çŤÍåijžåĹűåĚşéŤőå■ŮåŔĆæŢřãĂĆåřśåČŔäÿŃéÌćè£ŹæăůïijŽ

```
class MyMeta(type):
    # Optional
    @classmethod
    def __prepare__(cls, name, bases, *, debug=False,␣
↪synchronize=False):
```

```
        # Custom processing
        pass
        return super().__prepare__(name, bases)

    # Required
    def __new__(cls, name, bases, ns, *, debug=False,␣
↪synchronize=False):
        # Custom processing
        pass
        return super().__new__(cls, name, bases, ns)

    # Required
    def __init__(self, name, bases, ns, *, debug=False,␣
↪synchronize=False):
        # Custom processing
        pass
        super().__init__(name, bases, ns)
```

èőÍèőž

çžŹäÿĂäÿłåĚČçśżæůżåŁääŔŕéĂĽåĚşéŤőå■ŮåŔĆæŢřéIJĂèęĄä¡ăåőŇåĚĺåijĎæĞĆçśżåĹŻåžžçŽĎæĽĂa
åŻăäÿžè£ŹäžŻåŔĆæŢřäijŽèćńäijăéĂŠçżŹæŕŔäÿĂäÿłçŻÿåĚşçŽĎæŰźæşŢãĂĆ
__prepare__() æŰźæşŢåIJĺæĽĂæIJĽçśżåőŽäźĽåijĂåğŇæĽğèąŇåĽ■éęŰåĚĹèćńèřČçŤĺïijŇçŤĺæİěåĹŻ
éĂŽåÿÿæİěëőšïijŇè£ŹäÿłæŰźæşŢåŔłæŸŕçőĂå■ŢçŽĎè£ŤåŻđäÿĂäÿłå■ŮåĚÿæĹŰåĚűäžŰæŸăåřĎåŕźèśąãĂĆ
__new__() æŰźæşŢèćńçŤĺæİěåőđä¿ŃåŇŰæIJĂçżĹçŽĎçśżåŕźèśąãĂĆåőČåIJĺçśżçŽĎäÿżä¡ŞèćńæĽğèąŇåő
__init__() æŰźæşŢæIJĂåŘŐèćńèřČçŤĺïijŇçŤĺæİěæĽğèąŇåĚűäžŰçŽĎäÿĂäžŻåĹİåğŃåŇŰåůěä¡IJãĂĆ

å¡ŞæĹŚäžňædĎéĂäåĚČçśżçŽĎæŮŭåĂŹïijŇéĂŽåÿÿåŔłéIJĂèęĄåőŽäźĽäÿĂäÿł
__new__() æĹŰ __init__() æŰźæşŢïijŇä¡Ęäÿ■æŸŕäÿď äÿłéČ¡åőŽäźĽãĂĆ
ä¡ĘæŸŕïijŇåęĆædIJéIJĂèęĄæŐěåŔŮåĚűäžŰçŽĎåĚşéŤőå■ŮåŔĆæŢřçŽĎèŕİïijŇè£ŹäÿďäÿłæŰźæşŢåŕśèęĄå
éżŸèőď çŽĎ __prepare__() æŰźæşŢæŐěåŔŮäžżæĎŔçŽĎåĚşéŤőå■ŮåŔĆæŢřïijŇä¡ĘæŸŕäijŽå£¡çŢěåő
æĽĂäżěåŔłæIJĽå¡Şè£ŹäžŻécİåď ŰçŽĎåŔĆæŢřåŔŕèČ¡äijŽå¡śåŞ■åĹřçśżåŠ¡åŘ■çľžéÚť çŽĎåĹŻåžżæŮűä¡ăa
__prepare__() æŰźæşŢãĂĆ

éĂŽè£Ğä¡£çŤĺåijžåĹűåĚşéŤőå■ŮåŔĆæŢřïijŇåIJĺçśżçŽĎåĹŻåžžè£ĞçÍÑäÿ■æĹŚäžňå£ĚéążéĂŽè£ĞåĚş

ä¡£çŤĺåĚşéŤőå■ŮåŔĆæŢřéĚ■ç¡őäÿĂäÿłåĚČçśżè£ŸåŕŕäžěèğĘä¡IJåŕźçśżåŔŸéĞŔçŽĎäÿĂçğ■æŻ£äžčæU

```
class Spam(metaclass=MyMeta):
    debug = True
    synchronize = True
    pass
```

åřĘè£ŹäžŻåśđæĂğåőŽäźĽäÿžåŔĆæŢřçŽĎåě¡åď ĎåIJÍäžŐåőČäžňäÿ■äijŽæśąæ§ŞçśżçŽĎåŘ■çğřçľžéŮť
è£ŹäžŻåśđæĂğäžĚäžĚåŔłäžŐåśđäžŐçśżçŽĎåĹŻåžžéŸűæőţïijŇèĂŇäÿ■æŸŕçśżäÿ■çŽĎèŕ■åŘèæĽğèąŇéŸűa
åŔęåď ŰïijŇåőČäžňåIJÍ __prepare__() æŰźæşŢäÿ■æŸŕåŕŕäžěèćńèő£éŰőçŽĎïijŇåŻăäÿžè£ŹäÿłæŰźæşŢ
ä¡ĘæŸŕçśżåŔŸéĞŔåŔłèČ¡åIJÍåĚČçśżçŽĎ __new__() åŠŇ __init__()
æŰźæşŢäÿ■åŔŕèğĄãĂĆ

11.16 9.16 *argsåŠŇ**kwargsçŽĎåijžåĹűåŔĆæŢřç■¿åŘ■

éŮőéćŸ

ä¡ăæIJĽäÿĂäÿłåĞ¡æŢřæĹŰæŰźæşŢïijŇåőČä¡£çŤĺ*argsåŠŇ**kwargsä¡IJäÿžåŔĆæŢřïijŇè£Źæăůä¡£å¿Ů
ä¡ĘæIJĽæŮűåĂŹä¡ăæČşæčĂæ§ěäijăéĂŠè£ŻæİěçŽĎåŔĆæŢřæŸŕäÿ■æŸŕæ§Řäÿłä¡ăæČşèęĄçŽĎçśżåđŃãĂĆ

èğčåĘşæŰźæąĹ

åŕźäżżä¡ŢæűĽåŔŁåĹřæŞ■ä¡IJåĞ¡æŢřèřČçŤĺç■¿åŘ■çŽĎéŮőéćŸïijŇä¡ăéČ¡åžŤèŕěä¡£çŤĺ inspect æĺąåİŮäÿ■çŽĎç■¿åŘ■çĽźæĂğãĂĆ æĹŚäżňæIJĂäÿżèęĄåĚşæşĺäÿď äÿłçśżïijŽSignature åŠŇ Parameter ãĂĆäÿŃéİćæŸŕäÿĂäÿłåĹŻåżžåĞ¡æŢřåĽ■éİćçŽĎäžď äžŠä¿Ńå■ŘïijŽ

```
>>> from inspect import Signature, Parameter
>>> # Make a signature for a func(x, y=42, *, z=None)
>>> parms = [ Parameter('x', Parameter.POSITIONAL_OR_KEYWORD),
...          Parameter('y', Parameter.POSITIONAL_OR_KEYWORD,␣
↪default=42),
...          Parameter('z', Parameter.KEYWORD_ONLY, default=None) ]
>>> sig = Signature(parms)
>>> print(sig)
(x, y=42, *, z=None)
>>>
```

äÿĂæŮęä¡ăæIJĽäžĘäÿĂäÿłç■¿åŘ■åŕźèśąïijŇä¡ăåřśåŔŕäżěä¡£çŤĺåőČçŽĎ bind() æŰźæşŢå¿ĹåőźæŸŞçŽĎåřĘåőČçżŚåőŽåĹř *args åŠŇ **kwargs äÿŁåŐżãĂĆ äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎæijŤçď žïijŽ

```
>>> def func(*args, **kwargs):
...     bound_values = sig.bind(*args, **kwargs)
...     for name, value in bound_values.arguments.items():
...         print(name,value)
...
>>> # Try various examples
>>> func(1, 2, z=3)
x 1
y 2
z 3
>>> func(1)
x 1
>>> func(1, z=3)
x 1
z 3
>>> func(y=2, x=1)
x 1
y 2
>>> func(1, 2, 3, 4)
Traceback (most recent call last):
...
```

```
    File "/usr/local/lib/python3.3/inspect.py", line 1972, in _bind
        raise TypeError('too many positional arguments')
TypeError: too many positional arguments
>>> func(y=2)
Traceback (most recent call last):
...
    File "/usr/local/lib/python3.3/inspect.py", line 1961, in _bind
        raise TypeError(msg) from None
TypeError: 'x' parameter lacking default value
>>> func(1, y=2, x=3)
Traceback (most recent call last):
...
    File "/usr/local/lib/python3.3/inspect.py", line 1985, in _bind
        '{arg!r}'.format(arg=param.name))
TypeError: multiple values for argument 'x'
>>>
```

åŘŕäžěçIJŃåĞžæİěïijŇéĂŽè£ĞářEç■¿åŘ■åŠŇäijăéĂŠçŽĎåŔĆæŢřçżŚåőŽèţůæİěïijŇåŔŕäžěåijžåĹűåĞ¡a

äÿŃéİćæŸŕäÿĂäÿłåijžåĹűåĞ¡æŢřç■¿åŘ■æŻťåĚůä¡ŞçŽĎä¿Ńå■ŘãĂĆåIJĺäžčçăĄäÿ■ïijŇæĹŚäžňåIJĺå§žç
`__init__()` æŰźæşŢïijŇ çĎűåŘŐæĹŚäžňåijžåĹűæĽĂæIJĽçŽĎå■Řçśżå£Ěéążæŕŕä¿ŻäÿĂäÿłçĽźåőŽçŽĹ

```
from inspect import Signature, Parameter

def make_sig(*names):
    parms = [Parameter(name, Parameter.POSITIONAL_OR_KEYWORD)
             for name in names]
    return Signature(parms)

class Structure:
    __signature__ = make_sig()
    def __init__(self, *args, **kwargs):
        bound_values = self.__signature__.bind(*args, **kwargs)
        for name, value in bound_values.arguments.items():
            setattr(self, name, value)

# Example use
class Stock(Structure):
    __signature__ = make_sig('name', 'shares', 'price')

class Point(Structure):
    __signature__ = make_sig('x', 'y')
```

äÿŃéİćæŸŕä¡£çŤĺè£Źäÿł `Stock` çśżçŽĎçď žä¿ŃïijŽ

```
>>> import inspect
>>> print(inspect.signature(Stock))
(name, shares, price)
>>> s1 = Stock('ACME', 100, 490.1)
>>> s2 = Stock('ACME', 100)
Traceback (most recent call last):
```

```
...
TypeError: 'price' parameter lacking default value
>>> s3 = Stock('ACME', 100, 490.1, shares=50)
Traceback (most recent call last):
...
TypeError: multiple values for argument 'shares'
>>>
```

èőÍèőž

åIJÍæĹŚäžňéIJĂèęĄæđĎåžžéĂŽçŤÍåĞ¡æŢřåžŞãĂĄçijŰåĘŹèčĚéěřåŹÍæĹŰåőđçŐřäžčçŘĘçŽĎæŮűåĂŹ
*args åŠŇ **kwargs çŽĎä¡£çŤÍæŸŕå¿ĹæŹőéĄ■çŽĎãĂĆ
ä¡ĘæŸŕïijŇè£ŹæăůçŽĎåĞ¡æŢřæIJL'äÿĂäÿłçijžçĆźåřśæŸŕå¡Şä¡ăæČşèęĄåőđçŐřèĞłåůśçŽĎåŔĆæŢřæčĂéłŇ
è£ŹæŮűåĂŹæĹŚäžňåŔŕäžěéĂŽè£ĞäÿĂäÿłç■¿åŘ■åŕźèśąæİěçőĂåŇŰåőČãĂĆ

åIJÍæIJĂåŘŐçŽĎäÿĂäÿłæŰźæąĹåőđä¿Ńäÿ■ïijŇæĹŚäžňè£ŸåŔŕäžěéĂŽè£Ğä¡£çŤÍèĞłåőŽäźL'åĚČçśżæİ

```
from inspect import Signature, Parameter

def make_sig(*names):
    parms = [Parameter(name, Parameter.POSITIONAL_OR_KEYWORD)
             for name in names]
    return Signature(parms)

class StructureMeta(type):
    def __new__(cls, clsname, bases, clsdict):
        clsdict['__signature__'] = make_sig(*clsdict.get('_fields',
↪[]))
        return super().__new__(cls, clsname, bases, clsdict)

class Structure(metaclass=StructureMeta):
    _fields = []
    def __init__(self, *args, **kwargs):
        bound_values = self.__signature__.bind(*args, **kwargs)
        for name, value in bound_values.arguments.items():
            setattr(self, name, value)

# Example
class Stock(Structure):
    _fields = ['name', 'shares', 'price']

class Point(Structure):
    _fields = ['x', 'y']
```

å¡ŞæĹŚäžňèĞłåőŽäźL'ç■¿åŘ■çŽĎæŮűåĂŹïijŇårĘç■¿åŘ■å■ŸåĆĺåIJÍçL'źåőŽçŽĎåšđæĂğ
__signature__ äÿ■éĂŽåÿÿæŸŕå¿ĹæIJL'çŤÍçŽĎãĂĆ è£ŹæăůçŽĎèŕÍïijŇåIJÍä¡£çŤÍ
inspect æÍąåİŮæL'ğèąŇåĘĚçIJAçŽĎäžčçăĄåŕśèČ¡åŔŚçŐřç■¿åŘ■åźűåřĘåőČä¡IJäÿžèřČçŤÍçžęåőŽãĂĆ

```
>>> import inspect
>>> print(inspect.signature(Stock))
(name, shares, price)
>>> print(inspect.signature(Point))
(x, y)
>>>
```

11.17 9.17 åIJÍçśżäÿŁåijżåĹűä¡£çŤÍçijŰçÍŃèğĎçžę

éŮőécŸ

ä¡ăçŽĎçÍŃåžŘåŇĚåŘńäÿĂäÿłå¿Ĺåď ğçŽĎçśżçżğæL'£ä¡ŞçşżïijŇä¡ăåÿŇæIJŻåijżåĹűæL'ğèąŇæ§ŘäžŻçij

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăæČşçŻŚæŐğçśżçŽĎåőŽäźL'ïijŇéĂŽåÿÿåŔräżěéĂŽè£ĞåőŽäźL'äÿĂäÿłåĚČçśżãĂĆäÿĂäÿłå§ type åźűéĞ■åőŽäźL'åőČçŽĎ __new__() æŰźæşŢ æĹŰèĂĚæŸŕ __init__() æŰźæşŢãĂĆæŕŤåęĆïijŽ

```
class MyMeta(type):
    def __new__(self, clsname, bases, clsdict):
        # clsname is name of class being defined
        # bases is tuple of base classes
        # clsdict is class dictionary
        return super().__new__(cls, clsname, bases, clsdict)
```

åŘęäÿĂçğ■æŸŕïijŇåőŽäźL' __init__() æŰźæşŢïijŽ

```
class MyMeta(type):
    def __init__(self, clsname, bases, clsdict):
        super().__init__(clsname, bases, clsdict)
        # clsname is name of class being defined
        # bases is tuple of base classes
        # clsdict is class dictionary
```

äÿžäžĘä¡£çŤÍè£ŹäÿłåĚČçśżïijŇä¡ăéĂŽåÿÿèęĄåřĘåőČæŤ¿åĹřåĹřäÿĂäÿłéąűçžğçĹűçśżåőŽäźL'äÿ■ïijŇçĹ

```
class Root(metaclass=MyMeta):
    pass

class A(Root):
    pass

class B(Root):
    pass
```

åĚČçśžçŽĎäÿĂäÿłåĚşéŤőçĽźçĆźæŸŕåőČåĚĄèőÿä¡ååIJĺåőŽäźĽçŽĎæŮűåĂŹæčĂæ§ěçśžçŽĎåĘĚåőźã
__init__() æŰźæşŢäÿ■ïijŇ ä¡ăåŔŕäžěå¿Ĺè¡żæİ¿çŽĎæčĂæ§ěçśžå■ŮåĚÿãĂĄçĹűçśžç■Ľç■Ľ˜ãĂĆåźűäÿŤi
åŻăæ■ď˜ïijŇäÿĂäÿłæąĘæđűçŽĎæđĎåžžèĂĚåřśèČ¡åIJĺåď˜gåđŃçŽĎçżğæĽ£ä¡Şçşžäÿ■éĂŽè£ĞçźŹäÿĂäÿłéąű

ä¡IJäÿžäÿĂäÿłåĚůä¡ŞçŽĎåžŤçŤĺä¿Ńå■ŘïijŇäÿŃéİćåőŽäźĽäžĘäÿĂäÿłåĚČçśžïijŇåőČäijŽæŃŠçżİäżżä¡

```
class NoMixedCaseMeta(type):
    def __new__(cls, clsname, bases, clsdict):
        for name in clsdict:
            if name.lower() != name:
                raise TypeError('Bad attribute name: ' + name)
        return super().__new__(cls, clsname, bases, clsdict)

class Root(metaclass=NoMixedCaseMeta):
    pass

class A(Root):
    def foo_bar(self): # Ok
        pass

class B(Root):
    def fooBar(self): # TypeError
        pass
```

ä¡IJäÿžæŻť énŸçžğåŠŇåőđçŤĺçŽĎä¿Ńå■ŘïijŇäÿŃéİćæIJĽäÿĂäÿłåĚČçśžïijŇåőČçŤĺæİěæčĂæţŃéĞ■è¡¡

```
from inspect import signature
import logging

class MatchSignaturesMeta(type):

    def __init__(self, clsname, bases, clsdict):
        super().__init__(clsname, bases, clsdict)
        sup = super(self, self)
        for name, value in clsdict.items():
            if name.startswith('_') or not callable(value):
                continue
            # Get the previous definition (if any) and compare the␣
↪signatures
            prev_dfn = getattr(sup,name,None)
            if prev_dfn:
                prev_sig = signature(prev_dfn)
                val_sig = signature(value)
                if prev_sig != val_sig:
                    logging.warning('Signature mismatch in %s. %s !
↪= %s',
                                    value.__qualname__, prev_sig,␣
↪val_sig)

# Example
class Root(metaclass=MatchSignaturesMeta):
    pass
```

```
class A(Root):
    def foo(self, x, y):
        pass

    def spam(self, x, *, z):
        pass

# Class with redefined methods, but slightly different signatures
class B(A):
    def foo(self, a, b):
        pass

    def spam(self,x,z):
        pass
```

åęĆæđIJä¡ăè£ŘèąŇè£ŹæőţäżčçăĄïijŇåřśäijŽå¿ŮåĹřäÿŃéİćè£ŹæăůçŽĎè¿ŞåĞžçżŞæđIJïijŽ

```
WARNING:root:Signature mismatch in B.spam. (self, x, *, z) != (self,
↪ x, z)
WARNING:root:Signature mismatch in B.foo. (self, x, y) != (self, a,␣
↪b)
```

è£Źçğ■è■ęåŚŁä£ąæĄŕåŕźäžŐæ■ŢèŐůäÿĂäžŻå¿őåęŹçŽĎçĺŃåžŔbugæŸŕå¿ĹæIJĽçŤĺçŽĎãĂĆä¿ŃåęĆïij
éĆčäźĹå¡Şå■ŘçśżæŤźåŔŸåŔĆæŢřåŘ■å■ŮçŽĎæŮűåĂŹåřśäijŽèřČçŤĺåĞžéŤŹãĂĆ

èőĺèőž

åIJĺåď ğåđŃéİćåŘŚåŕźèśąçŽĎçĺŃåžŔäÿ■ïijŇéĂŽåÿÿåřĘçśżçŽĎåőŽäźĽæŤ¿åIJĺåĚČçśżäÿ■æŐğåĹűæŸŕå
åĚČçśżåŔŕäżěçŻŚæŐğçśżçŽĎåőŽäźĽïijŇè■ęåŚŁçijŰçĺŃäžžåŚŸæ§ŘäžŻæšąæIJĽæşĺæĎŔåĹřçŽĎåŔŕèČ¡åĞ

æIJĽäžžåŔŕèČ¡äijŽèŕťïijŇåČŔè£ŹæăůçŽĎéŤŹèŕŕåŔŕäżěéĂŽè£ĞçĺŃåžŔåĹĘæđŘåůěåĚůæĹŰIDEåŐżå
ä¡ĘæŸŕïijŇåęĆæđIJä¡ăåIJĺæđĎåżžäÿĂäÿłæąĘæđűæĹŰåĞ¡æŢřåžŞä¿ŻåĚűäżŰäžžä¡£çŤĺïijŇéĆčäźĹä¡ăæšąåŁ
åŻăæ■ď ïijŇåŕźäžŐè£Źçğ■çśżåđŃçŽĎçĺŃåžŔïijŇåęĆæđIJåŔŕäżěåIJĺåĚČçśżäÿ■åĄŽæčĂæţŃæĹŰèőÿåŔŕäżě

åIJĺåĚČçśżäÿ■éĂĽæŃľéĞ■æŰřåőŽäźĽ `__new__()` æŰźæşŢè£ŸæŸŕ `__init__()` æŰźæşŢåŔŰåĘşäžŐä¡ăæČşæĂŐæăůä¡£çŤĺçżŞæđIJçśżãĂĆ `__new__()` æŰźæşŢåIJĺçśżåĹŻåżžäźŃåĽ■èćńèřČçŤĺïijŇéĂŽåÿÿçŤĺäžŐéĂŽè£Ğæ§Řçğ■æŰźåijŔïijĹæŕŤåęĆéĂŽè£ĞæŤ
èĂŇ `__init__()` æŰźæşŢæŸŕåIJĺçśżèćńåĹŻåżžäźŃåŘŐèćńèřČçŤĺïijŇå¡Şä¡ăéIJĂèęĄåőŇæŢťæđĎåżžçśż
åIJĺæIJĂåŘŐäÿĂäÿłä¿Ńå■Řäÿ■ïijŇè£ŹæŸŕå£ĚèęĄçŽĎïijŇåŻăäÿžåőČä¡£çŤĺäžĘ `super()`
åĞ¡æŢřæİěæŘIJçť ćäźŃåĽ■çŽĎåőŽäźĽãĂĆ åőČåŔłèČ¡åIJĺçśżçŽĎåőđä¿ŃèćńåĹŻåżžäźŃåŘŐïijŇåźűäÿŤçŻ

æIJĂåŘŐäÿĂäÿłä¿Ńå■Řè£ŸæijŤçď žäžĘPythonçŽĎåĞ¡æŢřç■¿åŘ■åŕźèśąçŽĎä¡£çŤĺãĂĆ
åőđéŹĚäÿŁïijŇåĚČçśżåřĘæŕŔäÿłåŔŕèřČçŤĺåőŽäźĽæŤ¿åIJĺäÿĂäÿłçśżäÿ■ïijŇæŘIJçť ćåĽ■äÿĂäÿłåőŽäźĽïijĹ
çĎűåŘŐéĂŽè£Ğä¡£çŤĺ `inspect.signature()` æİěçőĂå■ŢçŽĎæŕŤè¿ČåőČäżňçŽĎèřČçŤĺç■¿åŘ■ãĂĆ

æIJĂåŘŐäÿĂçĆźïijŇäżčçăĄäÿ■æIJĽäÿĂèąŇä¡£çŤĺäžĘ `super(self, self)`
åźűäÿ■æŸŕæŐŠçĽĹéŤŹèŕŕãĂĆ å¡Şä¡£çŤĺåĚČçśżçŽĎæŮűåĂŹïijŇæĹŚäżňèęĄæŮűåĹżèőřä¡ŔäÿĂçĆźåřśæŸ
`self` åőđéŹĚäÿŁæŸŕäÿĂäÿłçśżåŕźèśąãĂĆ åŻăæ■ď ïijŇè£Źæİąèŕ■åŔěåĚűåőđåřśæŸŕçŤĺæİěåŕżæĽ¿ä¡■äžŐçż
`self` çĹűçśżçŽĎåőŽäźĽãĂĆ

11.18 9.18 äžěçijŰçÍNæŰźåijŔåőŽäźĽ'çśż

éŮőécŸ

ä¡ăåIJÍåĘŹäÿĂæőţäžčçăĄïijŇæIJĂçżĹéIJĂèęĄåĹŻåžžäÿĂäÿłæŰřçŽĎçśżåŕźèśąăĂĆä¡äèĂČèŹŚåřĘçśżçŽ
åźűäÿŤä¡£çŤÍåĞ¡æŢřæŕŤåęĆ `exec()` æİěæĽ'ğèąŇåőČïijŇä¡ĘæŸŕä¡äæČşåŕżæĽ'¿äÿĂäÿłæŻť'åŁääijŸéŻĚçŽ

èğčåĘşæŰźæąĹ

ä¡ăåŔŕäžěä¡£çŤÍåĞ¡æŢř `types.new_class()` æİěåĹİåğŃåŇŰæŰřçŽĎçśżåŕźèśąăĂĆ
ä¡ăéIJĂèęĄåĄŽçŽĎåŔłæŸŕæŔŘä¿ŻçśżçŽĎåŘ■å■ŮãĂĄçĹűçśżåĚČçżĎãĂĄåĚşéŤőå■ŮåŔĆæŢřïijŇäžěåŔŁ

```
# stock.py
# Example of making a class manually from parts

# Methods
def __init__(self, name, shares, price):
    self.name = name
    self.shares = shares
    self.price = price
def cost(self):
    return self.shares * self.price

cls_dict = {
    '__init__' : __init__,
    'cost' : cost,
}

# Make a class
import types

Stock = types.new_class('Stock', (), {}, lambda ns: ns.update(cls_
↪dict))
Stock.__module__ = __name__
```

è£Źçğ■æŰźåijŔäijŽæđĎåžžäÿĂäÿłæŹőéĂŽçŽĎçśżåŕźèśąïijŇåźűäÿŤæŇĽçĚğä¡äçŽĎæIJ§æIJŻåůěä¡IJïi

```
>>> s = Stock('ACME', 50, 91.1)
>>> s
<stock.Stock object at 0x1006a9b10>
>>> s.cost()
4555.0
>>>
```

è£Źçğ■æŰźæşŢäÿ■ïijŇäÿĂäÿłæŕŤè¿ČéŽ¿çŘĘèğčçŽĎåIJřæŰźæŸŕåIJÍèřČçŤÍåőŇ
`types.new_class()` åŕź `Stock.__module__` çŽĎèţŇåĂijãĂĆ
æŕŔæňąå¡ŞäÿĂäÿłçśżèćńåőŽäźĽåŘŐïijŇåőČçŽĎ `__module__`
åśđæĂğåŇĚåŘńåőŽäźĽåőČçŽĎæÍąåİŮåŘ■ãĂĆ è£ŹäÿłåŘ■å■ŮçŤÍäžŐçŤ§æĹŘ

__repr__() æŰźæşŢçŽĎè¿ŞåĞžãĂĆåőČåŘŇæăůäź§èćńçŤĺäžŐå¿Ĺåď'ŽåžŞïijŇæŕŤåęĆ pickle ãĂĆ åŻäæ■ď'ïijŇäÿžäžĘèőľä¡ăåĹŻåżžçŽĎçśżæŸŕâĂIJæ■čçąőâĂİçŽĎïijŇä¡ăéIJĂèęĄçąőä£Ìè£ŹäÿŁ

åęĆæđIJä¡ăæČşåĹŻåżžçŽĎçśżéIJĂèęĄäÿĂäÿłäÿ■åŘŇçŽĎåĚČçśżïijŇåŕŕäžěéĂŽè£Ğ types.new_class() çňňäÿL'äÿłåŔĆæŢřäijăéĂŠçżŹåőČãĂĆä¿ŃåęĆïijŽ

```
>>> import abc
>>> Stock = types.new_class('Stock', (), {'metaclass': abc.ABCMeta},
...                         lambda ns: ns.update(cls_dict))
...
>>> Stock.__module__ = __name__
>>> Stock
<class '__main__.Stock'>
>>> type(Stock)
<class 'abc.ABCMeta'>
>>>
```

çňňäÿL'äÿłåŔĆæŢřè£ŸåŕŕäžěåŇĚåŘńåĚűäžŰçŽĎåĚşéŤőå■ŮåŔĆæŢřãĂĆæŕŤåęĆïijŇäÿĂäÿłçśżçŽĎåő

```
class Spam(Base, debug=True, typecheck=False):
    pass
```

éĆčäźĹåŕŕäžěåřĘåĚűç£żèŕŚæĹŘåęĆäÿŇçŽĎ new_class() èřĈçŤĺå¡ćåijŔïijŽ

```
Spam = types.new_class('Spam', (Base,),
                       {'debug': True, 'typecheck': False},
                       lambda ns: ns.update(cls_dict))
```

new_class() çňňåŻŽäÿłåŔĆæŢřæIJĂçěđçğŸïijŇåőČæŸŕäÿĂäÿłçŤĺæİěæŐěåŔŮçśżåŚ¡åŘ■çľ'žéŮť'çŽ éĂŽåÿÿè£ŹæŸŕäÿĂäÿłæŹőéĂŽçŽĎå■ŮåĚÿïijŇä¡ĘæŸŕåőČåőđéŹĚäÿŁæŸŕ __prepare__() æŰźæşŢè£ŤåŽđçŽĎäżżæĎŘåŕźèśąïijŇè£ŹäÿłåIJĺ9.14åřŔèŁĆåůšçżŔäžŃçż■è£ĞäžĘãĂĆ è£ŹäÿłåĞ¡æŢřéIJĂèęĄä¡£çŤĺäÿŁéİćæijŤçd'žçŽĎ update() æŰźæşŢçżŹåŚ¡åŘ■çľ'žéŮť'åćđåŁååĘĚåőźãĂĆ

èőĺèőž

å¿Ĺåď'ŽæŮűåĂŹåęĆæđIJèČ¡æđĎéĂäæŰřçŽĎçśżåŕźèśąæŸŕå¿ĹæIJL'çŤĺçŽĎãĂĆ æIJL'äÿłå¿ĹçĘ§æĆL'çŽĎä¿Ńå■ŘæŸŕèřČçŤĺ collections.namedtuple() åĞ¡æŢřïijŇä¿ŃåęĆïijŽ

```
>>> Stock = collections.namedtuple('Stock', ['name', 'shares',
↪'price'])
>>> Stock
<class '__main__.Stock'>
>>>
```

namedtuple() ä¡£çŤĺ exec() èĂŇäÿ■æŸŕäÿŁéİćäžŃçż■çŽĎæŁĂæIJŕãĂĆä¡ĘæŸŕïijŇäÿŃéİćéĂŽè æĹŚäžňçŽť'æŐěåĹŻåżžäÿĂäÿłçśżïijŽ

```
import operator
import types
import sys

def named_tuple(classname, fieldnames):
    # Populate a dictionary of field property accessors
    cls_dict = { name: property(operator.itemgetter(n))
                 for n, name in enumerate(fieldnames) }

    # Make a __new__ function and add to the class dict
    def __new__(cls, *args):
        if len(args) != len(fieldnames):
            raise TypeError('Expected {} arguments'.
↪format(len(fieldnames)))
        return tuple.__new__(cls, args)

    cls_dict['__new__'] = __new__

    # Make the class
    cls = types.new_class(classname, (tuple,), {},
                          lambda ns: ns.update(cls_dict))

    # Set the module to that of the caller
    cls.__module__ = sys._getframe(1).f_globals['__name__']
    return cls
```

è£ŹæőţäżčçăĄçŽĎæIJĂåŘŐéČĺåĹĘä¡£çŤĺäžĘäÿĂäÿłæĽĂèřŞçŽĎâĂİæąĘæđűé■ŤæşŢâĂİïijŇéĂŽè£Ğè
sys._getframe() æİěèŐůåŔŰèřČçŤĺèĂĚçŽĎæĺąåİŮåŘ■ãĂĆ
åŔęåď ŰäÿĂäÿłæąĘæđűé■ŤæşŢä¿Ńå■ŘåIJĺ2.15åřŔèŁĆäÿ■æIJĽäżŃçż■è£ĞãĂĆ

äÿŃéİćçŽĎä¿Ńå■ŘæijŤçď žäžĘåĽ■éİćçŽĎäżčçăĄæŸŕåęĆä¡Ţåůěä¡IJçŽĎïijŽ

```
>>> Point = named_tuple('Point', ['x', 'y'])
>>> Point
<class '__main__.Point'>
>>> p = Point(4, 5)
>>> len(p)
2
>>> p.x
4
>>> p.y
5
>>> p.x = 2
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: can't set attribute
>>> print('%s %s' % p)
4 5
>>>
```

è£ŹéąźæŁĂæIJŕäÿĂäÿłå¿ĹéĞ■èęĄçŽĎæŰźéİćæŸŕåőČåŕźäžŐåĚČçśżçŽĎæ■čçąőä¡£çŤĺãĂĆ

ä¡ăåŔŕèČ¡åČŔéĂŽè£ĞçŻť æŐěåőđä¿ŃåŇŰäÿĂäÿłåĚČçśżæİěçŻť æŐěåĹŻåżžäÿĂäÿłçśżïijŽ

```
Stock = type('Stock', (), cls_dict)
```

è£Źçğ■æŰźæşŢçŽĎéŮőéćŸåIJĺäžŐåőČå£¡çŢěäžĘäÿĂäžŻåĚşéŤőæ■ěéłď ïijŇæŕŤåęĆåŕźäžŐåĚČçśżäÿ■ __prepare__() æŰźæşŢçŽĎèřČçŤĺãĂĆ éĂŽè£Ğä¡£çŤĺ types.new_class() ïijŇä¡ăåŔŕäżěä£İèŕĄæĽĂæIJĽçŽĎå£ĚèęĄåĹİåğŃåŇŰæ■ěéłď éČ¡èČ¡å¿ŮåĹřæĽğèąŇãĂĆ æŕŤåęĆïijŇtypes.new_class() çňňåŻŻäÿłåŔĆæŢřçŽĎåŻđèřČåĞ¡æŢřæŐěåŔŮ __prepare__() æŰźæşŢè£ŤåŻđçŽĎæŸăåřĎåŕźèśąãĂĆ

åęĆæđIJä¡ăäżĚäżĚåŔłæŸŕæČşæĽğèąŇåĞĘåď Ğæ■ěéłď ïijŇåŔŕäżěä¡£çŤĺ types.prepare_class() ãĂĆä¿ŃåęĆïijŽ

```
import types
metaclass, kwargs, ns = types.prepare_class('Stock', (), {'metaclass
↪': type})
```

åőČäijŽæ§ěæĽ¿åŘĹéĂĆçŽĎåĚČçśżåźűèřČçŤĺåőČçŽĎ __prepare__() æŰźæşŢãĂĆ çĎűåŘŐè£ŹäÿłåĚČçśżä£İå■ŸåőČçŽĎåĚşéŤőå■ŮåŔĆæŢřïijŇåĞĘåď ĞåŚ¡åŘ■çľžéŮť åŘŐèćńè

æŻť åď Žä£ąæĄŕ, èŕůåŔĆèĂČ PEP 3115 , äżěåŔŁ Python documentation .

11.19 9.19 åIJĺåőŽäźĽçŽĎæŮűåĂŹåĹİåğŃåŇŰçśżçŽĎæĹŘåŚŸ

éŮőéćŸ

ä¡ăæČşåIJĺçśżèćńåőŽäźĽçŽĎæŮűåĂŹåŕśåĹİåğŃåŇŰäÿĂéČĺåĹĘçśżçŽĎæĹŘåŚŸïijŇèĂŇäÿ■æŸŕèęĄç

èğčåĘşæŰźæąĹ

åIJĺçśżåőŽäźĽæŮűåŕśæĽğèąŇåĹİåğŃåŇŰæĹŰèő¿ç¡őæŞ■ä¡IJæŸŕåĚČçśżçŽĎäÿĂäÿłåĚÿåđŃåžŤçŤĺåIJ è£ŹæŮűåĂŹä¡ăåŔŕäżěæĽğèąŇäÿĂäžŻéćİåď ŰçŽĎæŞ■ä¡IJãĂĆ

äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇåĹľçŤĺè£ŹäÿłæĂİèůŕæİěåĹŻåżžçśżäijijäžŐ collections æĺąåİŮäÿ■çŽĎåŚ¡åŘ■åĚČçżĎçŽĎçśżïijŽ

```
import operator

class StructTupleMeta(type):
    def __init__(cls, *args, **kwargs):
        super().__init__(*args, **kwargs)
        for n, name in enumerate(cls._fields):
            setattr(cls, name, property(operator.itemgetter(n)))

class StructTuple(tuple, metaclass=StructTupleMeta):
    _fields = []
    def __new__(cls, *args):
        if len(args) != len(cls._fields):
```

```
            raise ValueError('{} arguments required'.format(len(cls.
↪_fields)))
        return super().__new__(cls,args)
```

è£ŹæőţäżčçăĄåŔŕäżěçŤĺæİěåőŽäźĽçőĂå■ŢçŽĎå§žäžŐåĚČçżĎçŽĎæŢřæ■őçżŞæđĎïijŇåęĆäÿŃæĽĂç

```
class Stock(StructTuple):
    _fields = ['name', 'shares', 'price']

class Point(StructTuple):
    _fields = ['x', 'y']
```

äÿŃéİćæijŤçď žåőČåęĆä¡Ţåůěä¡IJïijŽ

```
>>> s = Stock('ACME', 50, 91.1)
>>> s
('ACME', 50, 91.1)
>>> s[0]
'ACME'
>>> s.name
'ACME'
>>> s.shares * s.price
4555.0
>>> s.shares = 23
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
AttributeError: can't set attribute
>>>
```

èőĺèőž

è£ŹäÿĂåřŔèŁĆäÿ■ïijŇçśż `StructTupleMeta` èŐůåŔŰåĹřçśżåśđæĂğ `_fields` äÿ■çŽĎåśđæĂğåŘ■å■ŮåĹŮèąĺïijŇ çĎűåŘŐåřĘåőČäżňè¡ňæ■ćæĹŘçŻÿåžŤçŽĎåŔŕèő£éŮőçĽźåőŽåĚČçżĎæ
`operator.itemgetter()` åĹŻåżžäÿĂäÿłèő£éŮőåŹĺåĞ¡æŢřïijŇ çĎűåŘŐ `property()` åĞ¡æŢřåřĘåĚűè¡ňæ■ćæĹŘäÿĂäÿłåśđæĂğãĂĆ

æIJňèŁĆæIJĂéŽ¿æĞĆçŽĎéČĺåĹĘæŸŕç§ěéĄŞäÿ■åŘŇçŽĎåĹİåğŃåŇŰæ■ěéłď æŸŕäżĂäźĹæŮűåĂŹåŔŚ
`StructTupleMeta` äÿ■çŽĎ `__init__()` æŰźæşŢåŔłåIJĺæŕŔäÿłçśżèćńåőŽäźĽæŮűèćńèřČçŤĺäÿĂæňąã
`cls` åŔĆæŢřåřśæŸŕéĆčäÿłèćńåőŽäźĽçŽĎçśżãĂĆåőđéŹĚäÿŁïijŇäÿŁè£řäżčçăĄä¡£çŤĺäžĘ
`_fields` çśżåŔŸéĞŔæİěä£İå■ŸæŰřçŽĎèćńåőŽäźĽçŽĎçśżïijŇ
çĎűåŘŐçżŹåőČåĘ■æůżåŁääÿĂçĆźæŰřçŽĎäÿIJèě£ãĂĆ

`StructTuple` çśżä¡IJäÿžäÿĂäÿłæŹőéĂŽçŽĎå§žçśżïijŇä¿ŻåĚűäżŰä¡£çŤĺèĂĚæİěçżğæĽ£ãĂĆ
è£Źäÿłçśżäÿ■çŽĎ `__new__()` æŰźæşŢçŤĺæİěæđĎéĂăæŰřçŽĎåőđä¿ŃãĂĆ è£ŹéĞŇä¡£çŤĺ
`__new__()` åźűäÿ■æŸŕå¿ĹåÿÿèğĄïijŇäÿżèęĄæŸŕåŻääÿžæĹŚäżňèęĄä£őæŤźåĚČçżĎçŽĎèřČçŤĺç■¿åŘ■ïij
ä¡£å¿ŮæĹŚäżňåŔŕäżěåČŔæŹőéĂŽçŽĎåőđä¿ŃèřČçŤĺéĆčæăůåĹŻåżžåőđä¿ŃãĂĆåřśåČŔäÿŃéİćè£ŹæăůïijŽ

```
s = Stock('ACME', 50, 91.1) # OK
s = Stock(('ACME', 50, 91.1)) # Error
```

èů§ `__init__()` äÿ■åŘŇçŽĎæŸŕïijŇ`__new__()` æŰźæşŢåIJĺåőđä¿ŃèćńåĹŻåžžäźŃåĽ■èćńèğęåŔŚ
çŤśäžŐåĚČçżĎæŸŕäÿ■åŔŕä£őæŤźçŽĎïijŇæĽĂäžëäÿĂæŮęåőČäžňèćńåĹŻåžžäžĘåřśäÿ■åŔŕèČ¡åŕźåőČåĄŽä
`__init__()` äijŽåIJĺåőđä¿ŃåĹŻåžžçŽĎæIJĂåŘŐèćńèğęåŔŚïijŇ
è£Źæăůç ŽĎèŕÍæĹŚäžňåřśåŔŕäžëåĄŽæĹŚäžňæČşåĄŽçŽĎäžĘãĂĆè£Źäź§æŸŕäÿžäżĂäźĹ
`__new__()` æŰźæşŢåůšçżŔèćńåőŽäźĽäžĘãĂĆ

åř¡çőąæIJňèŁĆå¿Łç§■ïijŇè£ŸæŸŕéIJĂèęĄä¡ăèČ¡äžŢçżĘçăŤèŕżïijŇæůśåĚěæĂÍèĂČPythonçśżæŸŕåęĆä
è£ŸæIJĽåŕśæŸŕåĚČçśżåŠŇçśżçŽĎåŘĎäÿłäÿ■åŘŇçŽĎæŰźæşŢçľűçń§åIJĺäžĂäźĹæŮűåĂŹèćńèřČçŤĺãĂĆ

PEP 422 æŔŘä¿ŽäžĘäÿĂäÿłèğčåĘşæIJňèŁĆéŮőécŸçŽĎåŔęåď ŰäÿĂçğ■æŰźæşŢãĂĆ
ä¡ĘæŸŕïijŇæĹłæ■ćåĽræĹŚåĘŹè£ŹæIJňäźęçŽĎæŮűåĂŹïijŇåőČè£ŸæšąèćńéĞĞçžşåŠŇæŐěåŔŮãĂĆ
åř¡çőąåęĆæ■ď ïijŇåęĆæđIJä¡ăä¡£çŤĺçŽĎæŸŕPython 3.3æĹŰæŽť éńŸçŽĎçĽĹæIJňïijŇéĆčäźĹè£ŸæŸŕåĂijå

11.20 9.20 åĹľçŤĺåĞ¡æŢřæşÍèğčåőđçŐřæŰźæşŢéĞ■è¡¡

éŮőécŸ

ä¡ăåůšçżŔå■ęè£ĞæĂŐæăůä¡£çŤĺåĞ¡æŢřåŔĆæŢřæşÍèğčïijŇéĆčäźĹä¡ăåŔŕèČ¡äijŽæČşåĹľçŤĺåőČæÍěåőđ
ä¡ĘæŸŕä¡ăäÿ■çąőåőŽåžŤèŕěæĂŐæăůåŐžåőđçŐřïijĹæĹŰèĂĚåĹřåžŢèąŇå¿ŮéĂŽäÿ■ïijĽãĂĆ

èğčåĘşæŰźæąĹ

æIJňåřŔèŁĆçŽĎæŁĂæIJŕæŸŕå§žäžŐäÿĂäÿłçőĂå■ŢçŽĎæŁĂæIJŕïijŇéĆčåřśæŸŕPythonåĚĄèőÿåŔĆæŢ

```
class Spam:
    def bar(self, x:int, y:int):
        print('Bar 1:', x, y)

    def bar(self, s:str, n:int = 0):
        print('Bar 2:', s, n)

s = Spam()
s.bar(2, 3)  # Prints Bar 1: 2 3
s.bar('hello')  # Prints Bar 2: hello 0
```

äÿŃéIćæŸŕæĹŚäžñçňňäÿĂæ■ěçŽĎåřÍèŕŢïijŇä¡£çŤĺåĹřäžĘäÿĂäÿłåĚČçśżåŠŇæŔŘè£řåŹĺïijŽ

```
# multiple.py
import inspect
import types

class MultiMethod:
    '''
    Represents a single multimethod.
    '''
    def __init__(self, name):
        self._methods = {}
        self.__name__ = name
```

```
    def register(self, meth):
        '''
        Register a new method as a multimethod
        '''
        sig = inspect.signature(meth)

        # Build a type signature from the method's annotations
        types = []
        for name, parm in sig.parameters.items():
            if name == 'self':
                continue
            if parm.annotation is inspect.Parameter.empty:
                raise TypeError(
                    'Argument {} must be annotated with a type'.
↪format(name)
                )
            if not isinstance(parm.annotation, type):
                raise TypeError(
                    'Argument {} annotation must be a type'.
↪format(name)
                )
            if parm.default is not inspect.Parameter.empty:
                self._methods[tuple(types)] = meth
            types.append(parm.annotation)

        self._methods[tuple(types)] = meth

    def __call__(self, *args):
        '''
        Call a method based on type signature of the arguments
        '''
        types = tuple(type(arg) for arg in args[1:])
        meth = self._methods.get(types, None)
        if meth:
            return meth(*args)
        else:
            raise TypeError('No matching method for types {}'.
↪format(types))

    def __get__(self, instance, cls):
        '''
        Descriptor method needed to make calls work in a class
        '''
        if instance is not None:
            return types.MethodType(self, instance)
        else:
            return self

class MultiDict(dict):
    '''
```

```
    Special dictionary to build multimethods in a metaclass
    '''
    def __setitem__(self, key, value):
        if key in self:
            # If key already exists, it must be a multimethod or␣
↪callable
            current_value = self[key]
            if isinstance(current_value, MultiMethod):
                current_value.register(value)
            else:
                mvalue = MultiMethod(key)
                mvalue.register(current_value)
                mvalue.register(value)
                super().__setitem__(key, mvalue)
        else:
            super().__setitem__(key, value)

class MultipleMeta(type):
    '''
    Metaclass that allows multiple dispatch of methods
    '''
    def __new__(cls, clsname, bases, clsdict):
        return type.__new__(cls, clsname, bases, dict(clsdict))

    @classmethod
    def __prepare__(cls, clsname, bases):
        return MultiDict()
```

äÿžäžĘä¡£çŤĺè£ŹäÿłçśżïijŇä¡ăåŔŕäżěåČŔäÿŃéİćè£ŹæăůåĘŹïijŽ

```
class Spam(metaclass=MultipleMeta):
    def bar(self, x:int, y:int):
        print('Bar 1:', x, y)

    def bar(self, s:str, n:int = 0):
        print('Bar 2:', s, n)

# Example: overloaded __init__
import time

class Date(metaclass=MultipleMeta):
    def __init__(self, year: int, month:int, day:int):
        self.year = year
        self.month = month
        self.day = day

    def __init__(self):
        t = time.localtime()
        self.__init__(t.tm_year, t.tm_mon, t.tm_mday)
```

äÿŃéİćæŸŕäÿĂäÿłäžd'äžŠçd'žä¿ŃæİěéłŇèŕĄåőČèČ¡æ■čçąőçŽĎåůěä¡IJïijŽ

```
>>> s = Spam()
>>> s.bar(2, 3)
Bar 1: 2 3
>>> s.bar('hello')
Bar 2: hello 0
>>> s.bar('hello', 5)
Bar 2: hello 5
>>> s.bar(2, 'hello')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "multiple.py", line 42, in __call__
        raise TypeError('No matching method for types {}'.
 ↪format(types))
TypeError: No matching method for types (<class 'int'>, <class 'str
 ↪'>)
>>> # Overloaded __init__
>>> d = Date(2012, 12, 21)
>>> # Get today's date
>>> e = Date()
>>> e.year
2012
>>> e.month
12
>>> e.day
3
>>>
```

èőÍèőž

åÍęçŹ¡æÍèèőšïijŇçŻÿåŕźäžŐéĂŽåÿÿçŽĎäžčçăĄèĂŇåůšæIJňèŁĆä¡£çŤÍåĹřäžĘå¿Ĺåď’ŽçŽĎé■ŤæşŢäžčçă
ä¡ĘæŸŕïijŇåőČå■t’èČ¡èőľ’æĹŚäžňæůśåĚěçŘĘèğčåĚČçśżåŠŇæŔŔè£řåŹÍçŽĎåžŢåśĆåůěä¡IJåŐ§çŘĘïijŇ
åźűèČ¡åŁäæůśåŕźè£ŹäžŻæęĆå£ţçŽĎå■řèśąãĂĆåŻäæ■d’ïijŇåŕśçőŮä¡äåźűäÿ■äijŽçńŇå■şåŐżåžŢçŤÍæIJňèŁĆ
åőČçŽĎäÿĂäžŻåžŢåśĆæĂÍæČşå■t’äijŽå¡śåŞ■åĹřåĚűåőČæűĽ’åŔŁåĹřåĚČçśżãĂĄæŔŔè£řåŹÍåŠŇåĞ¡æŢřæş

æIJňèŁĆçŽĎåőđçŐřäÿ■çŽĎäÿżèęĄæĂÍèůřåĚűåőđæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆMutipleMeta
åĚČçśżä¡£çŤÍåőČçŽĎ `__prepare__()` æŰźæşŢ æÍèæŔŘä¿ŻäÿĂäÿłä¡IJäÿž `MultiDict`
åőđä¿ŃçŽĎèĞłåőŽäźĽå■ŮåĚÿãĂĆè£Źäÿłèů§æŹőéĂŽå■ŮåĚÿäÿ■äÿĂæäůçŽĎæŸŕïijŇ
`MultiDict` äijŽåIJÍåĚČçt’ãèćńèő¿ç¡őçŽĎæŮűåĂŹæčĂæ§ěæŸŕåŘęåůšçżŔå■ŸåIJÍïijŇåęĆædIJå■ŸåIJÍçŽĎ
`MultiMethod` åőđä¿Ńäÿ■åŘĹåźűãĂĆ

`MultiMethod` åőđä¿ŃéĂŽè£ĞædĎåžžäžŐçśżådŃç■¿åŘ■åĹřåĞ¡æŢřçŽĎæŸäåřĎæÍěæŤűéŻĘæŰźæşŢ
åIJÍè£ŹäÿłædĎåžžè£ĞçÍŃäÿ■ïijŇåĞ¡æŢřæşÍèğčèćńçŤÍæÍěæŤűéŻĘè£ŹäžŻç■¿åŘ■çĎűåŘŐædĎåžžè£ŹäÿłæŸ
è£Źäÿłè£ĞçÍŃåIJÍ `MultiMethod.register()` æŰźæşŢäÿ■åőđçŐřãĂĆ
è£Źçğ■æŸäåřĎçŽĎäÿĂäÿłåĚşéŤőçĽźçĆźæŸŕåŕźäžŐåď’ŽäÿłæŰźæşŢïijŇæĽ’ĂæIJĽåŔĆæŢřçśżådŃéČ¡å£Ěé

äÿžäžĘèőľ’ `MultiMethod` åőđä¿ŃæĺąæŃ§äÿĂäÿłèřČçŤÍïijŇåőČçŽĎ
`__call__()` æŰźæşŢèćńåőđçŐřäžĘãĂĆ è£ŹäÿłæŰźæşŢäžŐæĽ’ĂæIJĽæŐŠéŹd’ `slef`
çŽĎåŔĆæŢřäÿ■ædĎåžžäÿĂäÿłçśżådŃåĚČçżĎïijŇåIJÍåĘĚéČÍmapäÿ■æ§ěæĽ’¿è£ŹäÿłæŰźæşŢïijŇ
çĎűåŘŐèřČçŤÍçŻÿåžŤçŽĎæŰźæşŢãĂĆäÿžäžĘèČ¡èőľ’ `MultiMethod`

åőđä¿ŃåIJĺçśżåőŽäźĽæŮűæ■čçąőæŞ■ä¡IJïijŇ__get__() æŸŕå£Ěéążå¿ŮåőđçŐřçŽĎãĂĆ
åőČèćńçŤĺæİěæđĎåżžæ■čçąőçŽĎçżŚåőŽæŰźæşŢãĂĆæŕŤåęĆïijŽ

```
>>> b = s.bar
>>> b
<bound method Spam.bar of <__main__.Spam object at 0x1006a46d0>>
>>> b.__self__
<__main__.Spam object at 0x1006a46d0>
>>> b.__func__
<__main__.MultiMethod object at 0x1006a4d50>
>>> b(2, 3)
Bar 1: 2 3
>>> b('hello')
Bar 2: hello 0
>>>
```

äÿ■è£ĞæIJňèŁĆçŽĎåőđçŐřè£ŸæIJĽäÿĂäžŻéŹŘåĹűïijŇåĚűäÿ■äÿĂäÿłæŸŕåőČäÿ■èČ¡ä¡£çŤĺåĚşéŤőå■

```
>>> s.bar(x=2, y=3)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: __call__() got an unexpected keyword argument 'y'

>>> s.bar(s='hello')
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
TypeError: __call__() got an unexpected keyword argument 's'
>>>
```

äź§èőÿæIJĽåĚűäżŰçŽĎæŰźæşŢèČ¡æůżåŁäè£Źçğ■æŤŕæŇĄïijŇä¡ĘæŸŕåőČéIJĂèęĄäÿĂäÿłåőŇåĚĺäÿ■
éŮőéćŸåIJĺäžŐåĚşéŤőå■ŮåŔĆæŢřçŽĎåĞžçŐřæŸŕæšąæIJĽéąžåžŔçŽĎãĂĆå¡ŞåőČèů§ä¡■ç¡őåŔĆæŢřæůůåŔ
éĆčä¡äçŽĎåŔĆæŢřåřśäijŽåŔŸå¿ŮæŕŤè¿ČæůůäźśäžĘïijŇè£ŹæŮűåĂŹä¡ăäÿ■å¿Ůäÿ■åIJĺ
__call__() æŰźæşŢäÿ■åĚĹåŐżåĄŽäÿłæŐŠåžŔãĂĆ

åŘŇæăůåŕźäžŐçżğæĽ£äź§æŸŕæIJĽéŹŘåĹűçŽĎïijŇä¿ŃåęĆïijŇçśżäijijäÿŃéİćè£Źçğ■äżčçăĄåřśäÿ■èČ¡

```
class A:
    pass

class B(A):
    pass

class C:
    pass

class Spam(metaclass=MultipleMeta):
    def foo(self, x:A):
        print('Foo 1:', x)

    def foo(self, x:C):
        print('Foo 2:', x)
```

åŐ§åŽäæŸŕåŽääÿž `x : A` æşĺèğčäÿ■èČ¡æĹŘåŁ§åŇźéĚ■å■Řçśžåőđä¿ŃïijĹæŕŤåęĆBçŽĎåőđä¿ŃïijĽïijŇå

```
>>> s = Spam()
>>> a = A()
>>> s.foo(a)
Foo 1: <__main__.A object at 0x1006a5310>
>>> c = C()
>>> s.foo(c)
Foo 2: <__main__.C object at 0x1007a1910>
>>> b = B()
>>> s.foo(b)
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "multiple.py", line 44, in __call__
        raise TypeError('No matching method for types {}'.
 ↪format(types))
TypeError: No matching method for types (<class '__main__.B'>,)
>>>
```

ä¡IJäÿžä¡£çŤĺåĚČçśžåŠŇæşĺèğčçŽĎäÿĂçğ■æŻ£äžčæŰźæąĹïijŇåŔŕäžěéĂŽè£ĞæŔŔè£ŕåŹĺæİěåőđçŐřç

```
import types

class multimethod:
    def __init__(self, func):
        self._methods = {}
        self.__name__ = func.__name__
        self._default = func

    def match(self, *types):
        def register(func):
            ndefaults = len(func.__defaults__) if func.__defaults__␣
 ↪else 0
            for n in range(ndefaults+1):
                self._methods[types[:len(types) - n]] = func
            return self
        return register

    def __call__(self, *args):
        types = tuple(type(arg) for arg in args[1:])
        meth = self._methods.get(types, None)
        if meth:
            return meth(*args)
        else:
            return self._default(*args)

    def __get__(self, instance, cls):
        if instance is not None:
            return types.MethodType(self, instance)
        else:
            return self
```

äÿžäžĘä¡£çŤĺæŔŔè£řåŹĺçĽĹæIJňïijŇä¡ăéIJĂèęĄåČŔäÿŃéİćè£ŹæăůåĘŹïijŽ

```
class Spam:
    @multimethod
    def bar(self, *args):
        # Default method called if no match
        raise TypeError('No matching method for bar')

    @bar.match(int, int)
    def bar(self, x, y):
        print('Bar 1:', x, y)

    @bar.match(str, int)
    def bar(self, s, n = 0):
        print('Bar 2:', s, n)
```

æŔŔè£řåŹĺæŰźæąĹåŘŇæăůäź§æIJĽåĽ■éİćæŔŘåĹřçŽĎéŹŘåĹűïijĹäÿ■æŤŕæŇĄåĚşéŤőå■ŮåŔĆæŢřåŠ

æĽĂæIJĽäžŃçĽľéČ¡æŸŕåźşç■ĽçŽĎïijŇæIJĽåě¡æIJĽåİŔïijŇäź§èőÿæIJĂåě¡çŽĎåŁđæşŢåřśæŸŕåIJĺæŽ
äÿ■è£ĞæIJĽäžŻçĽźæőŁæČĚåĘţäÿŃè£ŸæŸŕæIJĽæĎŔäźĽçŽĎïijŇæŕŤåęĆå§žäžŐæĺąåijŔåŇźéĚ■çŽĎæŰźæ
äÿ¿äÿłä¿Ńå■ŘïijŇ8.21åřŔèŁĆäÿ■çŽĎèő£éŮőèĂĚæĺąåijŔåŔŕäżěä£őæŤźäÿžäÿĂäÿłä¡£çŤĺæŰźæşŢéĞ■è¡¡çŽĎ
ä¡ĘæŸŕïijŇéŹďäžĘè£ŹäÿłäżěåďŰïijŇéĂŽåÿÿäÿ■åžŤèŕěä¡£çŤĺæŰźæşŢéĞ■è¡¡ïijĹåřśçőĂå■ŢçŽĎä¡£çŤĺäÿ■å

åIJĺPythonçď¿åŇžåŕźäžŐåőđçŐřæŰźæşŢéĞ■è¡¡çŽĎèőĺèőžåůšçżŔçŤśæİěåůšäźĚãĂĆ
åŕźäžŐåijŢåŔŚè£ŹäÿłäžĽèőžçŽĎåŐ§åŻäïijŇåŔŕäżěåŔĆèĂČäÿŃGuido van
RossumçŽĎè£ŹçŕĞå■ŽåőćïijŽ Five-Minute Multimethods in Python

11.21 9.21 éĄ£åĚ■éĞ■åď■çŽĎåśđæĂğæŰźæşŢ

éŮőéćŸ

ä¡ăåIJĺçśżäÿ■éIJĂèęĄéĞ■åď■çŽĎåőŽäźĽäÿĂäžŻæĽğèąŇçŻÿåŘŇéĂżè¿ŚçŽĎåśđæĂğæŰźæşŢïijŇæŕŤ

èğčåĘşæŰźæąĹ

èĂČèŹŚäÿŃäÿĂäÿłçőĂå■ŢçŽĎçśżïijŇåőČçŽĎåśđæĂğçŤśåśđæĂğæŰźæşŢåŇĚèčĚïijŽ

```
class Person:
    def __init__(self, name ,age):
        self.name = name
        self.age = age

    @property
    def name(self):
        return self._name

    @name.setter
    def name(self, value):
        if not isinstance(value, str):
```

```
            raise TypeError('name must be a string')
        self._name = value

    @property
    def age(self):
        return self._age

    @age.setter
    def age(self, value):
        if not isinstance(value, int):
            raise TypeError('age must be an int')
        self._age = value
```

åŔŕäżěçIJŃåĹřïijŇäÿžäžĘåőđçŐřåśđæĂğåĂijçŽĎçśżåđŃæčĂæ§ěæĹŚäżňåĘŹäžĘå¿Ĺåď ŽçŽĎéĞ■åď■ä
åŔłèęĄä¡ăäżěåŘŐçIJŃåĹřçśżäijijè£ŹæăůçŽĎäżčçăĄïijŇä¡ăéČ¡åžŤèŕěæČşåŁđæşŢåŐżçőĂåŇŰåőČãĂĆ
äÿĂäÿłåŔŕèąŇçŽĎæŰźæşŢæŸŕåĹŻåżžäÿĂäÿłåĞ¡æŢřçŤĺæİěåőŽäźĽåśđæĂğåźűè£ŤåŻđåőČãĂĆä¿ŃåęĆïijŽ

```
def typed_property(name, expected_type):
    storage_name = '_' + name

    @property
    def prop(self):
        return getattr(self, storage_name)

    @prop.setter
    def prop(self, value):
        if not isinstance(value, expected_type):
            raise TypeError('{} must be a {}'.format(name, expected_
↪type))
        setattr(self, storage_name, value)

    return prop

# Example use
class Person:
    name = typed_property('name', str)
    age = typed_property('age', int)

    def __init__(self, name, age):
        self.name = name
        self.age = age
```

èőĺèőž

æIJňèŁĆæĹŚäżňæijŤçď žåĘĚéČĺåĞ¡æŢřæĹŰèĂĚéŮ■åŇĚçŽĎäÿĂäÿłéĞ■èęĄçĽźæĂğïijŇåőČäżňå¿Ĺå
typed_property() çIJŃäÿŁåŐżæIJĽçĆźéŽ¿çŘĘèğčïijŇåĚűåőđåőČæĽĂåĄŽçŽĎäżĚäżĚåřśæŸŕäÿžä¡ăç
åŻăæ■ď ïijŇå¡ŞåIJĺäÿĂäÿłçśżäÿ■ä¡£çŤĺåőČçŽĎæŮűåĂŹïijŇæŢĹæđIJèů§åřĘåőČéĞŇéİćçŽĎäżčçăĄæŤ¿åĹř
åř¡çőąåśđæĂğçŽĎ getter åŠŇ setter æŰźæşŢèő£éŮőäžĘæIJňåIJřåŔŸéĞŔåęĆ name ,
expected_type äżěåŔŁ storate_name ïijŇè£Źäÿłå¿Ĺæ■čåÿÿïijŇè£ŹäžŻåŔŸéĞŔçŽĎåĂijäijŽä£İå■Ÿ

æĹŚäžňè£ŸåŔŕäžěä¡£çŤĺ `functools.partial()` æİěçĺ■çĺ■æŤźåŔŸäÿŃè£Źäÿłä¿Ńå■ŘïijŇå¿ĹæIJI

```
from functools import partial

String = partial(typed_property, expected_type=str)
Integer = partial(typed_property, expected_type=int)

# Example:
class Person:
    name = String('name')
    age = Integer('age')

    def __init__(self, name, age):
        self.name = name
        self.age = age
```

åĚűåðđä¡ăåŔŕäžěåŔŚçŐřïijŇè£ŹéĞŇçŽĎäžčçăĄèů§8.13åřŔèŁĆäÿ■çŽĎçśžåđŃçşżçż§æŔŔè£řåŹĺäžčçă

11.22 9.22 åőŽäźĽäÿŁäÿŃæŰĞçőąçŘĘåŹĺçŽĎçőÅå■ŢæŰźæşŢ

éŮőécŸ

ä¡ăæČşèĞłåůśåŐżåőđçŐřäÿĂäÿłæŰřçŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇäžěä¿£ä¡£çŤĺwithèŕ■åŘěãĂĆ

èğčåĘşæŰźæąĹ

åőđçŐřäÿĂäÿłæŰřçŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹĺçŽĎæIJĂçőÅå■ŢçŽĎæŰźæşŢåřśæŸŕä¡£çŤĺ `contexlib` æĺąåİŮäÿ■çŽĎ `@contextmanager` èčĚéěřåŹĺãĂĆ äÿŃéİćæŸŕäÿĂäÿłåőđçŐřäžĘäžčçăĄåİÙèőąæŮűåŁ§èČ¡çŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹĺä¿Ńå■ŘïijŽ

```
import time
from contextlib import contextmanager

@contextmanager
def timethis(label):
    start = time.time()
    try:
        yield
    finally:
        end = time.time()
        print('{}: {}'.format(label, end - start))

# Example use
with timethis('counting'):
    n = 10000000
    while n > 0:
        n -= 1
```

åIJÍåĞ¡æŢř `timethis()` äÿ■ïijŇ`yield`äźŃåL’■çŽĎäżčçăĄäijŽåIJÍäÿŁäÿŃæŰĞçőąçŘĘåŹÍäÿ■ä¡IJäÿ `__enter__()` æŰźæşŢæL’ğèąŇïijŇ æL’ĂæIJL’åIJÍ `yield` äźŃåŘŐçŽĎäżčçăĄäijŽä¡IJäÿž `__exit__()` æŰźæşŢæL’ğèąŇãĂĆ åęĆæđIJåĞžçŐřäžĘåijĆåÿÿïijŇåijĆåÿÿäijŽåIJÍyield-èŕ■åŘéĆčéĞŇæŁŻåĞžãĂĆ

äÿŃéİćæŸŕäÿĂäÿłæŻť’åŁäéńŸçžğäÿĂçĆźçŽĎäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇåőđçŐřäžĘåĹŮèąĺåŕźèśąäÿŁç

```
@contextmanager
def list_transaction(orig_list):
    working = list(orig_list)
    yield working
    orig_list[:] = working
```

è£Źæőţäżčçă Ąç ŽĎä¡IJçŤÍæŸŕäżżä¡Ţåŕźå ĹŮèąĺçŽĎä£őæŤźåŘłæIJL’å¡ŞæL’ĂæIJL’äżčçăĄè£ŘèąŇåőŇæ äÿŃéİćæĹŚäžňæİěæijŤçd’žäÿĂäÿŃïijŽ

```
>>> items = [1, 2, 3]
>>> with list_transaction(items) as working:
...     working.append(4)
...     working.append(5)
...
>>> items
[1, 2, 3, 4, 5]
>>> with list_transaction(items) as working:
...     working.append(6)
...     working.append(7)
...     raise RuntimeError('oops')
...
Traceback (most recent call last):
    File "<stdin>", line 4, in <module>
RuntimeError: oops
>>> items
[1, 2, 3, 4, 5]
>>>
```

èőÍèőž

éĂŽåÿÿæČĚåĘţäÿŃïijŇåęĆæđIJèęĄåĘŹäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇä¡ăéIJĂèęĄåőŽäźL’äÿĂäÿłç `__enter__()` åŠŇäÿĂäÿł `__exit__()` æŰźæşŢïijŇåęĆäÿŃæL’Ăçd’žïijŽ

```
import time

class timethis:
    def __init__(self, label):
        self.label = label

    def __enter__(self):
        self.start = time.time()

    def __exit__(self, exc_ty, exc_val, exc_tb):
```

```
        end = time.time()
        print('{}: {}'.format(self.label, end - self.start))
```

åř¡çőąè£ŹäÿłäźŞäÿ■éŽ¿åĘŹïijŇä¡ĘæŸŕçŻÿæŕŤè¿ČåĘŹäÿĂäÿłçőĂå■ŢçŽĎä¡£çŤÍ `@contextmanager` æşĺèğčçŽĎåĞ¡æŢřèĂŇèĺĂè£ŸæŸŕçĺ■æŸ¿äźŔåŚşãĂĆ

`@contextmanager` åžŤèřëäžĚäżĚçŤĺæĺěåĘŹèĞłåŇĚåŘńçŽĎäÿŁäÿŃæŰĞçőąçŘĘåĞ¡æŢřãĂĆ åęĆæđIJä¡ăæIJL'äÿĂäžŻåŕźèśą(æŕŤåęĆäÿĂäÿłæŰĞäžűãĂĄç¡ŚçżIJè£đæŐěæĹŰéŤA)ïijŇéIJĂèęĄæŤŕæŇĄ `with` èŕ■åŘěïijŇéĆčäźĹä¡ăåŕśéIJĂèęĄå■ŢçŃňåőđçŐř `__enter__()` æŰźæşŢåŠŇ `__exit__()` æŰźæşŢãĂĆ

11.23 9.23 åIJĺåśĂéČĺåŔŸéĞŘå§§äÿ■æĽ'ğèąŇäżčçăĄ

éŮőéćŸ

ä¡ăæČşåIJĺä¡£çŤĺèŇČåŻť'åĘĚæL'ğèąŇæ§Řäÿłäżčçă ĄçL'Ğæőţ ïijŇåźűäÿŤåÿŇæIJŻåIJĺæL'ğèąŇåŘŐæL'Ă

èğčåĘşæŰźæąĹ

äÿžäžĘçŘĘèğčè£ŹäÿłéŮőéćŸïijŇåĚĹèŕŢèŕŢäÿĂäÿłçőĂå■ŢåIJżæŹŕăĂĆéęŮåĚĹïijŇåIJĺåĚĺåśĂåŚ¡åŘ■ç

```
>>> a = 13
>>> exec('b = a + 1')
>>> print(b)
14
>>>
```

çĎűåŘŐïijŇåĘ■åIJĺäÿĂäÿłåĞ¡æŢřäÿ■æL'ğèąŇåŘŇæăůçŽĎäżčçăĄïijŽ

```
>>> def test():
...     a = 13
...     exec('b = a + 1')
...     print(b)
...
>>> test()
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
    File "<stdin>", line 4, in test
NameError: global name 'b' is not defined
>>>
```

åŔŕäżěçIJŃåĞžïijŇæIJĂåŘŐæŁŻåĞžäžĘäÿĂäÿłNameErroråijĆåÿÿïijŇåřśèů§åIJĺ `exec()` èŕ■åŘěäżŐæšąæL'ğèąŇè£ĞäÿĂæăůãĂĆ èęĄæŸŕä¡ăæČşåIJĺåŘŐéĺćçŽĎèőąçőŮäÿ■ä¡£çŤĺåĹř `exec()` æL'ğèąŇçżŞæđIJçŽĎèŕİåřśäijŽæIJL'éŮőéćŸäžĘãĂĆ

äÿžäžĘä£őæ■čè£ŹæăůçŽĎéŤŹèŕŕïijŇä¡ăéIJĂèęĄåIJĺèřČçŤĺ `exec()` äźŃåL'■ä¡£çŤĺ `locals()` åĞ¡æŢřæİěå¿ŮåĹřäÿĂäÿłåśĂéČĺåŔŸéĞŘå■ŮåĚÿãĂĆ äźŃåŘŐä¡ăåřśèČ¡äżŐåśĂéČĺå■ŮåĚÿäÿ■èŐůåŘŰä£őæŤźè£ĞåŘŐçŽĎåŔŸéĞŘåĂijäžĘãĂĆä¿ŃåęĆïijŽ

```
>>> def test():
...     a = 13
...     loc = locals()
...     exec('b = a + 1')
...     b = loc['b']
...     print(b)
...
>>> test()
14
>>>
```

èőÍèőž

åőđéŹĚäÿŁåŕźäžŐ exec() çŽĎæ■čçąőä¡£çŤÍæŸŕæŕŤè¿ČéŽ¿çŽĎãĂĆåď ğåď ŽæŢřæČĚåĘţäÿŃå¡Şä¡ăè
exec() çŽĎæŮűåĂŹïijŇ è£ŸæIJĽåŔęåď ŰæŻť åě¡çŽĎèğčåĘşæŰźæąĹïijĹæŕŤåęĆèčĚéěřåŹÍãĂĄéŮ■åŇĚã

çĎűèĂŇïijŇåęĆæđIJä¡ääż■çĎűèęĄä¡£çŤÍ exec() ïijŇæIJňèŁĆåĹŮåĞžäžĘäÿĂäžŻåęĆä¡Ţæ■čçąőä¡£ç
éżŸèőď æČĚåĘţäÿŃïijŇexec() äijŽåIJÍèřČçŤÍèĂĚåśĂéČÍåŠŇåĚÍåśĂèŇČåŻť åĘĚæĽğèąŇäžčçăĄãĂĆçĎű
äijăéĂŠçżŹ exec() çŽĎåśĂéČÍèŇČåŻť æŸŕæŃůèť Ìåőđé ŹĚåśĂéČÍåŔŸéĞŔçżĎæĹŘçŽĎäÿĂäÿłå■ŮåĚÿãĂ
åŻăæ■ď ïijŇåęĆæđIJ exec() åęĆæđIJæĽğèąŇäžĘä£őæŤźæŞ■ä¡IJïijŇè£Źçğ■ä£őæŤźåŘŐçŽĎçżŞæđIJåŕźåő
äÿŃéÌćæŸŕåŔęåď ŰäÿĂäÿłæijŤçď žåőČçŽĎä¿Ńå■ŘïijŽ

```
>>> def test1():
...     x = 0
...     exec('x += 1')
...     print(x)
...
>>> test1()
0
>>>
```

äÿŁéÌćäżčçăĄéĞŇïijŇå¡Şä¡ăèřČçŤÍ locals() èŐůåŔŰåśĂéČÍåŔŸéĞŔæŮűïijŇä¡ăèŐůå¿ŮçŽĎæŸŕäi
exec() çŽĎåśĂéČÍåŔŸéĞŔçŽĎäÿĂäÿłæŃůèť ÌãĂĆ éĂŽè£ĞåIJÍäżčçăĄæĽğèąŇåŘŐåőąæ§ěè£Źäÿłå■ŮåĚÿ

```
>>> def test2():
...     x = 0
...     loc = locals()
...     print('before:', loc)
...     exec('x += 1')
...     print('after:', loc)
...     print('x =', x)
...
>>> test2()
before: {'x': 0}
after: {'loc': {...}, 'x': 1}
x = 0
>>>
```

äżŤçżĘèğĆår§æIJĂåŘŐäÿĂæ■ęçŽĎè¿ŞåĞžïijŇéŹď éÌđä¡ăårĘ loc

äÿ■èćńä£őæŤźåŘŐçŽĎåĂijæĽŃåŁĹèţŃåĂijçżŹxïijŇåŘęåĹŹxåŔŸéĞŔåĂijæŸŕäÿ■äijŽåŔŸçŽĎãĂĆ

åIJĺä¡£çŤĺ `locals()` çŽĎæŮűåĂŹïijŇä¡ăéIJĂèęĄæşĺæĎŔæŞ■ä¡IJéąžåžŔãĂĆærŔæňąåőČèćńèřČçŤĺçŹ `locals()` äijŽèŐůåŔŰåśĂéČĺåŔŸéĞŔåĂijäÿ■çŽĎåĂijåźűèęĘçŻŰå■ŮåĚÿäÿ■çŻÿåžŤçŽĎåŔŸéĞŔãĂĆèřůæşĺæĎŔèğĆår§äÿŃäÿŃéİćè£ŹäÿłèŕŢéłŇçŽĎè¿ŞåĞžçżŞæđIJïijŽ

```
>>> def test3():
...     x = 0
...     loc = locals()
...     print(loc)
...     exec('x += 1')
...     print(loc)
...     locals()
...     print(loc)
...
>>> test3()
{'x': 0}
{'loc': {...}, 'x': 1}
{'loc': {...}, 'x': 0}
>>>
```

æşĺæĎŔæIJĂåŘŐäÿĂæňąèřČçŤĺ `locals()` çŽĎæŮűåĂŹxçŽĎåĂijæŸŕåęĆä¡Ţèćńèę€çŻŰæŐĽçŽĎã

ä¡IJäÿž `locals()` çŽĎäÿĂäÿłæŻ£äžčæŰźæąĹïijŇä¡ăåŔŕäžëä¡£çŤĺä¡ăèĞłåůśçŽĎå■ŮåĚÿïijŇåźűåřĘåő `exec()` ãĂĆä¿ŃåęĆïijŽ

```
>>> def test4():
...     a = 13
...     loc = { 'a' : a }
...     glb = { }
...     exec('b = a + 1', glb, loc)
...     b = loc['b']
...     print(b)
...
>>> test4()
14
>>>
```

åđ˙ğéČĺåĹĘæČĚåĘţäÿŃïijŇè£Źçğ■æŰźåijŔæŸŕä¡£çŤĺ `exec()` çŽĎæIJĂä¡şåőđèůţãĂĆ ä¡ăåŔłéIJĂèęĄä£ÌèŕĄåĚĺåśĂåŠŇåśĂéČĺå■ŮåĚÿåIJĺåŘŐéİcäžčçăĄèő£éŰőæŰűåůšçżŔèćńåĹÌåğŃåŇŰãĂĆ

è£ŸæIJL˙äÿĂçĆźïijŇåIJĺä¡£çŤĺ `exec()` äźŃåL˙■ïijŇä¡ăåŔŕèČ¡éIJĂèęĄéŰőäÿŃèĞłåůśæŸŕåŘęæIJL˙åĚű åđ˙ğåđ˙ŽæŢřæČĚåĘţäÿŃå¡Şä¡ăèęĄèĂČèŹŚä¡£çŤĺ `exec()` çŽĎæŰűåĂŹïijŇ è£ŸæIJL˙åŘęåđ˙ŰæŻť˙åě¡çŽĎèğčåĘşæŰźæąĹïijŇærŤåęĆèčĚéěřåŹĺãĂĄéŮ■åŇĚãĂĄåĚČçśżïijŇæĹŰåĚűäżŰ

11.24 9.24 èğčæđŘäÿŐåĹĘæđŘPythonæžŘçăĄ

éŮőécŸ

ä¡ăæČşåĘŹèğčæđŘåźűåĹĘæđŘPythonæžŘäžčçăĄçŽĎçĺŃåžŔãĂĆ

èğčåĘşæŰźæąĹ

åď ğéČĺåĹĘçÍŃåžŘåŚŸç§ěéĄŞPythonèČ¡åď §èőąçőŮæĹŰæĽ ğèąŇå■Ůçňęäÿšå¡ćåijŔçŽĎæžŘäžčçăĄãĂ

```
>>> x = 42
>>> eval('2 + 3*4 + x')
56
>>> exec('for i in range(10): print(i)')
0
1
2
3
4
5
6
7
8
9
>>>
```

åř¡çőąåęĆæ■ď ïijŇast æĺąåİŮèČ¡èćńçŤĺæİěåřĘPythonæžŘçăĄçijŰèŕŚæĹŘäÿĂäÿłåŔŕèćńåĹĘæđŘçŽĎa

```
>>> import ast
>>> ex = ast.parse('2 + 3*4 + x', mode='eval')
>>> ex
<_ast.Expression object at 0x1007473d0>
>>> ast.dump(ex)
"Expression(body=BinOp(left=BinOp(left=Num(n=2), op=Add(),
right=BinOp(left=Num(n=3), op=Mult(), right=Num(n=4))), op=Add(),
right=Name(id='x', ctx=Load())))"

>>> top = ast.parse('for i in range(10): print(i)', mode='exec')
>>> top
<_ast.Module object at 0x100747390>
>>> ast.dump(top)
"Module(body=[For(target=Name(id='i', ctx=Store()),
iter=Call(func=Name(id='range', ctx=Load()), args=[Num(n=10)],
keywords=[], starargs=None, kwargs=None),
body=[Expr(value=Call(func=Name(id='print', ctx=Load()),
args=[Name(id='i', ctx=Load())], keywords=[], starargs=None,
kwargs=None))], orelse=[])])"
>>>
```

åĹĘæđŘæžŘçăĄæăŚéIJĂèęĄä¡äèĞłåůśæŻť åď ŽçŽĎå■ęäźăïijŇåőČæŸŕçŤśäÿĂçşżåĹŮASTèŁĆçĆźçżĎ
åĹĘæđŘè£ŹäžŻèŁĆçĆźæIJĂçőĂå■ŢçŽĎæŰźæşŢåřśæŸŕåőŽäźĽäÿĂäÿłèő£éŮőèĂĚçśżïijŇåőđçŐřå¿Łåď Ž
`visit_NodeName()` æŰźæşŢïijŇ `NodeName()` åŇźéĚ■éĆčäžŻä¡äæĎ§åĚť èűčçŽĎèŁĆçĆźãĂĆäÿŃéÍćæ

```
import ast

class CodeAnalyzer(ast.NodeVisitor):
    def __init__(self):
```

```
        self.loaded = set()
        self.stored = set()
        self.deleted = set()

    def visit_Name(self, node):
        if isinstance(node.ctx, ast.Load):
            self.loaded.add(node.id)
        elif isinstance(node.ctx, ast.Store):
            self.stored.add(node.id)
        elif isinstance(node.ctx, ast.Del):
            self.deleted.add(node.id)

# Sample usage
if __name__ == '__main__':
    # Some Python code
    code = '''
    for i in range(10):
        print(i)
    del i
    '''

    # Parse into an AST
    top = ast.parse(code, mode='exec')

    # Feed the AST to analyze name usage
    c = CodeAnalyzer()
    c.visit(top)
    print('Loaded:', c.loaded)
    print('Stored:', c.stored)
    print('Deleted:', c.deleted)
```

åęĆædIJä¡ăè£ŘèąŇè£ŹäýłçÍŃåžŔïijŇä¡ăäijŽå¿ŮåĹřäýŃéİćè£ŹæăůçŽĎè¿ŞåĞžïijŽ

```
Loaded: {'i', 'range', 'print'}
Stored: {'i'}
Deleted: {'i'}
```

æIJĂåŘŐïijŇASTåŔŕäżěéĂŽè£Ğ `compile()` åĞ¡æŢřæİěçijŰèŕŚåźűæL'ğèąŇãĂĆä¿ŃåęĆïijŽ

```
>>> exec(compile(top,'<stdin>', 'exec'))
0
1
2
3
4
5
6
7
8
9
>>>
```

èőÍèőž

å¡Şä¡ãèČ¡åď§åĹĘæđŘæžŘäžčçăĄåźűäżŐäÿ■èŐůåŔŰä£ąæĄŕçŽĎæŮűåĂŹïijŇä¡ăářśèČ¡åĘŹå¿Ĺåď'Žäžč
ä¿ŇåęĆïijŇçŻÿæŕŤçŻšçŽőçŽĎäijãéĂŠäÿĂäžŽäžčçăĄçĽĞæőţåĹřçśžäijij
exec() åĞ¡æŢřäÿ■ïijŇä¡ăåŔŕäžěåĚĹářĘåőČè¡ňæ■ćæĹŘäÿĂäÿłASTïijŇ
çĎűåŘŐèğĆår§åőČçŽĎçżĘèŁĆçIJŃåőČåĹřåžŢæŸŕæĂŐæăůåĄŽçŽĎãĂĆ
ä¡ãè£ŸåŔŕäžěåĘŹäÿĂäžŽåůěåĚůæİěæ§ěçIJŃæ§Řäÿłæĺąå İŮçŽĎåĚĺéČĺæžŘçăĄïijŇåźűäÿŤåIJĺæ■ď'å§žçąĂäÿ

éIJĂèęĄæşĺæĎŘçŽĎæŸŕïijŇåęĆæđIJä¡ăç§ěéĄŞèĞłåůšåIJĺåźšåŢěïijŇä¡ãè£ŸèČ¡åď'§éĞ■åĘŹASTæİěèą
äÿŃéİćæŸŕäÿĂäÿłècĚéěřåŹĺä¿Ńå■ŘïijŇåŔŕäžěéĂŽè£ĞéĞ■æŰřèğčæđŘåĞ¡æŢřä¡ŞæžŘçăĄãĂĄ
éĞ■åĘŹASTåźűéĞ■æŰřåĹŻåžžåĞ¡æŢřäžčçăĄåŕźèśąæİěåřĘåĚĺåśĂèő£éŮőåŔŸéĞŔéŹ■äÿžåĞ¡æŢřä¡Şä¡IJçŤ

```
# namelower.py
import ast
import inspect

# Node visitor that lowers globally accessed names into
# the function body as local variables.
class NameLower(ast.NodeVisitor):
    def __init__(self, lowered_names):
        self.lowered_names = lowered_names

    def visit_FunctionDef(self, node):
        # Compile some assignments to lower the constants
        code = '__globals = globals()\n'
        code += '\n'.join("{0} = __globals['{0}']".format(name)
                          for name in self.lowered_names)
        code_ast = ast.parse(code, mode='exec')

        # Inject new statements into the function body
        node.body[:0] = code_ast.body

        # Save the function object
        self.func = node

# Decorator that turns global names into locals
def lower_names(*namelist):
    def lower(func):
        srclines = inspect.getsource(func).splitlines()
        # Skip source lines prior to the @lower_names decorator
        for n, line in enumerate(srclines):
            if '@lower_names' in line:
                break

        src = '\n'.join(srclines[n+1:])
        # Hack to deal with indented code
        if src.startswith((' ','\t')):
            src = 'if 1:\n' + src
```

```
        top = ast.parse(src, mode='exec')

        # Transform the AST
        cl = NameLower(namelist)
        cl.visit(top)

        # Execute the modified AST
        temp = {}
        exec(compile(top,'','exec'), temp, temp)

        # Pull out the modified code object
        func.__code__ = temp[func.__name__].__code__
        return func
    return lower
```

äÿžäžĘä¡£çŤĺè£ŹäÿłäżčçăĄïijŇä¡ăåŔŕäżěåČŔäÿŃéİćè£ŹæăůåĘŹïijŽ

```
INCR = 1
@lower_names('INCR')
def countdown(n):
    while n > 0:
        n -= INCR
```

èčĚéěřåŹĺäijŽåřĘ `countdown()` åĞ¡æŢřéĞ■åĘŹäÿžçśżäijijäÿŃéİćè£Źæăůå■ŘïijŽ

```
def countdown(n):
    __globals = globals()
    INCR = __globals['INCR']
    while n > 0:
        n -= INCR
```

åIJĺæĂğèČ¡æţŃèŕŢäÿ■ïijŇåőČäijŽèőľåĞ¡æŢřè£ŘèąŇå£ń20%

çŐřåIJĺïijŇä¡ăæŸŕäÿ■æŸŕæČşäÿžä¡ăæĽĂæIJĽçŽĎåĞ¡æŢřéČ¡åŁääÿŁè£ŹäÿłèčĚéěřåŹĺåŚćïij§æĹŰèőÿä
ä¡ĘæŸŕïijŇè£Źå■ťæŸŕåŕźäžŐäÿĂäžŻéńŸçžğæŁĂæIJŕæŕŤåęĆASTæŞ■ä¡IJãĂĄæžŘçăĄæŞ■ä¡IJç■Ľç■ĽçŽĎ

æIJňèŁĆåŔŮåŔęåď ŰäÿĂäÿłåIJĺ `ActiveState` äÿ■åď ĎçŘĘPythonå■ŮèŁĆçăĄçŽĎçńăèŁĆçŽĎåŘŕç
ä¡£çŤĺASTæŸŕäÿĂäÿłæŻťåŁăéńŸçžğçĆźçŽĎæŁĂæIJŕïijŇåźűäÿŤäź§æŻťçőĂå■ŢäžŻãĂĆåŔĆèĂČäÿŃéİćä

11.25 9.25 æŃĘèğčPythonå■ŮèŁĆçăĄ

éŮőéćŸ

ä¡ăæČşéĂŽè£ĞåřĘä¡ăçŽĎäżčçăĄåŔ■çijŰèŕŚæĹŘä¡ŐçžğçŽĎå■ŮèŁĆçăĄæİěæ§ěçIJŃåőČåžŢåśĆçŽĎå

èğčåĘşæŰźæąĹ

`dis` æĺąåİŮåŔŕäżěèćńçŤĺæİěè¿ŞåĞžäżżä¡ŢPythonåĞ¡æŢřçŽĎåŔ■çijŰèŕŚçżŞæđIJãĂĆä¿ŃåęĆïijŽ

```
>>> def countdown(n):
... while n > 0:
...     print('T-minus', n)
...     n -= 1
... print('Blastoff!')
...
>>> import dis
>>> dis.dis(countdown)
...
>>>
```

èőÍèőž

å¡Şä¡ăæČşèęAç§ěéĄŞä¡ăçŽĎçÍŃåžŔåžŢåśĆçŽĎè£ŘèąŇæIJžåĹűçŽĎæŮűåĂŹïijŇdis æÍąåİŮæŸŕå¿ĹæIJL'çŤÍçŽĎãĂĆæŕŤåęĆåęĆæđIJä¡ăæČşèŕŢçİĂçŘĘèğčæĂğèČ¡çL'źå¿ĄãĂĆ èćń dis() åĞ¡æŢřèğčæđŘçŽĎåŐ§åğŃå■ŮèŁĆçăĄåęĆäÿŇæL'Ăçđ'žïijŽ

```
>>> countdown.__code__.co_code
b"x
 ↪'\x00|\x00\x00d\x01\x00k\x04\x00r)\x00t\x00\x00d\x02\x00|\x00\x00\x83
\x02\x00\x01|\x00\x00d\x03\x008}
 ↪\x00\x00q\x03\x00Wt\x00\x00d\x04\x00\x83
\x01\x00\x01d\x00\x00S"
>>>
```

åęĆæđIJä¡ăæČşèĞłåůśèğčéĞŁè£Źæőţäžčçă ĄïijŇä¡ăéIJĂèęĄä¡£çŤÍäÿĂäžŽåIJÍ opcode æÍąåİŮäÿ■åőŽäźL'çŽĎåÿÿéĞŔãĂĆä¿ŃåęĆïijŽ

```
>>> c = countdown.__code__.co_code
>>> import opcode
>>> opcode.opname[c[0]]
>>> opcode.opname[c[0]]
'SETUP_LOOP'
>>> opcode.opname[c[3]]
'LOAD_FAST'
>>>
```

åěĞæĂłçŽĎæŸŕïijŇåIJÍ dis æÍąåİŮäÿ■åźűæšąæIJL'åĞ¡æŢřèőľä¡ääžěçijŰçÍŃæŰźåijŔå¿Ĺåőźæ ŸŞçŽĎäÿ■è£ĞïijŇäÿŃéİćçŽĎçŤ§æĹŘåŹÍåĞ¡æŢřåŔŕäžěåřĘåŐ§åğŃå■ŮèŁĆçăĄåžŔåĹŮè¡ñæ■ćæĹŘ opcodes åŠŇåŔĆæŢřãĂĆ

```
import opcode

def generate_opcodes(codebytes):
    extended_arg = 0
    i = 0
    n = len(codebytes)
    while i < n:
        op = codebytes[i]
```

```
        i += 1
        if op >= opcode.HAVE_ARGUMENT:
            oparg = codebytes[i] + codebytes[i+1]*256 + extended_arg
            extended_arg = 0
            i += 2
            if op == opcode.EXTENDED_ARG:
                extended_arg = oparg * 65536
                continue
        else:
            oparg = None
        yield (op, oparg)
```

ä¡£çŤÍæŰźæşŢåęĆäÿŃïijŽ

```
>>> for op, oparg in generate_opcodes(countdown.__code__.co_code):
...     print(op, opcode.opname[op], oparg)
```

è£Źçğ■æŰźåijŕå¿ĹåřŚæIJL'äžžç§ěéĄŞïijŇä¡ăåŔŕäżěåĹl'çŤÍåőČæŻ£æ■ćäżä¡Ţä¡äæČşèęĄæŻ£æ■ćçŽĎäÿŃéİćæĹŚäżňçŤÍäÿĂäÿłçd'žä¿ŃæİěæijŤçd'žæŢt'äÿłè£ĞçÍŃïijŽ

```
>>> def add(x, y):
...     return x + y
...
>>> c = add.__code__
>>> c
<code object add at 0x1007beed0, file "<stdin>", line 1>
>>> c.co_code
b'|\x00\x00|\x01\x00\x17S'
>>>
>>> # Make a completely new code object with bogus byte code
>>> import types
>>> newbytecode = b'xxxxxxx'
>>> nc = types.CodeType(c.co_argcount, c.co_kwonlyargcount,
...     c.co_nlocals, c.co_stacksize, c.co_flags, newbytecode, c.co_
 ↪consts,
...     c.co_names, c.co_varnames, c.co_filename, c.co_name,
...     c.co_firstlineno, c.co_lnotab)
>>> nc
<code object add at 0x10069fe40, file "<stdin>", line 1>
>>> add.__code__ = nc
>>> add(2,3)
Segmentation fault
```

ä¡ăåŔŕäżěåČŔè£ŹæăůèĂ■åd'ğæŃŻèől'èğčéĞŁåŹĺåěŤæžČãĂĆä¡ĘæŸŕïijŇårźäžŐçijŰåĘŹæŻt'énŸçžğäi
äżŰäżňåŔŕèČ¡çIJ§çŽĎéIJĂèęĄéĞ■åĘŹå■ŮèŁĆçăĄãĂĆæIJňèŁĆæIJĂåŘŐçŽĎéČĺåĹĘæijŤçd'žäžĘè£ŹäÿłæŸ
this code on ActiveState

12 çňňå■ĄçńäïijŽæĺąåİŮäÿŐåŇĚ

æĺąåİŮäÿŐåŇĚæŸŕäżżä¡Ţåď ğåđŃçĺŃåžŔçŽĎæäÿå£ČïijŇåŕśè£đPythonåőĽèčĚçĺŃåžŔæIJňèžńäź§æŸŕ

Contents:

12.1 10.1 æđĎåżžäÿĂäÿłæĺąåİŮçŽĎåśĆçžğåŇĚ

éŮőéćŸ

ä¡ăæČşåřĘä¡ăçŽĎäżčçăĄçżĎçżĞæĹŘçŤśå¿Ĺåď ŽåĹĘåśĆæĺąåİŮæđĎæĹŘçŽĎåŇĚãĂĆ

èğčåĘşæŰźæąĹ

åřĄèčĚæĹŘåŇĚæŸŕå¿ĹçőĂå■ŢçŽĎãĂĆåIJĺæŰĞäżűçşżçż§äÿŁçżĎçżĞä¡ăçŽĎäżčçăĄïijŇåźűçąőä£İæŕ
ä¿ŃåęĆïijŽ

```
graphics/
    __init__.py
    primitive/
        __init__.py
        line.py
        fill.py
        text.py
    formats/
        __init__.py
        png.py
        jpg.py
```

äÿĂæŮęä¡ăåĄŽåĹřäžĘè£ŹäÿĂçĆźïijŇä¡ăåžŤèŕěèČ¡åď §æĽğèąŇåŘĎçğ■importèŕ■åŔěïijŇåęĆäÿŃïijŽ

```
import graphics.primitive.line
from graphics.primitive import line
import graphics.formats.jpg as jpg
```

èőĺèőž

åőŽäźĽæĺąåİŮçŽĎåśĆæňąçżŞæđĎåřśåČŔåIJĺæŰĞäżűçşżçż§äÿŁåżžçńŃçŻőå¡ŢçżŞæđĎäÿĂæăůåőźæŸ
æŰĞäżű__init__.pyçŽĎçŻőçŽĎæŸŕèęĄåŇĚåŘńäÿ■åŘŇè£ŘèąŇçžğåĹńçŽĎåŇĚçŽĎåŔŕéĂĽçŽĎåĹİåğŃåŇ
äÿ¿äÿłä¿Ńå■ŘïijŇåęĆæđIJä¡ăæĽğèąŇäžĘèŕ■åŔěimport graphicsïijŇ æŰĞäżűgraph-
ics/__init__.pyåřĘèćńåŕijåĚě,åżžçńŃgraphicsåŚ¡åŘ■çľžéŮťçŽĎåĘĚåőźãĂĆåČŔimport
graphics.format.jpgè£ŹæăůåŕijåĚěïijŇæŰĞäżűgraphics/__init__.pyåŠŇæŰĞäżűgraphics/formats/__init__.py

çżİåď ğéČĺåĹĘæŮűåĂŹèőľ__init__.pyçľžçİĂåřśåě¡ãĂĆä¡ĘæŸŕæIJĽäžŻæČĚåĘţäÿŃåŔŕèČ¡åŇĚåŘńäżč
äÿ¿äÿłä¿Ńå■ŘïijŇ__init__.pyèČ¡åď §çŤĺæİěèĞłåŁĺåŁăè¡¡å■ŘæĺąåİŮ:

```
# graphics/formats/__init__.py
from . import jpg
from . import png
```

åČŔè£Źæăůäÿ Ăäÿłæ ŰĞäžű,çŤĺæĹůåŔŕäžëäžĚäžĚéĂŽè£Ğimport graphics.formatsæİěäžčæŻ£import graphics.formats.jpgäžěåŔŁimport graphics.formats.pngãĂĆ

__init__.pyçŽĎåĚűäžŰåÿÿçŤĺçŤĺæşŢåŇĚæŃňåřĘåd'ŽäÿłæŰĞäžűåŘĹåźűåĹřäÿĂäÿłéÄžè¿ŚåŠ¡åŘ■çl'žé

æŢŔéŤŘçŽĎçÍŃåžŔåŚŚŸäijŽåŘŚçŐřïijŇå■şä¡£æšąæIJL'__init__.pyæŰĞäžűå■ŸåIJĺïijŇpythonäž■çĎűä

12.2 10.2 æŐğåĹűæĺąåİŮèćńåĚĺéČĺåŕijåĚěçŽĎåĘĚåőź

éŮőécŸ

å¡Şä¡£çŤĺâĂŹfrom module import *âĂŹ èŕ■åŔěæŮűïijŇåÿŇæIJŻåŕźäžŐæĺąåİŮæĹŰåŇĚåŕijåĞžçŽĎçňę

èğčåĘşæŰźæąĹ

åIJĺä¡äçŽĎæĺąåİŮäÿ■åőŽäźL'äÿĂäÿłåŔŸéĞŔ __all__ æİěæŸŐçąőåIJřåĹŮåĞžéIJĂèęĄåŕijåĞžçŽĎåĘĚå

äÿ¿äÿłä¿Ńå■Ř:

```
# somemodule.py
def spam():
    pass

def grok():
    pass

blah = 42
# Only export 'spam' and 'grok'
__all__ = ['spam', 'grok']
```

èőĺèőž

åř¡çőąåijžçČĹåŔ■åŕźä¡£çŤĺ âĂŸfrom module import *âĂŹ, ä¡ĘæŸŕåIJĺåőŽäźL'äžĘåd'ğéĞŔåŔŸéĞŔåŘ■çŽĎæĺąåİŮäÿ■écŚçźĄä¡£çŤĺãĂĆ åęĆæđIJä¡ääÿ■åĄŽäżżä¡ŢäžŃ, è£ŹæăůçŽĎåŕijåĚěåřĘäijŽåŕijåĚěæL'ĂæIJL'äÿ■äžěäÿŃåĹŠçž£åijĂåd't'çŽĎã åŘęäÿĂæŰźéIć,åęĆæđIJåőŽäźL'äžĘ __all__ , éĆčäźĹåŔłæIJL'èćńåĹŮäÿ¿åĞžçŽĎäÿIJèě£äijŽèćńåŕijåĞžãĂĆ

åęĆæđIJä¡ăåřĘ __all__ åőŽäźL'æĹŘäÿĂäÿłçl'žåĹŮèąĺ, æšąæIJL'äÿIJèě£åřĘèćńåŕijåĚěãĂĆ åęĆæđIJ __all__ åŇĚåŘńæIJłåőŽäźL'çŽĎåŘ■å■Ů, åIJĺåŕijåĚěæŮűåijŢètůAttributeErrorãĂĆ

12.3 10.3 ä¡£çŤĺçŻÿåŕźèůŕå¿ĎåŘ■åŕijåĚěåŇĚäÿ■å■ŘæĺąåİŮ

éŮőécŸ

åřĘäžčçăĄçżĎçżĞæĹŘåŇĚ,æČşçŤĺimportèŕ■åŔěäžŐåŔęäÿĂäÿłåŇĚåŘ■æšąæIJL'çąňçijŰçăĄè£ĞçŽĎå

èğčåĘşæŰźæąĹ

ä¡£çŤĺåŇĚçŽĎçŻÿåŕźåŕijåĚěïijŇä¡£äÿĂäÿłæĺąåİŮåŕijåĚěåŘŇäÿĂäÿłåŇĚçŽĎåŔęäÿĂäÿłæĺąåİŮ äÿ¿äÿłä¿Ńå■ŘïijŇåĄĞèő¿åIJĺä¡ăçŽĎæŰĞäžűçşżçż§äÿŁæIJL'mypackageåŇĚïijŇçżĎçżĞåęĆäÿŇïijŽ

```
mypackage/
    __init__.py
    A/
        __init__.py
        spam.py
        grok.py
    B/
        __init__.py
        bar.py
```

åęĆæđIJæĺąåİŮmypackage.A.spamèęĄåŕijåĚěåŘŇçŻőå¡ŢäÿŇçŽĎæĺąåİŮgrokïijŇåőČåžŤèŕěåŇĚæŃňçŽ

```
# mypackage/A/spam.py
from . import grok
```

åęĆæđIJæĺąåİŮmypackage.A.spamèęĄåŕijåĚěäÿ■åŘŇçŻőå¡ŢäÿŇçŽĎæĺąåİŮB.barïijŇåőČåžŤèŕěä¡£çŤĺ

```
# mypackage/A/spam.py
from ..B import bar
```

äÿd'äÿłimportèŕ■åŔěéČ¡æšąåŇĚåŘńéąűåśĆåŇĚåŘ■ïijŇèĂŇæŸŕä¡£çŤĺäžĘspam.pyçŽĎçŻÿåŕźèůŕå¿ĎãĂ

èőÍèőž

åIJĺåŇĚåĘĚïijŇæŮćåŘŕäžěä¡£çŤĺçŻÿåŕźèůŕå¿Ďäź§åŘŕäžěä¡£çŤĺçżÌåŕźèůŕå¿ĎæÌěåŕijåĚěãĂĆ äÿ¿äÿłä¿Ńå■ŘïijŽ

```
# mypackage/A/spam.py
from mypackage.A import grok # OK
from . import grok # OK
import grok # Error (not found)
```

åČŔmypackage.Aè£Źæăůä¡£çŤĺçżÌåŕźèůŕå¿ĎåŘ■çŽĎäÿ■åĹl'äźŃåd'ĎæŸŕè£ŹåřĘéąűåśĆåŇĚåŘ■çąňçijŮ äÿ¿äÿłä¿Ńå■ŘïijŇåęĆæđIJä¡ăæŤźåŔŸäžĘåŇĚåŘ■ïijŇä¡ăåřśå£ĚéążæčĂæ§ěæL'ĂæIJL'æŰĞäžűæÌěä£őæ■čæž åŘŇæăůïijŇçąňçijŰçăĄçŽĎåŘ■çğřäijŽä¡£çğżåĹĺäžčçăĄåŔŸå¿ŮåŽŕéŽ¿ãĂĆäÿ¿äÿłä¿Ńå■ŘïijŇäź§èőÿæIJL'ä åęĆæđIJä¡£çŤĺçŻÿåŕźåŕijåĚěïijŇéĆčäÿĂåĹĞéČ¡okïijŇçĎűèĂŇä¡£çŤĺçżÌåŕźèůŕå¿ĎåŘ■å¿ĹåŔŕèČ¡äijŽåĞžéŮ

importèŕ■åŔěçŽĎ . åŠŇ . . çIJŃèţůæİěå¿ĹæżŚçĺ¡, ä¡ĘåőČæŇĞåőŽçŻőå¡ŢåŘ■.äÿžå¡ŞåĽ■çŻőå¡ŢïijŇ..BäÿžçŻőå¡Ţ../BãĂĆè£Źçğ■èŕ■æşŢåŔłéĂĆçŤĺäžŐimport äÿ¿äÿłä¿Ńå■ŘïijŽ

```
from . import grok # OK
import .grok # ERROR
```

åř¡çőąä¡£çŤĺçŽÿåŕźåŕijåĚěçIJŃèţůæİěåČŔæŸŕæţŔèğĹæŰĞäžűçşżçż§ïijŇä¡ĘæŸŕäÿ■èČ¡åĹřåőŽäźĽåŇĚ

æIJĂåŘŐïijŇçŽÿåŕźåŕijåĚěåŔłéĂĆçŤĺäžŐåIJĺåŘĹéĂĆçŽĎåŇĚäÿ■çŽĎæĺąåİŮãĂĆård'åĚűæŸŕåIJĺéąűås ä¿ŃåęĆïijŽ

```
% python3 mypackage/A/spam.py # Relative imports fail
```

åŔęäÿĂæŰźéİćïijŇåęĆæđIJä¡ăä¡£çŤĺPythonçŽĎ-méĂĽéąźæİěæĽğèąŇåĚĹåĽ■çŽĎèĎŽæIJñïijŇçŽÿåŕ ä¿ŃåęĆïijŽ

```
% python3 -m mypackage.A.spam # Relative imports work
```

æŻť'åd'ŽçŽĎåŇĚçŽĎçŽÿåŕźåŕijåĚěçŽĎèČŇæŹŕç§ěèŕE,èŕůçIJŃ PEP 328 .

12.4 10.4 åřĘæĺąåİŮåĹĘåĽšæĹŘåd'Žäÿłæ ŰĞäžű

éŮőécŸ

ä¡ăæČşåřĘäÿĂäÿłæĺąåİŮåĹĘåĽšæĹŘåd'Žäÿłæ ŰĞäžűãĂĆä¡ĘæŸŕä¡ăäÿ■æČşåřĘåĹĘçężçŽĎæŰĞäžűçż

èğčåĘşæŰźæąĹ

çĺŃåžŔæĺąåİŮåŔŕäžěéĂŽè£ĞåŔŸæĹŘåŇĚæİěåĹĘåĽšæĹŘåd'ŽäÿłçŃñçńŃçŽĎæŰĞäžűãĂĆèĂČèŹŚäÿ

```
# mymodule.py
class A:
    def spam(self):
        print('A.spam')

class B(A):
    def bar(self):
        print('B.bar')
```

åĄĞèő¿ä¡ăæČşmymodule.pyåĹĘäÿžäÿd'äÿłæŰĞäžűïijŇæŕŔäÿłåőŽäźĽçŽĎäÿĂäÿłçśžãĂĆèęĄåĄŽåĹřè è£Źè£ŹäÿłçŻőå¡ŢäÿŃïijŇåĹŻåžžäžěäÿŃæŰĞäžűïijŽ

```
mymodule/
    __init__.py
    a.py
    b.py
```

åIJÍa.pyæŰĞäžűäÿ■æŔŠåĚěäžěäÿŃäžčçăĄïijŽ

```
# a.py
class A:
    def spam(self):
        print('A.spam')
```

åIJÍb.pyæŰĞäžűäÿ■æŔŠåĚěäžěäÿŃäžčçăĄïijŽ

```
# b.py
from .a import A
class B(A):
    def bar(self):
        print('B.bar')
```

æIJĂåŘŐïijŇåIJÍ __init__.py äÿ■ïijŇářĘ2äÿłæŰĞäžűçšŸåŘĹåIJÍäÿĂèţůïijŽ

```
# __init__.py
from .a import A
from .b import B
```

åęĆæđIJæŇL'çĚğè£ŹäžŻæ■ěéłd'ïijŇæL'Ăäžğç Ť§çŽĎåŇĚMyModuleářĘä¡IJäÿžäÿĂäÿłå■ŢäÿĂçŽĎéĂż

```
>>> import mymodule
>>> a = mymodule.A()
>>> a.spam()
A.spam
>>> b = mymodule.B()
>>> b.bar()
B.bar
>>>
```

èőÍèőž

åIJÍè£Źäÿłçńăè ŁĆäÿ■çŽĎäÿżèęĄéŮőécŸæŸŕäÿĂäÿłèő¿èőąéŮőécŸïijŇäÿ■çőąä¡ăæŸŕåŘęåÿŇæIJŻçŤÍæ

```
from mymodule.a import A
from mymodule.b import B
...
```

è£ŹæăůèČ¡åůěä¡IJïijŇä¡Ęè£ŹèőľçŤÍæĹůæL'£åŔŮæŻť'åd'ŽçŽĎèť§æŃĚïijŇçŤÍæĹůèęĄç§ěéĄŞäÿ■åŘŇ

```
from mymodule import A, B
```

åŕźåŘŐèĂĚèĂŇèÍĂïijŇèőľ'mymoduleæĹŘäÿžäÿĂäÿłåd'ğçŽĎæžŘæŰĞäžűæŸŕæIJĂåÿÿèğĄçŽĎãĂĆä¡Ę
è£ŹæăůåĄŽçŽĎåĚşéŤőæŸŕåĹŻåžžäÿĂäÿłåŇĚçŻőå¡ŢïijŇä¡£çŤÍ __init__.py
æŰĞäžűæÌěåřĘæŕŔéČÍåĹĘçšŸåŘĹåIJÍäÿĂèţůãĂĆ

å¡ŞäÿĂäÿłæÍąåÌŮèćńåĹĘåL'šïijŇä¡äéIJĂèęĄçL'źåĹńæşÍæĎŔäžd'åŔL'åijŢçŤÍçŽĎæŰĞäžűåŘ■ãĂĆäÿ¿äÿ
from .a import A æÌěèŐůåŔŰãĂĆ

æŢť äÿłçńäèŁĆéČ¡ä¡£çŤĺåŇĚçŽĎçŻÿåŕźåřijåĚěæİěéĄ£åĚ■åřĘéąűåśĆæĺąåİŮåŘ■çąňçijŰçăĄåĹřæžŘäżč

ä¡IJäÿžè£ŹäÿĂçńäèŁĆçŽĎåżűäijÿïijŇåřĘäżŃçż■åżűè£§åŕijåĚěãĂĆåęĆåŻ¿æL’Ăçd’žïijŇ__init__.pyæŰ
èęĄåĄŽåĹřè£ŹäÿĂçĆźïijŇ__init__.pyæIJL’çżĘå¿őçŽĎåŔŸåŇŰïijŽ

```
# __init__.py
def A():
    from .a import A
    return A()

def B():
    from .b import B
    return B()
```

åIJÍè£ŹäÿłçL’ĹæIJñäÿ■ïijŇçśżAåŠŇçśżBèćńæŻ£æ■ćäÿžåIJÍçňňäÿĂæňąèő£éŮőæŮűåŁăè¡¡æL’ĂéIJĂçŽĺ
ä¿ŃåęĆïijŽ

```
>>> import mymodule
>>> a = mymodule.A()
>>> a.spam()
A.spam
>>>
```

åžűè£§åŁăè¡¡çŽĎäÿżèęĄçijžçĆźæŸŕçżğæL’£åŠŇçśżåđŃæčĂæ§ěåŔŕèČ¡äijŽäÿ■æŰ■ãĂĆä¡ăåŔŕèČ¡äijŽ

```
if isinstance(x, mymodule.A): # Error
...

if isinstance(x, mymodule.a.A): # Ok
...
```

åžűè£§åŁăè¡¡çŽĎçIJ§åőđä¿Ńå■Ř, èğĄæăĞåĞĘåžŞ multiprocessing/__init__.py
çŽĎæžŘçăĄ.

12.5 10.5 åĹľçŤĺåŚ¡åŘ■çľžéŮťåŕijåĚěçŻőå¡ŢåĹĘæŢčçŽĎäżčçăĄ

éŮőécŸ

ä¡ăåŔŕèČ¡æIJL’åd’ğéĞŔçŽĎäżčçăĄïijŇçŤśäÿ■åŘŇçŽĎäžžæİěåĹĘæŢčåIJřçżt’æŁd’ãĂĆæŕŔäÿłéČĺåĹĘè

èğčåĘşæŰźæąĹ

äżŐæIJñèt’ĺäÿŁèőšïijŇä¡ăèęĄåőŽäźL’äÿĂäÿłéąűçžğPythonåŇĚïijŇä¡IJäÿžäÿĂäÿłåd’ğéŻĘåŘĹåĹĘåijĂç

åIJÍçż§äÿĂäÿ■åŘŇçŽĎçŻőå¡ŢéĞŇçż§äÿĂçŻÿåŘŇçŽĎåŚ¡åŘ■çľžéŮťïijŇä¡ĘæŸŕèęĄåĹăåŐżçŤĺæİěåřĿ

```
foo-package/
    spam/
        blah.py
```

```
bar-package/
    spam/
        grok.py
```

åIJÍè£Ź2äÿłçŻőå¡ŢéĞŇïijŇéČ¡æIJL’çİĂåĚśåŘŇçŽĎåŚ¡åŘ■çl’žéŮt’spamãĂĆåIJÍäżżä¡ŢäÿĂäÿłçŻőå¡ŢéC

èől’æĹŚäżñçIJŃçIJŃïijŇåęĆæđIJåřĘfoo-packageåŠŇbar-packageéČ¡åŁăåĹřpythonæÍąåİŮèůřå¿ĎåźűåřİèŕŢářijåĚëäijŽåŘŚçŤ§äżĂäźĹ

```
>>> import sys
>>> sys.path.extend(['foo-package', 'bar-package'])
>>> import spam.blah
>>> import spam.grok
>>>
```

äÿd’äÿłäÿ■åŘŇçŽĎåŇĚçŻőå¡ŢèćńåŕĹåźűåĹřäÿĂèţůïijŇä¡ääŘŕäżěåŕijåĚěspam.blahåŠŇspam.grokïijŇå

èőÍèőž

åIJÍè£ŹéĞŇåůëä¡IJçŽĎæIJžåĹűèćńçğřäÿžâĂIJåŇĚåŚ¡åŘ■çl’žéŮt’âĂİçŽĎäÿĂäÿłçL’źå¿ĄãĂĆäżŐæIJňè

åŇĚåŚ¡åŘ■çl’žéŮt’çŽĎåĚşéŤőæŸŕçąőä£ÍéąűçžğçŻőå¡Ţäÿ■ęšąæIJL’__init__.pyæŰĞäżűæİëä¡IJäÿžåĚśäÿ¿äÿłä¿Ńå■ŘïijŽ

```
>>> import spam
>>> spam.__path__
_NamespacePath(['foo-package/spam', 'bar-package/spam'])
>>>
```

åIJÍåőŽä¡■åŇĚçŽĎå■ŘçżĎäżűæŮűïijŇçŻőå¡Ţ__path__åřĘèćńçŤÍåĹř(ä¿ŃåęĆ, å¡ŞåŕijåĚěspam.grokæĹŰèĂĚspam.blahçŽĎæŮűåĂŹ).

åŇĚåŚ¡åŘ■çl’žéŮt’çŽĎäÿĂäÿłéĞ■èęĄçL’źçĆźæŸŕäżżä¡ŢäžžéČ¡åŕäżěçŤÍèĞłåůśçŽĎäżčçăĄæİěæL’l’åś

```
my-package/
    spam/
        custom.py
```

åęĆæđIJä¡ăåřĘä¡ăçŽĎäżčçăĄçŻőå¡ŢåŠŇåĚűäżŰåŇĚäÿĂèţůæůżåŁăåĹřsys.pathïijŇè£ŹåřĘæŮăçijİåIJřå

```
>>> import spam.custom
>>> import spam.grok
>>> import spam.blah
>>>
```

äÿĂäÿłåŇĚæŸŕåŘęèćńä¡IJäÿžäÿĂäÿłåŇĚåŚ¡åŘ■çl’žéŮt’çŽĎäÿżèęĄæŰźæşŢæŸŕæčĂæ§ěåĚű__file__åś

```
>>> spam.__file__
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
```

```
AttributeError: 'module' object has no attribute '__file__'
>>> spam
<module 'spam' (namespace)>
>>>
```

æŻť åď ŽçŽĎåŇĚåŚ¡åŘ■çľ žéŮť ä£ąæĄŕåŔŕäżěæ§ěçIJŃ PEP 420.

12.6 10.6 éĞ■æŰřåŁăè¡¡æĺąåİŮ

éŮőéćŸ

ä¡ăæČşéĞ■æŰřåŁăè¡¡åůšçżŔåŁăè¡¡çŽĎæĺąåİŮïijŇåŻăäÿžä¡ăåŕźåĚűæžŘçăĄè£ŻèąŇäžĘä£őæŤźãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤĺimp.reload()æİěéĞ■æŰřåŁăè¡¡åĚĹåĽ■åŁăè¡¡çŽĎæĺąåİŮãĂĆäÿ¿äÿłä¿Ńå■ŘïijŽ

```
>>> import spam
>>> import imp
>>> imp.reload(spam)
<module 'spam' from './spam.py'>
>>>
```

èőĺèőž

éĞ■æŰřåŁăè¡¡æĺąåİŮåIJĺåijĂåŔŚåŠŇèřČèŕŢè£ĞçĺŃäÿ■åÿÿåÿÿå¿ĹæIJĽçŤĺãĂĆä¡ĘåIJĺçŤ§äžğçŐŕåćČä

reload()æŞęéŹď äžĘæĺąåİŮåžŢåśĆå■ŮåĚÿçŽĎåĘĚåőźïijŇåźűéĂŽè£ĞéĞ■æŰřæĽğèąŇæĺąåİŮçŽĎæžŘ

åř¡çőąåęĆæ■ď ïijŇreload()æšąæIJĽæŻť æŰřåČŔâĂİfrom module import nameâĂİè£Źæăůä¡£çŤĺimportèŕ■åŔěåŕijåĚěçŽĎåőŽäźĽãĂĆäÿ¿äÿłä¿Ńå■ŘïijŽ

```
# spam.py
def bar():
    print('bar')

def grok():
    print('grok')
```

çŐřåIJĺåŘŕåŁĺäžď äžŠåijŔäijŽèŕİïijŽ

```
>>> import spam
>>> from spam import grok
>>> spam.bar()
bar
>>> grok()
```

```
grok
>>>
```

äÿ■éĂĂåĞžPythonä£őæŤźspam.pyçŽĎæžŘçăĄïijŇářĘgrok()åĞ¡æŢřæŤźæĹŘè£ŹæăůïijŽ

```
def grok():
    print('New grok')
```

çŐřåIJĺåŻđåĹřäžď'äžŠåijŔäijŽèŕİïijŇéĞ■æŰřåŁäè¡¡æĺąåİŮïijŇářİèŕŢäÿŃè£ŹäÿłåőđéłŇïijŽ

```
>>> import imp
>>> imp.reload(spam)
<module 'spam' from './spam.py'>
>>> spam.bar()
bar
>>> grok() # Notice old output
grok
>>> spam.grok() # Notice new output
New grok
>>>
```

åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇä¡ăçIJŃåĹřæIJL'2äÿłçL'ĹæIJňçŽĎgrok()åĞ¡æŢřèćńåŁäè¡¡ãĂĆéĂŽåÿÿæİèèŕť'ï

åŻäæ■d'ïijŇåIJÍçŤ§äžğçŐŕáćČäÿ■åŔŕèČ¡éIJĂèęĄéĄ£åĚ■éĞ■æŰřåŁäè¡¡æĺąåİŮãĂĆåIJÍäžď'äžŠçŐŕáćČ

12.7 10.7 è£ŘèąŇçŻőå¡ŢæĹŰåŐŃçijľæŰĞäžű

éŮőécŸ

æĆÍæIJL'äÿĂäÿłåůšæĹŘéŢ£äÿžåŇĚåŘńåď'Žäÿłæ ŰĞäżűçŽĎåžŤçŤÍïijŇåőČåůšè£IJäÿ■åĘ■æŸŕäÿĂäÿłçő

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăçŽĎåžŤçŤÍçÍŃåžŔåůšçżŔæIJL'åď'Žäÿłæ ŰĞäżűïijŇä¡ăåŔŕäžěæŁŁä¡ăçŽĎåžŤçŤÍçÍŃåžŔæŤ¿
äÿ¿äÿłä¿Ńå■ŘïijŇä¡ăåŔŕäžěåČŔè£ŹæăůåĹŻåžžçŻőå¡Ţïijž

```
myapplication/
    spam.py
    bar.py
    grok.py
    __main__.py
```

åęĆædIJ__main__.pyå■ŸåIJÍïijŇä¡ăåŔŕäžěçőĂå■ŢåIJřåIJÍéąűçžğçŻőå¡Ţè£ŘèąŇPythonèğčéĞŁåŹÍïijž

```
bash % python3 myapplication
```

èğčéĞŁåŹÍåřĘæL'ğèąŇ__main__.pyæŰĞäżűä¡IJäÿžäÿżçÍŃåžŔãĂĆ

åęĆædIJä¡ăåřĘä¡ăçŽĎäžčçăĄæL'ŞåŇĚæĹŘzipæŰĞäżűïijŇè£Źçğ■æŁĂæIJŕåŘŇæăůäź§éĂĆçŤÍïijŇäÿ¿ă

```
bash % ls
spam.py bar.py grok.py __main__.py
bash % zip -r myapp.zip *.py
bash % python3 myapp.zip
... output from __main__.py ...
```

èőÍèőž

åĹŻåžžäÿĂäÿłçŻőå¡ŢæĹŰzipæŰĞäżűåźűæůżåŁă__main__.pyæŰĞäżűæİěářĘäÿĂäÿłæŻť åď ğçŽĎPytho

çŤśäžŐçŻőå¡ŢåŠŇzipæŰĞäżűäÿŐæ■čåÿÿæŰĞäżűæIJL’äÿĂçĆźäÿ■åŘŇïijŇä¡ääŔřèČ¡è£ŸéIJĂèęĄáćđå

```
#!/usr/bin/env python3 /usr/local/bin/myapp.zip
```

12.8 10.8 èŕżåŔŰä¡■äžŐåŇĚäÿ■çŽĎæŢřæ■őæŰĞäżű

éŮőécŸ

ä¡ăçŽĎåŇĚäÿ■åŇĚåŘńäžčçăĄéIJĂèęĄåŐżèŕżåŔŰçŽĎæŢřæ■őæŰĞäżűãĂĆä¡ăéIJĂèęĄåř¡åŔřèČ¡åIJřçŢ

èğčåĘşæŰźæąĹ

åĄĞèő¿ä¡ăçŽĎåŇĚäÿ■çŽĎæŰĞäżűçżĎçżĞæĹŘåęĆäÿŃïijŽ

```
mypackage/
    __init__.py
    somedata.dat
    spam.py
```

çŐřåIJĺåĄĞèő¿spam.pyæŰĞäżűéIJĂèęĄèŕżåŔŰsomedata.datæŰĞäżűäÿ■çŽĎåĘĚåőźãĂĆä¡ăåŘŕäžěçŢĺà

```
# spam.py
import pkgutil
data = pkgutil.get_data(__package__, 'somedata.dat')
```

çŤśæ■d’äžğçŢ§çŽĎåŔŸéĞŔæŸŕåŇĚåŘńèřěæŰĞäżűçŽĎåŐ§ågŃåĘĚåőźçŽĎå■ŮèŁĆå■Ůçňęäÿšã ĂĆ

èőÍèőž

èęĄèŕżåŔŰæŢřæ■őæŰĞäżűïijŇä¡ääŔřèČ¡äijŽåĂ¿åŘŚäžŐçijŰåĘŹä¡£çŢĺåĘĚç¡őçŽĎI/
OåŁ§èČ¡çŽĎäžčçăĄïijŇåęĆopen()ãĂĆä¡ĘæŸřè£Źçğ■æŰźæşŢäź§æIJL’äÿĂäžŻéŮőécŸãĂĆ

éęŰåĚĹïijŇäÿĂäÿłåŇĚåŕźèğčéĞŁåŹĺçŽĎå¡ŞåL’■åůëä¡IJçŻőå¡ŢåĞääźŐæšąæIJL’æŐğåĹűæİČãĂĆåŻăæ

çňňäžŇïijŇåŇĚéĂŽåÿÿåőL’èčĚä¡IJäÿž.zipæĹŰ.eggæŰĞäżűïijŇè£ŹäžŻæŰĞäżűåźűäÿ■åČŔåIJĺæŰĞäżű

pkgutil.get_data()åĞ¡æŢřæŸŕäÿĂäÿłèŕżåŔŰæŢřæ■őæŰĞäžűçŽĎéńŸçžĝåůěåĚůïijŇäÿ■çŤĺçőąåŇĚæŸŕ

get_data()çŽĎçňňäÿĂäÿłåŔĆæŢřæŸŕåŇĚåŘńåŇĚåŘ■çŽĎå■Ůçňęäÿšã ĂĆä¡ăåŔŕäżěçŻť'æŐëä¡£çŤĺåŇĚ

12.9 10.9 åřĘæŰĞäžűåď'źåŁăåĚěåĹřsys.path

éŮőécŸ

ä¡ăæŮăæşŢåŕijåĚëä¡ăçŽĎPythonäżčçăĄåŻăäÿžåőČæĽ'ĂåIJĺçŽĎçŻőå¡Ţäÿ■åIJĺsys.pathéĞŇãĂĆä¡ăæČş

èğčåĘşæŰźæąĹ

æIJL'äÿd'çğ■åÿÿçŤĺçŽĎæŰźåijŔåřĘæŰřçŻőå¡Ţæůżå Łăå Ĺřsys.pathãĂĆçňňäÿĂçğ■ïijŇä¡ăåŔŕäżěä¡£çŤĺ

```
bash % env PYTHONPATH=/some/dir:/other/dir python3
Python 3.3.0 (default, Oct 4 2012, 10:17:33)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more␣
↪information.
>>> import sys
>>> sys.path
['', '/some/dir', '/other/dir', ...]
>>>
```

åIJĺèĞłåőŽäźL'ăžŤçŤĺçĺŃåžŔäÿ■ïijŇè£ŹæăůçŽĎçŐŕåćČåŔŸéĞŔåŔŕåIJĺçĺŃåžŔåŔŕåŁĺæŮűèő¿ç¡őæĹŰ

çňňäžŇçğ■æŰźæşŢæŸŕåĹŻåžžäÿĂäÿł.pthæŰĞäžűïijŇåřĘçŻőå¡ŢåĹŮäÿ¿åĞžæÌěïijŇåČŔè£Źæăůïijž

```
# myapplication.pth
/some/dir
/other/dir
```

è£Źäÿł.pthæŰĞäžűéIJĂèęĄæŤ¿åIJĺæ§ŘäÿłPythonçŽĎsite-packagesçŻőå¡ŢïijŇéĂŽåÿÿä¡■äžŐ/usr/local/lib/python3.3/site-packages æĹŰèĂĚ ~/.local/lib/python3.3/sitepackagesãĂĆå¡Şèğčé ĞŁåŹĺåŘŕåŁĺæŮűïijŇ.pthæŰĞäžűéĞŇåĹŮäÿ¿åĞžæÌěçŽĎå■ŸåIJ

èőĺèőž

æŕŤèţůèť źåŁŻåIJŕæĽ¿æŰĞäžűïijŇä¡ăåŔŕèČ¡äijŽåĂ¿åŘŚäžŐåĘŹäÿĂäÿłäžčçăĄæĽŃåŁĺèřČèŁĆsys.pat

```
import sys
sys.path.insert(0, '/some/dir')
sys.path.insert(0, '/other/dir')
```

èŹ¡çĎűè£ŹèČ¡âĂIJåůëä¡IJâĂÌïijŇåőČæŸŕåIJĺåőđèůţäÿ■æđĄäÿžèĎĘåijśïijŇäžŤår¡éĞŔéĄ£åĚ■ä¡£çŤĺã

```
import sys
from os.path import abspath, join, dirname
sys.path.insert(0, join(abspath(dirname(__file__)), 'src'))
```

è£ŹåřĘsrcçŻőå¡Ţæůżå£ăåĹřpathéĞŇïijŇåŠŇæĽğèąŇæŔŠåĚèæ■ěéłd'çŽĎäžčçăĄåIJĺåŘŇäÿĂäÿłçŻőå¡T

site-packagesçŻőå¡ŢæŸŕçňňäÿL'æŰźåŇĚåŠŇæĺąåİŮåőL'èčĚçŽĎçŻőå¡ŢãĂĆåęĆæđIJä¡ăæL'ŃåŁĺåőL'èč
packagesçŻőå¡ŢãĂĆèŹ¡çĎűçŤĺäžŐéĚ■ç¡őpathçŽĎ.pthæŰĞäžűå£ĚéążæŤ¿ç¡őåIJĺsite-
packageséĞŇïijŇä¡ĘåőČéĚ■ç¡őçŽĎèůřå¿ĎåŘŕäžěæŸŕçşżçż§äÿŁäżżä¡Ţä¡ăåÿŇæIJŻçŽĎçŻőå¡ŢãĂĆåŻăæ■d'

12.10 10.10 éĂŽè£Ğå■Ůçňęäÿšå Ř■åŕijåĚěæĺąåİŮ

éŮőécŸ

ä¡ăæČşåŕijåĚěäÿĂäÿłæĺąåİŮïijŇä¡ĘæŸŕæĺąåİŮçŽĎåŘ■å■ŮåIJĺå■Ůçňęäÿšé ĞŇãĂĆä¡ăæČşåŕźå■Ůçňęäÿ

èğčåĘşæŰźæąĹ

ä¡£çŤĺimportlib.import_module()åĞ¡æŢřæİěæĽŃåŁĺåŕijåĚěåŘ■å■Ůäÿžå■ŮçňęäÿšçżŹåĞžçŽĎäÿĂäÿłæ

```
>>> import importlib
>>> math = importlib.import_module('math')
>>> math.sin(2)
0.9092974268256817
>>> mod = importlib.import_module('urllib.request')
>>> u = mod.urlopen('http://www.python.org')
>>>
```

import_moduleåŘłæŸŕçőĂå■ŢåIJřæĽğèąŇåŠŇimportçŻÿåŘŇçŽĎæ■ěéłd'ïijŇä¡ĘæŸŕè£ŤåŻđçŤ§æĹŘç

åęĆæđIJä¡ăæ■čåIJĺä¡£çŤĺçŽĎåŇĚïijŇimport_module()äź§åŘŕçŤĺäžŐçŻÿåŕźåŕijåĚěãĂĆä¡ĘæŸŕïijŇä¡äé

```
import importlib
# Same as 'from . import b'
b = importlib.import_module('.b', __package__)
```

èőĺèőž

ä¡£çŤĺimport_module()æĽŃåŁĺåŕijåĚěæĺąåİŮçŽĎéŮőécŸéĂŽåÿÿåĞžçŐřåIJĺäžěæ§ŘçğæŰźåijŔçijŰå

åIJĺæŮğçŽĎäžčçăĄïijŇæIJL'æŮűä¡ääijŽçIJŃåĹřçŤĺäžŐåŕijåĚěçŽĎåĘĚåžžåĞ¡æŢř__import__()ãĂĆår¡ç
éĂŽåÿÿæŻt'åőźæŸŞä¡£çŤĺãĂĆ

èĞłåőŽäźL'åŕijåĚěè£ĞçÍŃçŽĎénŸçžğåőđä¿ŃèğĄ10.11åřŔèŁĆ

12.11 10.11 éĂŽè£ĞéŠľå■Řè£lJçÍŃåŁăè¡¡æÍąåİŮ

éŮőécŸ

ä¡ăæČşèĞłåőŽäźĽPythonçŽĎimportèŕ■åŔëïijŇä¡£å¿ŮåőČèČ¡äžŐè£IJçÍŃæIJžåŹÍäÿŁéÍćéĂŔæŸŐçŽĎå

èğčåĘşæŰźæąĹ

éęŰåĚĹèęĄæŔŘåĞžæİěçŽĎæŸŕåőĽåĚÍéŮőécŸãĂĆæIJňèŁĆèőÍèőžçŽĎæĂİæČşåęĆæđIJæšąæIJL'äÿĂ
äź§åřśæŸŕèŕť'ïijŇæĹŚäžňçŽĎäÿżèęĄçŻőçŽĎæŸŕæůśåĚěåĹĘæđŘPythonçŽĎimportèŕ■åŔěæIJžåĹűãĂĆ
åęĆæđIJä¡ăçŘĘèğčäžĘæIJňèŁĆåĘĚéČÍåŐ§çŘĘïijŇä¡ăåřśèČ¡åd'§äÿžåĚűäżŰäżżä¡ŢçŻőçŽĎèĂŇèĞłåőŽäźĽi
æIJL'äžĘè£ŹäžŻïijŇèőľ'æĹŚäžňçżğçż■åŘŚåL'■èţřãĂĆ

æIJňèŁĆæäÿå£ČæŸŕèő¿èőąåŕijåĚěèŕ■åŔěçŽĎæL'ľåśŢåŁ§èČ¡ãĂĆæIJL'å¿Ĺåd'Žçğ■æŰźæşŢåŔŕäżěåĄŽ
äÿ■è£ĞäÿžäžĘæijŤçd'žçŽĎæŰźä¿£ïijŇæĹŚäžňåijĂăğŃåĚĹæđĎéĂääÿŃéİćè£ŹäÿłPythonäžčçăĄçżŞæđĎïijŽ

```
testcode/
    spam.py
    fib.py
    grok/
        __init__.py
        blah.py
```

è£ŹäžŻæŰĞäżűçŽĎåĘĚåőźåźűäÿ■éĞ■èęĄïijŇäÿ■è£ĞæĹŚäžňåIJÍæŕŔäÿłæŰĞäżűäÿ■æŤ¿åĚěäžĘåřŚéĞ
è£Źæăůä¡ăåŔŕäżěæţŃèŕŢåőČäžňåźűæ§ěçIJŃå¡ŞåőČäžňèćńåŕijåĚěæŮűçŽĎè¿ŞåĞžãĂĆä¿ŃåęĆïijŽ

```
# spam.py
print("I'm spam")

def hello(name):
    print('Hello %s' % name)

# fib.py
print("I'm fib")

def fib(n):
    if n < 2:
        return 1
    else:
        return fib(n-1) + fib(n-2)

# grok/__init__.py
print("I'm grok.__init__")

# grok/blah.py
print("I'm grok.blah")
```

è£ŹéĞŇçŽĎçŽőçŽĎæŸŕåĚĄèőÿè£ŹäžŻæŰĞäżűä¡IJäÿžæÍąåİŮèćńè£IJçÍŃèő£éŮőãĂĆ
äź§èőÿæIJĂçőĂå■ŢçŽĎæŰźåijŔåřśæŸŕårĘåőČäžňåŔŚåÿČåĹřäÿĂäÿłwebæIJ■åŁąåŹÍäÿŁéİćãĂĆåIJÍtestcode

```
bash % cd testcode
bash % python3 -m http.server 15000
Serving HTTP on 0.0.0.0 port 15000 ...
```

æIJ■åŁąåŹÍè£ŘèąŇètůæİěåŘŐåĘ■åŘŕåŁÍäÿĂäÿłå■ŢçŃňçŽĎPythonèğčéĞŁåŹÍãĂĆçąőä£İä¡ăåŔŕäžěä¡£çŤÍ urllib èő£éŮőåĹřè£IJçÍŇæŰĞäžűãĂĆä¿ŃåęĆïijŽ

```
>>> from urllib.request import urlopen
>>> u = urlopen('http://localhost:15000/fib.py')
>>> data = u.read().decode('utf-8')
>>> print(data)
# fib.py
print("I'm fib")

def fib(n):
    if n < 2:
        return 1
    else:
        return fib(n-1) + fib(n-2)
>>>
```

äžŐè£ŹäÿłæIJ■åŁąåŹÍåŁăè¡¡æžŘäžčçăĄæŸŕæŐěäÿŃæİěæIJňèŁĆçŽĎå§žçąĂãĂĆäÿžäžĘæŻ£äžčæL'ŃåŁÍçŽĎéĂŽè£Ğ urlopen() æİěæŤűéŻĘæžŘæŰĞäžűïijŇæĹŚäžňéĂŽè£ĞèĞłåőŽäźL'importèŕ■åŔěæİěåIJÍåŘŐåŔřèĞłåŁÍåÿőæĹŚäžňåĄŽåĹřãĂĆ

åŁăè¡¡è£IJçÍŇæÍąåİŮçŽĎçňňäÿĂçğ■æŰźæşŢæŸŕåĹŻåžžäÿĂäÿłæŸ¿çd'žçŽĎåŁăè¡¡åĞ¡æŢřæİěåőŇæĹŘ

```
import imp
import urllib.request
import sys

def load_module(url):
    u = urllib.request.urlopen(url)
    source = u.read().decode('utf-8')
    mod = sys.modules.setdefault(url, imp.new_module(url))
    code = compile(source, url, 'exec')
    mod.__file__ = url
    mod.__package__ = ''
    exec(code, mod.__dict__)
    return mod
```

è£ŹäÿłåĞ¡æŢřäijŽäÿŃè¡¡æžŘäžčçăĄïijŇåźűä¡£çŤÍ compile() åřĘåĚűçijŰèŕŠåĹřäÿĂäÿłäžčçăĄåŕźèśąäÿ■ïijŇ çĎűåŘŐåIJÍäÿĂäÿłæŰřåĹŻåžžçŽĎæÍąåİŮåŕźèśąçŽĎå■ŮåĚÿä

```
>>> fib = load_module('http://localhost:15000/fib.py')
I'm fib
>>> fib.fib(10)
89
>>> spam = load_module('http://localhost:15000/spam.py')
I'm spam
>>> spam.hello('Guido')
```

```
Hello Guido
>>> fib
<module 'http://localhost:15000/fib.py' from 'http://
 ↪localhost:15000/fib.py'>
>>> spam
<module 'http://localhost:15000/spam.py' from 'http://
 ↪localhost:15000/spam.py'>
>>>
```

æ■čåęĆä¡ăæĽĂèğĄïijŇåŕźäžŐçőĂå■ŢçŽĎæĺąåİŮè£ŹäÿłæŸŕèąŇå¿ŮéĂŽçŽĎãĂĆ
äÿ■è£ĞåőČåźűæšąæIJĽåţŇåĚěåĹřéĂŽåÿÿçŽĎimportèŕ■åŔěäÿ■ïijŇåęĆæđIJèęĄæŤŕæŇĄæŻť'éńŸçžğçŽĎçż

äÿĂäÿłæŻť'éĚůçŽĎåĄŽæşŢæŸŕåĹŽåżžäÿĂäÿłèĞłåőŽäźĽ'åŕijåĚěåŹĺãĂĆçňňäÿĂçğ■æŰźæşŢæŸŕåĹŽåż

```
# urlimport.py
import sys
import importlib.abc
import imp
from urllib.request import urlopen
from urllib.error import HTTPError, URLError
from html.parser import HTMLParser

# Debugging
import logging
log = logging.getLogger(__name__)

# Get links from a given URL
def _get_links(url):
    class LinkParser(HTMLParser):
        def handle_starttag(self, tag, attrs):
            if tag == 'a':
                attrs = dict(attrs)
                links.add(attrs.get('href').rstrip('/'))
    links = set()
    try:
        log.debug('Getting links from %s' % url)
        u = urlopen(url)
        parser = LinkParser()
        parser.feed(u.read().decode('utf-8'))
    except Exception as e:
        log.debug('Could not get links. %s', e)
    log.debug('links: %r', links)
    return links

class UrlMetaFinder(importlib.abc.MetaPathFinder):
    def __init__(self, baseurl):
        self._baseurl = baseurl
        self._links = { }
        self._loaders = { baseurl : UrlModuleLoader(baseurl) }

    def find_module(self, fullname, path=None):
```

```
        log.debug('find_module: fullname=%r, path=%r', fullname,␣
↪path)
        if path is None:
            baseurl = self._baseurl
        else:
            if not path[0].startswith(self._baseurl):
                return None
            baseurl = path[0]
        parts = fullname.split('.')
        basename = parts[-1]
        log.debug('find_module: baseurl=%r, basename=%r', baseurl,␣
↪basename)

        # Check link cache
        if basename not in self._links:
            self._links[baseurl] = _get_links(baseurl)

        # Check if it's a package
        if basename in self._links[baseurl]:
            log.debug('find_module: trying package %r', fullname)
            fullurl = self._baseurl + '/' + basename
            # Attempt to load the package (which accesses __init__.
↪py)
            loader = UrlPackageLoader(fullurl)
            try:
                loader.load_module(fullname)
                self._links[fullurl] = _get_links(fullurl)
                self._loaders[fullurl] = UrlModuleLoader(fullurl)
                log.debug('find_module: package %r loaded',␣
↪fullname)
            except ImportError as e:
                log.debug('find_module: package failed. %s', e)
                loader = None
            return loader
        # A normal module
        filename = basename + '.py'
        if filename in self._links[baseurl]:
            log.debug('find_module: module %r found', fullname)
            return self._loaders[baseurl]
        else:
            log.debug('find_module: module %r not found', fullname)
            return None

    def invalidate_caches(self):
        log.debug('invalidating link cache')
        self._links.clear()

# Module Loader for a URL
class UrlModuleLoader(importlib.abc.SourceLoader):
    def __init__(self, baseurl):
```

```
        self._baseurl = baseurl
        self._source_cache = {}

    def module_repr(self, module):
        return '<urlmodule %r from %r>' % (module.__name__, module._
↪_file__)

    # Required method
    def load_module(self, fullname):
        code = self.get_code(fullname)
        mod = sys.modules.setdefault(fullname, imp.new_
↪module(fullname))
        mod.__file__ = self.get_filename(fullname)
        mod.__loader__ = self
        mod.__package__ = fullname.rpartition('.')[0]
        exec(code, mod.__dict__)
        return mod

    # Optional extensions
    def get_code(self, fullname):
        src = self.get_source(fullname)
        return compile(src, self.get_filename(fullname), 'exec')

    def get_data(self, path):
        pass

    def get_filename(self, fullname):
        return self._baseurl + '/' + fullname.split('.')[-1] + '.py'

    def get_source(self, fullname):
        filename = self.get_filename(fullname)
        log.debug('loader: reading %r', filename)
        if filename in self._source_cache:
            log.debug('loader: cached %r', filename)
            return self._source_cache[filename]
        try:
            u = urlopen(filename)
            source = u.read().decode('utf-8')
            log.debug('loader: %r loaded', filename)
            self._source_cache[filename] = source
            return source
        except (HTTPError, URLError) as e:
            log.debug('loader: %r failed. %s', filename, e)
            raise ImportError("Can't load %s" % filename)

    def is_package(self, fullname):
        return False

# Package loader for a URL
class UrlPackageLoader(UrlModuleLoader):
```

```
    def load_module(self, fullname):
        mod = super().load_module(fullname)
        mod.__path__ = [ self._baseurl ]
        mod.__package__ = fullname

    def get_filename(self, fullname):
        return self._baseurl + '/' + '__init__.py'

    def is_package(self, fullname):
        return True

# Utility functions for installing/uninstalling the loader
_installed_meta_cache = { }
def install_meta(address):
    if address not in _installed_meta_cache:
        finder = UrlMetaFinder(address)
        _installed_meta_cache[address] = finder
        sys.meta_path.append(finder)
        log.debug('%r installed on sys.meta_path', finder)

def remove_meta(address):
    if address in _installed_meta_cache:
        finder = _installed_meta_cache.pop(address)
        sys.meta_path.remove(finder)
        log.debug('%r removed from sys.meta_path', finder)
```

äÿŃéÍćæŸŕäÿĂäÿłäžd'äžŠäijŽèŕİïijŇæijŤçd'žäžĘåęĆä¡Ţä¡£çŤĺåL'■éÍćçŽĎäżčçăĄïijŽ

```
>>> # importing currently fails
>>> import fib
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'
>>> # Load the importer and retry (it works)
>>> import urlimport
>>> urlimport.install_meta('http://localhost:15000')
>>> import fib
I'm fib
>>> import spam
I'm spam
>>> import grok.blah
I'm grok.__init__
I'm grok.blah
>>> grok.blah.__file__
'http://localhost:15000/grok/blah.py'
>>>
```

è£ŹäÿłçL'źæőŁçŽĎæŰźæąĹäijŽåőL'èčĚäÿĂäÿłçL'źåĹńçŽĎæ§ěæL'¿åŹÍ

`UrlMetaFinder` åőđä¿ŃïijŇ ä¡IJäÿž `sys.meta_path`

äÿ■æIJĂåŘŐçŽĎåőđä¡ŞãĂĆ å¡ŞæĺąåİŮèćńåŕijåĚěæŮűïijŇäijŽä¿İæ■ő `sys.meta_path`

äÿ■çŽĎæ§ěæL’¿åŹĺáőŽä¡■æĺąåİŮãĂĆ åIJĺè£Źäÿłä¿Ńå■Řäÿ■ïijŇUrlMetaFinder åőđä¿ŃæŸŕæIJĂåŘŐäÿĂäÿłæ§ěæL’¿åŹĺæŰźæąĹïijŇ å¡ŞæĺąåİŮåİJĺäżżä¡ŢäÿĂäÿłæŹőéĂŽåIJřæŰźéČ¡æL’¿äÿ

ä¡IJäÿžåÿÿèğĄçŽĎåőđçŐřæŰźæąĹïijŇUrlMetaFinder çśżåŇĚèčĚåIJĺäÿĂäÿłçŤĺæĹůæŇĞåőŽçŽĎURLäÿŁãĂĆ åIJĺåĘĚéČĺïijŇæ§ěæL’¿åŹléĂŽè£ĞæŁŞåŔŰæŇĞåčåŕijåĚěçŽĎæŮűåĂŹïijŇæĺąåİŮåŘ■äijŽèů§åůšæIJL’çŽĎéŞ¿æŐěä¡IJåŕźæŕŤãĂĆåęĆæđIJæL’¿åĹřäžĘäÿĂäÿłåÑäÿĂäÿłå■ŢçŃňçŽĎ UrlModuleLoader çśżèćńçŤĺæİěäżŐè£IJçÍŃæIJžåŹĺäÿŁåŁãè¡¡æžŘäżčçăĄåźűåĹŻåżžè£ŹéĞŇçijŞå■ŸéŞ¿æŐěçŽĎäÿĂäÿłåŐ§åŽäæŸŕéĄ£åĚ■äÿ■å£ĚèęĄçŽĎHTTPèŕůæśĆéĞ■åd’■åŕijåĚěãĂĆ

èĞłåőŽäźL’åŕijåĚěçŽĎçňňäžŇçğ■æŰźæşŢæŸŕçijŰåĘŹäÿĂäÿłéŠľå■ŘçŻt’æŐěątŇåĚěåĹř sys.path åŔŸéĞŘäÿ■åŐżïijŇ èŕĘåĹńæ§ŘäžŻçŻőå¡ŢåŚ¡åŘ■æĺąåijŔãĂĆ åIJĺ urlimport.py äÿ■æůżåŁãåęĆäÿŇçŽĎçśżåŠŇæŤŕæŇĄåĞ¡æŢřijŽ

```
# urlimport.py
# ... include previous code above ...
# Path finder class for a URL
class UrlPathFinder(importlib.abc.PathEntryFinder):
    def __init__(self, baseurl):
        self._links = None
        self._loader = UrlModuleLoader(baseurl)
        self._baseurl = baseurl

    def find_loader(self, fullname):
        log.debug('find_loader: %r', fullname)
        parts = fullname.split('.')
        basename = parts[-1]
        # Check link cache
        if self._links is None:
            self._links = [] # See discussion
            self._links = _get_links(self._baseurl)

        # Check if it's a package
        if basename in self._links:
            log.debug('find_loader: trying package %r', fullname)
            fullurl = self._baseurl + '/' + basename
            # Attempt to load the package (which accesses __init__.
↪py)
            loader = UrlPackageLoader(fullurl)
            try:
                loader.load_module(fullname)
                log.debug('find_loader: package %r loaded',␣
↪fullname)
            except ImportError as e:
                log.debug('find_loader: %r is a namespace package',␣
↪fullname)
                loader = None
            return (loader, [fullurl])

        # A normal module
        filename = basename + '.py'
        if filename in self._links:
```

```
            log.debug('find_loader: module %r found', fullname)
            return (self._loader, [])
        else:
            log.debug('find_loader: module %r not found', fullname)
            return (None, [])

    def invalidate_caches(self):
        log.debug('invalidating link cache')
        self._links = None

# Check path to see if it looks like a URL
_url_path_cache = {}
def handle_url(path):
    if path.startswith(('http://', 'https://')):
        log.debug('Handle path? %s. [Yes]', path)
        if path in _url_path_cache:
            finder = _url_path_cache[path]
        else:
            finder = UrlPathFinder(path)
            _url_path_cache[path] = finder
        return finder
    else:
        log.debug('Handle path? %s. [No]', path)

def install_path_hook():
    sys.path_hooks.append(handle_url)
    sys.path_importer_cache.clear()
    log.debug('Installing handle_url')

def remove_path_hook():
    sys.path_hooks.remove(handle_url)
    sys.path_importer_cache.clear()
    log.debug('Removing handle_url')
```

èęĄä¡£çŤĺè£Źäÿłèůŕå¿Ďæ§ěæL’¿åŹĺïijŇä¡ăåŔłéIJĂèęĄåIJĺ `sys.path` äÿ■åŁăåĚěURLéŞ¿æŐěãĂĆä¿ŃåęĆïijŽ

```
>>> # Initial import fails
>>> import fib
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'

>>> # Install the path hook
>>> import urlimport
>>> urlimport.install_path_hook()

>>> # Imports still fail (not on path)
>>> import fib
Traceback (most recent call last):
```

```
    File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'

>>> # Add an entry to sys.path and watch it work
>>> import sys
>>> sys.path.append('http://localhost:15000')
>>> import fib
I'm fib
>>> import grok.blah
I'm grok.__init__
I'm grok.blah
>>> grok.blah.__file__
'http://localhost:15000/grok/blah.py'
>>>
```

åĚşéŤőçĆźåřśæŸŕ `handle_url()` åĞ¡æŢřïijŇåőČèćńæůżåŁääĹřäžĘ `sys.path_hooks` åŔŸéĞŔäÿ■ãĂĆ å¡Ş `sys.path` çŽĎåőđä¡Şèćńåđ'ĎçŘĘæŮűïijŇäijŽèřČçŢĺ `sys.path_hooks` äÿ■çŽĎåĞ¡æŢřãĂĆ åęĆæđIJäżżä¡ŢäÿĂäÿłåĞ¡æŢřè£ŤåŻđäžĘäÿĂäÿłæ§ěæL'¿åŹĺåŕźèśą `sys.path` åőđä¡ŞåŁäè¡¡æĺąåİŮãĂĆ

è£IJçĺŃæĺąåİŮåŁäè¡¡èů§åĚűäżŰçŽĎåŁäè¡¡ä¡£çŤĺæŰźæşŢåĞääźŐæŸŕäÿĂæăůçŽĎãĂĆä¿ŃåęĆïijŽ

```
>>> fib
<urlmodule 'fib' from 'http://localhost:15000/fib.py'>
>>> fib.__name__
'fib'
>>> fib.__file__
'http://localhost:15000/fib.py'
>>> import inspect
>>> print(inspect.getsource(fib))
# fib.py
print("I'm fib")

def fib(n):
    if n < 2:
        return 1
    else:
        return fib(n-1) + fib(n-2)
>>>
```

èőĺèőž

åIJĺèŕęçżĘèőĺèőžäźŃåL'■ïijŇæIJL'çĆźèęĄåijžèřČçŽĎæŸŕïijŇPythonçŽĎæĺąåİŮãĂĄåŇĚåŠŇåŕijåĚěæIJ å■şä¡£çżŔéłŇäÿřåŕŇçŽĎPythonçĺŃåžŔåŠŸäź§å¿ĹåřŚèČ¡çš¿éĂŽåőČäžñãĂĆ æĹŚåIJĺè£ŹéĞŇæŐĺè■ŘäÿĂäžŽåĂijçŽĎåŐżèŕżçŽĎæŰĞæąčåŠŇäźęçś■ïijŇåŇĚæŃň importlib module åŠŇ PEP 302. æŰĞæąčåĘĚåőźåIJĺè£ŹéĞŇäÿ■äijŽèćńéĞ■åd'■æŔŘåĹŕïijŇäÿ■è£ĞæĹŚåIJĺè£Z

éęŰåĚĹïijŇåęĆæđIJä¡ăæČşåĹŽåžžäÿĂäÿłæŰřçŽĎæĺąåİŮåŕźèśąïijŇä¡£çŤĺ `imp.new_module()` åĞ¡æŢřïijŽ

```
>>> import imp
>>> m = imp.new_module('spam')
>>> m
<module 'spam'>
>>> m.__name__
'spam'
>>>
```

æÍąåİŮåŕźèśąéĂŽåÿÿæIJL’äÿĂäžŻæIJ§æIJŻåśđæĂğïijŇåŇĚæŃň `__file__` ïijĹè£ŘèąŇæÍąåİŮåŁäè¡¡èí■åŘěçŽĎæŰĞäžűåŘ■ïijL’ åŠŇ `__package__` (åŇĚåŘ■)ãĂĆ

åĚűæňąïijŇæÍąåİŮäijŽèćńèğčéĞŁåŹÍçijŞå■Ÿèţůæİěã ĂĆæÍąåİŮçijŞå■ŸåŘŕäžěåIJÍå■ŮåĚÿ `sys.modules` äÿ■èćńæL’¿åĹřãĂĆ åŻääÿžæIJL’äžĘè£Źäÿłçij Şå■ŸæIJžåĹűïijŇéĂŽåÿÿåŘŕäžěårĘçijŞå■ŸåŠ

```
>>> import sys
>>> import imp
>>> m = sys.modules.setdefault('spam', imp.new_module('spam'))
>>> m
<module 'spam'>
>>>
```

åęĆæđIJçżŹåőŽæÍąåİŮåůšçżŘå■ŸåIJÍéĆčäźĹåřśäijŽçŻt’æŐèèŐůå¿ŮåůšçżŕèćńåĹŻåżžè£ĞçŽĎæÍąåİŮï

```
>>> import math
>>> m = sys.modules.setdefault('math', imp.new_module('math'))
>>> m
<module 'math' from '/usr/local/lib/python3.3/lib-dynload/math.so'>
>>> m.sin(2)
0.9092974268256817
>>> m.cos(2)
-0.4161468365471424
>>>
```

çŤśäžŐåĹŻåżžæÍąåİŮå¿ĹçőĂå■Ţïij Ňå¿Ĺåőźæ ŸŞçijŰåĘŹçőĂå■ŢåĞ¡æŢřæŕŤåęĆçňňäÿĂéČĺåĹĘçŽĎ `load_module()` åĞ¡æŢřãĂĆ è£ŹäÿłæŰźæąĹçŽĎäÿĂäÿłçijžçĆźæŸŕå¿ĹéŽ¿åď ĎçŘĘåď■æİĆæČĚåĘţæŕŤäÿžäžĘåď ĎçŘĘäÿĂäÿłåŇĚïijŇä¡äèęĄéĞ■æŰřåőđçŐřæŹőéĂŽimportèí■åŘěçŽĎåžŢåśĆéĂżè¿ŚïijĹæŕŤåęĆæL’ğèąŇéĆčäžŻæŰĞäžűïijŇèő¿ç¡őèůřå¿Ďç■L’ïijL’ãĂĆè£Źäÿłåď ■æİĆæĂğářśæŸŕäÿžäżĂäźĹæIJĂåě¡çŻt’æŐ

æL’l’åśŢimportèí■åŘěå¿ĹçőĂå■Ţïij Ňä¡ĘæŸŕäijŽæIJL’å¿Ĺåď Žçğżå ŁĺæŞ■ä¡IJãĂĆ æIJĂéńŸåśĆäÿŁïijŇåŕijåĚěæŞ■ä¡IJèćńäÿĂäÿłä¡■äžŐsys.meta_pathåĹŮèąĺäÿ■çŽĎâĂIJåĚČèůřå¿ĎâĂİæ§ěæ åęĆæđIJä¡äè¿ŞåĞžåőČçŽĎåĂijïijŇäijŽçIJŇåĹřäÿŇéÍćè£ŹæăůïijŽ

```
>>> from pprint import pprint
>>> pprint(sys.meta_path)
[<class '_frozen_importlib.BuiltinImporter'>,
<class '_frozen_importlib.FrozenImporter'>,
<class '_frozen_importlib.PathFinder'>]
>>>
```

å¡ŞæL’ğèąŇäÿĂäÿłèí■åŘěæŕŤåęĆ `import fib` æŮűïijŇèğčéĞŁåŹĺäijŽéĄ■åŐĘsys.mata_pathäÿ■çŽĎ èřČçŤĺåőČäžňçŽĎ `find_module()` æŰźæşŢåőŽä¡■æ■čçąőçŽĎæÍąåİŮåŁäè¡¡åŹĺãĂĆ

åŔŕäžěéĂŽè£ĞåőđéłŇæİěçIJŃçIJŃïijŽ

```
>>> class Finder:
...     def find_module(self, fullname, path):
...         print('Looking for', fullname, path)
...         return None
...
>>> import sys
>>> sys.meta_path.insert(0, Finder()) # Insert as first entry
>>> import math
Looking for math None
>>> import types
Looking for types None
>>> import threading
Looking for threading None
Looking for time None
Looking for traceback None
Looking for linecache None
Looking for tokenize None
Looking for token None
>>>
```

æşÍæĎŔçIJŃ `find_module()` æŰźæşŢæŸŕæĂŐæăůåIJÍæŕŔäÿĂäÿłåŕijåĚěåřśèćńèğęåŔŚçŽĎãĂĆè£ŹäÿłæŰźæşŢäÿ■çŽĎpathåŔĆæŢřçŽĎä¡IJçŤÍæŸŕåď˙ĎçŘĘåŇĚãĂĆåď˙ŽäÿłåŇĚèćńåŕijåĚěïijŇåřśæŸŕäÿĂäÿłåŔŕåIJÍåŇĚçŽĎ `__path__` åšđæĂğäÿ■æL'¿åĹřçŽĎèůŕå¿ĎåĹŮèąÍãĂĆ èęĄæL'¿åĹřåŇĚçŽĎå■ŘçżĎäžűåřśèęĄæčĂæ§ěè£ŹäžŻèůŕå¿ĎãĂæŕŤåęĆæşÍæĎŔåŕźäžŐ `xml.etree` åŠŇ `xml.etree.ElementTree` çŽĎèůŕå¿ĎéĚ■ç¡őïijŽ

```
>>> import xml.etree.ElementTree
Looking for xml None
Looking for xml.etree ['/usr/local/lib/python3.3/xml']
Looking for xml.etree.ElementTree ['/usr/local/lib/python3.3/xml/
↪etree']
Looking for warnings None
Looking for contextlib None
Looking for xml.etree.ElementPath ['/usr/local/lib/python3.3/xml/
↪etree']
Looking for _elementtree None
Looking for copy None
Looking for org None
Looking for pyexpat None
Looking for ElementC14N None
>>>
```

åIJÍ `sys.meta_path` äÿŁæ§ěæL'¿åŹÍçŽĎä¡■ç¡őå¿ĹéĞ■èęĄïijŇåřĘåőČäžŐéŸ§åď˙ç䞺åĹřéŸ§år¿ïijŇ

```
>>> del sys.meta_path[0]
>>> sys.meta_path.append(Finder())
>>> import urllib.request
>>> import datetime
```

çŐřåIJĺä¡ăçIJŃäÿ■åĹřäżżä¡Ţè¿ŞåĞžäžĘïijŃåŻääÿžåŕijåĚèècńsys.meta_pathäÿ■çŽĎåĚűäżŰåőđä¡Şåď'Đç
è£ŹæŮűåĂŹïijŃä¡ääŔłæIJL'åIJĺåŕijåĚěäÿ■å■ŸåIJĺæĺąåİŮçŽĎæŮűåĂŹæL'■èČ¡çIJŃåĹřåőČècńèğęåŔŚïijŽ

```
>>> import fib
Looking for fib None
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'
>>> import xml.superfast
Looking for xml.superfast ['/usr/local/lib/python3.3/xml']
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'xml.superfast'
>>>
```

ä¡ääźŃåL'■åőL'èčĚè£ĞäÿĂäÿłæ■ŢèŐůæIJłç§ěæĺąåİŮçŽĎæ§ěæL'¿åŹĺïijŃè£ŹäÿłæŸŕ
UrlMetaFinder çśżçŽĎåĚşéŤőãĂĆ äÿĂäÿł UrlMetaFinder åőđä¿ŃècńæůżåŁääĹř
sys.meta_path çŽĎæIJńář¿ïijŃä¡IJäÿžæIJĂåŘŐäÿĂäÿłæ§ěæL'¿åŹĺæŰźæąĹãĂĆ
åęĆædIJècńèřůæśĆçŽĎæĺąåİŮåŘ■äÿ■èČ¡åőŽä¡■ïijŃářśäijŽècńè£Źäÿłæ§ěæL'¿åŹĺåď'ĐçŘĘæŐL'ãĂĆ
åď'ĐçŘĘåŇĚçŽĎæŰűåĂŹéIJĂèęĄæşĺæĎŔïijŃåIJĺpathåŔĆæŢřäÿ■æŇĞåőŽçŽĎåĂijéIJĂèęĄècńæčĂæ§ěïijŃ
åęĆædIJäÿ■æŸŕïijŃèřěå■ŘæĺąåİŮå£Ěéąžå¡ŠåśđäžŐåĚűäżŰæ§ěæL'¿åŹĺåźűècńå£¡çŢěæŐL'ãĂĆ

åŕźäžŐåŇĚçŽĎåĚűäżŰåď'ĐçŘĘåŔŕåIJĺ UrlPackageLoader
çśżäÿ■èćńæL'¿åĹřãĂĆ è£Źäÿłçśżäÿ■äijŽåŕijåĚěåŇĚåŘ■ïijŃèĂŇæŸŕåŐžåŁäè¡¡åŕźåžŢçŽĎ
__init__.py æŰĞäżűãĂĆ åőČäź§äijŽèő¿ç¡őæĺąåİŮçŽĎ __path__
åśđæĂğïijŃè£ŹäÿĂæ■ěå¿ĹéĞ■èęĄïijŃ åŻääÿžåIJĺåŁäè¡¡åŇĚçŽĎå■ŘæĺąåİŮæŮűè£ŹäÿłåĂijäijŽècńäijäçżŹå
find_module() èřČçŤĺãĂĆ å§žäžŐèůŕå¿ĐçŽĎåŕijåĚěéŠl'å■ŘæŸŕè£ŹäžŽæĂİæČşçŽĎäÿĂäÿłæL'l'åśŢïijŃ
æĹŚäžňéČ¡ç§ěéĄŞïijŃsys.path æŸŕäÿĂäÿłPythonæ§ěæL'¿æĺąåİŮçŽĎçŻő¡ŢåĹŮèąĺïijŃä¿ŃåęĆïijŽ

```
>>> from pprint import pprint
>>> import sys
>>> pprint(sys.path)
['',
'/usr/local/lib/python33.zip',
'/usr/local/lib/python3.3',
'/usr/local/lib/python3.3/plat-darwin',
'/usr/local/lib/python3.3/lib-dynload',
'/usr/local/lib/...3.3/site-packages']
>>>
```

åIJĺ sys.path äÿ■çŽĎærŔäÿĂäÿłåőđä¡ŞéČ¡äijŽècńéćĺåď'ŰçŽĎçżŚåőŽåĹřäÿĂäÿłæ§ěæL'¿åŹĺårźèśąäÿ
ä¡ääŔŕäžěéĂŽè£Ğæ§ěçIJŃ sys.path_importer_cache
åŐżçIJŃäÿŃè£ŹäžŽæ§ěæL'¿åŹĺïijŽ

```
>>> pprint(sys.path_importer_cache)
{'.': FileFinder('.'),
'/usr/local/lib/python3.3': FileFinder('/usr/local/lib/python3.3'),
'/usr/local/lib/python3.3/': FileFinder('/usr/local/lib/python3.3/
↪'),
'/usr/local/lib/python3.3/collections': FileFinder('...python3.3/
↪collections'),
```

```
'/usr/local/lib/python3.3/encodings': FileFinder('...python3.3/
 ↪encodings'),
'/usr/local/lib/python3.3/lib-dynload': FileFinder('...python3.3/
 ↪lib-dynload'),
'/usr/local/lib/python3.3/plat-darwin': FileFinder('...python3.3/
 ↪plat-darwin'),
'/usr/local/lib/python3.3/site-packages': FileFinder('...python3.3/
 ↪site-packages'),
'/usr/local/lib/python33.zip': None}
>>>
```

sys.path_importer_cache æŕŤ sys.path äijŽæŽť'åd'ğçĆźïijŇ åŽääÿžåőČäijŽäÿžæĽĂæIJĽèćńåŁäè¡¡äžčçăĄçŽĎçŽőå¡Ţèőřå¡ŢåőČäžńçŽĎæ§ěæĽ¿åŹĺãĂĆ è£ŹåŇĚæŃňåŇĚçŽĎå■ŘçŽőå¡ŢïijŇè£ŹäžŽéĂŽåÿÿåIJĺ sys.path äÿ■æŸŕäÿ■å■ŸåIJĺçŽĎãĂĆ

èęĄæĽğèąŇ import fib ïijŇäijŽéąžåžŔæčĂæ§ě sys.path äÿ■çŽĎçŽőå¡ŢãĂĆ åŕźäžŐæŕŔäÿłçŽőå¡ŢïijŇåŘ■çğřâĂIJfibâĂİäijŽèćńäijăçżŹçŽÿåžŤçŽĎ sys.path_importer_cache äÿ■çŽĎæ§ěæĽ¿åŹĺãĂĆ è£ŹäÿłåŔŕäżěèőľä¡ăåĹŽåžžèĞłåůśçŽĎæ§ěæĽ¿åŹĺåźűå

```
>>> class Finder:
... def find_loader(self, name):
...     print('Looking for', name)
...     return (None, [])
...
>>> import sys
>>> # Add a "debug" entry to the importer cache
>>> sys.path_importer_cache['debug'] = Finder()
>>> # Add a "debug" directory to sys.path
>>> sys.path.insert(0, 'debug')
>>> import threading
Looking for threading
Looking for time
Looking for traceback
Looking for linecache
Looking for tokenize
Looking for token
>>>
```

åIJĺè£ŹéĞŇïijŇä¡ăåŔŕäżěäÿžåŘ■å■ŮâĂIJdebugâĂİåĹŽåžžäÿĂäÿłæŰřçŽĎçijŞå■Ÿåőđä¡Şåźűår̨åőČèő¿ sys.path äÿŁçŽĎçňňäÿĂäÿłãĂĆ åIJĺæĽĂæIJĽæŐěäÿŃæİěçŽĎårijåĚěäÿ■ïijŇä¡ääijŽçIJŃåĹŕä¡äçŽĎæ§ěä äÿ■è£ĞïijŇçŤśäžŐåőČè£ŤåŽđ (None, [])ïijŇéĆčäźĹåď'ĎçŘĘè£ŽçĺŃäijŽçżğçż■åď'ĎçŘĘäÿŃäÿĂäÿłåőđä¡Şã

sys.path_importer_cache çŽĎä¡£çŤĺèćńäÿĂäÿłå■ŸåĆĺåIJĺ sys.path_hooks äÿ■çŽĎåĞ¡æŢřåĹŮèąĺæŐğåĹűãĂĆ èŕŢèŕŢäÿŃéİćçŽĎä¿Ńå■ŘïijŇåőČäijŽæÿĚéŹď'çijŞå■ŸåźűçżŹ sys.path_hooks æůżåŁääÿĂäÿłæŰřçŽĎèůŕå¿ĎæčĂæ§ěåĞ¡æŢř

```
>>> sys.path_importer_cache.clear()
>>> def check_path(path):
...     print('Checking', path)
...     raise ImportError()
```

```
...
>>> sys.path_hooks.insert(0, check_path)
>>> import fib
Checked debug
Checking .
Checking /usr/local/lib/python33.zip
Checking /usr/local/lib/python3.3
Checking /usr/local/lib/python3.3/plat-darwin
Checking /usr/local/lib/python3.3/lib-dynload
Checking /Users/beazley/.local/lib/python3.3/site-packages
Checking /usr/local/lib/python3.3/site-packages
Looking for fib
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'
>>>
```

æ■čåęĆä¡ăæĽĂèğĄïijŇcheck_path() åĞ¡æŢřèćńæŕŔäÿł sys.path äÿ■çŽĎåőđä¡ŞěřČçŢĺãĂĆ äÿ■éą¿ïijŇçŢśäžŐæŁŻåĞžäžĘ ImportError åijĆåÿÿïijŇ åŢěéČ¡äÿ■äijŽåŔŚçŢ§äžĘïijĹäžĔäżĎåřĘæčĂæ§ěè¡ňçğżåĹřsys.path_hooksçŽĎäÿŇäÿĂäÿłåĞ¡æŢřïijL'ãĂĆ

ç§ěéĄŞäžĘæĂŐæăůsys.pathæŸŕæĂŐæăůèćńåď'ĎçŘĘçŽĎïijŇä¡ăářśèČ¡æđĎåżžäÿĂäÿłèĞłåőŽäźL'èůŕå¿

```
>>> def check_url(path):
...     if path.startswith('http://'):
...         return Finder()
...     else:
...         raise ImportError()
...
>>> sys.path.append('http://localhost:15000')
>>> sys.path_hooks[0] = check_url
>>> import fib
Looking for fib # Finder output!
Traceback (most recent call last):
    File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'

>>> # Notice installation of Finder in sys.path_importer_cache
>>> sys.path_importer_cache['http://localhost:15000']
<__main__.Finder object at 0x10064c850>
>>>
```

è£Źářśæ ŸŕæIJňèŁĆæIJĂåŘŐéČĺåĹĘçŽĎåĚşéŤőçĆźãĂĆäžŇåőđäÿŁïijŇäÿĂäÿłçŤĺæİěåIJĺsys.pathäÿ■æ
å¡ŞåőČäżňèćńçćřåĹřçŽĎæŮűůåĂŹïijŇäÿĂäÿłæŰřçŽĎ UrlPathFinder åőđä¿ŃèćńåĹŻåžžåźűèćńæŤ¿åĚě sys.path_importer_cache. äźŇåŘŐïijŇæĽĂæIJL'éIJĂèęĄæčĂæ§ě sys.path çŽĎářijåĚěŕ■åŕěéČ¡äijŽä¡£çŤĺä¡ăçŽĎèĞłåőŽäźL'æ§ě

å§žäžŐèůŕå¿ĎářijåĚěçŽĎåŇĚåď'ĎçŘEçĺ■å¿őæIJL'çĆźåď'■æİĆïijŇåźűäÿŤèů§ find_loader() æŰźæşŢè£ŤåŻđåĂijæIJL'åĚşãĂĆ åŕźäžŐçőĂå■ŢæĺąåİŮïijŇfind_loader() è£ŤåŻđäÿĂäÿłåĚČçżĎ(loader, None)ïijŇ åĚűäÿ■çŽĎload-eræŸŕäÿĂäÿłçŤĺäžŐářijåĚěæĺąåİŮçŽĎåŁăè¡¡åŹĺåőđä¿ŃãĂĆ

åŕźäžŐäÿĂäÿłæŹőéĂŽçŽĎåŇĚïijŇfind_loader() è£ŤåŻđäÿĂäÿłåĚČçżĎ(loader, path)ïijŇ åĚűäÿ■çŽĎloaderæŸŕäÿĂäÿłçŤĺäžŐåŕijåĚěåŇĚïijĹåźűæĽğèąŇ__init__.pyïijĽçŽĎåŁãè¡¡åŹĺåőđä¿pathæŸŕäÿĂäÿłäijŽåĹİåğŃåŇŰåŇĚçŽĎ __path__ åśđæĂğçŽĎçŽőå¡ŢåĹŮèąĺãĂĆä¿ŃåęĆïijŇåęĆæđIJå§žçąĂURLæŸŕ http://localhost:15000 åźűäÿŤäÿĂäÿłçŤĺæĹůæĽğèąŇ import grok , éĆčäźĹ find_loader() è£ŤåŻđçŽĎpathåřśäijŽæŸŕ [âĂŸhttp://localhost:15000/grokâĂŹ]

find_loader() è£ŸèęĄèČ¡åd'ĎçŘĘäÿĂäÿłåŚ¡åŘ■çl'žéŮt'åŇĚãĂĆäÿĂäÿłåŚ¡åŘ■çl'žéŮt'åŇĚäÿ■æIJL'äÿĂäÿłåŘĹæşŢçŽĎåŇĚçŻőå¡ŢåŘ■ïijŇä¡ĘæŸŕäÿ■å■ŸåIJĺ__init__.pyæŰè£ŹæăůçŽĎèŕÏïijŇfind_loader() å£ĚéążèŁŤåŻđäÿĂäÿłåĚČçżĎ(None, path)ïijŇpathæŸŕäÿĂäÿłçŻőå¡ŢåĹŮèąĺïijŇçŤśåőČæİěæđĎåżžåŇĚçŽĎåőŽäźĽæIJL'__init__.pyæŰĞäžűçŽĎ__path__åŕźäžŐè£Źçğ■æČĚåĘţïijŇåŕijåĚěæIJžåĹűäijŽçżğçż■åL'■èąŇåŐżæčĂæ§ěsys.pathäÿ■çŽĎçŽőå¡ŢãĂĆåęĆæđIJæL'¿åĹřäžĘåŚ¡åŘ■çl'žéŮt'åŇĚïijŇæL'ĂæIJL'çŽĎçżŞæđIJèůŕå¿ĎèćńåŁäåĹřäÿĂèţůæİěæđĎåżžæIJĂåĚşäžŐåŚ¡åŘ■çl'žéŮt'åŇĚçŽĎæŻt'åd'Žä£qæĄŕèŕůåŔĆèĂČ10.5åřŔèŁĆãĂĆ

æL'ĂæIJL'çŽĎåŇĚéČ¡åŇĚåŘńäžĘäÿĂäÿłåĘĚéČĺèůŕå¿Ďèő¿ç¡őïijŇåŔŕäžěåIJĺ__path__åśđæĂğäÿ■çIJŃ

```
>>> import xml.etree.ElementTree
>>> xml.__path__
['/usr/local/lib/python3.3/xml']
>>> xml.etree.__path__
['/usr/local/lib/python3.3/xml/etree']
>>>
```

äźŃåL'■æŔŘåĹřïijŇ__path__çŽĎèő¿ç¡őæŸŕéĂŽè£Ğ find_loader() æŰźæşŢè£ŤåŻđåĂijæŐğåĹűçŽĎãĂĆ äÿ■è£ĞïijŇ__path__æŐëäÿŃæİěäź§èćńsys.path_hooksäÿ■çŽĎåĞ¡æŢåŻăæ■d'ïijŇä¡ĘåŇĚçŽĎå■ŘçżĎäžűèćńåŁäè¡¡åŘŐïijŇä¡■äžŐ__path__äÿ■çŽĎåőđä¡ŞäijŽèćń handle_url() åĞ¡æŢřæčĂæ§ěãĂĆ è£ŹäijŽåŕijèĞt'æŰřçŽĎ UrlPathFinder åőđä¿ŃèćńåĹŻåžžåźűäÿŤèćńåŁääĚěåĹř sys.path_importer_cache äÿ■ãĂĆ

è£ŸæIJL'äÿłéŽ¿çĆźåřśæŸŕ handle_url() åĞ¡æŢřäžěåŘŁåőČèů§åĘĚéČĺä¡£çŤĺçŽĎ _get_links() åĞ¡æŢřäźŃéŮt'çŽĎäžd'äžŠãĂĆ åęĆæđIJä¡äçŽĎæ§ěæL'¿åŹĺåőđçŐŕéIJĂèęĄä¡£çŤĺåĹřåĚűæIJL'åŘŕèČ¡è£ŹäžŻæĹąåİŮäijŽåIJĺæ§ěæL'¿åŹĺæŞ■ä¡IJæIJ§éŮt'è£ŻèąŇæŻt'åd'ŽçŽĎåŕijåĚěãĂĆ åőČåŘŕäžěåŕijèĞt' handle_url() åŠŇåĚűäžŰæ§ěæL'¿åŹĺéČĺåĹĘéŽůåĚěäÿĂçğ■éĂŠå¡Šå¿łçŐŕçŁűæĂĄãäÿžäžĘèğčéĞŁè£Źçğ■åŘŕèČ¡æĂğïijŇåőđçŐřäÿ■æIJL'äÿĂäÿłèćńåĹŻåžžçŽĎæ§ěæL'¿åŹĺçijŞå■ŸïijĹæŕŘäÿĂåőČåŘŕäžěéĄ£åĚ■åĹŻåžžéĞ■åd'■æ§ěæL'¿åŹĺçŽĎéŮőéćŸãĂĆ åŘęåd'ŰïijŇäÿŃéİćçŽĎäžčçăĄçL'ĞæőţåŘŕäžěçąőä£İæ§ěæL'¿åŹĺäÿ■äijŽåIJĺåĹİåğŃåŇŰéŞ¿æŐěéŽĘåŘĹçŽĎ

```
# Check link cache
if self._links is None:
    self._links = [] # See discussion
    self._links = _get_links(self._baseurl)
```

æIJĂåŘŐïijŇæ§ěæL'¿åŹĺçŽĎ invalidate_caches() æŰźæşŢæŸŕäÿĂäÿłåůěåĚůæŰźæşŢïijŇçŤĺæİěæÿĚçŘĘåĘĚéČĺçijŞå■ŸãĂĆ è£ŹäÿłæŰźæşŢåĘ■çŤĺæĹůèřČçŤĺ importlib.invalidate_caches() çŽĎæŮűåĂŹèćńèğęåŔŠãĂĆ åęĆæđIJä¡äæČşèől'URLåŕijåĚěèĂĚéĞ■æŰřèŕžåŘŰéŞ¿æŐěåĹŮèąĺçŽĎèŕÍåŘŕä

åŕźæŕŤäÿŃäÿd'çğ■æŰźæąĹïijĹä£őæŤźsys.meta_pathæĹŰä¡£çŤĺäÿĂäÿłèůŕå¿ĎéŠl'å■ŘïijL'ãĂĆä¡£çŤĺsys.meta_pathçŽĎåŕijåĚěèĂĚåŘŕäžěæŃL'çĚğèĞłåůśçŽĎéIJĂèęĄèĞłçŤśåd'ĎçŘĘæĺąåİŮãĂĆä¿ŃåęĆïijŇåőČäžňåŘŕäžěäžŐæŢřæ■őåžŞäÿ■åŕijåĚěæĹŰäžěäÿ■åŘŃäžŐäÿĂèĹňæĺąåİŮ/åŃĚåd'ĎçŘĘæŰźåi

è£Źçğ■èĞłçŤśåŘŇæăůæĎŘåŚşçİĂåŕijåĚěèĂĚéIJĂèęĄèĞłåůśè£Żèą ŇåĘĚéČĺçŽĎäÿĂäžŻçőąçŘĘãĂĆ
åŘęåď Űïij Ňå§žäžŐèůŕå¿ĎçŽĎéŠľå■ŘåŘłæŸŕéĂĆçŤĺäžŐåŕźsys.pathçŽĎåď ĎçŘĘãĂĆ
éĂŽè£Ğè£Źçğ■æĽľåśŢåŁãè¡¡çŽĎæĺąåİŮèů§æŹőéĂŽæŰźåijŔåŁãè¡¡çŽĎçĽźæĂğæŸŕäÿĂæăůçŽĎãĂĆ

åęĆæđIJåĹřçŐřåIJĺäÿžæ■ćä¡ãè£ŸæŸŕäÿ■æŸŕå¿ĹæŸŐçŹ¡ïijŇéĆčäźĹåŘŕäžěéĂŽè£ĞåćđåŀääÿĂäžŻæŮ

```
>>> import logging
>>> logging.basicConfig(level=logging.DEBUG)
>>> import urlimport
>>> urlimport.install_path_hook()
DEBUG:urlimport:Installing handle_url
>>> import fib
DEBUG:urlimport:Handle path? /usr/local/lib/python33.zip. [No]
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ImportError: No module named 'fib'
>>> import sys
>>> sys.path.append('http://localhost:15000')
>>> import fib
DEBUG:urlimport:Handle path? http://localhost:15000. [Yes]
DEBUG:urlimport:Getting links from http://localhost:15000
DEBUG:urlimport:links: {'spam.py', 'fib.py', 'grok'}
DEBUG:urlimport:find_loader: 'fib'
DEBUG:urlimport:find_loader: module 'fib' found
DEBUG:urlimport:loader: reading 'http://localhost:15000/fib.py'
DEBUG:urlimport:loader: 'http://localhost:15000/fib.py' loaded
I'm fib
>>>
```

æIJĂåŘŐïijŇåžžèőőä¡ãèŁśçĆźæŮűéŮt'çIJŃçIJŃ PEP 302 äžěåŘŁimportlibçŽĎæŰĞæąčãĂĆ

12.12 10.12 åŕijåĚěæĺąåİŮçŽĎåŘŇæŮűä£őæŤźæĺąåİŮ

éŮőécŸ

ä¡ăæČşçżŹæ§Řäÿłåůšå■ŸåIJĺæĺąåİŮäÿ■çŽĎåĞ¡æŢřæůżåŁãècĚéěřåŹĺãĂĆ
äÿ■è£ĞïijŇåĽ■æŘŘæŸŕè£ŹäÿłæĺąåİŮåůšçżŔèćńåŕijåĚěåźűäÿŤècńä¡£çŤĺè£ĞãĂĆ

èğčåĘşæŰźæąĹ

è£ŹéĞŇéŮőécŸçŽĎæIJňèťĺåřśæŸŕä¡ăæČşåIJĺæĺąåİŮècńåŁãè¡¡æŮűæĽğèąŇæ§Řäÿłåŀĺä¡IJãĂĆ
åŘŕèČ¡æŸŕä¡ăæČşåIJĺäÿĂäÿłæĺąåİŮècńåŁãè¡¡æŮűèğęåŔŚæ§ŘäÿłåŻđèřČåĞ¡æŢřæİěéĂŽç§ëä¡ããĂĆ

è£ŹäÿłéŮőécŸåŘŕäžëä¡£çŤĺ10.11åřŔèŁĆäÿ■åŘŇæăůçŽĎåŕijåĚěéŠľå■ŘæIJžåĹűæİěåőđçŐřãĂĆäÿŃéİć

```
# postimport.py
import importlib
import sys
```

```python
from collections import defaultdict

_post_import_hooks = defaultdict(list)

class PostImportFinder:
    def __init__(self):
        self._skip = set()

    def find_module(self, fullname, path=None):
        if fullname in self._skip:
            return None
        self._skip.add(fullname)
        return PostImportLoader(self)

class PostImportLoader:
    def __init__(self, finder):
        self._finder = finder

    def load_module(self, fullname):
        importlib.import_module(fullname)
        module = sys.modules[fullname]
        for func in _post_import_hooks[fullname]:
            func(module)
        self._finder._skip.remove(fullname)
        return module

def when_imported(fullname):
    def decorate(func):
        if fullname in sys.modules:
            func(sys.modules[fullname])
        else:
            _post_import_hooks[fullname].append(func)
        return func
    return decorate

sys.meta_path.insert(0, PostImportFinder())
```

è£ŹæăůïijŇä¡ăåřśåŔŕäżěä¡£çŤĺ `when_imported()` èčĚéěřåŹĺäžĘïijŇä¿ŃåęĆïijŽ

```python
>>> from postimport import when_imported
>>> @when_imported('threading')
... def warn_threads(mod):
...     print('Threads? Are you crazy?')
...
>>>
>>> import threading
Threads? Are you crazy?
>>>
```

ä¡IJäÿžäÿĂäÿłæŻťåőđéŹĚçŽĎä¿Ńå■ŘïijŇä¡ăåŔŕèČ¡æČşåIJĺåůšå■ŸåIJĺçŽĎåőŽäźĽäÿŁéİćæůżåŁăèčĚé

```
from functools import wraps
from postimport import when_imported

def logged(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        print('Calling', func.__name__, args, kwargs)
        return func(*args, **kwargs)
    return wrapper

# Example
@when_imported('math')
def add_logging(mod):
    mod.cos = logged(mod.cos)
    mod.sin = logged(mod.sin)
```

èőÍèőž

æIJňèŁĆæŁĂæIJŕä¿İèţŰäžŐ10.11åřŔèŁĆäÿ■èőšè£řè£ĞçŽĎåŕijåĚééŠľå■ŘïijŇåźűçÍ■ä¡IJä£őæŤźãĂĆ

`@when_imported` èčĚéěřåŹÍçŽĎä¡IJçŤÍæŸŕæşÍåĘŇåIJÍåŕijåĚěæŮűèćńæ£Ăæť żçŽĎåď ĎçŘĘåŹÍåĞ¡æ èŕěèčĚéěřåŹÍæčĂæ§ěsys.modulesæİěæ§ěçIJŃæĺąåİŮæŸŕåŘęçIJ§çŽĎåůšçżŔèćńåŁäè¡¡äžĘãĂĆ åęĆæđIJæŸŕçŽĎèŕÍïijŇèřěåď ĎçŘĘåŹÍèćńçńŇå■şèřČçŤÍãĂĆäÿ■çĎűïijŇåď ĎçŘĘåŹÍèćńæůżåŁääĹř `_post_import_hooks` å■ŮåĚÿäÿ■çŽĎäÿĂäÿłåĹŮèąĺäÿ■åŐżãĂĆ `_post_import_hooks` çŽĎä¡IJçŤÍåřśæŸŕæŤűéŻĘæĽĂæIJĽçŽĎäÿžæŕŔäÿłæĺąåİŮæşÍåĘŇçŽĎåď ĎçŘĘäÿĂäÿłæĺąåİŮåŔŕäžěæşÍåĘŇåď Žäÿłåď ĎçŘĘåŹÍãĂĆ

èęĄèőľæĺąåİŮåŕijåĚěåŘŐèğęåŔŚæůżåŁäçŽĎåŁĺä¡IJïijŇ`PostImportFinder` çśżèćńèő¿ç¡őäÿžsys.meta_pathçňňäÿĂäÿłåĚČçť ããĂĆ åőČäijŽæ■ŢèŐůæĽĂæIJĽæĺąåİŮåŕijåĚěæŞ■ä¡IJãĂĆ

æIJňèŁĆäÿ■çŽĎ `PostImportFinder` çŽĎä¡IJçŤÍåźűäÿ■æŸŕåŁäè¡¡æĺąåİŮïijŇèĂŇæŸŕèĞłåÿęåŕijåĚ åőđéŹĚçŽĎåŕijåĚèèćńåğŤæť ¿çżŹä¡■äžŐsys.meta_pathäÿ■çŽĎåĚűäžŰæ§ěæĽ¿åŹÍãĂĆ `PostImportLoader` çśżäÿ■çŽĎ `imp.import_module()` åĞ¡æŢřèćńéĂŠå¡ŠçŽĎèřČçŤÍãĂĆ äÿžäžĘéĄ£åĚ■éŹůåĚěæŮăçž£å¿łçŐŕïijŇ`PostImportFinder` ä£ÍæŇĄäžĘäÿĂäÿłæĽĂæIJĽèćńåŁäè¡¡è£ĞçŽĎæĺąåİŮéŻĘåŘĹãĂĆ åęĆæđIJäÿĂäÿłæĺąåİŮåŘ■å■ŸåIJÍåřśäijŽçŻť æŐèèćńå£¡çŢěæŐĽãĂĆ

å¡ŞäÿĂäÿłæĺąåİŮèćń `imp.import_module()` åŁäè¡¡åŘŐïijŇ æĽĂæIJĽåIJÍ_post_import_hooksèćńæşÍåĘŇçŽĎåď ĎçŘĘåŹÍèćńèřČçŤÍïijŇä¡£çŤÍæŰřåŁäè¡¡æĺąåİŮä¡IJäÿžã

æIJĽäÿĂçĆźéIJĂèęĄæşÍæĎŔçŽĎæŸŕæIJňæIJžäÿ■éĂĆçŤÍäžŐéĆčäžŻéĂŽè£Ğ `imp.reload()` èćńæŸ¿åijŔåŁäè¡¡çŽĎæĺąåİŮãĂĆ äź§åŕśæŸŕèŕťïijŇåęĆæđIJä¡ååŁäè¡¡äÿĂäÿłäźŇåĽ■åůšèćńåŁäè åŘęåď ŰïijŇèęĄæŸŕä¡ääžŐsys.modulesäÿ■åĹäéŹď æĺąåİŮçĎűåŘŐåĘ■éĞ■æŰřåŕijåĚěïijŇåď ĎçŘĘåŹÍåŔĹäï

æŻť åď ŽåĚşäžŐåŕijåĚěåŘŐéŠľå■Ŕä£ąæĄŕèŕůåŔĆèĂČ PEP 369.

12.13 10.13 åőĽèčĚçğĄæIJĽçŽĎåŇĚ

éŮőécŸ

ä¡ăæČşèęĄåőĽèčĚäÿĂäÿłçňňäÿĽæŰźåŇĚïijŇä¡ĘæŸŕæšąæIJĽæİČéŹŘářĘåőČåőĽèčĚåĹřçşżçż§Python
æĹŰèĂĚïijŇä¡ăåŔŕèČ¡æČşèęĄåőĽèčĚäÿĂäÿłä¿ŻèĞłåůśä¡£çŤĺçŽĎåŇĚïijŇèĂŇäÿ■æŸŕçşżçż§äÿŁéİćæĽĂa

èğčåĘşæŰźæąĹ

PythonæIJĽäÿĂäÿłçŤĺæĹůåőĽèčĚçŻőå¡Ţïij ŇéĂŽåÿÿçśżäijijâĂİ~/.local/lib/python3.3/site-packagesâĂİãĂĆ èęĄåijžåĹűåIJĺè£Źäÿłçżőå¡Ţäÿ■åőĽèčĚåŇĚïijŇåŔŕä¡£çŤĺåőĽèčĚéĂĽ'éąźâĂIJ–userâĂİãĂ

```
python3 setup.py install --user
```

æĹŰèĂĚ

```
pip install --user packagename
```

åIJĺsys.pathäÿ■çŤĺæĹůçŽĎâĂIJsite-packagesâĂİçŻőå¡Ţä¡■äžŐçşżçż§çŽĎâĂIJsite-packagesâĂİçŻőå¡ŢäźŃåĽ■ãĂĆ åŻăæ■ď'ïijŇä¡ăåőĽèčĚåIJĺéĞŇéİćçŽĎåŇĚářśæŕŤçşżçż§åůšåőĽèčĚçŽĎåŇïijĹář¡çőąåźűäÿ■æĂżæŸŕè£Źæăûïij ŇèęĄåŔŰåĘşäžŐçňňäÿĽæŰźåŇĚçőąçŘĘåŹĺïijŇæŕŤåęĆdistributeæĹŰpi

èőĺèőž

éĂŽåÿÿåŇĚäijŽèćńåőĽèčĚåĹřçşżçż§çŽĎsite-packagesçŻőå¡Ţäÿ■åŐżïijŇèůŕå¿ĎçśżäijijâĂIJ/usr/local/lib/python3.3/site-packagesâĂİãĂĆ äÿ■è£ĞïijŇè£ŹæăůåĄŽéIJĂèęĄæIJĽçőąçŘĘåŚŸæİČéŹŘåźűäÿŤä¡£çŤĺsudoåŚ¡äžď'ãĂĆ åřśçőŮä¡ăæIJĽè£ŹæăůçŽĎæİČéŹŘåŐżæĽ'ğèąŇåŚ¡äžď'ïijŇä¡£çŤĺsudoåŐżåőĽèčĚäÿĂäÿłæŰřçŽĎïijŇåŔŕèČ

åőĽèčĚåŇĚåĹřçŤĺæĹůçŻőå¡Ţäÿ■éĂŽåÿÿæŸŕäÿĂäÿłæIJĽæŢĹçŽĎæŰźæąĹïijŇåőČåĚĄèőÿä¡ăåĹŻåžžäŔęåď'ŰïijŇä¡ăè£ŸåŔŕäžěåĹŻåžžäÿĂäÿłèŹŽæŃ§çŐŕåćČïijŇè£ŹäÿłæĹŚäžňåIJĺäÿŃäÿĂèŁĆäijŽèőšåĹřă

12.14 10.14 åĹŻåžžæŰřçŽĎPythonçŐŕåćČ

éŮőécŸ

ä¡ăæČşåĹŻåžžäÿĂäÿłæŰřçŽĎPythonçŐŕåćČïijŇçŤĺæİěåőĽèčĚæĺąåİŮåŠŇåŇĚãĂĆ äÿ■è£ĞïijŇä¡ăäÿ■æČşåőĽèčĚäÿĂäÿłæŰřçŽĎPythonåĚŃéŽĘïijŇäź§äÿ■æČşåŕźçşżçż§PythonçŐŕåćČäžğçŤ§

èğčåĘşæŰźæąĹ

ä¡ăåŔŕäžěä¡£çŤĺ `pyvenv` åŚ¡äžď'åĹŻåžžäÿĂäÿłæŰřçŽĎâĂIJèŹŽæŃ§âĂİçŐŕåćČãĂĆ è£ŹäÿłåŚ¡äžď'èćńåőĽèčĚåIJĺPythonèğčéĞŁåŹĺåŘŇäÿĂçŻőå¡ŢïijŇæĹŰWindowsäÿŁéİćçŽĎScriptsçŻőå¡Ţäÿ

```
bash % pyvenv Spam
bash %
```

äijăçżŹ pyvenv åŚ¡äżď'çŽĎåŘ■å■ŮæŸŕåřĘèęĄèćńåĹŻåżžçŽĎçŻőå¡ŢåŘ■ãĂĆå¡ŞèćńåĹŻåżžåŘŐïijŇS

```
bash % cd Spam
bash % ls
bin include lib pyvenv.cfg
bash %
```

åIJĺbinçŻőå¡Ţäÿ■ïijŇä¡ăäijŽæĽ'¿åĹřäÿĂäÿłåŔŕäżěä¡£çŤĺçŽĎPythonèğčéĞŁåŹĺïijŽ

```
bash % Spam/bin/python3
Python 3.3.0 (default, Oct 6 2012, 15:45:22)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more␣
↪information.
>>> from pprint import pprint
>>> import sys
>>> pprint(sys.path)
['',
'/usr/local/lib/python33.zip',
'/usr/local/lib/python3.3',
'/usr/local/lib/python3.3/plat-darwin',
'/usr/local/lib/python3.3/lib-dynload',
'/Users/beazley/Spam/lib/python3.3/site-packages']
>>>
```

è£ŹäÿłèğčéĞŁåŹĺçŽĎçĽźçĆźåřśæŸŕäżŰçŽĎsite-packagesçŻőå¡Ţèćńèő¿ç¡őäÿžæŰřåĹŻåżžçŽĎçŐŕåćČ
åęĆæđIJä¡ăèęĄåőĽèčĚçňňäÿĽæŰźåŇĚïijŇåőČäżňäijŽèćńåőĽèčĚåIJĺéĆčéĞŇïijŇèĂŇäÿ■æŸŕéĂŽåÿÿçşżçż
packagesçŻőå¡ŢãĂĆ

èőĺèőž

åĹŻåżžèŹŽæŃ§çŐŕåćČéĂŽåÿÿæŸŕäÿžäžĘåőĽèčĚåŠŇçőąçŘĘçňňäÿĽæŰźåŇĚãĂĆ
æ■čåęĆä¡ăåIJĺä¿Ńå■Řäÿ■çIJŃåĹřçŽĎéĆčæăůïijŇsys.path
åŔŸéĞŔåŇĚåŘńæİèèĞłäžŐçşżçż§PythonçŽĎçŻőå¡ŢïijŇ èĂŇ site-
packagesçŻőå¡ŢåůšçżŔèćńéĞ■åőŽä¡■åĹřäÿĂäÿłæŰřçŽĎçŻőå¡ŢãĂĆ

æIJĽäžĘäÿĂäÿłæŰřçŽĎèŹŽæŃ§çŐŕåćČïijŇäÿŃäÿĂæ■ěåřśæŸŕåőĽèčĚäÿĂäÿłåŇĚçőąçŘĘåŹĺïijŇæŕŤå
ä¡ĘåőĽèčĚè£ŹæăůçŽĎåůěåĚůåŠŇåŇĚçŽĎæŮűåĂŹïijŇä¡ăéIJĂèęĄçąőä£İä¡ăä¡£çŤĺçŽĎæŸŕèŹŽæŃ§çŐŕåćČ
åőČäijŽåřĘåŇĚåőĽèčĚåĹřæŰřåĹŻåżžçŽĎsite-packagesçŻőå¡Ţäÿ■åŐżãĂĆ

åř¡çőąäÿĂäÿłèŹŽæŃ§çŐŕåćČçIJŃäÿŁåŐżæŸŕPythonåőĽèčĚçŽĎäÿĂäÿłåď■åĹűïijŇ
äÿ■è£ĞåőČåőđéŹĚäÿŁåŔłåŇĚåŘńäžĘåřŚéĞŔåĞăäÿłæŰĞäżűåŠŇäÿĂäžŻçňęåŔůéŞ¿æŐěãĂĆ
æĽĂæIJĽæăĞåĞĘåžŞåĞ¡æŰĞäżűåŠŇåŔŕæĽğèąŇèğčéĞŁåŹĺéČ¡æİěèĞłåŐ§æİěçŽĎPythonåőĽèčĚãĂĆ
åŻăæ■ď'ïijŇåĹŻåżžè£ŹæăůçŽĎçŐŕåćČæŸŕå¿ĹåőźæŸŞçŽĎïijŇåźűäÿŤåĞăäźŐäÿ■äijŽæűĹèĂŮæIJžåŹĺèţĎæ

éżŸèőď'æČĚåĘţäÿŃïijŇèŹŽæŃ§çŐŕåćČæŸŕçľžçŽĎïijŇäÿ■åŇĚåŘńäżżä¡Ţéćİåď'ŰçŽĎçňňäÿĽæŰźåžŞ
åŔŕäżěä¡£çŤĺâĂIJ–system-site-packagesâĂİéĂĽ'éąźæİěåĹŻåżžèŹŽæŃ§çŐŕåćČïijŇä¿ŃåęĆïijŽ

```
bash % pyvenv --system-site-packages Spam
bash %
```

èů§åď ŽåĚşäžŐ `pyvenv` åŠŇèŹŽæŃ§çŐŕåćČçŽĎä£ąæĄŕåŘŕäžěåŔĆèĂČ PEP 405.

12.15 10.15 åĹĘåŔŚåŇĚ

éŮőécŸ

ä¡ăåůšçżŘçijŰåĘŹäžĘäÿĂäÿłæIJL'çŤĺçŽĎåžŞïijŇæČşářĘåőČåĹĘäžńçżŹåĚűäżŰäžžãĂĆ

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăæČşåĹĘåŔŚä¡ăçŽĎäżčçăĄïijŇçňňäÿĂäżűäžŃåřśæŸŕçżŹåőČäÿĂäÿłåŤŕäÿĂçŽĎåŘ■å■ŮïijŇä¿ŃåęĆïijŇäÿĂäÿłåĚÿåđŃçŽĎåĞ¡æŢřåžŞåŇĚäijŽçśżäijijäÿŃéİćè£ŹæăůïijŽ

```
projectname/
    README.txt
    Doc/
        documentation.txt
    projectname/
        __init__.py
        foo.py
        bar.py
        utils/
            __init__.py
            spam.py
            grok.py
    examples/
        helloworld.py
        ...
```

èęĄèől'ä¡ăçŽĎåŇĚåŔŕäžěåŔŚåÿČåĞžåŐżïijŇéęŰåĚĹä¡ăèęĄçijŰåĘŹäÿĂäÿł `setup.py` ïijŇçśżäijijäÿŃéİćè£ŹæăůïijŽ

```
# setup.py
from distutils.core import setup

setup(name='projectname',
    version='1.0',
    author='Your Name',
    author_email='you@youraddress.com',
    url='http://www.you.com/projectname',
    packages=['projectname', 'projectname.utils'],
)
```

äÿŃäÿĂæ■ěïijŇåřśæŸŕåĹŻåžžäÿĂäÿł `MANIFEST.in` æŰĞäžűïijŇåĹŮåĞžæL'ĂæIJL'åIJÍä¡ăçŽĎåŇĚäÿ

```
# MANIFEST.in
include *.txt
recursive-include examples *
recursive-include Doc *
```

çąőä£İ `setup.py` åŠŇ `MANIFEST.in` æŰĞäżűæŤ¿åIJĺä¡ăçŽĎåŇĚçŽĎæIJĂéąűçžğçŻőå¡Ţäÿ■ãĂĆ
äÿĂæŮęä¡ăåůšçżŕŕåĄŽäžĘè£ŹäžŻïijŇä¡ăåřśåŕŕäžěåČŕäÿŇéİćè£ŹæăůæL'ğèąŇåŚ¡äžď'æİěåĹŽåžžäÿĂäÿłæžŘ

```
% bash python3 setup.py sdist
```

åőČäijŽåĹŻåžžäÿĂäÿłæŰĞäżűæŕŤåęĆâĂİprojectname-1.0.zipâĂİ æĹŰ
âĂIJprojectname-1.0.tar.gzâĂİ, åĚůä¡Şä¿İètŰäžŐä¡ăçŽĎçşżçż§åźşåŘřãĂĆåęĆæđIJäÿĂåĹĞæ■čåÿÿïijŇ
è£ŹäÿłæŰĞäżűåřśåŕŕäžěåŔŚéĂĄçżŹåĹńäžžä¡£çŤĺæĹŰèĂĚäÿŁäijäèĞş Python Package Index.

èőĺèőž

åŕźäžŐçžŕPythonäżčçăĄïijŇçijŰåEŹäÿĂäÿłæŹőéĂŽçŽĎ `setup.py`
æŰĞäżűéĂŽåÿÿå¿ĹçőĂå■ŢãĂĆ äÿĂäÿłåŔŕèČ¡çŽĎéŮőécŸæŸŕä¡ăå£ĚéążæL'ŃåŁĺåĹŮåĞżæL'ĂæIJL'æđĎæ
äÿĂäÿłåÿÿèğĄéŤŹèŕŕåřśæŸŕäżĚäžĚåŔłåĹŮåĞžäÿĂäÿłåŇĚçŽĎæIJĂéąűçžğçŻőå¡ŢïijŇå£ŸèőřäžĘåŇĚåŘńåŇ
è£Źäź§æŸŕäÿžäżĂäźĹåIJĺ `setup.py` äÿ■åŕźäžŐåŇĚçŽĎèŕť'æŸŐåŇĚåŘńäžĘåĹŮèąĺ
`packages=['projectname', 'projectname.utils']`

åď'ğéČĺåĹĘPythonçĺŃåžŔåŠŸéČ¡ç§ěéĄŞïijŇæIJL'å¿Ĺåď'ŽçňňäÿL'æŰźåŇĚçőąçŘĘåŹĺä¿ŻéĂL'æŃľ'ïijŇ
æIJL'äžŻæŸŕäÿžäžĚæŻ£äžčæăĞåĞĘåžŞäÿ■çŽĎdistutilsãĂĆæşĺæĎŔåęĆæđIJä¡ää¿ÌètŰè£ŹäžŻåŇĚïijŇ
çŤĺæĹůåŔŕèČ¡äÿ■èČ¡åőL'èčĚä¡ăçŽĎè¡ŕäżűïijŇéŹď'éİđäžŰäžňåůšçżŔäžŃåĚĹåőL'èčĚè£ĞæL'ĂéIJĂèęĄçŽĎå
æ■čåŻäåęĆæ■d'ïijŇä¡ăæŻt'åžŤèŕěæŮűåĹżèőřä¡ŔèűŁçőĂå■Ţèűŀåě¡çŽĎéĄŞçŘĘãĂĆ
æIJĂåě¡èől'ä¡ăçŽĎäžčçăĄä¡£çŤĺæăĞåĞĘçŽĎPython 3åőL'èčĚãĂĆ
åęĆæđIJåĚűäžŰåŇĚäź§éIJĂèęĄçŽĎèŕİïijŇåŔŕäžěéĂŽè£ĞäÿĂäÿłåŔŕéĂL'éąźæİěæŤŕæŇĄãĂĆ

åŕźäžŐæűL'åŘŁåĹřCæL'l'åśŢçŽĎäžčçăĄæL'ŞåŇĚäÿŐåĹĘåŔŚåŕśæŻt'åď'■æİĆçĆźäžĘãĂĆ
çňň15çńååŕźåĚşäžŐCæL'l'åśŢçŽĎè£ŹæŰźéİćç§ěèŕĘæIJL'äÿĂäžŻèŕęçżĘèőšèğčïijŇçL'źåĹńæŸŕåIJĺ15.2åřŔè

13 çňňå■ĄäÿĂçńăïijŽç¡ŚçżIJäÿŐWebçijŰçĺŃ

æIJňçńăæŸŕåĚşäžŐåIJĺç¡ŚçżIJåžŢçŤĺåŠŇåĹĘåÿČåijŔåžŢçŤĺäÿ■ä¡£çŤĺçŽĎåŘĎçğ■äÿżécŸãĂĆäÿżécŸ

Contents:

13.1 11.1 ä¡IJäÿžåőćæĹůçńŕäÿŐHTTPælJ■åŁąäžď'äžŠ

éŰőécŸ

ä¡ăéIJĂèęĄéĂŽè£ĞHTTPå■ŔèőőäžěåőćæĹůçńŕçŽĎæŰźåijŔèő£éŰőåď'Žçğ■æIJ■åŁąãĂĆä¿ŃåęĆïijŇäÿ

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎäžŃæČĚæİěèŕťïijŇéĂŽåÿÿä¡£çŤÍ `urllib.request` æĺąåİŮåřšåď§äžĘãĂĆä¿ŃåęĆïijŇåŘŚéĂĄäÿĂäÿłçőĂå■ŢçŽĎHTTP GETèřůæśĆåĹřè£IJçÍŃçŽĎæIJ■åŁąäÿŁïijŇåŘŕäžěè£ŹæăůåĄŽïijŽ

```
from urllib import request, parse

# Base URL being accessed
url = 'http://httpbin.org/get'

# Dictionary of query parameters (if any)
parms = {
   'name1' : 'value1',
   'name2' : 'value2'
}

# Encode the query string
querystring = parse.urlencode(parms)

# Make a GET request and read the response
u = request.urlopen(url+'?' + querystring)
resp = u.read()
```

åęĆædIJä¡ăéIJĂèęĄä¡£çŤÍPOSTæŰźæşŢåIJÍèřůæśĆäÿżä¡Şäÿ■åŘŚéĂĄæ§ěèŕćåŘĆæŢřïijŇåŘŕäžěåřĘåŘ `urlopen()` åĞ¡æŢřïijŇåřšåČŔè£ŹæăůïijŽ

```
from urllib import request, parse

# Base URL being accessed
url = 'http://httpbin.org/post'

# Dictionary of query parameters (if any)
parms = {
   'name1' : 'value1',
   'name2' : 'value2'
}

# Encode the query string
querystring = parse.urlencode(parms)

# Make a POST request and read the response
u = request.urlopen(url, querystring.encode('ascii'))
resp = u.read()
```

åęĆædIJä¡ăéIJĂèęĄåIJÍåŘŚåĞžçŽĎèřůæśĆäÿ■æŘŘä¿ŻäÿĂäžŻèĞłåőŽäźL’çŽĎHTTPåď’ťïijŇä¿ŃåęĆä£ `user-agent` å■Ůæőţ,åŘŕäžěåĹŻåžžäÿĂäÿłåŇĚåŘńå■Ůæőţå Ăijç ŽĎå■ŮåĚÿïijŇåźűåĹŻåžžäÿĂäÿłRequestå `urlopen()` ïijŇåęĆäÿŃïijŽ

```
from urllib import request, parse
...
```

```
# Extra headers
headers = {
    'User-agent' : 'none/ofyourbusiness',
    'Spam' : 'Eggs'
}

req = request.Request(url, querystring.encode('ascii'),␣
↪headers=headers)

# Make a request and read the response
u = request.urlopen(req)
resp = u.read()
```

åęĆædIJéIJĂèęĄäžd'äžŠçŽĎæIJ■åŁąæŕŤäÿŁéİćçŽĎä¿Ńå■ŘéČ¡èęĄåd'■æİĆïijŇäź§èőÿåžŤèŕěåŐżçIJŃç
requests åžŞïijĹhttps://pypi.python.org/pypi/requestsïijL'ãĂĆä¿ŃåęĆïijŇäÿŃéİćè£Źäÿłçd'žä¿ŃéĞĞçŤĺreques

```
import requests

# Base URL being accessed
url = 'http://httpbin.org/post'

# Dictionary of query parameters (if any)
parms = {
   'name1' : 'value1',
   'name2' : 'value2'
}

# Extra headers
headers = {
    'User-agent' : 'none/ofyourbusiness',
    'Spam' : 'Eggs'
}

resp = requests.post(url, data=parms, headers=headers)

# Decoded text returned by the request
text = resp.text
```

åĚşäžŐrequestsåžŞïijŇäÿĂäÿłåĂijå¿ŮäÿĂæŔŘçŽĎçL'źæĂğåřśæŸŕåőČèČ¡äžěåd'Žçğ■æŰźåijŔäżŐèŕůa
`resp.text` åÿęçżŹæĹŚäžňçŽĎæŸŕäžěUnicodeèğčçăĄçŽĎåŞ■åžŤæŰĞæIJñãĂĆä¡ĘæŸŕïijŇåęĆædIJåŐżèć
`resp.content` ïijŇářśäijŽå¿ŮåĹřåŐ§ąğŃçŽĎäžŇè£ŻåĹűæŢřæ■őãĂĆåŔęäÿĂæŰźéİćïijŇåęĆædIJèő£éŮć
`resp.json` ïijŇéĆčäźĹářśäijŽå¿ŮåĹřJSONæăijåijŔçŽĎåŞ■åžŤåĘĚåőźãĂĆ

äÿŃéİćè£Źäÿłçd'žä¿ŃåĹl'çŤÍ `requests` åžŞåŘŚèţůäÿĂäÿłHEAD-èŕůæśĆïijŇåźűäžŐåŞ■åžŤäÿ■æŔŘåŘŰåĞžäÿĂäžŻHTTPåd'ť'æŢřæ■őçŽĎå■Ůæőţïij Ž

```
import requests

resp = requests.head('http://www.python.org/index.html')
```

```
status = resp.status_code
last_modified = resp.headers['last-modified']
content_type = resp.headers['content-type']
content_length = resp.headers['content-length']
```

äÿŃéÌćæŸŕäÿĂäÿłåĹľçŤÍrequestséĂŽè£Ğå§žæIJňèŏď'èŕĄçŹżå¡ŢPypiçŽĎä¿Ńå■ŘïijŽ

```
import requests

resp = requests.get('http://pypi.python.org/pypi?:action=login',
                    auth=('user','password'))
```

äÿŃéÌćæŸŕäÿĂäÿłåĹľçŤÍrequestsåřĘHTTP cookiesäżŐäÿĂäÿłèŕůæśĆäijåéĂŠåĹřåŔęäÿĂäÿłçŽĎä¿Ńå■

```
import requests

# First request
resp1 = requests.get(url)
...

# Second requests with cookies received on first requests
resp2 = requests.get(url, cookies=resp1.cookies)
```

æIJĂåŘŐä¡Ęåźűéİđæ IJĂäÿ■éĞ■èęĄçŽĎäÿĂäÿłä¿Ńå■ŘæŸŕçŤÍrequestsäÿŁäijååĘĚåőźïijŽ

```
import requests
url = 'http://httpbin.org/post'
files = { 'file': ('data.csv', open('data.csv', 'rb')) }

r = requests.post(url, files=files)
```

èőÍèőž

åŕźäžŐçIJ§çŽĎå¿ĹçőĂå■ŢHTTPåőćæĹůçńŕäžčçăĄïijŇçŤÍåĘĚç¡őçŽĎ `urllib` æÍąåİŮéĂŽåÿÿåřśèűşåď§äžĘãĂĆä¡ĘæŸŕïijŇåęĆædIJä¡äèęĄåĄŽçŽĎäÿ■äžĚäžĚåŔłæŸŕçőĂå■ŢçŽĎGETæÍ `requests` åď'ğæŸ¿èžńæL'ŃçŽĎæŮűåĂŹäžĘãĂĆ

ä¿ŃåęĆïijŇåęĆædIJä¡ăåĘşåőŽåİŽæŇĄä¡£çŤÍæăĞåĞĚçŽĎçÍŃåžŔåžŞèĂŇäÿ■èĂČèŹŚåČŔ `requests` è£ŹæăůçŽĎçňňäÿL'æŰźåžŞïijŇéĆčäźĹäź§èőÿåřśäÿ■å¿Ůäÿ■ä¡£çŤÍåžŢåśĆçŽĎ `http.client` æÍąåİŮæİěåőđçŐřèĞłåůśçŽĎäžčçăĄãĂĆærŤæŰźèŕť'ïijŇäÿŃéÌćçŽĎäžčçăĄåśŢçđ'žäžĘåęĆä

```
from http.client import HTTPConnection
from urllib import parse

c = HTTPConnection('www.python.org', 80)
c.request('HEAD', '/index.html')
resp = c.getresponse()

print('Status', resp.status)
```

```
for name, value in resp.getheaders():
    print(name, value)
```

åŘŇæăůåIJřïijŇåęĆæđIJå£Ěéążçij ŰåĘŹæűL’åŔŁäžčçŘĘãĂĄèőd’èŕĄãĂĄcookiesäżěåŔŁåĚűäżŰäÿĂäžZ
urllib åřśæŸ¿å¿ŮçL’źåĹńåĹńæL’■åŠŇåŢřåŮęãĂĆæŕŤæŰźèŕt’ïijŇäÿŃéİćè£Źäÿłçd’žä¿Ńáőđç ŐřåIJĺPython

```
import urllib.request

auth = urllib.request.HTTPBasicAuthHandler()
auth.add_password('pypi','http://pypi.python.org','username',
 ↪'password')
opener = urllib.request.build_opener(auth)

r = urllib.request.Request('http://pypi.python.org/pypi?
 ↪:action=login')
u = opener.open(r)
resp = u.read()

# From here. You can access more pages using opener
...
```

åİęçŹ¡èŕt’ïijŇæL’ĂæIJL’çŽĎè£ŹäžŻæŞ■ä¡IJåIJĺ requests
åžŞäÿ■éČ¡åŔŸå¿ŮçőĂå■ŢçŽĎåd’ŽãĂĆ

åIJĺåijĂåŔŚè£ĞçĺŃäÿ■æţŃèŕŢHTTPåőćæĹůçńŕäžčçăĄåÿÿåÿÿæŸŕå¿Ĺäžd’äžžæšöäÿğçŽĎïijŇåŻääÿžæL
//httpbin.orgïijL’ãĂĆè£ŹäÿłçńŹçĆźäijŽæŐěæŤűåŘŚåĞžçŽĎèřůæśĆïijŇçĎűåŘŐäžěJSONçŽĎå¡ćåijŔåřĘçŻÿ

```
>>> import requests
>>> r = requests.get('http://httpbin.org/get?name=Dave&n=37',
...     headers = { 'User-agent': 'goaway/1.0' })
>>> resp = r.json
>>> resp['headers']
{'User-Agent': 'goaway/1.0', 'Content-Length': '', 'Content-Type': '
 ↪',
'Accept-Encoding': 'gzip, deflate, compress', 'Connection':
'keep-alive', 'Host': 'httpbin.org', 'Accept': '*/*'}
>>> resp['args']
{'name': 'Dave', 'n': '37'}
>>>
```

åIJĺèęĄåŘŇäÿĂäÿłçIJ§æ■čçŽĎçńŹçĆźè£ŻèąŇäžd’äžŠåL’■ïijŇåĚĹåIJĺ httpbin.org
è£ŹæăůçŽĎç¡ŚçńŹäÿŁåĄŽåőđéłŇåÿÿåÿÿæŸŕåŔŕåŘŰçŽĎåŁđæşŢãĂĆårđ’åĚűæŸŕå¡ŞæĹŚäžňéİćårź3æňąçŹź

år¡çőąæIJňèŁĆæšąæIJL’æűL’åŔŁïijŇ request åžŞè£Ÿårźèőÿåd’ŽénŸçžğçŽĎHTTPåőćæĹůçńŕå■Řèőő
requests æĺąåİŮçŽĎæŰĞæąčïijĹhttp://docs.python-requests.
org)èt’ĺéĞŘå¿ĹénŸïijĹåİęçŹ¡èŕt’æŕŤåIJĺè£Źç§■ç§■çŽĎäÿĂèŁĆçŽĎçŕĞáźĚäÿ■æL’ĂæŔŘä¿ŻçŽĎäżżä¡Ţä£ą

13.2 11.2 åĹŻåžžTCPæIJ■åŁąåŹÍ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłæIJ■åŁąåŹĺïijŇéĂŽè£ĞTCPå■Ŕèőőå ŠŇåőćæĹůçńŕéĂŽä£ąãĂĆ

èğčåĘşæŰźæąĹ

åĹŻåžžäÿĂäÿłTCPæIJ■åŁąåŹÍçŽĎäÿĂäÿłçőĂå■ŢæŰźæşŢæŸŕä¡£çŤÍ `socketserver` åžŞãĂĆä¿ŃåęĆïijŇäÿŃéÍćæŸŕäÿĂäÿłçőĂå■ŢçŽĎåžŤç■ŤæIJ■åŁąåŹÍïijŽ

```
from socketserver import BaseRequestHandler, TCPServer

class EchoHandler(BaseRequestHandler):
    def handle(self):
        print('Got connection from', self.client_address)
        while True:

            msg = self.request.recv(8192)
            if not msg:
                break
            self.request.send(msg)

if __name__ == '__main__':
    serv = TCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

åIJÍè£ŹæőţäžčçăĄäÿ■ïijŇä¡ăåőŽäźL'äžĘäÿĂäÿłçL'źæőŁçŽĎåď'ĎçŘĘçśżïijŇåőđçŐřäžĘäÿĂäÿł `handle()` æŰźæşŢïijŇçŤÍæİěäÿžåőćæĹůçńŕè£đæŐěæIJ■åŁąãĂĆ `request` åśđæĂğæŸŕåőćæĹůçńŕsocketïijŇ`client_address` æIJL'åőćæĹůçńŕåIJřåİĂãĂĆ äÿžäžĘæţŃèŕŢè£ŹäÿłæIJ■åŁąåŹÍïijŇè£ŘèąŇåőČåźűæL'ŞåijĂåŔęåď'ŰäÿĂäÿłPythonè£ŻçÍŃè£đæŐěè£Źäÿłæ

```
>>> from socket import socket, AF_INET, SOCK_STREAM
>>> s = socket(AF_INET, SOCK_STREAM)
>>> s.connect(('localhost', 20000))
>>> s.send(b'Hello')
5
>>> s.recv(8192)
b'Hello'
>>>
```

å¿Ĺåď'ŽæŮűåĂŹïijŇåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎåőŽäźL'äÿĂäÿłäÿ■åŘŇçŽĎåď'ĎçŘĘåŹĺãĂĆäÿŃéÍćæŸŕäÿĂ `StreamRequestHandler` å§žçśžåřĘäÿĂäÿłçśżæŰĞäžűæŐěåŔčæŤ¿ç¡őåIJĺåžŢåśĆsocketäÿŁçŽĎä¿Ńå■Ř

```
from socketserver import StreamRequestHandler, TCPServer

class EchoHandler(StreamRequestHandler):
    def handle(self):
        print('Got connection from', self.client_address)
```

```
        # self.rfile is a file-like object for reading
        for line in self.rfile:
            # self.wfile is a file-like object for writing
            self.wfile.write(line)

if __name__ == '__main__':
    serv = TCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

èőÍèőž

socketserver åŘŕäžěèőľ'æĹŚäžňå¿Ĺåőźæ ŸŞçŽĎåĹŻåžžçőÄå■ŢçŽĎTCPæIJ■åŁąåŹĺãĂĆ ä¡ĘæŸŕïijŇä¡äéIJĂèęĄæşĺæĎŘçŽĎæŸŕïijŇéżŸèőď'æČĚåĘţäÿŃè£Źçğ■æIJ■åŁąåŹĺæŸŕå■Ţçž£çĺŃçŽĎïijŇäÿ åęĆæđIJä¡äæČşåď'ĎçŘĘåď'ŽäÿłåőćæĺůçńŕïijŇåŘŕäžěåĹİåğŇåŇŰäÿĂäÿł ForkingTCPServer æĹŰèĂĚæŸŕ ThreadingTCPServer åŕźèśąãĂĆä¿ŃåęĆïijŽ

```
from socketserver import ThreadingTCPServer


if __name__ == '__main__':
    serv = ThreadingTCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

ä¡£çŤĺforkæĹŰçž£çĺŃæIJ■åŁąåŹĺæIJL'äÿłæ¡IJåIJĺéŮőécŸåřśæŸŕåőČäžňäijŽäÿžæŕŘäÿłåőćæĹůçńrè£đæ çŤśäžŐåőćæĹůçńrè£đæŐěæŢřæŸŕæšąæIJL'éŹŘåĹűçŽĎïijŇåŽäæ■d'äÿĂäÿłæĄűæĎŘçŽĎézŚåőćåŘŕäžěåŘŇ

åęĆæđIJä¡äæŃĚå£Čè£ŹäÿłéŮőécŸïijŇä¡ääŘŕäžěåĹŻåžžäÿĂäÿłécĎåĚĹåĹĘéĚ■åď'ğåřŘçŽĎåůěä¡IJçž£ç ä¡ääĚĹåĹŻåžžäÿĂäÿłæŹőéĂŽçŽĎéIđçž£çĺŃæIJ■åŁąåŹĺïijŇçĎűåŘŐåIJĺäÿĂäÿłçž£çĺŃæśääÿ■ä¡£çŤĺ serve_forever() æŰźæşŢæİěåŘŕåŁĺåőČäžñäĆ

```
if __name__ == '__main__':
    from threading import Thread
    NWORKERS = 16
    serv = TCPServer(('', 20000), EchoHandler)
    for n in range(NWORKERS):
        t = Thread(target=serv.serve_forever)
        t.daemon = True
        t.start()
    serv.serve_forever()
```

äÿĂèĹňæİěèőšïijŇäÿĂäÿł TCPServer åIJĺåőđä¿ŃåŇŰçŽĎæŮűåĂŹäijŽçżŚåőŽåźűæ£Äæť'żçŽÿåžŤçŽ socket ãĂĆ äÿ■è£ĞïijŇæIJL'æŮűåĂŹä¡äæČşéĂŽè£Ğèő¿ç¡őæ§ŘäžŽéĂL'éąźåŐżèřČæŢť'åžŢäÿŃçŽĎ *socket*' ïijŇåŘŕäžěèő¿ç¡őåŔĆæŢř bind_and_activate=False ãĂĆåęĆäÿŃïijŽ

```
if __name__ == '__main__':
    serv = TCPServer(('', 20000), EchoHandler, bind_and_
↪activate=False)
    # Set up various socket options
    serv.socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR,␣
↪True)
```

```
    # Bind and activate
    serv.server_bind()
    serv.server_activate()
    serv.serve_forever()
```

äÿŁéÍćçŽĎ socket éĂĽ'éąźæŸŕäÿĂäÿłéÍđåÿÿæŹőéĄ■çŽĎéĚ■ç¡őéąźïijŇåőČåĚĄèőÿæIJ■åŁąåŹĺéĞ■æ
çŤśäžŐèęĄèćńçżŔåÿÿä¡£çŤĺåĹřïijŇåőČèćńæŤ¿ç¡őåĹřçśżåŔŸéĞŔäÿ■ïijŇåŔŕäžěçŻť'æŐěåIJÍ
TCPServer äÿŁéÍćèő¿ç¡õãĂĆ åIJÍåőđä¿ŃåŇŰæIJ■åŁąåŹĺçŽĎæŮűåĂŹåŐżèő¿ç¡őåőČçŽĎåĂijïijŇåęĆäÿŃ

```
if __name__ == '__main__':
    TCPServer.allow_reuse_address = True
    serv = TCPServer(('', 20000), EchoHandler)
    serv.serve_forever()
```

åIJÍäÿŁéÍćçđ'žä¿Ńäÿ■ïijŇæĹŚäžňæijŤçđ'žäžĘäÿđ'çğ■äÿ■åŘŇçŽĎåđ'ĎçŘĘåŹĺå§žçśżïijĹ
BaseRequestHandler åŠŇ StreamRequestHandler ïijL'ãĂĆ
StreamRequestHandler æŻť'åŁăçĄţæť'żçĆźïijŇèČ¡éĂŽè£Ğèŏ¿ç¡őåĚűäžŰçŽĎçśżåŔŸéĞŔæÍěæŤŕæŇA

```
import socket

class EchoHandler(StreamRequestHandler):
    # Optional settings (defaults shown)
    timeout = 5                         # Timeout on all socket␣
↪operations
    rbufsize = -1                       # Read buffer size
    wbufsize = 0                        # Write buffer size
    disable_nagle_algorithm = False     # Sets TCP_NODELAY socket␣
↪option
    def handle(self):
        print('Got connection from', self.client_address)
        try:
            for line in self.rfile:
                # self.wfile is a file-like object for writing
                self.wfile.write(line)
        except socket.timeout:
            print('Timed out!')
```

æIJĂåŘŐïijŇè£ŸéIJĂèęĄæşĺæĎŔçŽĎæŸŕåůĺåď'ğéČĺåĹĘPythonçŽĎéńŸåśÇç¡ŚçżIJæĺąåİŮïijĹærŤåęĆI
RPCç■L'ïijL'éČ¡æŸŕåżžçńŇåIJÍ socketserver åŁ§èČ¡äźŇäÿŁãĂĆ
äź§åřśæŸŕèŕť'ïijŇçŻť'æŐëä¡£çŤĺ socket åžŞæİěåődçŐřæIJ■åŁąåŹĺäź§åźűäÿ■æŸŕå¿ĹéŽ¿ãĂĆ
äÿŃéÍćæŸŕäÿĂäÿłä¡£çŤĺ socket çŻť'æŐëçijŰçĺŃåődçŐřçŽĎäÿĂäÿłæIJ■åŁąåŹĺçőĂå■Ţä¿Ńå■ŘïijŽ

```
from socket import socket, AF_INET, SOCK_STREAM

def echo_handler(address, client_sock):
    print('Got connection from {}'.format(address))
    while True:
        msg = client_sock.recv(8192)
        if not msg:
            break
```

```
        client_sock.sendall(msg)
    client_sock.close()

def echo_server(address, backlog=5):
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(address)
    sock.listen(backlog)
    while True:
        client_sock, client_addr = sock.accept()
        echo_handler(client_addr, client_sock)

if __name__ == '__main__':
    echo_server(('', 20000))
```

13.3 11.3 åĹŻåżžUDPæIJ■åŁąåŹÍ

éŮőécŸ

ä¡ăæČşåőđçŐřäÿĂäÿłå§žäžŐUDPå■Řèőőç ŽĎæIJ■åŁąåŹÍæİěäÿŐåőćæĹůçńŕéĂŽä£ąãĂĆ

èğčåĘşæŰźæąĹ

èů§TCPäÿĂæăůïijŇUDPæIJ■åŁąåŹÍäź§åŘŕäžěéĂŽè£Ğä¡£çŤÍ `socketserver` åžŞå¿Ĺåőźæ ŸŞçŽĎèćńåĹŻåżžãĂĆ ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎæŮűéŮťæIJ■åŁąåŹÍïijŽ

```
from socketserver import BaseRequestHandler, UDPServer
import time

class TimeHandler(BaseRequestHandler):
    def handle(self):
        print('Got connection from', self.client_address)
        # Get message and client socket
        msg, sock = self.request
        resp = time.ctime()
        sock.sendto(resp.encode('ascii'), self.client_address)

if __name__ == '__main__':
    serv = UDPServer(('', 20000), TimeHandler)
    serv.serve_forever()
```

èů§äźŃåL’■äÿĂæăůïijŇä¡ăåĚĹåőŽäźL’äÿĂäÿłåőđçŐř `handle()` çL’źæőŁæŰźæşŢçŽĎçśžïijŇäÿžåőćæĹůçńŕè£đæŐěæIJ■åŁąãĂĆ è£ŹäÿłçśżçŽĎ `request` åśđæĂğæŸŕäÿĂäÿłåŇĚåŘńäžĘæŢřæ■őæŁěåŠŇåžŢåśĆsock-etåŕźèśąçŽĎåĚČçżĎãĂĆ`client_address` åŇĚåŘńäžĘåőćæĹůçńŕåIJřåİĂãĂĆ

æĹŚäžňæİěæţŃèŕŢäÿŃè£ŹäÿłæIJ■åŁąåŹÍïijŇéęŰåĚĹè£ŘèąŇåőČïijŇçĎűåŘŐæL’Şåij ÄåŔęåd’ŰäÿĂäÿł

```
>>> from socket import socket, AF_INET, SOCK_DGRAM
>>> s = socket(AF_INET, SOCK_DGRAM)
>>> s.sendto(b'', ('localhost', 20000))
0
>>> s.recvfrom(8192)
(b'Wed Aug 15 20:35:08 2012', ('127.0.0.1', 20000))
>>>
```

èőÍèőž

äÿĂäÿłåĚÿåđŃçŽĎUDPæIJ■åŁąåŹÍæŐěæŤűåĹřè¿¿çŽĎæŢřæ■őæŁě(æűĹæĄŕ)åŠŇåőćæĹůçńŕåIJřåİĂãĂ
åőČèęĄçżŹåőćæĹůçńŕåŻđåŔŚäÿĂäÿłæŢřæ■őæŁěãĂĆåŕźäžŐæŢřæ■őæŁěçŽĎäijåéĂĄïijŇ
ä¡ăåžŤèŕěä¡£çŤÍsocketçŽĎ `sendto()` åŠŇ `recvfrom()` æŰźæşŢãĂĆ
åř¡çőąäijåçż§çŽĎ `send()` åŠŇ `recv()` äź§åŔŕäžěè¿¿åĹřåŘŇæăůçŽĎæŢĹæđIJïijŇ
ä¡ĘæŸŕåĽ■éİćçŽĎäÿd'äÿłæŰźæşŢåŕźäžŐUDPè£đæŐěèĂŇèÍĂæŻt'æŹőéĄ■ãĂĆ

çŤśäžŐæšąæIJL'åžŢåśĆçŽĎè£đæŐěïijŇUPDæIJ■åŁąåŹÍçŻÿåŕźäžŐTCPæIJ■åŁąåŹÍæİěèőšåőđçŐřętůæİ
äÿ■è£GïijŇUDPåd'l'çŤ§æŸŕäÿ■åŔŕéİăçŽĎïijĹåŻääÿžéĂŽä£ąæšąæIJL'åżžçńŃè£đæŐěïijŇæűĹæĄŕåŔŕèČ¡äÿ
åŻăæ■d'éIJĂèęĄçŤśä¡äèĞłåůśæİěåĘşåőŽèřěæĂŐæăůåd'ĎçŘĘäÿćåd'śæűĹæĄŕçŽĎæČĚåĘţãĂĆè£Źäÿłåůšçżl
äÿ■è£ĞéĂŽåÿÿæİèèŕt'ïijŇåęĆæđIJåŔŕéİăæĂğåŕźäžŐä¡ăçÍŃåžŔå¿ĹéĞ■èęĄïijŇä¡ăéIJĂèęĄåĂ§åŁl'äžŐåžŔåÍ
UDPéĂŽåÿÿèćńçŤÍåIJÍéĆčäžŻåŕźäžŐåŔŕéİăäijàè¿ŞèęĄæśĆäÿ■æŸŕå¿ĹéńŸçŽĎåIJžåŘĹãĂĆä¿ŃåęĆïijŇåIJÍå
æŮăéIJĂè£ŤåŻđæĄćåd'■äÿćåd'śçŽĎæŢřæ■őåŇĚïijĹçÍŃåžŔåŘłéIJĂçőĂå■ŢçŽĎå£¡çŢěáőČåźűçżğçż■åŘŚåĹ

`UDPServer` çśżæŸŕå■Ţçž£çÍŃçŽĎïijŇäź§åřśæŸŕèŕt'äÿĂæňąåŘłèČ¡äÿžäÿĂäÿłåőćæĹůçńŕè£đæŐěæIJ■
åőđéŹĚä¡£çŤÍäÿ■ïijŇè£ŹäÿłæŮăèőžæŸŕåŕźäžŐUDPè£ŸæŸŕTCPéČ¡äÿ■æŸŕäżĂäźĹåd'ğéŮőécŸãĂĆ
åęĆæđIJä¡ăæČşèęĄåźűåŘŚæŞ■ä¡IJïijŇåŔŕäžěåőđä¿ŃåŇŰäÿĂäÿł `ForkingUDPServer`
æĹŰ `ThreadingUDPServer` åŕźèśąïijŽ

```
from socketserver import ThreadingUDPServer

   if __name__ == '__main__':
    serv = ThreadingUDPServer(('',20000), TimeHandler)
    serv.serve_forever()
```

çŻt'æŐěä¡£çŤÍ `socket` æİěåőđçŐřäÿĂäÿłUDPæIJ■åŁąåŹÍäź§äÿ■éŽ¿ïijŇäÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŽ

```
from socket import socket, AF_INET, SOCK_DGRAM
import time

def time_server(address):
    sock = socket(AF_INET, SOCK_DGRAM)
    sock.bind(address)
    while True:
        msg, addr = sock.recvfrom(8192)
        print('Got message from', addr)
        resp = time.ctime()
        sock.sendto(resp.encode('ascii'), addr)
```

```
if __name__ == '__main__':
    time_server(('', 20000))
```

13.4 11.4 éĂŽè£ĞCIDRåIJřåİĂçŤ§æĹŘåŕźåžŤçŽĎIPåIJřåİĂéŻĘ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłCIDRç¡ŚçżIJåIJřåİĂæŕŤåęĆâĂIJ123.45.67.89/27âĂİïijŇä¡ăæČşåřĘåĚűè¡ňæ■ćæĹŘåőČa
ïijĹæŕŤåęĆïijŇâĂIJ123.45.67.64âĂİ, âĂIJ123.45.67.65âĂİ, âĂę, âĂIJ123.45.67.95âĂİ)ïijL'

èğčåĘşæŰźæąĹ

åŘŕäżěä¡£çŤÍ `ipaddress` æĺąåİŮå¿ĹåőźæŸŞçŽĎåőđçŐřè£ŹæăůçŽĎèőąçőŮãĂĆä¿ŃåęĆïijŽ

```
>>> import ipaddress
>>> net = ipaddress.ip_network('123.45.67.64/27')
>>> net
IPv4Network('123.45.67.64/27')
>>> for a in net:
...     print(a)
...
123.45.67.64
123.45.67.65
123.45.67.66
123.45.67.67
123.45.67.68
...
123.45.67.95
>>>

>>> net6 = ipaddress.ip_network('12:3456:78:90ab:cd:ef01:23:30/125')
>>> net6
IPv6Network('12:3456:78:90ab:cd:ef01:23:30/125')
>>> for a in net6:
...     print(a)
...
12:3456:78:90ab:cd:ef01:23:30
12:3456:78:90ab:cd:ef01:23:31
12:3456:78:90ab:cd:ef01:23:32
12:3456:78:90ab:cd:ef01:23:33
12:3456:78:90ab:cd:ef01:23:34
12:3456:78:90ab:cd:ef01:23:35
12:3456:78:90ab:cd:ef01:23:36
12:3456:78:90ab:cd:ef01:23:37
>>>
```

`Network` äź§åĚĄèőÿåČŔæŢřçżĎäÿĂæăůçŽĎçť'ćåijŢåŔŰåĂijïijŇä¿ŃåęĆïijŽ

```
>>> net.num_addresses
32
>>> net[0]

IPv4Address('123.45.67.64')
>>> net[1]
IPv4Address('123.45.67.65')
>>> net[-1]
IPv4Address('123.45.67.95')
>>> net[-2]
IPv4Address('123.45.67.94')
>>>
```

åŔęåď'ŰïijŇä¡ăè£ŸåŔŕäžěæL'ğèąŇç¡ŚçżIJæĹŘåŚŸæčĂæ§ěäźŃçśżçŽĎæŞ■ä¡IJïijŽ

```
>>> a = ipaddress.ip_address('123.45.67.69')
>>> a in net
True
>>> b = ipaddress.ip_address('123.45.67.123')
>>> b in net
False
>>>
```

äÿĂäÿłIPåIJřåİĂåŠŇç¡ŚçżIJåIJřåİĂèČ¡éĂŽè£ĞäÿĂäÿłIPæŐěåŔčæİěæŇĞåőŽïijŇä¿ŃåęĆïijŽ

```
>>> inet = ipaddress.ip_interface('123.45.67.73/27')
>>> inet.network
IPv4Network('123.45.67.64/27')
>>> inet.ip
IPv4Address('123.45.67.73')
>>>
```

èőÍèőž

ipaddress æÍąåİŮæIJL'å¿Ĺåď'ŽçśżåŔŕäžěèąÍçď'žIPåIJřåİĂãĂĄç¡ŚçżIJåŠŇæŐěåŔčãĂĆ
å¡Şä¡ăéIJĂèęĄæŞ■ä¡IJç¡ŚçżIJåIJřåİĂïijĹæŕŤåęĆèğčædŘãĂĄæL'Şå■řãĂĄéłŇèŕĄç■L'ïijL'çŽĎæŮűåĂŹäijŽå

èęĄæşÍæĎŔçŽĎæŸŕïijŇipaddress æÍąåİŮèů§åĚűäžŰäÿĂäžŽåŠŇç¡ŚçżIJçŽÿåĚşçŽĎæÍąåİŮæŕŤåęĆ
socket åžŞäžď'éŻĘå¿ĹåřŚãĂĆ æL'ĂäžěïijŇä¡ääÿ■èČ¡ä¡£çŤÍ IPv4Address
çŽĎåőđä¿ŃæİěäžčæŻ£äÿĂäÿłåIJřåİĂå■ŮçňęäÿšïijŇä¡ăéęŰåĚĹå¿ŮæŸ¿åijŔçŽĎä¡£çŤÍ
str() è¡ňæ■ćåőČãĂĆä¿ŃåęĆïijŽ

```
>>> a = ipaddress.ip_address('127.0.0.1')
>>> from socket import socket, AF_INET, SOCK_STREAM
>>> s = socket(AF_INET, SOCK_STREAM)
>>> s.connect((a, 8080))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Can't convert 'IPv4Address' object to str implicitly
```

```
>>> s.connect((str(a), 8080))
>>>
```

æŻť åď ŽçŻÿåĚşåĘĚåőźïijŇèŕůåŔĆèĂČ An Introduction to the ipaddress Module

13.5 11.5 åĹŻåžžäÿĂäÿłçőĂå■ŢçŽĎRESTæŐěåŔč

éŮőécŸ

ä¡ăæČşä¡£çŤĺäÿĂäÿłçőĂå■ŢçŽĎRESTæŐěåŔčéĂŽè£Ğç¡ŚçżIJè£IJçÍŃæŐğåĹűæĹŰèő£éŮőä¡ăçŽĎåžŤ

èğčåĘşæŰźæąĹ

æđĎåžžäÿĂäÿłRESTéčŐæăijçŽĎæŐěåŔčæIJĂçőĂå■ŢçŽĎæŰźæşŢæŸŕåĹŻåžžäÿĂäÿłå§žäžŐWSGIæă 3333ïijL'çŽĎå¿ĹářŔçŽĎåžŞïijŇäÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŽ

```
# resty.py

import cgi

def notfound_404(environ, start_response):
    start_response('404 Not Found', [ ('Content-type', 'text/plain
↪') ])
    return [b'Not Found']

class PathDispatcher:
    def __init__(self):
        self.pathmap = { }

    def __call__(self, environ, start_response):
        path = environ['PATH_INFO']
        params = cgi.FieldStorage(environ['wsgi.input'],
                                  environ=environ)
        method = environ['REQUEST_METHOD'].lower()
        environ['params'] = { key: params.getvalue(key) for key in␣
↪params }
        handler = self.pathmap.get((method,path), notfound_404)
        return handler(environ, start_response)

    def register(self, method, path, function):
        self.pathmap[method.lower(), path] = function
        return function
```

äÿžäžĘä¡£çŤĺè£ŹäÿłèřČåžęåŹĺïijÑä¡ääŔłéIJÄèęĄçijŰåĘŹäÿ■åŘŇçŽĎåď ĎçŘĘåŹĺïijÑářšåČŔäÿŃéİćè£

```
import time

_hello_resp = '''\
```

```
<html>
  <head>
     <title>Hello {name}</title>
   </head>
   <body>
     <h1>Hello {name}!</h1>
   </body>
</html>'''

def hello_world(environ, start_response):
    start_response('200 OK', [ ('Content-type','text/html')])
    params = environ['params']
    resp = _hello_resp.format(name=params.get('name'))
    yield resp.encode('utf-8')

_localtime_resp = '''\
<?xml version="1.0"?>
<time>
  <year>{t.tm_year}</year>
  <month>{t.tm_mon}</month>
  <day>{t.tm_mday}</day>
  <hour>{t.tm_hour}</hour>
  <minute>{t.tm_min}</minute>
  <second>{t.tm_sec}</second>
</time>'''

def localtime(environ, start_response):
    start_response('200 OK', [ ('Content-type', 'application/xml')␣
↪])
    resp = _localtime_resp.format(t=time.localtime())
    yield resp.encode('utf-8')

if __name__ == '__main__':
    from resty import PathDispatcher
    from wsgiref.simple_server import make_server

    # Create the dispatcher and register functions
    dispatcher = PathDispatcher()
    dispatcher.register('GET', '/hello', hello_world)
    dispatcher.register('GET', '/localtime', localtime)

    # Launch a basic server
    httpd = make_server('', 8080, dispatcher)
    print('Serving on port 8080...')
    httpd.serve_forever()
```

èęĄæţŃèŕŢäÿŃè£ŹäÿłæIJ■åŁąåŹĺïijŇä¡ăåŔŕäžěä¡£çŤĺäÿĂäÿłæţŔèğĹåŹĺæĹŰ `urllib` åŠŇåőČäžďäžŠãĂĆä¿ŃåęĆïijŽ

```
>>> u = urlopen('http://localhost:8080/hello?name=Guido')
>>> print(u.read().decode('utf-8'))
<html>
  <head>
     <title>Hello Guido</title>
   </head>
   <body>
     <h1>Hello Guido!</h1>
   </body>
</html>

>>> u = urlopen('http://localhost:8080/localtime')
>>> print(u.read().decode('utf-8'))
<?xml version="1.0"?>
<time>
  <year>2012</year>
  <month>11</month>
  <day>24</day>
  <hour>14</hour>
  <minute>49</minute>
  <second>17</second>
</time>
>>>
```

èőÍèőž

åIJÍçijŰåĘŹRESTæŐěåŔčæŮűïijŇéĂŽåÿÿéČ¡æŸŕæIJ■åŁąäžŐæŹőéĂŽçŽĎHTTPèŕůæśĆãĂĆä¡ĘæŸŕè
è£ŹäžŻæŢřæ■őäžěåŘĎçğ■æăĞåĞĘæăijåijŔçijŰçăĄïijŇæŕŤåęĆXMLãĂĄJSONæĹŰCSVãĂĆ
åř¡çőąçÍŃåžŔçIJŃäÿŁåŐžå¿ĹçőĂå■ŢïijŇä¡ĘæŸŕäžěè£Źçğ■æŰźåijŔæŔŘä¿ŻçŽĎAPIåŕźäžŐå¿Ĺåď'ŽåžŤçŤĺç

ä¿ŃåęĆïijŇéŢ£æIJ§è£ŘèąŇçŽĎçÍŃåžŔåŔŕèČ¡äijŽä¡£çŤĺäÿĂäÿłREST
APIæÌěåőđçŐřçŻŚæŐğæĹŰèŕŁæŰ■ãĂĆ åď'ğæŢřæ■őåžŤçŤĺçÍŃåžŔåŔŕäžěä¡£çŤĺRESTæÌěæđĎåžžäÿĂäÿłæ
RESTè£ŸèČ¡çŤĺæÌěæŐğåĹűçąňäžűěő¿åď'ĞæŕŤåęĆæIJžåŹĺäžžãĂĄäijăæĎ$åŹĺãĂĄåůěåŐĆæĹŰçĄŕæşąãĂĆ
æŻť'éĞ■èęĄçŽĎæŸŕïijŇREST APIåůšçżŔèćńåď'ğéĞŘåőćæĹůçńŕçijŰçÍŃçŐŕåćČæĽ'ĂæŤŕæŇĄïijŇæŕŤåęĆ-
Javascript, Android, iOSç■L'ãĂĆ åŽăæ■d'ïijŇåĹl'çŤĺè£Źçğ■æŐěåŘčåŔŕäžěèől'ä¡ăåijĂåŘŠåĞžæŻť'åŁăåď'■æ

äÿžäžĘåőđçŐřäÿĂäÿłçőĂå■ŢçŽĎRESTæŐěåŘčïijŇä¡ăåŔłéIJÀèől'ä¡ăçŽĎçÍŃåžŔäžčçăĄæżąèűşPython
WSGIèćńæăĞåĞĘåžŞæŤŕæŇĄïijŇåŘŇæŰűäź§èćńçżÌåď'ğéČĺåĹEçňňäÿĽ'æŰźwebæąĘæđűæŤŕæŇĄãĂĆ
åŽăæ■d'ïijŇåęĆæđIJä¡ăçŽĎäžčçăĄéĄţå¿łè£ŹäÿłæăĞåĞĘïijŇåIJĺåŘŐéÍćçŽĎä¡£çŤĺè£ĞçÍŃäÿ■åřśäijŽæŻť'åŁ

åIJÍWSGIäÿ■ïijŇä¡ăåŔŕäžěåČŔäÿŃéÍćè£ŹæăůçžęåőŽçŽĎæŰźåijŔäžěäÿĂäÿłåŔŕèŘçŤĺåŕźèśąå¡ćåijŔæÍ

```
import cgi

def wsgi_app(environ, start_response):
    pass
```

environ åśđæĂğæŸŕäÿĂäÿłå■ŮåĚÿïijŇåŇĚåŘńäžĘäžŐwebæIJ■åŁąåŹĺåęĆA-
pache[åŔĆèĂČInternet RFC 3875]æŔŘä¿ŽçŽĎCGIæŐěåŘčäÿ■èŐůåŔŰçŽĎåĂijăĂĆ
èęĄåřĘè£ŹäžŻäÿ■åŘŇçŽĎåĂijæŔŘåŔŰåĞžæÌěïijŇä¡ăåŔŕäžěåČŔè£ŹäźĹè£ŹæăůåĘŹïijŽ

```
def wsgi_app(environ, start_response):
    method = environ['REQUEST_METHOD']
    path = environ['PATH_INFO']
    # Parse the query parameters
    params = cgi.FieldStorage(environ['wsgi.input'],␣
↪environ=environ)
```

æĹŚäżňåśŢçď žäžĘäÿĂäžŻåÿÿèğĄçŽĎåĂijãĂĆenviron['REQUEST_METHOD'] äžčèąĺèŕůæśĆçśżåđŃåęĆGETãĂĄPOSTãĂĄHADç■L'ãĂĆ environ['PATH_INFO'] èąĺçď'žèćńèŕůæśĆèţĎæžŘçŽĎèůŕå¿ĎãĂĆ èřČçŤÍ cgi.FieldStorage() åŔŕäžëäżŐèŕůæśĆäÿ■æŔŘåŔŰæ§ěèŕćåŔĆæŢřåźűåřĘåőČäżňæŤ¿åĚëäÿĂäÿłçśżå■ŮåĚÿåŕźèśąäÿ■äžëä¿£åŘŐ

start_response åŔĆæŢřæŸŕäÿĂäÿłäÿžäžĘåĹİåğŃåŇŰäÿĂäÿłèŕůæśĆåŕźèśąèĂŇå£Ěéążèćńèř ČçŤÍçňňäÿĂäÿłåŔĆæŢřæŸŕè£ŤåŻđçŽĎHTTPçŁűæĂĄåĂijïijŇçňňäžŇäÿłåŔĆæŢřæŸŕäÿĂäÿł(åŘ■,åĂij)åĚČçżĎ

```
def wsgi_app(environ, start_response):
    pass
    start_response('200 OK', [('Content-type', 'text/plain')])
```

äÿžäžĘè£ŤåŻđæŢřæ■őïijŇäÿĂäÿłWSGIçÍŃåžŔå£Ěéążè£ŤåŻđäÿĂäÿłå■ŮèŁĆå■ŮçňęäÿšåžŔåĹŮãĂĆåĚ

```
def wsgi_app(environ, start_response):
    pass
    start_response('200 OK', [('Content-type', 'text/plain')])
    resp = []
    resp.append(b'Hello World\n')
    resp.append(b'Goodbye!\n')
    return resp
```

æĹŰèĂĚïijŇä¡ăè£ŸåŔŕäžëä¡£çŤÍ yield ïijŽ

```
def wsgi_app(environ, start_response):
    pass
    start_response('200 OK', [('Content-type', 'text/plain')])
    yield b'Hello World\n'
    yield b'Goodbye!\n'
```

è£ŹéĞŇèęĄåijžèřČçŽĎäÿĂçĆźæŸŕæIJĂåŘŐè£ŤåŻđçŽĎå£ĚéążæŸŕå■ŮèŁĆå■Ůçňęäÿšã ĂĆåęĆæđIJè£ å¡ŞçĎűïijŇåźűæšąæIJL'èęĄæśĆä¡ăè£ŤåŻđçŽĎäÿĂåőŽæŸŕæŰĞæIJñïijŇä¡ăåŔŕäžëå¿Ĺè¡żæİ¿çŽĎçijŰåĘŹäÿ

år¡çőąWSGIçÍŃåžŔéĂŽåÿÿècńåőŽäźL'æĹŘäÿĂäÿłåĞ¡æŢřïijŇäÿ■è£Ğä¡ăäź§åŔŕäžëä¡£çŤÍçśżåőđä¿ŃæİÍ __call__() æŰźæşŢãĂĆä¿ŃåęĆïijŽ

```
class WSGIApplication:
    def __init__(self):
        ...
    def __call__(self, environ, start_response)
       ...
```

æĹŚäżňåůšçżŔåIJÍäÿŁéİćä¡£çŤÍè£Źçğ■æŁĂæIJŕåĹŻåžž PathDispatcher çśżãĂĆ è£ŹäÿłåĹĘåŘŠåŹÍäžĚäžĚåŘłæŸŕçőąçŘĘäÿĂäÿłå■ŮåĚÿïijŇårĘ(æŰźæşŢ,èůŕå¿Ď)åŕźæŸăåřĎåĹråď'ĎçŘĘåŹÍ

å¡ŞäÿĂäÿłèřůæśĆåĹřæİěæŮűïijŇåőČçŽĎæŰźæşŢåŠŇèůřå¿ĎèćńæŕŕåŕŰåĞžæİěïijŇçĎűåŘŐèćńåĹĘåŔŚåĹ
åŔęåď´ŰïijŇäżżä¡Ţæ§ěèŕćåŔŸéĞŔäijŽèćńèğčæđŘåŘŐæŤ¿åĹřäÿĂäÿłå■ŮåĚÿäÿ■ïijŇäžě
`environ['params']` å¡ćåijŕå■ŸåĆĺãĂĆ åŘŐéİćè£Źäÿłæ■ěéłď´åď´łåÿÿèğĄïijŇæĽĂäżěåżżèőőä¡ăåIJĺåĹĘ
ä¡£çŤĺåĹĘåŔŚåŹĺçŽĎæŮűåĂŹïijŇä¡ăåŔłéIJĂçőĂå■ŢçŽĎåĹŻåżżäÿĂäÿłåőđä¿ŃïijŇçĎűåŘŐéĂŽè£ĞåőČæş
çijŰåĘŹè£ŹäżŽåĞ¡æŢřåžŤèŕěèűĚçžğçőĂå■ŢäžĘïijŇåŔłèęĄä¡ăéĄţå¿ł
`start_response()` åĞ¡æŢřçŽĎçijŰåĘŹèğĎåĹŹïijŇåźűäÿŤæIJĂåŘŐè£ŤåŻđå■ŮèŁĆå■Ůçňęäÿšå■şåŔŕã

å¡ŞçijŰåĘŹè£Źçğ■åĞ¡æŢřçŽĎæŮűåĂŹè£ŸéIJĂæşłæĎŔçŽĎäÿĂçĆźåřśæŸŕåŕźäžŐå■Ůçňęäÿšæĺąæİ£çŽ
æšąäžżæĎ£æĎŔåĘŹéĆčçğ■åĹřåď´ĎæůůåŘĹçİĂ `print()` åĞ¡æŢř ãĂĄXM-
LåŠŇåď´ğéĞŔæăijåijŔåŇŰæŞ■ä¡IJçŽĎäžčçăĄãĂĆ æĹŚäžňäÿŁéİćä¡£çŤĺäžĘäÿĽ´åijŢåŔůåŇĚåŔńçŽĎécĎåĚ
è£Źçğ■æŰźåijŔçŽĎåŔŕäžěèőľ´æĹŚäžňå¿Ĺåőźæ˘ŞçŽĎåIJĺäžěåŘŐä£őæŤźè¿ŞåĞžæăijåijŔ(åŔłéIJĂèęĄä£őæ

æIJĂåŘŐïijŇä¡£çŤĺWSGIè£ŸæIJĽäÿĂäÿłå¿ĹéĞ■èęĄçŽĎéČĺåĹĘåřśæŸŕæšąæIJĽäžĂäźĹåIJřæŰźæŸŕé
åŻääÿžæăĞåĞĘåŕźäžŐæIJ■åŁąåŹĺåŠŇæąĘæđűæŸŕäÿ■çńŇçŽĎïijŇä¡ăåŔŕäžěåřĘä¡ăçŽĎçĺŇåžŔæŤ¿åĚěäżżä
æĹŚäžňä¡£çŤĺäÿŃéİćçŽĎäžčçăĄæţŃèŕŢæţŃèŕŢæIJňèŁĆäžčçăĄïijŽ

```
if __name__ == '__main__':
    from wsgiref.simple_server import make_server

    # Create the dispatcher and register functions
    dispatcher = PathDispatcher()
    pass

    # Launch a basic server
    httpd = make_server('', 8080, dispatcher)
    print('Serving on port 8080...')
    httpd.serve_forever()
```

äÿŁéİćäžčçăĄåĹŻåżżäžĘäÿĂäÿłçőĂå■ŢçŽĎæIJ■åŁąåŹĺïijŇçĎűåŘŐä¡ăåŕśåŔŕäžěæİěæţŃèŕŢäÿŃä¡ăçŽĎ
æIJĂåŘŐïijŇå¡Şä¡ăåĞĘåď´Ğè£ŻäÿĂæ■ěæĽľ´åśŢä¡ăçŽĎçĺŇåžŔçŽĎæŮűåĂŹïijŇä¡ăåŔŕäžěä£őæŤźè£Źäÿłäžč

WSGIæIJňèžńæŸŕäÿĂäÿłå¿ĹåŕŔçŽĎæăĞåĞĘãĂĆåŻăæ■ď´åőČåźűæšąæIJĽæŔŘä¿ŻäÿĂäžŻénŸçžğçŽĎ
è£ŹäžŻä¡ăèĞłåůśåőđçŐřęţůæİěäź§äÿ■éŽ¿ãĂĆäÿ■è£ĞåęĆæđIJä¡ăæČşèęĄæŻť´åď´ŽçŽĎæŤŕæŇĄïijŇåŔŕäžěè
`WebOb` æĹŰèĂĚ `Paste`

13.6 11.6 éĂŽè£ĞXML-RPCåőđçŐřçőĂå■ŢçŽĎè£IJçĺŃèřČçŤĺ

éŮőécŸ

ä¡ăæČşæĽ¿åĹřäÿĂäÿłçőĂå■ŢçŽĎæŰźåijŔåŐżæĽğèąŇè£ŘèąŇåIJĺè£IJçĺŃæIJžåŹĺäÿŁéİćçŽĎPythonçĺ

èğčåĘşæŰźæąĹ

åőđçŐřäÿĂäÿłè£IJçĺŃæŰźæşŢèřČçŤĺçŽĎæIJĂçőĂå■ŢæŰźåijŔæŸŕä¡£çŤĺXML-
RPCãĂĆäÿŃéİćæĹŚäžňæijŤçď´žäÿĂäÿŃäÿĂäÿłåőđçŐřäžĘéŤő-
åĂijå■ŸåĆĺåŁ§èČ¡çŽĎçőĂå■ŢæIJ■åŁąåŹĺïijŽ

```
from xmlrpc.server import SimpleXMLRPCServer

class KeyValueServer:
    _rpc_methods_ = ['get', 'set', 'delete', 'exists', 'keys']
    def __init__(self, address):
        self._data = {}
        self._serv = SimpleXMLRPCServer(address, allow_none=True)
        for name in self._rpc_methods_:
            self._serv.register_function(getattr(self, name))

    def get(self, name):
        return self._data[name]

    def set(self, name, value):
        self._data[name] = value

    def delete(self, name):
        del self._data[name]

    def exists(self, name):
        return name in self._data

    def keys(self):
        return list(self._data)

    def serve_forever(self):
        self._serv.serve_forever()

# Example
if __name__ == '__main__':
    kvserv = KeyValueServer(('', 15000))
    kvserv.serve_forever()
```

äÿŃéÌćæĹŚäżñäżŐäÿĂäÿłåőćæĹůçńŕæIJžåŹĺäÿŁéÌćæÌěèõ£éŮőæIJ■åŁąåŹĺïijŽ

```
>>> from xmlrpc.client import ServerProxy
>>> s = ServerProxy('http://localhost:15000', allow_none=True)
>>> s.set('foo', 'bar')
>>> s.set('spam', [1, 2, 3])
>>> s.keys()
['spam', 'foo']
>>> s.get('foo')
'bar'
>>> s.get('spam')
[1, 2, 3]
>>> s.delete('spam')
>>> s.exists('spam')
False
>>>
```

èőĺèőž

XML-RPC åŔŕäżěèőľæĹŚäżňå¿ĹåőźæŸŞçŽĎæđĎéĂääÿĂäÿłçőĂå■ŢçŽĎè£IJçĺŃèřČçŤĺæIJ■åŁąãĂĆä¡
éĂŽè£ĞåőČçŽĎæŰźæşŢ register_function() æĺěæşĺåĘŇåĞ¡æŢřïijŇçĎűåŘŐä¡£çŤĺæŰźæşŢ
serve_forever() åŘŕåŁĺåőČãĂĆ åIJĺäÿŁéĺćæĹŚäżňåřĘè£ŹäžŻæ■ěéłď æŤ¿åIJĺäÿĂèţůåĘŹåĹřäÿĂäÿłçśż

```
from xmlrpc.server import SimpleXMLRPCServer
def add(x,y):
    return x+y

serv = SimpleXMLRPCServer(('', 15000))
serv.register_function(add)
serv.serve_forever()
```

XML-RPCæŽť éIJšåĞžæĺěçŽĎåĞ¡æŢřåŔłèČ¡éĂĆçŤĺäžŐéČĺåĹĘæŢřæ■őçśżåđŃïijŇæŕŤåęĆå■Ůçňęäÿš
åŕźäžŐåĚűäżŰçśżåđŃåřśå¿ŮéIJĂèęĄåĄŽäžŻéćĺåď ŰçŽĎåŁ§èŕ¿äžĘãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăæČşéĂŽè£Ğ XML-RPC äijăéĂŠäÿĂäÿłåŕźèśąåőđä¿ŃïijŇåőđéŹĚäÿŁåŔłæIJĽäżŰçŽĎ

```
>>> class Point:
...     def __init__(self, x, y):
...             self.x = x
...             self.y = y
...
>>> p = Point(2, 3)
>>> s.set('foo', p)
>>> s.get('foo')
{'x': 2, 'y': 3}
>>>
```

çśżäijijçŽĎïijŇåŕźäžŐäžŇè£ŻåĹűæŢřæ■őçŽĎåď ĎçŘĘäź§èů§ä¡ăæČşèśąçŽĎäÿ■åď łäÿĂæăůïijŽ

```
>>> s.set('foo', b'Hello World')
>>> s.get('foo')
<xmlrpc.client.Binary object at 0x10131d410>

>>> _.data
b'Hello World'
>>>
```

äÿĂèĹňæĺěèőšïijŇä¡ăäÿ■åžŤèŕěåřĘ XML-RPC æIJ■åŁąäżěåĚňåĚśAPIçŽĎæŰźåijŔæŽť éIJšåĞžæĺěãĂĆ
åŕźäžŐè£Źçğ■æČĚåĘţïijŇéĂŽåÿÿåĹĘåÿČåijŔåžŤçŤĺçĺŃåžŔäijŽæŸŕäÿĂäÿłæŻť åě¡çŽĎéĂĽæŃľãĂĆ

XML-RPCçŽĎäÿĂäÿłçijžçĆźæŸŕåőČçŽĎæĂğèČ¡ãĂĆSimpleXMLRPCServer
çŽĎåőđçŐřæŸŕå■Ţçž£çĺŃçŽĎïijŇ æĽĂäżěåőČäÿ■éĂĆåŘĹäžŐåď ğåđŃçĺŃåžŔïijŇåř¡çőąæĹŚäżňåIJĺ11.2åř
åŔęåď ŰïijŇçŤśäžŐ XML-RPC åřĘæĽĂæIJĽæŢřæ■őéČ¡åžŔåĹŮåŇŰäÿžXMLæăijåijŔïijŇæĽĂäżěåőČäijŽ
ä¡ĘæŸŕåőČäź§æIJĽäijŸçĆźïijŇè£Źçğ■æŰźåijŔçŽĎçijŰçăĄåŔŕäżěèćńçżĺåď ğéČĺåĹĘåĚűäżŰçijŰçĺŃèŕ■èĺĂ
éĂŽè£Ğä¡£çŤĺè£Źçğ■æŰźåijŔïijŇåĚűäżŰèŕ■èĺĂçŽĎåőćæĹůçńŕçĺŃåžŔéČ¡èČ¡èő£éŮőä¡ăçŽĎæIJ■åŁąãĂĆ

èŹ¡çĎűXML-RPCæIJĽå¿Ĺåď ŽçijžçĆźïijŇä¡ĘæŸŕåęĆæđIJä¡ăéIJĂèęĄå£ńéĂ§æđĎåżžäÿĂäÿłçőĂå■Ţè£
æIJĽæŮűåĂŹïijŇçőĂå■ŢçŽĎæŰźæąĹåřśåůšçżŔèűşåď §äžĘãĂĆ

13.7 11.7 åIJÍäÿ■åŘŇçŽĎPythonèğčéĞŁåŹÍäźŃéŮťäžďäžŠ

éŮőécŸ

ä¡ăåIJÍäÿ■åŘŇçŽĎæIJžåŹÍäÿŁéİćè£ŘèąŇçİĂåď'ŽäÿłPythonèğčéĞŁåŹÍåőđä¿ŃïijŇåźűåÿŇæIJŻèČ¡åď'§å

èğčåĘşæŰźæąĹ

éĂŽè£Ğä¡£çŤÍ `multiprocessing.connection` æÍąåİŮåŔŕäžěå¿ĹåőźæŸŞçŽĎåőđçŐřèğčéĞŁåŹÍ
äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎåžŤç■ŤæIJ■åŁąåŹÍä¿Ńå■ŘïijŽ

```
from multiprocessing.connection import Listener
import traceback

def echo_client(conn):
    try:
        while True:
            msg = conn.recv()
            conn.send(msg)
    except EOFError:
        print('Connection closed')

def echo_server(address, authkey):
    serv = Listener(address, authkey=authkey)
    while True:
        try:
            client = serv.accept()

            echo_client(client)
        except Exception:
            traceback.print_exc()

echo_server(('', 25000), authkey=b'peekaboo')
```

çĎűåŘŐåőćæĹůçńŕè£đæŐěæIJ■åŁąåŹÍåźűåŔŠéĂĄæűĹæĄŕçŽĎçőĂå■Ţçď'žä¿ŃïijŽ

```
>>> from multiprocessing.connection import Client
>>> c = Client(('localhost', 25000), authkey=b'peekaboo')
>>> c.send('hello')
>>> c.recv()
'hello'
>>> c.send(42)
>>> c.recv()
42
>>> c.send([1, 2, 3, 4, 5])
>>> c.recv()
[1, 2, 3, 4, 5]
>>>
```

èů§åžŢåśĆsocketäÿ■åŘŇçŽĎæŸŕïijŇæŕŔäÿłæűĹæĄŕäijŽåőŇæŢť'ä£Ìå■ŸïijĹæŕŔäÿĂäÿłéĂŽè£Ğsend()
åŘęåď'ŰïijŇæL'ĂæIJL'åŕźèśąäijŽéĂŽè£ĞpickleåžŘåĹŮåŇŰãĂĆåŻãæ■d'ïijŇäżżä¡ŢåĚijåőźpickleçŽĎåŕźèśą

èőÍèőž

çŽőåL'■æIJL'å¿Ĺåď'ŽçŤÍæÌěåőđçŐřåŘĎçğ■æűĹæĄŕäijàè¿ŞçŽĎåŇĚåŠŇåĞ¡æŢřåžŞïijŇæŕŤåęĆZeroMQ
ä¡ãè£ŸæIJL'åŘęåď'ŰäÿĂçğ■éĂL'æŃľåŕśæŸŕèĞłåůśåIJĺåžŢåśĆsocketå§žçąĂäźŇäÿŁæÌěåőđçŐřäÿĂäÿłæűĹæ
ä¡ĘæŸŕä¡ăæČşèęĄçőĂå■ŢäÿĂçĆźçŽĎæŰźæąĹïijŇéĆčäźĹè£ŹæŮűåĂŹ
multiprocessing.connection åŕśæť¿äÿŁçŤÍåIJžäžĘãĂĆ
äżĚäżĚä¡£çŤÍäÿĂäžŽçőĂå■ŢçŽĎèŕ■åŘěå■şåŔŕåőđçŐřåď'ŽäÿłèğčéĞŁåŹĺäźŇéŮť'çŽĎæűĹæĄŕéĂŽä£ąãĂĆ

åęĆæđIJä¡ăçŽĎèğčéĞŁåŹĺè£ŘèąŇåIJÍåŘŇäÿĂåŘřæIJžåŹĺäÿŁéİćïijŇéĆčäźĹä¡ăåŔŕäžěä¡£çŤÍåŘęåď'ŰçŽ
èęĄæČşä¡£çŤÍUNIXå§§åëŮæŐěå■ŮæÌěåĹŽåžžäÿĂäÿłè£đæŐěïijŇåŔłéIJĂçőĂå■ŢçŽĎårĘåIJřåİĂæŤźåĘŹäÿ

```
s = Listener('/tmp/myconn', authkey=b'peekaboo')
```

èęĄæČşä¡£çŤÍWindowsåŚ¡åŘ■çőąéĄŞæÌěåĹŽåžžè£đæŐěïijŇåŔłéIJĂåČŔäÿŇéİćè£ŹæăůäĄ£çŤÍäÿĂäÿła

```
s = Listener(r'\\.\pipe\myconn', authkey=b'peekaboo')
```

äÿĂäÿłéĂŽçŤÍåĞĘåĹŹæŸŕïijŇä¡ăäÿ■èęĄä¡£çŤÍ multiprocessing
æÌěåőđçŐřäÿĂäÿłåŕźåď'ŰçŽĎåĚňåĚśæIJ■åŁąãĂĆ Client() åŠŇ Listener()
äÿ■çŽĎ authkey åŔĆæŢřçŤÍæÌěèőď'èŕĄåŔŚèţůè£đæŐěçŽĎçżĹçńŕçŤÍæĹůãĂĆ
åęĆæđIJåŕĘéŠěäÿ■åŕźäijŽäžğçŤ§äÿĂäÿłåijĆåÿÿãĂĆæ■d'åď'ŰïijŇèŕěæĺąåIÙæIJĂéĂĆåŘĹçŤÍæÌěåžžçńŇéŢ£
ä¿ŇåęĆïijŇäÿď'äÿłèğčéĞŁåŹĺäźŇéŮť'åŘŕåŁĺåŘŐåŕśåijĂäğŇåžžçńŇè£đæŐěåźűåIJĺåď'ĎçŘĘæ§ŘäÿłéŮőécŸ

åęĆæđIJä¡ăéIJĂèęĄåŕźåžŢåśĆè£đæŐěåĄŽæŽť'åď'ŽçŽĎæŐğåĹűïijŇæŕŤåęĆéIJĂèęĄæŤŕæŇĄèűĚæŮűã
ä¡ăæIJĂåě¡ä¡£çŤÍåŘęåď'ŰçŽĎåžŞæĹŰèĂĚæŸŕåIJĺéńŸåśĆsocketäÿŁæÌěåőđçŐřè£ŹäžŽçL'źæĂğãĂĆ

13.8 11.8 åőđçŐřè£IJçÍŇæŰźæşŢèřČçŤÍ

éŮőécŸ

ä¡ăæČşåIJÍäÿĂäÿłæűĹæĄŕäijàè¿ŞåśĆåęĆ sockets ãĂĄmultiprocessing
connections æĹŰ ZeroMQ çŽĎå§žçąĂäźŇäÿŁåőđçŐřäÿĂäÿłçőĂå■ŢçŽĎè£IJçÍŇè£ĞçÍŇèřČçŤÍïijĹRPC

èğčåĘşæŰźæąĹ

åřĘåĞ¡æŢřèřůæśĆãĂĄåŘĆæŢřåŠŇè£ŤåŻđåĂijä¡£çŤÍpickleçijŰçăĄåŘŐïijŇåIJÍäÿ■åŘŇçŽĎèğčéĞŁåŹ
äÿŇéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎPRCåď'ĎçŘĘåŹÍïijŇåŘŕäžěèćńæŢť'åŘĹåĹřäÿĂäÿłæIJ■åŁąåŹĺäÿ■åŐžïijŽ

```
# rpcserver.py

import pickle
class RPCHandler:
    def __init__(self):
        self._functions = { }
```

```
    def register_function(self, func):
        self._functions[func.__name__] = func

    def handle_connection(self, connection):
        try:
            while True:
                # Receive a message
                func_name, args, kwargs = pickle.loads(connection.
↪recv())
                # Run the RPC and send a response
                try:
                    r = self._functions[func_name](*args,**kwargs)
                    connection.send(pickle.dumps(r))
                except Exception as e:
                    connection.send(pickle.dumps(e))
        except EOFError:
            pass
```

èęĄä¡£çŤĺè£Źäÿłåđ'ĎçŘĘåŹĺïijŇä¡ăéIJĂèęĄåřĘåőČåŁăåĚěåĹřäÿĂäÿłæűĹæĄŕæIJ■åŁąåŹĺäÿ■ãĂĆä¡ăæä¡ĘæŸŕä¡£çŤĺ multiprocessing åžŞæŸŕæIJĂçőĂå■ŢçŽĎãĂĆäÿŃéİćæŸŕäÿĂäÿłR-PCæIJ■åŁąåŹĺä¿Ńå■ŘïijŽ

```
from multiprocessing.connection import Listener
from threading import Thread

def rpc_server(handler, address, authkey):
    sock = Listener(address, authkey=authkey)
    while True:
        client = sock.accept()
        t = Thread(target=handler.handle_connection, args=(client,))
        t.daemon = True
        t.start()

# Some remote functions
def add(x, y):
    return x + y

def sub(x, y):
    return x - y

# Register with a handler
handler = RPCHandler()
handler.register_function(add)
handler.register_function(sub)

# Run the server
rpc_server(handler, ('localhost', 17000), authkey=b'peekaboo')
```

äÿžäžĘäżŐäÿĂäÿłè£IJçĺŃåőćæĹůçńŕèő£éŮőæIJ■åŁąåŹĺïijŇä¡ăéIJĂèęĄåĹŻåżžäÿĂäÿłåŕźåžŤçŽĎçŤĺæİ

```
import pickle

class RPCProxy:
    def __init__(self, connection):
        self._connection = connection
    def __getattr__(self, name):
        def do_rpc(*args, **kwargs):
            self._connection.send(pickle.dumps((name, args,␣
↪kwargs)))
            result = pickle.loads(self._connection.recv())
            if isinstance(result, Exception):
                raise result
            return result
        return do_rpc
```

èęĄä¡£çŤĺè£ŹäÿłäžčçŘĘçśżïijŇä¡äéIJĂèęĄåřĘåĚűåŇĚèčĚåĹřäÿĂäÿłæIJ■åŁąåŹĺçŽĎè£đæŐěäÿŁéİćïijŇ

```
>>> from multiprocessing.connection import Client
>>> c = Client(('localhost', 17000), authkey=b'peekaboo')
>>> proxy = RPCProxy(c)
>>> proxy.add(2, 3)

5
>>> proxy.sub(2, 3)
-1
>>> proxy.sub([1, 2], 4)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "rpcserver.py", line 37, in do_rpc
    raise result
TypeError: unsupported operand type(s) for -: 'list' and 'int'
>>>
```

èęĄæşĺæĎŘçŽĎæŸŕå¿Ĺåd'ŽæűĹæĄŕåśĆïijĹæŕŤåęĆ `multiprocessing` ïijL'åůšçżŔä¡£çŤĺpickleåžŘåĹŮåŇŰäžĘæŢřæ■őãĂĆ åęĆæđIJæŸŕè£Źæăůç ŽĎèŕİïijŇåŕź `pickle.dumps()` åŠŇ `pickle.loads()` çŽĎèřČçŤĺèęĄåŐżæŐL'ãĂĆ

èőĺèőž

`RPCHandler` åŠŇ `RPCProxy` çŽĎå§žæIJňæĂİèůŕæŸŕå¿ĹæŕŤè¿ČçőĂå■ŢçŽĎãĂĆ åęĆæđIJäÿĂäÿłåőćæĹůçńŕæČşèęĄèřČçŤĺäÿĂäÿłè£IJçĺŇåĞ¡æŢřïijŇæŕŤåęĆ `foo(1, 2, z=3)` ,äžčçŘĘçśżåĹŻåžžäÿĂäÿłåŇĚåŘńäžĘåĞ¡æŢřåŘ■åŠŇåŔĆæŢřçŽĎåĚČçżĎ `('foo', (1, 2), {'z': 3})` ãĂĆ è£ŹäÿłåĚČçżĎèćńpickleåžŘåĹŮåŇŰåŘŐéĂŽè£Ğç¡ŚçżIJè£đæŐěåŔŚçŤ§åČ è£ŹäÿĂæ■ěåIJĺ `RPCProxy` çŽĎ `__getattr__()` æŰźæşŢè£ŤåŻđçŽĎ `do_rpc()` éŮ■åŇĚäÿ■åőŇæĹŘãĂĆ æIJ■åŁąåŹĺæŐěæŤűåŘŐéĂŽè£ĞpickleåŘ■åžŘåĹŮåŇŰæűĹæĄŕïijŇæ§ěæL'¿åĞ¡æ æL'ğèąŇçżŞædIJ(æĹŰåijĆåÿÿ)èćńpickleåžŘåĹŮåŇŰåŘŐè£ŤåŻđåŔŚéĂĄçżŹåőćæĹůçńŕãĂĆæĹŚäžňçŽĎåő `multiprocessing` è£ŻèąŇéĂŽä£ąãĂĆ äÿ■è£ĞïijŇè£ŹçğæŰźåijŔåŕŕäżěéĂĆçŤĺäžŐåĚűäžŰäżżä¡ŢæűĹ äžĚäżĚåŔłéIJĂèęĄåřĘè£đæŐěåŕźèśąæ■ćæĹŘåŘĹéĂĆçŽĎZeroMQçŽĎsocketåŕźèśąå■şåŔŕãĂĆ

çŤśäžŐåžŢåśĆéIJÄèęĄä¿İèţŰpickleïijŇéĆčäźĹåőL’åĚĺéŮőécŸåřśéIJÄèęĄèĂČèŹŚäžĘ
ïijĹåŽääÿžäÿĂäÿłèĄłæŸŐçŽĎéżŠåőćåŔŕäžěåĹŽåžžçL’źåőŽçŽĎæűĹæĄŕïijŇèČ¡åd’§èőľäżżæĎŔåĞ¡æŢřéĂŽ
åŽäæ■d’ä¡äæřÿè£IJäÿ■èęĄåĚĄèőÿæİěèĞłäÿ■ä£ąäżżæĹŰæIJłèőd’èŕĄçŽĎåőćæĹůçńŕçŽĎRPCãĂĆçL’źåĹńæ
è£Źçğ■åŔłèČ¡åIJĺåĘĚéČĺèćńä¡£çŤĺïijŇä¡■äžŐéŸšçĄńáćŹåŘŐéİćåźűäÿŤäÿ■èęĄåŕźåd’ŰæŽť éIJšãĂĆ

ä¡IJäÿžpickleçŽĎæŻ£äżčïijŇä¡äáź§èőÿåŔŕäžěèĂČèŹŚä¡£çŤĺJSONãĂĄXMLæĹŰäÿĂäžŻåĚűäžŰçŽĎçi
ä¿ŇåęĆïijŇæIJñæIJžåőđä¿ŇåŔŕäžěå¿Ĺåőźæ ŸŞçŽĎæŤźåĘŹæĹŘJSONçijŰçăĄæŰźæąĹãĂĆè£ŸéIJÄèęĄåřĘ
`pickle.loads()` åŠŇ `pickle.dumps()` æŻ£æ■ćæĹŘ `json.loads()` åŠŇ `json.dumps()` å■şåŔŕïijŽ

```
# jsonrpcserver.py
import json

class RPCHandler:
    def __init__(self):
        self._functions = { }

    def register_function(self, func):
        self._functions[func.__name__] = func

    def handle_connection(self, connection):
        try:
            while True:
                # Receive a message
                func_name, args, kwargs = json.loads(connection.
↪recv())
                # Run the RPC and send a response
                try:
                    r = self._functions[func_name](*args,**kwargs)
                    connection.send(json.dumps(r))
                except Exception as e:
                    connection.send(json.dumps(str(e)))
        except EOFError:
             pass

# jsonrpcclient.py
import json

class RPCProxy:
    def __init__(self, connection):
        self._connection = connection
    def __getattr__(self, name):
        def do_rpc(*args, **kwargs):
            self._connection.send(json.dumps((name, args, kwargs)))
            result = json.loads(self._connection.recv())
            return result
        return do_rpc
```

åőđçŐřRPCçŽĎäÿĂäÿłæŕŤè¿Čåd’■æİĆçŽĎéŮőécŸæŸŕåęĆä¡ŢåŐžåd’ĎçŘĘåijĆåÿÿãĂĆèĞşåřŚïijŇå¡Ş
åŽäæ■d’ïijŇè£ŤåŽđçżŹåőćæĹůçńŕçŽĎåijĆåÿÿæL’ĂäžčèąĺçŽĎåŘńäźL’åŕśèęĄåě¡åě¡èő¿ëőąäžĘãĂĆ
åęĆæđIJä¡ää¡£çŤĺpickleïijŇåijĆåÿÿåŕźèśąåőđä¿ŇåIJĺåőćæĹůçńŕèČ¡ècńåŔ■åžŘåĹŮåŇŰåźűæŁŽåĞžãĂĆåęĆ

äÿ■è£ĜèĞşåřŚïijŇä¡ăåžŤèŕěåIJĺåŞ■åžŤäÿ■è£ŤåŻđåijĆåÿÿå■Ůçňęäÿšã ĂĆæĹŚäžňåIJĺJSONçŽĎä¿Ńå■Řäÿ■å

åŕźäžŐåĚűäżŰçŽĎRPCåőđçŐřä¿Ńå■ŘïijŇæĹŚæŐĺè■Řä¡ăçIJŃçIJŃåIJĺXML-RPCäÿ■ä¡£çŤĺçŽĎ `SimpleXMLRPCServer` åŠŇ `ServerProxy` çŽĎåőđçŐřïijŇ äźş§åřśæŸŕ11.6åřŔèŁĆäÿ■çŽĎåĘĚåőźãĂĆ

13.9 11.9 çőĂå■ŢçŽĎåőćæĹůçńŕèŏđ'èŕĄ

éŮőécŸ

ä¡ăæČşåIJĺåĹĘåÿČåijŔçşżçż§äÿ■åőđçŐřäÿĂäÿłçŏĂå■ŢçŽĎåőćæĹůçńŕè£đæŐěèŏđ'èŕĄåŁ§èČ¡ïijŇåŔĹä

èğčåĘşæŰźæąĹ

åŔŕäžěåĹl'çŤĺ `hmac` æĺąåİŮåőđçŐřäÿĂäÿłè£đæŐěæŔąæL'ŃïijŇäžŐèĂŇåőđçŐřäÿĂäÿłçŏĂå■ŢèĂŇénŸ

```
import hmac
import os

def client_authenticate(connection, secret_key):
    '''
    Authenticate client to a remote service.
    connection represents a network connection.
    secret_key is a key known only to both client/server.
    '''
    message = connection.recv(32)
    hash = hmac.new(secret_key, message)
    digest = hash.digest()
    connection.send(digest)

def server_authenticate(connection, secret_key):
    '''
    Request client authentication.
    '''
    message = os.urandom(32)
    connection.send(message)
    hash = hmac.new(secret_key, message)
    digest = hash.digest()
    response = connection.recv(len(digest))
    return hmac.compare_digest(digest,response)
```

å§žæIJňåŐ§çŘĘæŸŕå¡Şè£đæŐěåžžçńŇåŘŐïijŇæIJ■åŁąåŹĺçżŹåőćæĹůçńŕåŔŚéĂĄäÿĂäÿłéŽŔæIJžçŽĎ `os.urandom()` è£ŤåŻđåĂijïijL'ãĂĆ åőćæĹůçńŕåŠŇæIJ■åŁąåŹĺåŘŇæŮűåĹl'çŤĺh-macåŠŇäÿĂäÿłåŔŁæIJL'åŔŇæŰźç§ěéĄŞçŽĎåŕĘéŠěæİèèőąçőŮåĞžäÿĂäÿłåŁăåŕĘåŞĹåÿŇåĂijãĂĆçĎűåŘŐåæIJ■åŁąåŹĺéĂŽè£ĞæŕŤè¿Čè£ŹäÿłåĂijåŠŇèĞłåůśèőąçőŮçŽĎæŸŕåŘęäÿĂèĞt'æİěåĘşåőŽæŐěåŔŮæĹŰæŃŠ `hmac.compare_digest()` åĞ¡æŢřãĂĆ ä¡£çŤĺè£ŹäÿłåĞ¡æŢřåŔŕäžěéĄ£åĚ■éĄ■åĹřæŮűéŮt'åĹĘæđŘæŤäÿžäžĘä¡£çŤĺè£ŹäžŽåĞ¡æŢřïijŇä¡ăéIJĂèęĄåřĘåőČéŽĘæĹŘåĹřåůšæIJL'çŽĎç¡ŚçżIJæĹŰæűĹæĄŕäžčçăĄäÿ■

```
from socket import socket, AF_INET, SOCK_STREAM

secret_key = b'peekaboo'
def echo_handler(client_sock):
    if not server_authenticate(client_sock, secret_key):
        client_sock.close()
        return
    while True:

        msg = client_sock.recv(8192)
        if not msg:
            break
        client_sock.sendall(msg)

def echo_server(address):
    s = socket(AF_INET, SOCK_STREAM)
    s.bind(address)
    s.listen(5)
    while True:
        c,a = s.accept()
        echo_handler(c)

echo_server(('', 18000))
```

Within a client, you would do this:

```
from socket import socket, AF_INET, SOCK_STREAM

secret_key = b'peekaboo'

s = socket(AF_INET, SOCK_STREAM)
s.connect(('localhost', 18000))
client_authenticate(s, secret_key)
s.send(b'Hello World')
resp = s.recv(1024)
```

èőÍèőž

hmac èőd'èŕĄçŽĎäÿĂäÿłåÿÿèğĄä¡£çŤĺåIJžæŹŕæŸŕåĘĚéČĺæűĹæĄŕéĂŽä£ąçşżçż§åŠŇè£ŻçÍŃéŮť'éĂŽ
ä¿ŃåęĆĭijŇåęĆæđIJä¡ăçijŰåĘŹçŽĎçşżçż§æűĽ'åŔŁåĹŕäÿĂäÿłéŻĘç¿d'äÿ■åd'Žäÿłåd'ĎçŘĘåŹĺäźŃéŮť'çŽĎéĂ
ä¡ăåŔŕäžěä¡£çŤĺæIJňèŁĆæŰźæąĹæİěçąőä£İåŔłæIJL'èćńåĚĄèőÿçŽĎè£ŻçÍŃäźŃéŮť'æL'■èČ¡å¡ijæ■d'éĂŽä£ą
äžŃåőđäÿŁïijŇå§žäžŐ hmac çŽĎèőd'èŕĄèćń multiprocessing
æĺąåİŮä¡£çŤĺæİěåőđçŐřå■Řè£ŻçÍŃçŻť'æŐěçŽĎéĂŽä£ąãĂĆ

è£ŸæIJL'äÿĂçĆźéIJĂèęĄåijžèřČçŽĎæŸŕè£đæŐěèőd'èŕĄåŠŇåŁăárĘæŸŕäÿd'çăĄäžŃãĂĆ
èőd'èŕĄæĹŘåŁ§äźŃåŘŐçŽĎéĂŽä£ąæűĹæĄŕæŸŕäžěæŸŐæŰĞå¡ćåijŔåŘŚéĂĄçŽĎïijŇäżżä¡Ţäžžåŕłèę Ą æČ

hmacèőd'èŕĄçőŮæşŢå§žäžŐåŞĹåÿŇåĞ¡æŢřåęĆMD5åŠŇSHA-
1ïijŇåĚşäžŐè£ŹäÿłåIJĺIETF RFC 2104äÿ■æIJL'èŕęçżĘäżŃçż■ãĂĆ

13.10 11.10 åIJÍç¡ŚçżIJæIJ■åŁąäÿ■åŁååĚěSSL

éŮőécŸ

ä¡äæČşåőđçŐřäÿĂäÿłå§žäžŐsocketsçŽĎç¡ŚçżIJæIJ■åŁąïijŇåőćæĹůçńřåŠŇæIJ■åŁąåŹĺéĂŽè£ĞSSLå■Ř

èğčåĘşæŰźæąĹ

ssl æĺąåİŮèČ¡äÿžåžŢåśĆsocketè£đæŐěæůżåŁăSSLçŽĎæŤŕæŇĄãĂĆ ssl.wrap_socket() åĞ¡æŢřæŐěåŘŮäÿĂäÿłåůšå■ŸåIJÍçŽĎsocketä¡IJäÿžåŘĆæŢřáźűä¡£çŤÍSSLåśĆæİěåŇĚèč ä¿ŇåęĆïijŇäÿŇéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎåžŤç■ŤæIJ■åŁąåŹĺïijŇèČ¡åIJÍæIJ■åŁąåŹÍçńřäÿžæĽĂæIJĽåőćæĹ

```
from socket import socket, AF_INET, SOCK_STREAM
import ssl

KEYFILE = 'server_key.pem'   # Private key of the server
CERTFILE = 'server_cert.pem' # Server certificate (given to client)

def echo_client(s):
    while True:
        data = s.recv(8192)
        if data == b'':
            break
        s.send(data)
    s.close()
    print('Connection closed')

def echo_server(address):
    s = socket(AF_INET, SOCK_STREAM)
    s.bind(address)
    s.listen(1)

    # Wrap with an SSL layer requiring client certs
    s_ssl = ssl.wrap_socket(s,
                            keyfile=KEYFILE,
                            certfile=CERTFILE,
                            server_side=True
                            )
    # Wait for connections
    while True:
        try:
            c,a = s_ssl.accept()
            print('Got connection', c, a)
            echo_client(c)
        except Exception as e:
            print('{}: {}'.format(e.__class__.__name__, e))

echo_server(('', 20000))
```

äÿŃéÌćæĹŚäžňæijŤçd’žäÿĂäÿłåőćæĹůçńŕè£đæŐěæIJ■åŁąåŹĺçŽĎäžd’äžŠä¿Ńå■ŘãĂĆåőćæĹůçńŕäijŽèŕ

```
>>> from socket import socket, AF_INET, SOCK_STREAM
>>> import ssl
>>> s = socket(AF_INET, SOCK_STREAM)
>>> s_ssl = ssl.wrap_socket(s,
                cert_reqs=ssl.CERT_REQUIRED,
                ca_certs = 'server_cert.pem')
>>> s_ssl.connect(('localhost', 20000))
>>> s_ssl.send(b'Hello World?')
12
>>> s_ssl.recv(8192)
b'Hello World?'
>>>
```

è£Źçğ■çŻt’æŐěåd’ĎçŘĘåžŢåśĆsocketæŰźåijŔæIJL’äÿłéŮőécŸåřśæŸřåőČäÿ■èČ¡å¿Ĺåě¡çŽĎèů§æăĞåČ
ä¿ŃåęĆïijÑçżÌåd’ğéČĺåĹĘæIJ■åŁąåŹĺäžčçăĄïijĹHTTPãĂĄXML-
RPCç■L’ïijL’åőđéŹĚäÿŁæŸřå§žäžŐ socketserver åžŞçŽĎãĂĆ
åőćæĹůçńŕäžčçăĄåIJĺäÿĂäÿłè¿ČéńŸåśĆäÿŁåőđçŐřãĂĆæĹŚäžňéIJĂèęĄåŔęåd’ŰäÿĂçğ■çĺ■å¿őäÿ■åŘŇçŽĎ

éęŰåĚĹïijÑåŕźäžŐæIJ■åŁąåŹĺèĂŇèĺĂïijÑåŔŕäžěéĂŽè£ĞåČŔäÿŃéÌćè£Źæăůä¡£çŤĺäÿĂäÿłmixinçśžæÌě

```
import ssl

class SSLMixin:
'''
Mixin class that adds support for SSL to existing servers based
on the socketserver module.
'''
def __init__(self, *args,
             keyfile=None, certfile=None, ca_certs=None,
             cert_reqs=ssl.CERT_NONE,
             **kwargs):
    self._keyfile = keyfile
    self._certfile = certfile
    self._ca_certs = ca_certs
    self._cert_reqs = cert_reqs
    super().__init__(*args, **kwargs)

def get_request(self):
    client, addr = super().get_request()
    client_ssl = ssl.wrap_socket(client,
                                 keyfile = self._keyfile,
                                 certfile = self._certfile,
                                 ca_certs = self._ca_certs,
                                 cert_reqs = self._cert_reqs,
                                 server_side = True)
    return client_ssl, addr
```

äÿžäžĘä¡£çŤĺè£ŹäÿłmixinçśžïijÑä¡ăåŔŕäžěårĘåőČèů§åĚűäžŰæIJ■åŁąåŹĺçśžæůůåŘĹãĂĆä¿ŃåęĆïijÑäÿ
RPCæIJ■åŁąåŹĺä¿Ńå■ŘïijŽ

```
# XML-RPC server with SSL

from xmlrpc.server import SimpleXMLRPCServer

class SSLSimpleXMLRPCServer(SSLMixin, SimpleXMLRPCServer):
    pass

Here's the XML-RPC server from Recipe 11.6 modified only slightly␣
↪to use SSL:

import ssl
from xmlrpc.server import SimpleXMLRPCServer
from sslmixin import SSLMixin

class SSLSimpleXMLRPCServer(SSLMixin, SimpleXMLRPCServer):
    pass

class KeyValueServer:
    _rpc_methods_ = ['get', 'set', 'delete', 'exists', 'keys']
    def __init__(self, *args, **kwargs):
        self._data = {}
        self._serv = SSLSimpleXMLRPCServer(*args, allow_none=True,␣
↪**kwargs)
        for name in self._rpc_methods_:
            self._serv.register_function(getattr(self, name))

    def get(self, name):
        return self._data[name]

    def set(self, name, value):
        self._data[name] = value

    def delete(self, name):
        del self._data[name]

    def exists(self, name):
        return name in self._data

    def keys(self):
        return list(self._data)

    def serve_forever(self):
        self._serv.serve_forever()

if __name__ == '__main__':
    KEYFILE='server_key.pem'     # Private key of the server
    CERTFILE='server_cert.pem'   # Server certificate
    kvserv = KeyValueServer(('', 15000),
                            keyfile=KEYFILE,
                            certfile=CERTFILE)
```

```
    kvserv.serve_forever()
```

ä¡£çŤÍè£ŹäÿłæIJ■åŁąåŹÍæŮűïijŇä¡ăåŔŕäžěä¡£çŤÍæŹőéĂŽçŽĎ `xmlrpc.client` æÍąåİŮæİěè£đæŐěåőČãĂĆ åŔŀéIJĂèęĄåIJÍURLäÿ■æŇĞåőŽ `https:` å■şåŔŕïijŇä¿ŃåęĆïijŽ

```
>>> from xmlrpc.client import ServerProxy
>>> s = ServerProxy('https://localhost:15000', allow_none=True)
>>> s.set('foo','bar')
>>> s.set('spam', [1, 2, 3])
>>> s.keys()
['spam', 'foo']
>>> s.get('foo')
'bar'
>>> s.get('spam')
[1, 2, 3]
>>> s.delete('spam')
>>> s.exists('spam')
False
>>>
```

åŕźäžŐSSLåőćæĹůçńŕæİěèőšäÿĂäÿłæŕŤè¿Čåd'■æİĆçŽĎéŮőéćŸæŸŕåęĆä¡Ţçąőèőd'æIJ■åŁąåŹÍèŕĄäźęäÿ■åźÿçŽĎæŸŕïijŇæŽĆæŮűë£ŸæšąæIJL'äÿĂäÿłæăĞåĞĘæŰźæşŢæİěèğčåĘşè£ŹäÿłéŮőéćŸïijŇéIJĂèęĄèĞłäÿ■è£ĞïijŇäÿŃéİćçżŹåĞžäÿĂäÿłä¿Ńå■ŘïijŇçŤÍæİěåžžçńŃäÿĂäÿłåőL'åĚÍçŽĎXML-RPCè£đæŐěæİěçąőèőd'æIJ■åŁąåŹÍèŕĄäźęïijŽ

```
from xmlrpc.client import SafeTransport, ServerProxy
import ssl

class VerifyCertSafeTransport(SafeTransport):
    def __init__(self, cafile, certfile=None, keyfile=None):
        SafeTransport.__init__(self)
        self._ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLSv1)
        self._ssl_context.load_verify_locations(cafile)
        if certfile:
            self._ssl_context.load_cert_chain(certfile, keyfile)
        self._ssl_context.verify_mode = ssl.CERT_REQUIRED

    def make_connection(self, host):
        # Items in the passed dictionary are passed as keyword
        # arguments to the http.client.HTTPSConnection()␣
↪constructor.
        # The context argument allows an ssl.SSLContext instance to
        # be passed with information about the SSL configuration
        s = super().make_connection((host, {'context': self._ssl_
↪context}))

        return s

# Create the client proxy
s = ServerProxy('https://localhost:15000',
```

```
                transport=VerifyCertSafeTransport('server_cert.pem
↪'),
                allow_none=True)
```

æIJ■åŁąåŹÍåřĘèŕĄäźęåŔŠéĂĄçżŹåőćæĹůçńŕïijŇåőćæĹůçńŕæİěçąőèőd'åőČçŽĎåŘĹæşŢæĂğãĂĆè£ŹçğåęĆæđIJæIJ■åŁąåŹÍæČşèęĄçąőèőd'åőćæĹůçńŕïijŇåŔŕäžěåřĘæIJ■åŁąåŹÍåŘŕåŁÍäžčçăĄä£őæŤźåęĆäÿŇïijŽ

```
if __name__ == '__main__':
    KEYFILE='server_key.pem'    # Private key of the server
    CERTFILE='server_cert.pem' # Server certificate
    CA_CERTS='client_cert.pem' # Certificates of accepted clients

    kvserv = KeyValueServer(('', 15000),
                            keyfile=KEYFILE,
                            certfile=CERTFILE,
                            ca_certs=CA_CERTS,
                            cert_reqs=ssl.CERT_REQUIRED,
                            )
    kvserv.serve_forever()
```

äÿžäžĘèől'XML-RPCåőćæĹůçńŕåŔŠéĂĄèŕĄäźęïijŇä£őæŤź ServerProxy çŽĎåĹİåğŃåŇŰäžčçăĄåęĆäÿŇïijŽ

```
# Create the client proxy
s = ServerProxy('https://localhost:15000',
                transport=VerifyCertSafeTransport('server_cert.pem',
                                                  'client_cert.pem',
                                                  'client_key.pem'),
                allow_none=True)
```

èőÍèőž

èŕŢçİĂåŐżè£ŘèąŇæIJňèŁĆçŽĎäžčçăĄèČ¡æţŃèŕŢä¡ăçŽĎçşżçż§éĚ■ç¡őèČ¡åŁŻåŠŇçŘĘèğčSSLãĂĆ åŔŕèČ¡æIJĂåd'ğçŽĎæŇŠæĹŸæŸŕåęĆä¡ŢäÿĂæ■ěæ■ěçŽĎèŐůåŘŰåĹİåğŃéĚ■ç¡őkeyãĂĄèŕĄäźęåŠŇåĚűäžŰ

æĹŚèğčéĞŁäÿŇåĹřåžŢéIJĂèęĄåŢëïijŇærŔäÿĂäÿłSSLè£đæŐěçŽĹçńŕäÿĂèĹňéČ¡äijŽæIJL'äÿĂäÿłçğĄé è£ŹäÿłèŕĄäźęåŇĚåŘńäžĘåĚňéŠěåźűåIJÍæŕŔäÿĂæňąè£đæŐěçŽĎæŮűåĂŹéČ¡äijŽåŘŚéĂĄçżŹåŕźæŰźãĂĆ åŕźäžŐåĚňåĚśæIJ■åŁąåŹÍïijŇåőČäžňçŽĎèŕĄäźęéĂŽåÿÿæŸŕèćńæİČåÍĄèŕĄäźęæIJžædĎæŕŤåęĆVerisignãĂA äÿžäžĘçąőèőd'æIJ■åŁąåŹÍç■¿åŘ■ïijŇåőćæĹůçńŕåŻđä£Ìå■ŸäÿĂäż¡åŇĚåŘńäžĘä£ąäżżæŐĹæİČæIJžædĎçŽĎ ä¿ŃåęĆïijŇwebæţŔèğĹåŹÍä£Ìå■ŸäžĘäÿżèęĄçŽĎèőd'èŕĄæIJžædĎçŽĎèŕĄäźęïijŇåźűä¡£çŤÍåőČæİěäÿžæŕŔä åŕźæIJňåŕŔèŁĆçd'žä¿ŃèĂŇèÍĂïijŇåŘłæŸŕäÿžäžĘæţŃèŕŢïijŇæĹŠäžňåŘŕäžěåĹŻåżżèĞłç■¿åŘ■çŽĎèŕĄäźęïij

bash % openssl req -new -x509 -days 365 -nodes -out server_cert.pem
-keyout server_key.pem

Generating a 1024 bit RSA private key âĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂęâĂę++++++
âĂę++++++

writing new private key to âĂŸserver_key.pemâĂŹ

You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank For some fields there will be a default value, If you enter âĂŸ.âĂŹ, the field will be left blank.

Country Name (2 letter code) [AU]:US State or Province Name (full name) [Some-State]:Illinois Locality Name (eg, city) []:Chicago Organization Name (eg, company) [Internet Widgits Pty Ltd]:Dabeaz, LLC Organizational Unit Name (eg, section) []: Common Name (eg, YOUR name) []:localhost Email Address []: bash %

åIJĺåĹŻåżžèŕĄäźęçŽĎæŮűåĂŹïijŇåŘĎäÿłåĂijçŽĎèő¿åőŽåŔŕäżěæŸŕäżżæĎŔçŽĎïijŇä¡ĘæŸŕâĂİComm
NameâĂIJçŽĎåĂijéĂŽåÿÿèęĄåŇĚåŘńæIJ■åŁąåŹĺçŽĎDNSäÿżæIJżåŘ■ãĂĆ
åęĆæđIJä¡ääŔłæŸŕåIJĺæIJñæIJżæţŃèŕŢïijŇéĆčäźĹåřśä¡£çŤĺâĂİlocalhostâĂIJïijŇåŘęåĹŹä¡£çŤĺæIJ■åŁąåŹĺç

âĂŤ–BEGIN RSA PRIVATE KEYâĂŤ– MIICXQIBAAK-
BgQCZrCNLoEyAKF+f9UNcFaz5Osa6jf7qkbUl8si5xQrY3ZYC7juu
nL1dZLn/VbEFIITaUOgvBtPv1qUWTJGwga62VSG1oFE0ODIx3g2Nh4sRf+rySsx2
L4442nx0z4O5vJQ7k6eRNHAZUUnCL50+YvjyLyt7ryLSjSuKhCcJsbZgPwIDAQAB
AoGAB5evrr7eyL4160tM5rHTeATlaLY3UBOe5Z8XN8Z6gLiB/ucSX9AysviVD/6F
3oD6z2aL8jbeJc1vHqjt0dC2dwwm32vVl8mRdyoAsQpWmiqXrkvP4Bsl04VpBeHw
Qt8xNSW9SFhceL3LEvw9M8i9MV39viih1ILyH8OuHdvJyFECQQDLEjl2d2ppxND9
PoLqVFAirDfX2JnLTdWbc+M11a9Jdn3hKF8TcxfEnFVs5Gav1MusicY5KB0ylYPb
YbTvqKc7AkEAwbnRBO2VYEZsJZp2X0IZqP9ovWokkpYx+PE4+c6MySDgaMcigL7v
WDIHJG1CHudD09GbqENasDzyb2HAIW4CzQJBAKDdkv+xoW6gJx42Auc2WzTcUHCA
eXR/+BLpPrhKykzbvOQ8YvS5W764SUO1u1LWs3G+wnRMvrRvlMCZKgggBjkCQQCG
Jewto2+a+WkOKQXrNNScCDE5aPTmZQc5waCYq4UmCZQcOjkUOiN3ST1U5iuxRqfb
V/yX6fw0qh+fLWtkOs/JAkA+okMSxZwqRtfgOFGBfwQ8/iKrnizeanTQ3L6scFXI
CHZXdJ3XQ6qUmNxNn7iJ7S/LDawo1QfWkCfD9FYoxBlg âĂŤ–END RSA
PRIVATE KEYâĂŤ–

æIJ■åŁąåŹĺèŕĄäźęæŰĞäżűserver_cert.pemåĘĚåőźçśżäijijäÿŃéİćè£Źæăůïijž

âĂŤ–BEGIN CERTIFICATEâĂŤ– MIIC+DCCAmGgAwIBAgIJAPMd+vi45js3MA0GCSqGSIb3DQEI
BAYTAlVTMREwDwYDVQQIEwhJbGxpbm9pczEQMA4GA1UEBxMHQ2hpY2FnbzEUMBIG
A1UEChMLRGFiZWF6LCBMTEMxEjAQBgNVBAMTCWxvY2FsaG9zdDAeFw0xMzAxMTEx
ODQyMjdaFw0xNDAxMTExODQyMjdaMFwxCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhJ
bGxpbm9pczEQMA4GA1UEBxMHQ2hpY2FnbzEUMBIGA1UEChMLRGFiZWF6LCBMTEMx
EjAQBgNVBAMTCWxvY2FsaG9zdDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA
mawjS6BMgChfn/VDXBWs+TrGuo3+6pG1JfLIucUK2N2WAu47rpy9XWS5/1WxBSCE
2lDoLwbT79alFkyRsIGutlUhtaBRNDgyMd4NjYeLEX/q8krMdi+OONp8dM+DubyU

O5OnkTRwGVFJwi+dPmL48i8re68i0o0rioQnCbG2YD8CAwEAAaOBwTCBvjAdBgNV
HQ4EFgQUrtoLHHgXiDZTr26NMmgKJLJLFtIwgY4GA1UdIwSBhjCBg4AUrtoLHHgX
iDZTr26NMmgKJLJLFtKhYKReMFwxCzAJBgNVBAYTAlVTMREwDwYDVQQIEwhJbGxp
bm9pczEQMA4GA1UEBxMHQ2hpY2FnbzEUMBIGA1UEChMLRGFiZWF6LCBMTEMxEjAQ
BgNVBAMTCWxvY2FsaG9zdIIJAPMd+vi45js3MAwGA1UdEwQFMAMBAf8wDQYJKoZI
hvcNAQEFBQADgYEAFci+dqvMG4xF8UTnbGVvZJPIzJDRee6Nbt6AHQo9pOdAIMAu

WsGCplSOaDNdKKzl+b2UT2Zp3AIW4Qd51bouSNnR4M/gnr9ZD1ZctFd3jS+C5XRp
D3vvcW5lAnCCC80P6rXy7d7hTeFu5EYKtRGXNvVNd/06NALGDflrrOwxF3Y=
âĂŤ–END CERTIFICATEâĂŤ–

åIJÍæIJ■åŁąåŹÍçńŕäžčçăĄäÿ■ïijŇçğĄéŠěåŠŇèŕĄäźęæŰĞäžűäijŽèćńäijäçżŹSSLçŻÿåĚşçŽĎåŇĚèčĚåĞ¡
çğĄéŠěåžŤèŕěåIJÍä‡Ìå■ŸåIJÍæIJ■åŁąåŹÍäÿ■ïijŇåźűåŁääžěåőL’åĚÍä‡ÌæŁď ãĂĆ

åIJÍåőćæĹůçńŕäžčçăĄäÿ■ïijŇéIJĂèęĄä‡Ìå■ŸäÿĂäÿłåŘĹæşŢèŕĄäźęæŐĹæİČæŰĞäžűæİěçąőèőď æIJ■åŁ
åęĆæđIJä¡ăæšąæIJL’è£ŹäÿłæŰĞäžűïijŇä¡ăåŔŕäżěåIJÍåőćæĹůçńŕåď■åĹűäÿĂäż¡æIJ■åŁąåŹÍçŽĎèŕĄäźęåźűä¡
è£đæŐěåžžçńŇåŘŐïijŇæIJ■åŁąåŹÍäijŽæŔŘä¿ŻåőČçŽĎèŕĄäźęïijŇçĎűåŘŐä¡ăåřśèČ¡ä¡£çŤÍåůšçżŔä£Ìå■ŸçŽ

æIJ■åŁąåŹÍäź§èČ¡éĂL’æŃl’æŸŕåŘęèęĄçąőèőď åőćæĹůçńŕçŽĎèžńäż¡ãĂĆåęĆæđIJèęĄè£ŹæăůåĄŽçŽĎ
æIJ■åŁąåŹÍäź§éIJĂèęĄä‡Ìå■ŸäÿĂäÿłèćńä£ąäżżèŕĄäźęæŐĹæİČæŰĞäžűæİěçąőèőď åőćæĹůçńŕèŕĄäźęãĂĆ

åęĆæđIJä¡ăèęĄåIJÍçIJ§åőđçŐŕåćČäÿ■äÿžä¡ăçŽĎç¡ŚçżIJæIJ■åŁąåŁääÿŁSSLçŽĎæŤŕæŇĄïijŇè£ŹåŕŔèŁ
ä¡ăè£ŸåžŤèŕěåŔĆèĂČåĚűäżŰçŽĎæŰĞæąčïijŇåĄŽåě¡èŁśèť źäÿ■åřŚæŮűéŮť æİěæţŃèŕŢåőČæ■čåÿÿåůěä¡IJ
^_^

13.11 11.11 è£ŽçÍŇéŮť äijăéĂŠSocketæŰĞäžűæŔŘè£řçňę

éŮőécŸ

ä¡ăæIJL’åď ŽäÿłpythonèğčéĞŁåŹÍè£ŽçÍŇåIJÍåŘŇæŮűè£ŘèąŇïijŇä¡ăæČşåřĘæ§Řäÿłæ L’ŞåijĂçŽĎæŰĞä
æŕŤåęĆïijŇåĄĞèő¿æIJL’äÿłæIJ■åŁąåŹÍè£ŽçÍŃçŻÿåžŤè£đæŐěèŕůæśĆïijŇä¡ĘæŸŕåőđéŹĚçŽĎçŻÿåžŤéĂżè¿Ş

èğčåĘşæŰźæąĹ

äÿžäžĘåIJÍåď Žäÿłè£ŽçÍŃäÿ■äijăéĂŠæŰĞäžűæŔŘè£řçňęïijŇä¡ăéęŰåĚĹéIJĂèęĄåřĘåőČäžňè£đæŐěåĹřă
èĂŇåIJÍwindowsäÿŁéİćä¡ăéIJĂèęĄä¡£çŤÍåŚ¡åŘ■çőąéĄŞãĂĆäÿ■è£Ğä¡ăæŮăéIJĂçIJ§çŽĎéIJĂèęĄåŐżæŞ■ä¡
éĂŽåÿÿä¡£çŤÍ `multiprocessing` æĺąåİŮæİěåĹŻåżžè£ŹæăůçŽĎè£đæŐěäijŽæŻť’åőźæŸŞäÿĂäžŽãĂĆ

äÿĂæŮęäÿĂäÿłè£đæŐěèćńåĹŻåżžïijŇä¡ăåŔŕäžěä¡£çŤÍ `multiprocessing.reduction` äÿ■çŽĎ `send_handle()` åŠŇ `recv_handle()` åĞ¡æŢřåIJÍäÿ■åŘŇçŽĎåď ĎçŘĘåŹÍçŻť’æŐěäijăéĂŠæŰĞäžűæŔŘè£řçňęãĂĆ
äÿŃéIćçŽĎä¿Ńå■ŘæijŤçď žäžĘæIJĂå§žæIJňçŽĎçŤÍæşŢïijŽ

```
import multiprocessing
from multiprocessing.reduction import recv_handle, send_handle
import socket

def worker(in_p, out_p):
    out_p.close()
    while True:
        fd = recv_handle(in_p)
        print('CHILD: GOT FD', fd)
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM,␣
↪fileno=fd) as s:
            while True:
                msg = s.recv(1024)
```

```
                if not msg:
                    break
                print('CHILD: RECV {!r}'.format(msg))
                s.send(msg)

def server(address, in_p, out_p, worker_pid):
    in_p.close()
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, True)
    s.bind(address)
    s.listen(1)
    while True:
        client, addr = s.accept()
        print('SERVER: Got connection from', addr)
        send_handle(out_p, client.fileno(), worker_pid)
        client.close()

if __name__ == '__main__':
    c1, c2 = multiprocessing.Pipe()
    worker_p = multiprocessing.Process(target=worker, args=(c1,c2))
    worker_p.start()

    server_p = multiprocessing.Process(target=server,
                   args=(('', 15000), c1, c2, worker_p.pid))
    server_p.start()

    c1.close()
    c2.close()
```

åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇäÿd'äÿłè£ŻçÍŃèćńåĹŻåžžåźűéĂŽè£ĞäÿĂäÿł multiprocessing çőąéĄŞè£đæŐěèţůæİěãĂĆ æIJ■åŁąåŹÍè£ŻçÍŃæL'ŞåijĂäÿĂäÿłsocketåźűç■L'å¿ĚåőćæĿ åůěä¡IJè£ŻçÍŃäžĚäžĚä¡£çŤÍ recv_handle() åIJÍçőąéĄŞäÿŁéİćç■L'å¿ĚæŐěæŤűäÿĂäÿłæŰĞäžűæŔŔè£řçňęéĂŽè£Ğ å¡ŞæIJ■åŁąåŹÍæŐěæŤűåĹřäÿĂäÿłè£đæŐëïijŇåőČåřĘäžğçŤ§çŽĎsocketæŰĞäžűæŔŔè£řçňęéĂŽè£Ğ send_handle() äijäéĂŠçżŹåůěä¡IJè£ŻçÍŃãĂĆ åůěä¡IJè£ŻçÍŃæŐěæŤűåĹřsocketåŘŐåŘŠåőćæĹůçńřåŻđå

åęĆæđIJä¡ăä¡£çŤÍTelnetæĹŰçśžäijijåůěåĚůè£đæŐěåĹřæIJ■åŁąåŹÍïijŇäÿŃéİćæŸřäÿĂäÿłæijŤçd'žä¿Ńå■

```
bash % python3 passfd.py SERVER: Got connection from (âĂŸ127.0.0.1âĂŹ,
55543) CHILD: GOT FD 7 CHILD: RECV bâĂŹHellornâĂŹ CHILD: RECV
bâĂŹWorldrnâĂŹ
```

æ■d'ä¿ŃæIJĂéĞ■èęĄçŽĎéČÍåĹĘæŸřæIJ■åŁąåŹÍæŐěæŤűåĹřçŽĎåőćæĹůçńřsocketåőđéŹĚäÿŁèćńåŘęå æIJ■åŁąåŹÍäžĚäžĚåŘłæŸřářĘåĚűè¡ňæL'ŃåźűåĕşéŮ■æ■d'è£đæŐěïijŇçĎűåŘŐç■L'å¿ĚäÿŃäÿĂäÿłè£đæŐěãĂ

èőÍèőž

åŕźäžŐåd'ğéČÍåĹĘçÍŃåžŘåŚŸæİèèőšåIJÍäÿ■åŘŇè£ŻçÍŃäźŃéŮt'äijäéĂŠæŰĞäžűæŔŔè£řçňęåě¡åČŔæša ä¡ĘæŸřïijŇæIJL'æŮűåĂŹåőČæŸřædĎåžžäÿĂäÿłåŘŕæL'l'åśŢçşżçž§çŽĎå¿ĹæIJL'çŤÍçŽĎåůěåĚůãĂĆä¿ŃåęĆ ä¡ååŔŕäžěæIJL'åd'ŽäÿłpythonèğčéĞŁåŹÍåőđä¿ŃïijŇářĘæŰĞäžűæŔŔè£řçňęäijäéĂŠçżŹåĚűåőČèğčéĞŁåŹÍæ

send_handle() åŠŇ recv_handle() åĞ¡æŢřåŔłèČ¡åd'§çŤĺäžŐ multiprocessing è£đæŐěãĂĆ ä¡£çŤĺåőČäžňæİěäžčæŻ£çőąéĄŞçŽĎä¡£çŤĺïijĹåŔĆèĂČ11.7èŁĆïijL'ïijŇä¿ŇåęĆïijŇä¡ăåŔŕäżěèől'æIJ■åŁąåŹĺåŠŇåůěä¡IJèĂĚåŘĎèĞłäžěå■ŢçŃňçŽĎçĺŃåžŔæİěåŘŕåŁĺãĂĆäÿŃéĺćæŸ

```
# servermp.py
from multiprocessing.connection import Listener
from multiprocessing.reduction import send_handle
import socket

def server(work_address, port):
    # Wait for the worker to connect
    work_serv = Listener(work_address, authkey=b'peekaboo')
    worker = work_serv.accept()
    worker_pid = worker.recv()

    # Now run a TCP/IP server and send clients to worker
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, True)
    s.bind(('', port))
    s.listen(1)
    while True:
        client, addr = s.accept()
        print('SERVER: Got connection from', addr)

        send_handle(worker, client.fileno(), worker_pid)
        client.close()

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 3:
        print('Usage: server.py server_address port', file=sys.
↪stderr)
        raise SystemExit(1)

    server(sys.argv[1], int(sys.argv[2]))
```

è£ŘèąŇè£ŹäÿłæIJ■åŁąåŹĺïijŇåŔłéIJĂèęĄæĽğèąŇ *python3 servermp.py /tmp/servconn 15000* ïijŇäÿŃéĺćæŸŕçŻÿåžŤçŽĎåůěä¡IJèĂĚäžčçăĄïijŽ

```
# workermp.py

from multiprocessing.connection import Client
from multiprocessing.reduction import recv_handle
import os
from socket import socket, AF_INET, SOCK_STREAM

def worker(server_address):
    serv = Client(server_address, authkey=b'peekaboo')
    serv.send(os.getpid())
    while True:
        fd = recv_handle(serv)
```

```
        print('WORKER: GOT FD', fd)
        with socket(AF_INET, SOCK_STREAM, fileno=fd) as client:
            while True:
                msg = client.recv(1024)
                if not msg:
                    break
                print('WORKER: RECV {!r}'.format(msg))
                client.send(msg)

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 2:
        print('Usage: worker.py server_address', file=sys.stderr)
        raise SystemExit(1)

    worker(sys.argv[1])
```

èęĄè£ŘèąŇåůěä¡IJèĂĚïijŇæĽ'ğèąŇæĽ'ğèąŇåŚ¡äżď *python3 workermp.py /tmp/servconn* . æŢĹæđIJèů§ä¡£çŤĺPipe()ä¿Ńå■ŘæŸŕåőŇåĚĺäÿĂæăůçŽĎãĂĆ æŰĞäżűæŔŔè£řçňęçŽĎäijăéĂŠäijŽæűĽ'åŔŁåĹřUNIXå§§åěŮæŐěå■ŮçŽĎåĹŻåżžåŠŇåěŮæŐěå■ŮçŽĎ sendmsg() æŰźæşŢãĂĆ äÿ■è£Ğè£Źçğ■æŁĂæIJŕåźűäÿ■åÿÿèğĄïijŇäÿŃéİćæŸŕä¡£çŤĺåěŮæŐěå■Ůæİěäijăé

```
# server.py
import socket

import struct

def send_fd(sock, fd):
    '''
    Send a single file descriptor.
    '''
    sock.sendmsg([b'x'],
                 [(socket.SOL_SOCKET, socket.SCM_RIGHTS, struct.
↪pack('i', fd))])
    ack = sock.recv(2)
    assert ack == b'OK'

def server(work_address, port):
    # Wait for the worker to connect
    work_serv = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM)
    work_serv.bind(work_address)
    work_serv.listen(1)
    worker, addr = work_serv.accept()

    # Now run a TCP/IP server and send clients to worker
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, True)
    s.bind(('',port))
    s.listen(1)
    while True:
```

```
        client, addr = s.accept()
        print('SERVER: Got connection from', addr)
        send_fd(worker, client.fileno())
        client.close()

if __name__ == '__main__':
    import sys
    if len(sys.argv) != 3:
        print('Usage: server.py server_address port', file=sys.
↪stderr)
        raise SystemExit(1)

    server(sys.argv[1], int(sys.argv[2]))
```

äÿŃéÌćæŸŕä¡£çŤĺåěŮæŐěå■ŮçŽĎåůëä¡IJèĂĚåőđçŐřïijŽ

```
# worker.py
import socket
import struct

def recv_fd(sock):
    '''
    Receive a single file descriptor
    '''
    msg, ancdata, flags, addr = sock.recvmsg(1,
                                     socket.CMSG_LEN(struct.
↪calcsize('i')))

    cmsg_level, cmsg_type, cmsg_data = ancdata[0]
    assert cmsg_level == socket.SOL_SOCKET and cmsg_type == socket.
↪SCM_RIGHTS
    sock.sendall(b'OK')

    return struct.unpack('i', cmsg_data)[0]

def worker(server_address):
    serv = socket.socket(socket.AF_UNIX, socket.SOCK_STREAM)
    serv.connect(server_address)
    while True:
        fd = recv_fd(serv)
        print('WORKER: GOT FD', fd)
        with socket.socket(socket.AF_INET, socket.SOCK_STREAM,␣
↪fileno=fd) as client:
            while True:
                msg = client.recv(1024)
                if not msg:
                    break
                print('WORKER: RECV {!r}'.format(msg))
                client.send(msg)
```

```
if __name__ == '__main__':
    import sys
    if len(sys.argv) != 2:
        print('Usage: worker.py server_address', file=sys.stderr)
        raise SystemExit(1)

    worker(sys.argv[1])
```

åęĆædIJä¡ăæČşåIJÍä¡ăçŽĎçÍŃåžŔäÿ■äijăéĂŠæŰĞäžűæŔŔè£řçňęïijŇåżžèőőä¡ăåŔĆéŸĚåĚűäžŰäÿĂäžŽ
æŕŤåęĆ `Unix Network Programming by W. Richard Stevens (Prentice Hall, 1990)`. åIJÍWindowsäÿŁäijăéĂŠæŰĞäžűæŔŔè£řçňęèů§UnixæŸŕäÿ■äÿĂæăůçŽĎïijŇåżžèőőä¡ăçă
`multiprocessing.reduction` äÿ■çŽĎæžŘäžčçăĄçIJŃçIJŃåĚűåůëä¡IJåŐ§çŘĘãĂĆ

13.12 11.12 çŘĘèğčäžŃäžűéľśåŁÍçŽĎIO

éŮőécŸ

ä¡ăåžŤèŕěåůšçżŔåŘňè£Ğå§žäžŐäžŃäżűéľśåŁÍæĹŰåijĆæ■ěI/OçŽĎåŇĚïijŇä¡ĘæŸŕä¡ăè£Ÿäÿ■èČ¡åőŇåĚ
æĹŰèĂĚæŸŕåęĆædIJä¡£çŤĺåőČçŽĎèŕÍäijŽåŕźä¡ăçŽĎçÍŃåžŔäžğçŤ§äżĂäźĹå¡śåŞ■ãĂĆ

èğčåĘşæŰźæąĹ

äžŃäżűéľśåŁÍI/OæIJňèť´ÍäÿŁæİěèőšåřśæŸŕåřĘå§žæIJňI/OæŞ■ä¡IJïijĹæŕŤåęĆèŕżåŠŇåĘŹïijL'è¡ňåŇŰäÿž
ä¿ŃåęĆïijŇå¡ŞæŢřæ■őåIJÍæ§ŘäÿłsocketäÿŁèćńæŐěåŘŮåŘŐïijŇåőČäijŽè¡ňæ■ćæĹŘäÿĂäÿł
`receive` äžŃäżűïijŇçĎűåŘŐèćńä¡ăåőŽäźL'çŽĎåŻđèřČæŰźæşŢæĹŰåĞ¡æŢřæİěåď´ĎçŘĘãĂĆ
ä¡IJäÿžäÿĂäÿłåŔŕèČ¡çŽĎèţůåğŇçĆźïijŇäÿĂäÿłäžŃäżűéľśåŁÍçŽĎæąĘæđůåŔŕèČ¡äijŽäžěäÿĂäÿłåőđçŐřäžĘä

```
class EventHandler:
    def fileno(self):
        'Return the associated file descriptor'
        raise NotImplemented('must implement')

    def wants_to_receive(self):
        'Return True if receiving is allowed'
        return False

    def handle_receive(self):
        'Perform the receive operation'
        pass

    def wants_to_send(self):
        'Return True if sending is requested'
        return False

    def handle_send(self):
        'Send outgoing data'
        pass
```

è£Źäÿłçśżç ŽĎåőđä¿Ńä¡IJäÿžæŔŠäžűèćńæŤ¿åĚěçśżäijijäÿŃéIćè£Źæăůç ŽĎäžŃäžűå¿łçŐŕäÿ■ïijŽ

```
import select

def event_loop(handlers):
    while True:
        wants_recv = [h for h in handlers if h.wants_to_receive()]
        wants_send = [h for h in handlers if h.wants_to_send()]
        can_recv, can_send, _ = select.select(wants_recv, wants_
↪send, [])
        for h in can_recv:
            h.handle_receive()
        for h in can_send:
            h.handle_send()
```

äžŃäžűå¿łçŐŕçŽĎåĚşéŤőéČĺåĹĘæŸŕ `select()` èřČçŤĺïijŇåőČäijŽäÿ■æŰ■è¡őèŕćæŰĞäžűæŔŔè£řçňę åIJĺèřČçŤĺ `select()` äźŃåĽ■ïijŇæŮűéŮť å¿łçŐŕäijŽèŕćéŮőæĽĂæIJĽçŽĎåď ĎçŘĘåŹĺæİěåĘşåőŽåŞłäÿĂä çĎűåŘŐåőČåřĘçżŞæđIJåĹŮèąĺæŔŘä¿ŻçżŹ `select()` ãĂĆçĎűåŘŐ `select()` è£ŤåŽđåĞĘåď ĞæŐěåŘŮæĹŰåŘŚéĂĄçŽĎåŕźèśąçżĎæĹŘçŽĎåĹŮèąĺãĂĆ çĎűåŘŐçŻÿåžŤçŽĎ `handle_receive()` æĹŰ `handle_send()` æŰźæşŢèćńèğęåŘŚãĂĆ

çijŰåĘŹåžŤçŤĺçÍŃåžŘçŽĎæŮűåĂŹïijŇ`EventHandler` çŽĎåőđä¿ŃäijŽèćńåĹŻåžžãĂĆä¿ŃåęĆïijŇäÿŃéIćæŸŕäÿď äÿłçőĂå■ŢçŽĎå§žäžŐUDPç¡ŚçżIJæIJ■åŁąçŽĎåd

```
import socket
import time

class UDPServer(EventHandler):
    def __init__(self, address):
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        self.sock.bind(address)

    def fileno(self):
        return self.sock.fileno()

    def wants_to_receive(self):
        return True

class UDPTimeServer(UDPServer):
    def handle_receive(self):
        msg, addr = self.sock.recvfrom(1)
        self.sock.sendto(time.ctime().encode('ascii'), addr)

class UDPEchoServer(UDPServer):
    def handle_receive(self):
        msg, addr = self.sock.recvfrom(8192)
        self.sock.sendto(msg, addr)

if __name__ == '__main__':
    handlers = [ UDPTimeServer(('',14000)), UDPEchoServer(('',
↪15000))  ]
```

```
    event_loop(handlers)
```

æţŃèŕŢè£Źæőţäžčçă Ąïij Ňèŕ ŢçİĂäžŐåŔęåď Űäÿ Ăäÿł Pythonè ğčé ĞŁåŹÍè£đæŐěåőČïijŽ

```
>>> from socket import *
>>> s = socket(AF_INET, SOCK_DGRAM)
>>> s.sendto(b'',('localhost',14000))
0
>>> s.recvfrom(128)
(b'Tue Sep 18 14:29:23 2012', ('127.0.0.1', 14000))
>>> s.sendto(b'Hello',('localhost',15000))
5
>>> s.recvfrom(128)
(b'Hello', ('127.0.0.1', 15000))
>>>
```

åőđçŐřäÿ Ăäÿł TCPæIJ■åŁąåŹÍäijŽæŻťåŁååď■æİĆäÿ ĂçĆźïijŇåŻääÿžæŕŔäÿ ĂäÿłåőćæĹůçńŕéČ¡èęĄåĹİäÿ ŃéİćæŸŕäÿ Ăäÿł TCPåžŤç■ŤåőćæĹůçńŕä¿Ńå■ŘïijŽ

```
class TCPServer(EventHandler):
    def __init__(self, address, client_handler, handler_list):
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_
↪STREAM)
        self.sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR,
↪ True)
        self.sock.bind(address)
        self.sock.listen(1)
        self.client_handler = client_handler
        self.handler_list = handler_list

    def fileno(self):
        return self.sock.fileno()

    def wants_to_receive(self):
        return True

    def handle_receive(self):
        client, addr = self.sock.accept()
        # Add the client to the event loop's handler list
        self.handler_list.append(self.client_handler(client, self.
↪handler_list))

class TCPClient(EventHandler):
    def __init__(self, sock, handler_list):
        self.sock = sock
        self.handler_list = handler_list
        self.outgoing = bytearray()

    def fileno(self):
        return self.sock.fileno()
```

```
    def close(self):
        self.sock.close()
        # Remove myself from the event loop's handler list
        self.handler_list.remove(self)

    def wants_to_send(self):
        return True if self.outgoing else False

    def handle_send(self):
        nsent = self.sock.send(self.outgoing)
        self.outgoing = self.outgoing[nsent:]

class TCPEchoClient(TCPClient):
    def wants_to_receive(self):
        return True

    def handle_receive(self):
        data = self.sock.recv(8192)
        if not data:
            self.close()
        else:
            self.outgoing.extend(data)

if __name__ == '__main__':
    handlers = []
    handlers.append(TCPServer(('',16000), TCPEchoClient, handlers))
    event_loop(handlers)
```

TCPä¿Ńå■ŘçŽĎåĚşéŤőçĆźæŸŕäżŐåď'ĎçŘĘåŹĺäÿ■åĹŮèąĺåćđåŁäåŠŇåĹăéŹď'åőćæĹůçńŕçŽĎæŞ■ä¡IJã
åŕźæŕŔäÿĂäÿłè£đæŐěïijŇäÿĂäÿłæŰřçŽĎåď'ĎçŘĘåŹĺèćńåĹŻåžžåźůåŁäåĹřåĹŮèąĺäÿ■ãĂĆå¡Şè£đæŐěèćńåĚ
åęĆæđIJä¡äè£ŘèąŇçÍŃåžŘåźűèŕŢçİĂçŤÍTelnetæĹŰçśžäijijåůěåĚůè£đæŐěïijŇåőČäijŽåřĘä¡äåŔŠéĂĄçŽĎæű

èőÍèőž

åőđéŹĚäÿŁæL'ĂæIJL'çŽĎäžŃäžűél'śåŁÍæąĘæđűåŐ§çŘĘèů§äÿŁéÍćçŽĎä¿Ńå■ŘçŻÿåůőæŮäåĞããĂĆåőđ
ä¡ĘæŸŕåIJÍæIJĂæăÿå£ČçŽĎéČĺåĹĘïijŇéČ¡äijŽæIJL'äÿĂäÿłè¡őèŕćçŽĎå¿łçŐŕæİěæčĂæ§ěæt'żåŁÍsocketïijŇå

äžŃäžűél'śåŁÍI/OçŽĎäÿĂäÿłåŔŕèČ¡åě¡åd'ĎæŸŕåőČèČ¡åd'ĎçŘĘéİđåÿÿåd'ğçŽĎåźűåŔŚè£đæŐěïijŇèĂŇ
äź§åřśæŸŕèŕt'ïijŇselect() èřČçŤĺïijĹæĹŰåĚűäžŰç■L'æŢĹçŽĎïijL'èČ¡çŻŚåŘňåd'ğéĞŘçŽĎsocketåźűåŞ■
åIJĺå¿łçŐŕäÿ■äÿĂæňąåd'ĎçŘĘäÿĂäÿłäžŃäžűïijŇåźűäÿ■éIJĂèęĄåĚűäžŰçŽĎåźűåŔŚæIJžåĹűãĂĆ

äžŃäžűél'śåŁÍI/OçŽĎçijžçĆźæŸŕæšąæIJL'çIJ§æ■čçŽĎåŘŇæ■ěæIJžåĹűãĂĆ
åęĆæđIJäżżä¡ŢäžŃäžűåd'ĎçŘĘåŹĺæŰźæşŢéŸżąąđæĹŰæL'ğèąŇäÿĂäÿłèĂŮæŮűèőąçőŮïijŇåőČäijŽéŸżąąđa
èřČçŤĺéĆčäžŻåźűäÿ■æŸŕäžŃäžűél'śåŁĺéčŐæăijçŽĎåžŞåĞ¡æŢřäź§äijŽæIJL'éŮőécŸïijŇåŘŇæăůèęĄæŸŕæ§Ł

åŕźäžŐéŸżåąđæĹŰèĂŮæŮűèőąçőŮçŽĎéŮőécŸåŘŕäżěéĂŽè£ĞåřĘäžŃäžűåŘŚéĂĄäÿłåĚűäžŰå■ŢçŃňç
äÿ■è£ĞïijŇåIJĺäžŃäžűå¿łçŐŕäÿ■åijŢåĚěåd'Žçž£çĺŃåŠŇåd'Žè£ŻçĺŃæŸŕæŕŤè¿ČæčŸæL'ŃçŽĎïijŇ
äÿŃéİćçŽĎä¿Ńå■ŘæijŤçd'žäžĘåęĆä¡Ţä¡£çŤĺ concurrent.futures
æĹąåİŮæİěåőđçŐŕïijŽ

```
from concurrent.futures import ThreadPoolExecutor
import os

class ThreadPoolHandler(EventHandler):
    def __init__(self, nworkers):
        if os.name == 'posix':
            self.signal_done_sock, self.done_sock = socket.
↪socketpair()
        else:
            server = socket.socket(socket.AF_INET, socket.SOCK_
↪STREAM)
            server.bind(('127.0.0.1', 0))
            server.listen(1)
            self.signal_done_sock = socket.socket(socket.AF_INET,
                                                  socket.SOCK_
↪STREAM)
            self.signal_done_sock.connect(server.getsockname())
            self.done_sock, _ = server.accept()
            server.close()

        self.pending = []
        self.pool = ThreadPoolExecutor(nworkers)

    def fileno(self):
        return self.done_sock.fileno()

    # Callback that executes when the thread is done
    def _complete(self, callback, r):

        self.pending.append((callback, r.result()))
        self.signal_done_sock.send(b'x')

    # Run a function in a thread pool
    def run(self, func, args=(), kwargs={},*,callback):
        r = self.pool.submit(func, *args, **kwargs)
        r.add_done_callback(lambda r: self._complete(callback, r))

    def wants_to_receive(self):
        return True

    # Run callback functions of completed work
    def handle_receive(self):
        # Invoke all pending callback functions
        for callback, result in self.pending:
            callback(result)
            self.done_sock.recv(1)
        self.pending = []
```

åIJÍäžčçăĄäÿ■ïijŇrun() æŰźæşŢèćńçŤĺæİěářĘåůëä¡IJæŔŘäžďçżŹåŻđèřČåĞ¡æŢřæśäïijŇåďĎçŘĘåőŇ
åőđéŹĚåůëä¡IJèćńæŔŘäžďçżŹ ThreadPoolExecutor åőđä¿ŃãĂĆ

äÿ■è£ĞäÿĂäÿłéŽ¿çĆźæŸŕå■ŔèřČèőąçőŮçżŞæđIJåŠŇäžŃäżűå¿łçŐŕïijŇäÿžäžĘèğčåĘşåőČïijŇæĹŚäżňåĹŻåż
å¡Şçž£çĺŃæśăåőŇæĹŘåůěä¡IJåŘŐïijŇåőČäijŽæĽğèąŇçśżäÿ■çŽĎ _complete()
æŰźæşŢãĂĆ è£ŹäÿłæŰźæşŢåĘ■æ§ŘäÿłsocketäÿŁåĘŹåĚěå■ŮèŁĆäźŃåĽ■äijŽèőšæŇĆèţůçŽĎåŻđèřČåĞ¡æŢ
fileno() æŰźæşŢè£ŤåŻđåŔęåď ŰçŽĎéĆčäÿłsocketãĂĆ åŻăæ■ď ïijŇè£Źäÿłå■ŮèŁĆèćńåĘŹåĚěæŮűïijŇå
çĎűåŘŐ handle_receive() æŰźæşŢèćńæ£Ăæť żåźűäÿžæĽĂæIJĽäźŃåĽ■æŔŘäžď çŽĎåůěä¡IJæĽğèąŇ
åİęçŹ¡èőšïijŇèŕť äžĘè£ŹäźĹåď Žè£đæĹŚèĞłåůśéČ¡æŹŢäžĘãĂĆ
äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎæIJ■åŁąåŹĺïijŇæijŤçď žäžĘåęĆä¡Ţä¡£çŤĺçž£çĺŃæśăæİěåőđçŐřèĂŮæŮűçŽĎè

```
# A really bad Fibonacci implementation
def fib(n):
    if n < 2:
        return 1
    else:
        return fib(n - 1) + fib(n - 2)

class UDPFibServer(UDPServer):
    def handle_receive(self):
        msg, addr = self.sock.recvfrom(128)
        n = int(msg)
        pool.run(fib, (n,), callback=lambda r: self.respond(r,␣
↪addr))

    def respond(self, result, addr):
        self.sock.sendto(str(result).encode('ascii'), addr)

if __name__ == '__main__':
    pool = ThreadPoolHandler(16)
    handlers = [ pool, UDPFibServer(('',16000))]
    event_loop(handlers)
```

è£ŘèąŇè£ŹäÿłæIJ■åŁąåŹĺïijŇçĎűåŘŐèŕŢçİĂçŤĺåĚűåőČPythonçĺŃåžŔæİěæţŃèŕŢåőČïijŽ

```
from socket import *
sock = socket(AF_INET, SOCK_DGRAM)
for x in range(40):
    sock.sendto(str(x).encode('ascii'), ('localhost', 16000))
    resp = sock.recvfrom(8192)
    print(resp[0])
```

ä¡ăåžŤèŕěèČ¡åIJĺäÿ■åŘŇçłŮåŔčäÿ■éĞ■åď■çŽĎæĽğèąŇè£ŹäÿłçĺŃåžŔïijŇåźűäÿŤäÿ■äijŽå¡śåŞ■åĹřåĚ

åůšçżŔéŸĚèŕżåőŇäžĘè£ŹäÿĂåřŔèŁĆïijŇéĆčäźĹä¡ăåžŤèŕěä¡£çŤĺè£ŹéĞŇçŽĎäżčçăĄåŘŮïij§äź§èőÿäÿ
äÿ■è£ĞïijŇåęĆæđIJä¡ăçŘĘèğčäžĘå§žæIJňåŐ§çŘĘïijŇä¡ăåřśèČ¡çŘĘèğčè£ŹäžŻæąĘæđűæĽĂä¡£çŤĺçŽĎæäÿ
ä¡IJäÿžåŕźåŻđèřČåĞ¡æŢřçijŰçĺŃçŽĎæŻ£äżčïijŇäžŃäżűéľśåŁĺçijŰçăĄæIJĽæŮűåĂŹäijŽä¡£çŤĺåĹřå■ŔçĺŃï

13.13 11.13 åŔŚéĂĄäÿŐæŐěæŤűåď ğåđŃæŢřçżĎ

éŮőéćŸ

ä¡ăèęĄéĂŽè£Ğç¡ŚçżIJè£đæŐěåŔŚéĂĄåŠŇæŐěåŔŮè£đçż■æŢřæ■őçŽĎåď ğåđŃæŢřçżĎïijŇåźűåř¡éĞŔ

èğčåĘşæŰźæąĹ

äÿŃéİćçŽĎåĞ¡æŢřåĹľçŤĺ memoryviews æİěåŔŚéĂĄåŠŇæŐěåŔŮåď˙ğæŢřçżĎïijŽ

```
# zerocopy.py

def send_from(arr, dest):
    view = memoryview(arr).cast('B')
    while len(view):
        nsent = dest.send(view)
        view = view[nsent:]

def recv_into(arr, source):
    view = memoryview(arr).cast('B')
    while len(view):
        nrecv = source.recv_into(view)
        view = view[nrecv:]
```

äÿžäžĘæţŃèŕŢçĺŃåžŔïijŇéęŰåĚĹåĹŻåžžäÿĂäÿłéĂŽè£Ğsocketè£đæŐěçŽĎæIJ■åŁąåŹĺåŠŇåőćæĹůçńŕç

```
>>> from socket import *
>>> s = socket(AF_INET, SOCK_STREAM)
>>> s.bind(('', 25000))
>>> s.listen(1)
>>> c,a = s.accept()
>>>
```

åIJĺåőćæĹůçńŕïijĹåŔęåď˙ŰäÿĂäÿłèğčéĞŁåŹĺäÿ■ïijL˙ïijŽ

```
>>> from socket import *
>>> c = socket(AF_INET, SOCK_STREAM)
>>> c.connect(('localhost', 25000))
>>>
```

æIJňèŁĆçŽĎçŻőæăĞæŸŕä¡ăèČ¡éĂŽè£Ğè£đæŐěäijăè¿ŞäÿĂäÿłèűĚåď˙ğæŢřçżĎãĂĆè£ŹçğæČĚåĘţçŽĎ
array æĺąåİŮæĹŰ numpy æĺąåİŮæİěåĹŻåžžæŢřçżĎïijŽ

```
# Server
>>> import numpy
>>> a = numpy.arange(0.0, 50000000.0)
>>> send_from(a, c)
>>>

# Client
>>> import numpy
>>> a = numpy.zeros(shape=50000000, dtype=float)
>>> a[0:10]
array([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.])
>>> recv_into(a, c)
>>> a[0:10]
array([ 0.,  1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9.])
```

```
>>>
```

èőÍèőž

åIJÍæŢřæ■őåŕĘéŻĘåđŃåĹĘåÿČåijŔèőąçőŮåŠŇåźşèąŇèőąçőŮçÍŃåžŔäÿ■ïijŇèĞłåůśåĘŹçÍŃåžŔæÍěåőđç
äÿ■è£ĞïijŇèęĄæŸŕä¡ăçąőåőđæČşè£Źæăůå ĄŽïijŇä¡ăåŔŕèČ¡éIJĂèęĄåŕĘä¡ăçŽĎæŢřæ■őè¡ñæ■ćæĹŘåŐ§ågŃ
ä¡ăåŔŕèČ¡è£ŸéIJĂèęĄåŕĘæŢřæ■őåĹĞåL'šæĹŘåđ'Žäÿłå İŮïijŇåŻăäÿžåđ'ğéČÍåĹĘåŠŇç¡ŚçżIJçŻÿåĚşçŽĎåĞ¡

äÿĂçğ■æŰźæşŢæŸŕä¡£çŤÍæ§Řçğ■æIJžåĹűåžŔåĹŮåŇŰæŢřæ■őâĂŤâĂŤåŔŕèČ¡åřĘåĚűè¡ñæ■ćæĹŘäÿĂ
äÿ■è£ĞïijŇè£ŹæăůæIJĂçżĹäijŽåĹŻåžžæŢřæ■őçŽĎäÿĂäÿłåđ'■åĹűãĂĆ
åřśçőŮä¡ăåŔłæŸŕéŽűçćŐçŽĎåĄŽè£ŹäžŻïijŇä¡ăçŽĎäžčçăĄæIJĂçżĹè£ŸæŸŕäijŽæIJL'åđ'ğéĞŘçŽĎåřŔåđŃå

æIJňèŁĆéĂŽè£Ğä¡£çŤÍåĘĚå■ŸèğĘåŻ¿åśŢçđ'žäžĘäÿĂäžŻé■ŤæşŢæŞ■ä¡IJãĂĆ
æIJňèť'Íäÿŀïij Ňäÿ Ăäÿłå ĘĚå■ŸèğĘåŻ¿åřśæŸŕäÿĂäÿłåůšå■ŸåIJÍæŢřçżĎçŽĎèęĘçŽŰåśĆãĂĆäÿ■äžĚäžĚæŸŕé
åĘĚå■ŸèğĘåŻ¿è£ŸèČ¡äžěäÿ■åŘŇçŽĎæŰźåijŔè¡ñæ■ćæĹŘäÿ■åŘŇçśžåđŃæIěèąÍçŐřæŢřæ■őãĂĆ
è£Źäÿłåřśæ Ÿŕäÿ Ńéİćè£Źäÿłér■åŔěçŽĎçŽőçŽĎïijŽ

```
view = memoryview(arr).cast('B')
```

åőČæŐěåŔŮäÿĂäÿłæŢřçżĎ arråźűåřĘåĚűè¡ñæ■ćäÿžäÿĂäÿłæŮăçňęåŔůå■ŮèŁĆçŽĎåĘĚå■ŸèğĘåŻ¿ãĂĆ
æŕŤåęĆ `socket.send()` æĹŰ `send.recv_into()` ãĂĆ
åIJÍåĘĚéČÍïijŇè£ŹäžŻæŰźæşŢèČ¡åď§çŻť'æŐěæŞ■ä¡IJè£Źäÿłå ĘĚå■ŸåŇžå§§ãĂĆä¿ŃåęĆïijŇ`sock.send()` çŻť'æŐěäžŐåĘĚå■ŸäÿåŔŞçŤ§æŢřæ■őèĂŇäÿ■éIJĂèęĄåď■åĹűãĂĆ `send.recv_into()` ä¡£çŤÍè£ŹäÿłåĘĚå■ŸåŇžå§§ä¡IJäÿžæŐěåŔŮæŞ■ä¡IJçŽĎè¿ŞåĚěçijŞåĘšåŇžãĂĆ

åL'l'äÿŃçŽĎäÿĂäÿłéŽ¿çĆźåŕśæŸŕsocketåĞ¡æŢřåŔŕèČ¡åŔłæŞ■ä¡IJéČÍåĹĘæŢřæ■õãĂĆ
éĂŽåÿÿæİěèőšïijŇæĹŚäžňå¿Ůä¡£çŤÍå¿Ĺåđ'Žäÿ■åŘŇçŽĎ `send()` åŠŇ `recv_into()`
æİěäijåè¿ŞæŢť'äÿłæŢřçżĎãĂĆ äÿ■çŤÍæŃĚå£ČïijŇæŕŔæňąæŞ■ä¡IJåŘŐïijŇèğĘåŻ¿äijŽéĂŽè£ĞåŔŚéĂĄæĹŰ
æŰřçŽĎèğĘåŻ¿åŘŇæăůäźş§æŸŕåĘĚå■ŸèęĘçŽŰåśĆãĂĆåŽăæ■d'ïijŇè£ŸæŸŕæšąæIJL'äžżä¡ŢçŽĎåđ'■åĹűæŞ

è£ŹéĞŇæIJL'äÿłéŮőécŸåŕśæŸŕæŐěåŔŮèĂĚå£ĚéążäžŇåĚĹç§ěéĄŞæIJL'åđ'ŽåŕŚæŢřæ■őèęĄècńåŔŚéĂ
äžěä¿£åőĆèČ¡écĎåĹĘéĚ■äÿĂäÿłæŢřçżĎæĹŰèĂĚçąőä£İåőĆèČ¡åřĘæŐěåŔŮçŽĎæŢřæ■őæŤ¿åĚěäÿĂäÿłåůš
åęĆæđIJæšąåŁđæşŢç§ěéĄŞçŽĎèŕÍïijŇåŔŚéĂĄèĂĚåŕśå¿ŮåĚĹåřĘæŢřæ■őåđ'ğåŕŔåŔŚéĂĄè£ĞæİěïijŇçĎűå

14 çňňå■ĄäžŇçńăïijŽåźűåŔŚçijŰçÍŃ

åŕźäžŐåźűåŔŚçijŰçÍŃ, PythonæIJL'åđ'Žçğ■éŢ£æIJ§æŤŕæŇĄçŽĎæŰźæşŢ,
åŃĚæŃňåđ'Žçž£çÍŃ, èřČçŤÍå■Řè£ŻçÍŃ, äžěåŔŁåŘĎçğ■åŘĎæăůçŽĎåĚşäžŐçŤ§æĹŘåŹĺåĞ¡æŢřçŽĎæŁĂåů
è£ŹäÿĂçńäåřĘäijŽçżŹåĞžåźűåŔŚçijŰçÍŃåŘĎçğ■æŰźéİćçŽĎæŁĂåůğ,
åŃĚæŃňéĂŽçŤÍçŽĎåđ'Žçž£çÍŃæŁĂæIJŕäžěåŔŁåźűèąŇèőąçőŮçŽĎåőđçŐřæŰźæşŢ.

åČŔçżŔéłŇäÿřáŕŃçŽĎçÍŃåžŔåŚŸæL'Ăç§ěéĄŞçŽĎéĆčæăů,
åđ'ğåőűæŃĚå£ČåźűåŕśçŽĎçÍŃåžŔæIJL'æ¡IJåIJÍçŽĎå■śéŹl'. åŽăæ■d',
æIJňçńăçŽĎäÿżèęĄçŽőæăĞäźŃäÿĂæŸŕçżŹåĞžæŻť'åŁăåŔŕä£ąèţŰåŠŇæŸŞěŕČèŕŢçŽĎäžčçăĄ.

Contents:

14.1 12.1 åŘŕåŁÍäÿŐåĄIJæ■ćçž£çÍŃ

éŮőécŸ

ä¡ăèęĄäÿžéIJĂèęĄåźűåŔŚæL'ğèąŇçŽĎäžčçăĄåĹŻåžž/éŤĂæŕĄçž£çÍŃ

èğčåĘşæŰźæąĹ

threading åžŞåŔŕäżěåIJÍå■ŢçŃňçŽĎçž£çÍŃäÿ■æL'ğèąŇäżżä¡ŢçŽĎåIJÍ Python äÿ■åŔŕäżěèřČçŤÍçŽĎåŕźèśąãĂĆä¡ăåŔŕäżěåĹŻåžžäÿĂäÿł Thread åŕźèśąåźűåřĘä¡ăèęĄæL'ğèąŇçŽĎåŕźèśąäžě target åŔĆæŢřçŽĎå¡ćåijŔæŔŘä¿ŻçžŹèŕěåŕźèśąãĂĆ äÿŃéÍćæŸŕäÿĂäÿłçőĂå■ŢçŽĎä¿Ńå■ŘïijŽ

```
# Code to execute in an independent thread
import time
def countdown(n):
    while n > 0:
        print('T-minus', n)
        n -= 1
        time.sleep(5)

# Create and launch a thread
from threading import Thread
t = Thread(target=countdown, args=(10,))
t.start()
```

å¡Şä¡ăåĹŻåžžåě¡äÿĂäÿłçž£çÍŃåŕźèśąåŘŐïijŇèŕěåŕźèśąåźűäÿ■äijŽçńŃå■şæL'ğèąŇïijŇéŹd'éÍđä¡ăèřČçŤÍä start() æŰźæşŢïijĹå¡Şä¡ăèřČçŤÍ start() æŰźæşŢæŮűïijŇåőČäijŽèřČçŤÍä¡ăäijăéĂŠè£ŻæÍěçŽĎåĞ¡æŢ POSIX çž£çÍŃæĹŰèĂĚäÿĂäÿł Windows çž£çÍŃïijL'ïijŇè£ŹäžŽçž£çÍŃåřĘçŤśæŞ■ä¡IJçşżçż§æÍěåĚÍæÍČçőąçĹ

```
if t.is_alive():
    print('Still running')
else:
    print('Completed')
```

ä¡ăäź§åŔŕäżěåřĘäÿĂäÿłçž£çÍŃåŁăåĚěåĹřå¡ŞåL'■çž£çÍŃïijŇåźűç■L'å¿ĚåőČçżĹæ■ćïijŽ

```
t.join()
```

PythonèğčéĞŁåŹÍçŻt'åĹřæL'ĂæIJL'çž£çÍŃéČ¡çżĹæ■ćåL'■äż■ä‡ÍæŇĄè£ŘèąŇãĂĆåŕźäžŐéIJĂèęĄéŢ£æ ä¿ŃåęĆïijŽ

```
t = Thread(target=countdown, args=(10,), daemon=True)
t.start()
```

åŘŐåŔřçž£çÍŃæŮăæŞç■L'å¿ĚïijŇäÿ■è£ĞïijŇè£ŹäžŽçž£çÍŃäijŽåIJÍäÿżçž£çÍŃçżĹæ■ćæŮűèĞłåŁÍéŤĂ éŹd'äžĘåęĆäÿŁæL'Ăçd'žçŽĎäÿd'äÿłæŞ■ä¡IJïijŇåźűæšąæIJL'åd'łåd'ŽåŔŕäžěåŕźçž£çÍŃåĄŽçŽĎäžŃæČĚãĂĆ

```
class CountdownTask:
    def __init__(self):
        self._running = True

    def terminate(self):
        self._running = False

    def run(self, n):
        while self._running and n > 0:
            print('T-minus', n)
            n -= 1
            time.sleep(5)

c = CountdownTask()
t = Thread(target=c.run, args=(10,))
t.start()
c.terminate() # Signal termination
t.join()      # Wait for actual termination (if needed)
```

åęĆædIJçž£çÍŃæL'ğèąŇäÿĂäžŻåČŔI/Oè£Źæăůç ŽĎéŸżåąđæŞ■ä¡IJïijŇéĆčäźĹéĂŽè£Ğè¡őèŕćæİěçżĹæ■ä¿Ńå■ŘåęĆäÿŇïijŽ

```
class IOTask:
    def terminate(self):
        self._running = False

    def run(self, sock):
        # sock is a socket
        sock.settimeout(5)         # Set timeout period
        while self._running:
            # Perform a blocking I/O operation w/ timeout
            try:
                data = sock.recv(8192)
                break
            except socket.timeout:
                continue
            # Continued processing
            ...
        # Terminated
        return
```

èőÍèőž

çŤśäžŐåĚĺåśĂèğčéĞŁéŤĄïijĹGILïijL'çŽĎåŐ§åŻãïijŇPython çŽĎçž£çÍŃèćńéŹŘåĹűåĹřåŘŇäÿĂæŮűåĹżåŔĺåĚĄèőÿäÿĂäÿłçž£çÍŃæL'ğèąŇè£ŹæăůäÿĂäÿłæL'ğèąŇæĺąåđŇçŽĎçž£çÍŃæŻť éĂĆçŤĺäžŐåď ĎçŘĘI/OåŠŇåĚűäżŮéIJĂèęĄåźűåŔŚæL'ğèąŇçŽĎéŸżåąđæŞ■ä¡IJïijĹærŤåęĆ

æIJL'æŮűä¡ăäijŽçIJŃåĹřäÿŃè¿źè£Źçğ■éĂŽè£ĞçżğæL'£ `Thread` çśżæİěåőđçŐřçŽĎçž£çÍŃïijŽ

```
from threading import Thread

class CountdownThread(Thread):
    def __init__(self, n):
        super().__init__()
        self.n = 0
    def run(self):
        while self.n > 0:

            print('T-minus', self.n)
            self.n -= 1
            time.sleep(5)

c = CountdownThread(5)
c.start()
```

ăř¡çőąè£Źæăůäź§åŔŕäżěåůěä¡IJïijŇä¡Ęè£Źä¡£å¿Ůä¡ăçŽĎäżčçăĄä¿İèţŰäžŐ threading åžŞïijŇæĽĂäżěä¡ăçŽĎè£ŹäžŻäżčçăĄåŔłèČ¡åIJĺçž£çĺŃäÿŁäÿŃæŰĞäÿ■ä¡£çŤĺãĂĆäÿŁæŰĞæ threading åžŞæŮăåĚşçŽĎïijŇè£Źæăůåŕśä¡£å¿Ůè£ŹäžŻäżčçăĄåŔŕäżěèćńçŤĺåIJĺåĚűäżŰçŽĎäÿŁäÿŃæŰĞ multiprocessing æĺąåİŮåIJĺäÿĂäÿłå■ŢçŃňçŽĎè£ŻçĺŃäÿ■æĽğèąŇä¡ăçŽĎäżčçăĄïijŽ

```
import multiprocessing
c = CountdownTask(5)
p = multiprocessing.Process(target=c.run)
p.start()
```

åĘ■æňąéĞ■çŤşïijŇè£ŹæőţäżčçăĄäżĚéĂĆçŤĺäžŐ CountdownTask çśżæŸŕäżěçŃňçńŃäžŐåőđéŹĚçŽĎåźűåŔŚæĽŃæőţïijĹåď'Žçž£çĺŃãĂĄåď'Žè£ŻçĺŃç■Ľ'ç■Ľ'ïijĽ'åőđçŐřçŽĎæ

14.2 12.2 åĹďæŰ■çž£çĺŃæŸŕåŘęåůšçżŔåŘŕåŁĺ

éŮőéćŸ

ä¡ăåůšçżŔåŘŕåŁĺäžĘäÿĂäÿłçž£çĺŃïijŇä¡ĘæŸŕä¡ăæČşç§ěéĄŞåőČæŸŕäÿ■æŸŕçIJ§çŽĎåůšçżŔåijĂåğŃè£

èğčåĘşæŰźæąĹ

çž£çĺŃçŽĎäÿĂäÿłåĚşéŤőçĽźæĂğæŸŕæŕŔäÿłçž£çĺŃéČ¡æŸŕçŃňçńŃè£ŘèąŇäÿŤçŁűæĂĄäÿ■åŔŕéćĎæţ threading åžŞäÿ■çŽĎ Event åŕźèśąãĂĆ Event åŕźèśąåŇĚåŘńäÿĂäÿłåŔŕçŤśçž£çĺŃèő¿ç¡őçŽĎä£ąåŔůæ åŕźèśąäÿ■çŽĎä£ąåŔůæăĞå£Ůèćńèő¿ç¡őäÿžåĄĞãĂĆåęĆæđIJæIJĽçž£çĺŃç■Ľ'å¿ĚäÿĂäÿł event åŕźèśąïijŇèĂŇè£Źäÿł event åŕźèśąçŽĎæăĞå£ŮäÿžåĄĞïijŇéĆčäźĹè£Źäÿłçž£çĺŃåŕĘäijŽèćńäÿĂçŻť'éŸ event åŕźèśąçŽĎä£ąåŔůæăĞå£Ůèő¿ç¡őäÿžçIJ§ïijŇåőČåŕĘåŤď'éĘŠæĽĂæIJĽ'ç■Ľ'å¿Ěè£Źäÿł event åŕźèśąçŽĎçž£çĺŃãĂĆåęĆæđIJäÿĂäÿłçž£çĺŃç■Ľ'å¿ĚäÿĂäÿłåůšçżŔèćńèő¿ç¡őäÿžçIJ§çŽĎ event åŕźèśąïijŇéĆčäźĹåőČåŕĘå£¡çŢěè£ŹäÿłäžŃäżűïijŇçżğçż■æĽğèąŇãĂĆ äÿŃè¿źçŽĎäżčçăĄåśŢçď'žäžĘåęĆä¡Ţä¡£çŤĺ Event æİěå■ŔèřČçž£çĺŃçŽĎåŘŕåŁĺïijŽ

```
from threading import Thread, Event
import time

# Code to execute in an independent thread
def countdown(n, started_evt):
    print('countdown starting')
    started_evt.set()
    while n > 0:
        print('T-minus', n)
        n -= 1
        time.sleep(5)

# Create the event object that will be used to signal startup
started_evt = Event()

# Launch the thread and pass the startup event
print('Launching countdown')
t = Thread(target=countdown, args=(10,started_evt))
t.start()

# Wait for the thread to start
started_evt.wait()
print('countdown is running')
```

å¡Şä¡äæL'ğèąŇè£ŹæőţäžčçăĄïijŇâĂIJcountdown is runningâĂİ æĂżæŸŕæŸ¿çd'žåIJÍ âĂIJcountdown startingâĂİ äźŃåŘŐæŸ¿çd'žãĂĆè£ŹæŸŕçŤśäžŐä¡£çŤÍ event æİěå■ŔèřČçž£çÍŃïijŇä¡£å¿Ůäÿżçž£çÍŃèęĄç■L'åĹř `countdown()` åĞ¡æŢřè¿ŞåĞžåŘŕåŁÍä£qæĄŕåŘŐïijŇæL'■èČ¡çżğçż■æL'ğèąŇãĂĆ

èőÍèőž

event åŕźèśąæIJĂåě¡å■Ţæňąä¡£çŤÍïijŇåřśæŸŕèŕt'ïijŇä¡ăåĹŻåžžäÿĂäÿł event åŕźèśąïijŇèőľæ§Řäÿł£çÍŃç■L'å¿Ěè£ŹäÿłåŕźèśąïijŇäÿĂæŮęè£Źäÿłåŕźèśąèćńèő¿ç¡őäÿžçIJ§ïijŇä¡ăåřśåžŤèŕè `clear()` æŰźæşŢæİěéĞ■ç¡ő event åŕźèśąïijŇä¡ĘæŸŕå¿ĹéŽ¿çąőä£ĺåőL'åĚĺåIJřæÿĚçŘĘ event åŕźèśąåźűåŕźåőČéĞ■æŰřèţŃåĂijãĂĆå¿ĹåŘŕèČ¡äijŽåŘŚçŤ§éŤŹè£ĞäžŃäžűãĂĄæ■żéŤĄæĹŰèĂĚåĚűä event åŕźèśąçŽĎäžčçăĄäijŽåIJÍçž£çÍŃåĘ■æňąç■L'å¿Ěè£ŹäÿłåŕźèśąäźŃåL'■æL'ğèąŇïijL'ãĂĆåęĆæđIJäÿĂäÿłçž£çÍŃéIJĂèęĄäÿ■åĄIJåIJřéĞ■åd'■ä¡£çŤÍ event åŕźèśąïijŇä¡ăæIJĂåě¡ä¡£çŤÍ `Condition` åŕźèśąæİěäžčæŽ£ãĂĆäÿŃéİćçŽĎäžčçăĄä¡£çŤÍ `Condition` åŕźèśąåődçŐřäžĘäÿĂäÿłåŚÍæIJ§åőŽæŮűåŹÍïijŇærŔå¡ŞåőŽæŮűåŹÍèűĚæŮűçŽĎæŮűåĂŹïijŇåĹ

```
import threading
import time

class PeriodicTimer:
    def __init__(self, interval):
        self._interval = interval
        self._flag = 0
        self._cv = threading.Condition()
```

```
    def start(self):
        t = threading.Thread(target=self.run)
        t.daemon = True

        t.start()

    def run(self):
        '''
        Run the timer and notify waiting threads after each interval
        '''
        while True:
            time.sleep(self._interval)
            with self._cv:
                 self._flag ^= 1
                 self._cv.notify_all()

    def wait_for_tick(self):
        '''
        Wait for the next tick of the timer
        '''
        with self._cv:
            last_flag = self._flag
            while last_flag == self._flag:
                self._cv.wait()

# Example use of the timer
ptimer = PeriodicTimer(5)
ptimer.start()

# Two threads that synchronize on the timer
def countdown(nticks):
    while nticks > 0:
        ptimer.wait_for_tick()
        print('T-minus', nticks)
        nticks -= 1

def countup(last):
    n = 0
    while n < last:
        ptimer.wait_for_tick()
        print('Counting', n)
        n += 1

threading.Thread(target=countdown, args=(10,)).start()
threading.Thread(target=countup, args=(5,)).start()
```

eventåŕźèśąçŽĎäÿĂäÿłéĞ■èęĄçĽźçĆźæŸŕå¡ŞåőČèćńèő¿ç¡őäÿžçIJ§æŮűäijŽåŤď éĘŠæĽĂæIJĽç■Ľå¿Ě
Condition åŕźèśąæİěæŻ£äżčãĂĆèĂČèŹŚäÿĂäÿŃè£Źæőţä¡£çŤĺä£ąåŔůéĞŔåőđçŐřçŽĎäżčçăĄïijŽ

```
# Worker thread
def worker(n, sema):
    # Wait to be signaled
    sema.acquire()

    # Do some work
    print('Working', n)

# Create some threads
sema = threading.Semaphore(0)
nworkers = 10
for n in range(nworkers):
    t = threading.Thread(target=worker, args=(n, sema,))
    t.start()
```

è£ŘèąŇäÿŁè¿źçŽĎäžčçăĄåřĘäijŽåŘŕåŁíäÿĂäÿłçž£çÍŃæśäïijŇä¡ĘæŸŕåźűæšąæIJL'äżĂäźĹäžŃæČĚåŔŚ

```
>>> sema.release()
Working 0
>>> sema.release()
Working 1
>>>
```

çijŰåĘŹæűL'åŔŁåĹråd'ğéĞŔçŽĎçž£çÍŃéŮt'åŘŇæ■ěéŮőécŸçŽĎäžčçăĄäijŽèől'ä¡ăçŮŻäÿ■æňšçŤ§ãĂĆ

14.3 12.3 çž£çÍŃéŮť'éĂŽä£ą

éŮőécŸ

ä¡ăçŽĎçÍŃåžŔäÿ■æIJL'åd'Žäÿłçž£çÍŃïijŇä¡ăéIJĂèęĄåIJÍè£ŹäžŻçž£çÍŃäźŃéŮt'åőL'åĚÍåIJřäžd'æ■ćä£ąa

èğčåĘşæŰźæąĹ

äżŐäÿĂäÿłçž£çÍŃåŘŚåŔęäÿĂäÿłçž£çÍŃåŘŚéĂĄæŢřæ■őæIJĂåőL'åĚÍçŽĎæŰźåijŔåŔŕèČ¡åřśæŸŕä¡£çŤÍ
queue åžŞäÿ■çŽĎéŸ§åĹŮäžEãĂĆåĹŻåžžäÿĂäÿłèćńåd'Žäÿłçž£çÍŃåĚšäžńçŽĎ
Queue åŕźèšąïijŇè£ŹäžŻçž£çÍŃéĂŽè£Ğä¡£çŤÍ put() åŠŇ get()
æŞ■ä¡IJæÌěåŘŚéŸ§åĹŮäÿ■æůżåŁăæĹŰèĂĚåĹăéŹd'åĚČçt'ăãĂĆ ä¿ŃåęĆïijŽ

```
from queue import Queue
from threading import Thread

# A thread that produces data
def producer(out_q):
    while True:
        # Produce some data
        ...
        out_q.put(data)
```

```
# A thread that consumes data
def consumer(in_q):
    while True:
# Get some data
        data = in_q.get()
        # Process the data
        ...

# Create the shared queue and launch both threads
q = Queue()
t1 = Thread(target=consumer, args=(q,))
t2 = Thread(target=producer, args=(q,))
t1.start()
t2.start()
```

Queue åŕźèśąåůšçżŔåŇĚåŘńäžĘå£ĚèęĄçŽĎéŤĄïijŇæĽ Ăäžëä¡ăåŔŕäžěéĂŽè£ĞåőČåIJĺåď Žäÿłçž£çíŃé
å¡Şä¡£çŤĺéŸ§åĹŮæŮűïijŇå■ŔèřČçŤ§äžğèĂĚåŠŇæűĹèť źèĂĚçŽĎåĚşéŮ■éŮőécŸåŔŕèČ¡äijŽæIJL äÿĂäžŹé

```
from queue import Queue
from threading import Thread

# Object that signals shutdown
_sentinel = object()

# A thread that produces data
def producer(out_q):
    while running:
        # Produce some data
        ...
        out_q.put(data)

    # Put the sentinel on the queue to indicate completion
    out_q.put(_sentinel)

# A thread that consumes data
def consumer(in_q):
    while True:
        # Get some data
        data = in_q.get()

        # Check for termination
        if data is _sentinel:
            in_q.put(_sentinel)
            break

        # Process the data
        ...
```

æIJñä¿Ńäÿ■æIJL äÿĂäÿłçL źæőŁçŽĎåIJřæŰźïijŽæűĹèť źèĂĚåIJĺèŕžåĹřè£ŹäÿłçL źæőŁåĂijäźŃåŘŐçńŃ

åř¡çőąéŸ§åĹŮæŸŕæIJĂåÿÿèğĄçŽĎçž£çĺŃéŮťéĂŽä£ąæIJžåĹűïijŇä¡ĘæŸŕäż■çĎűåŔŕäżěèĞłåůśéĂŽè£ĞåĹŽ
Condition åŔŸéĞŔæİěåŇĚèčËä¡ăçŽĎæŢřæ■őçżŞæđĎãĂĆäÿŃè¿źè£Źäÿłä¿Ňå■ŘæijŤçď žäžĘåęĆä¡ŢåĹŻ

```
import heapq
import threading

class PriorityQueue:
    def __init__(self):
        self._queue = []
        self._count = 0
        self._cv = threading.Condition()
    def put(self, item, priority):
        with self._cv:
            heapq.heappush(self._queue, (-priority, self._count,␣
↪item))
            self._count += 1
            self._cv.notify()

    def get(self):
        with self._cv:
            while len(self._queue) == 0:
                self._cv.wait()
            return heapq.heappop(self._queue)[-1]
```

ä¡£çŤĺéŸ§åĹŮæİèè£ŻèąŇçž£çĺŃéŮťéĂŽä£ąæŸŕäÿĂäÿłå■ŢåŘŚãĂĄäÿ■çąőåőŽçŽĎè£ĞçĺŃãĂĆéĂŽåÿ
task_done() åŠŇ join() ïijŽ

```
from queue import Queue
from threading import Thread

# A thread that produces data
def producer(out_q):
    while running:
        # Produce some data
        ...
        out_q.put(data)

# A thread that consumes data
def consumer(in_q):
    while True:
        # Get some data
        data = in_q.get()

        # Process the data
        ...
        # Indicate completion
        in_q.task_done()

# Create the shared queue and launch both threads
q = Queue()
t1 = Thread(target=consumer, args=(q,))
```

```
t2 = Thread(target=producer, args=(q,))
t1.start()
t2.start()

# Wait for all produced items to be consumed
q.join()
```

åęĆædIJäÿĂäÿłçž£çÍŃéIJĂèęĄåIJÍäÿĂäÿłâĂIJæűĹèť źèĂĚâĂİçž£çÍŃåď ĎçŘĘåőŇçĽźåőŽçŽĎæŢřæ■ő

Event æŤ¿åĹřäÿĂèţůä¡£çŤĺïijŇè£ŹæăůâĂIJçŤ§ăžğèĂĚâĂİåřśåŔřäżěéĂŽè£Ğè£ŹäÿłEventåŕźèśąæİěçŻŚæţ

```
from queue import Queue
from threading import Thread, Event

# A thread that produces data
def producer(out_q):
    while running:
        # Produce some data
        ...
        # Make an (data, event) pair and hand it to the consumer
        evt = Event()
        out_q.put((data, evt))
        ...
        # Wait for the consumer to process the item
        evt.wait()

# A thread that consumes data
def consumer(in_q):
    while True:
        # Get some data
        data, evt = in_q.get()
        # Process the data
        ...
        # Indicate completion
        evt.set()
```

èőÍèőž

å§žäžŐçőĂå■ŢéŸ§åĹŮçijŰåĘŹåď Žçž£çÍŃçÍŃåžŘåIJÍåď ŽæŢřæČĚåĘţäÿŃæŸŕäÿĂäÿłæŕŤè¿ČæŸŐæŹ

ä¡£çŤÍçž£çÍŃéŸ§åĹŮæIJL äÿĂäÿłèęĄæşÍæĎŔçŽĎéŮőécŸæŸŕïijŇåŘŚéŸ§åĹŮäÿ■æůžåŁăæŢřæ■őéąźæŮűå

```
from queue import Queue
from threading import Thread
import copy

# A thread that produces data
def producer(out_q):
    while True:
        # Produce some data
        ...
```

```
        out_q.put(copy.deepcopy(data))

# A thread that consumes data
def consumer(in_q):
    while True:
        # Get some data
        data = in_q.get()
        # Process the data
        ...
```

Queue åŕźèśąæŔŘä¿ŻäÿĂäžŻåIJĺå¡ŞåL’■äÿŁäÿŃæŰĞå¿ĹæIJL’çŤĺçŽĎéŹĎåŁăçL’źæĂğãĂĆæŕŤåęĆåIJ
Queue åŕźèśąæŮűæŔŘä¿ŻåŔŕéĂL’çŽĎ size åŔĆæŢřæİěéŹŘåĹűåŔŕäžěæůżåŁäåĹřéŸ§åĹŮäÿ■çŽĎåĚČçť’ă
âĂIJæűĹèť’źâĂİçŽĎéĂ§åžęå£ńïijŇéĆčäźĹä¡£çŤĺåŽžåőŽåď’ğåřŔçŽĎéŸ§åĹŮåřśåŔŕäžěåIJĺéŸ§åĹŮåůšæżąçŽ
get() åŠŇ put() æŰźæşŢéČ¡æŤŕæŇĄéÍđéŸżàąđæŰźåijŔåŠŇèő¿åőŽèűĚæŮűïijŇä¿ŃåęĆïijŽ

```
import queue
q = queue.Queue()

try:
    data = q.get(block=False)
except queue.Empty:
    ...

try:
    q.put(item, block=False)
except queue.Full:
    ...

try:
    data = q.get(timeout=5.0)
except queue.Empty:
    ...
```

è£ŹäžŻæŞ■ä¡IJéČ¡åŔŕäžěçŤĺæİěéĄ£åĚ■å¡ŞæĽ’ğèąŇæ§ŘäžŻçĽ’źåőŽéŸ§åĹŮæŞ■ä¡IJæŮűåŔŚçŤ§æŮăé
put() æŰźæşŢåŠŇäÿĂäÿłåŻžåőŽåď’ğåřŔçŽĎéŸ§åĹŮäÿĂèţůä¡£çŤĺïijŇè£ŹæăůåŞéŸ§åĹŮåůšæżąæŮűåřśà

```
def producer(q):
    ...
    try:
        q.put(item, block=False)
    except queue.Full:
        log.warning('queued item %r discarded!', item)
```

åęĆæđIJä¡ăèŕŢåŻ¿èőľæűĹèť’źèĂĚçž£çĺŃåIJĺæĽ’ğèąŇåČŔ q.get()
è£ŹæăůçŽĎæŞ■ä¡IJæŮűïijŇèűĚæŮűèĞłåŁĺçžĹæ■ćäžěä¿£æčĂæ§ěçžĹæ■ćæăĞå£ŮïijŇä¡ăåžŤèŕěä¡£çŤĺ
q.get() çŽĎåŔŕéĂL’åŔĆæŢř timeout ïijŇåęĆäÿŃïijŽ

```
_running = True

def consumer(q):
```

```
    while _running:
        try:
            item = q.get(timeout=5.0)
            # Process item
            ...
        except queue.Empty:
            pass
```

æIJĂåŘŐïijŇæIJĽ `q.qsize()` ïijŇ `q.full()` ïijŇ `q.empty()` ç■Ľåőđç ŤÍæŰźæşŢåŔŕäžěèŐůåŔŰäÿĂäÿłéŸ§åĹŮçŽĎå¡ŞåĽ■åđ'ğářŔåŠŇçŁűæĂĄãĆä¡ĘèęĄæşÍæĎŔïijŇè `empty()` åĹđ'æŰ■åĞžè£ŹäÿłéŸ§åĹŮäÿžçľžïijŇä¡ĘåŘŇæŮűåŔęåđ'ŰäÿĂäÿłçž£çÍŃåŔŕèČ¡åůšçżŘåŘŚè£Źä

14.4 12.4 çżŹåĚşéŤőéČÍåĹĘåŁäéŤĄ

éŮőécŸ

ä¡ăéIJĂèęĄåŕźåđ'Žçž£çÍŃçÍŃåžŔäÿ■çŽĎäÿt'çŢŇåŇžåŁäéŤĄäżěéĄ£åĚ■çńđäžĽ'æİąäžűãĂĆ

èğčåĘşæŰźæąĹ

èęĄåIJÍåđ'Žçž£çÍŃçÍŃåžŔäÿ■åőĽ'åĚÍä¡£çŤÍåŔŕåŔŸåŕźèśąïijŇä¡ăéIJĂèęĄä¡£çŤÍ threading åžŞäÿ■çŽĎ `Lock` åŕźèśąïijŇåřśåČŔäÿŃè¿źè£ŹäÿłäżŃå■Řè£ŹæăůïijŽ

```
import threading

class SharedCounter:
    '''
    A counter object that can be shared by multiple threads.
    '''
    def __init__(self, initial_value = 0):
        self._value = initial_value
        self._value_lock = threading.Lock()

    def incr(self,delta=1):
        '''
        Increment the counter with locking
        '''
        with self._value_lock:
             self._value += delta

    def decr(self,delta=1):
        '''
        Decrement the counter with locking
        '''
        with self._value_lock:
             self._value -= delta
```

Lock åŕźèśąåŠŇ with èŕ■åŔěåİŮäÿĂèţůä¡£çŤĺåŔŕäžěä£İèŕĄäžŠæŰěæL’ğèąŇïijŇåřśæŸŕæŕŔæňąåŔłæI
with èŕ■åŔěåŇĚåŘńçŽĎäžčçăĄåİŮãĂĆwith èŕ■åŔěäijŽåIJĺè£Źäÿłäžčçă Ą åİŮæL’ğèąŇåL’■èĞłåŁĺèŐůåŔŰéŤ

èőĺèőž

çž£çĺŃèřČåžęæIJňèť’ĺäÿŁæŸŕäÿ■çąőåőŽçŽĎïijŇåŻäæ■d’ïijŇåIJĺåd’Žçž£çĺŃçĺŃåžŔäÿ■éŤŹèŕŕåIJŕä¡£çŤ
åIJĺäÿĂäžŻâĂIJèĂĄçŽĎâĂİ Python äžčçăĄäÿ■ïijŇæŸ¿åijŔèŐůåŔŰåŠŇéĞŁæŤ¿éŤĄæŸŕå¿Ĺåÿÿèğ Ą çŽĎãĂ

```
import threading

class SharedCounter:
    '''
    A counter object that can be shared by multiple threads.
    '''
    def __init__(self, initial_value = 0):
        self._value = initial_value
        self._value_lock = threading.Lock()

    def incr(self,delta=1):
        '''
        Increment the counter with locking
        '''
        self._value_lock.acquire()
        self._value += delta
        self._value_lock.release()

    def decr(self,delta=1):
        '''
        Decrement the counter with locking
        '''
        self._value_lock.acquire()
        self._value -= delta
        self._value_lock.release()
```

çŻÿæŕŤäžŐè£Źçğ■æŸ¿åijŔèřČçŤĺçŽĎæŰźæşŢïijŇwith èŕ■åŔěæŻť’åŁääijŸéŻĚïijŇäź§æŻť’äÿ■åőźæŸŞ
release() æŰźæşŢæĹŰèĂĚçĺŃåžŔåIJĺèŐůå¿ŮéŤĄäźŃåŘŐäžğçŤ§åijĆåÿÿè£Źäÿd’çğ■æČĚåĘţïijĹä¡£çŤĺ
with èŕ■åŔěåŔŕäžěä£İèŕĄåIJĺè£Źäÿd’çğ■æČĚåĘţäÿŃäż■èČ¡æ■čçąőéĞŁæŤ¿éŤĄïijL’ãĂĆ
äÿžäžĘéĄ£åĚ■åĞžçŐřæ■żéŤĄçŽĎæČĚåĘţïijŇä¡£çŤĺéŤĄæIJžåĹűçŽĎçĺŃåžŔåžŤèŕěèő¿åőŽäÿžæŕŔäÿłçž£çĺ
åIJĺ threading åžŞäÿ■è£ŸæŔŔä¿ŻäžĘåĚűäžŰçŽĎåŘŇæ■ěåŐ§èŕ■ïijŇæŕŤåęĆ RLock
åŠŇ Semaphore åŕźèśąãĂĆä¡ĘæŸŕæäźæ■őäžěå¿ĂçżŔéłŇïijŇè£ŹäžŻåŐ§èŕ■æŸŕçŤĺäžŐäÿĂäžŻçL’źæőŁçŽ
RLock ïijĹåŔŕéĞ■åĚěéŤĄïijL’åŔŕäžěèćńåŘŇäÿĂäÿłçž£çĺŃåd’Žæňąè ŐůåŔŰïijŇäÿżèęĄçŤĺæÌěåőđçŐřå§žäžĆ
SharedCounter çśżïijŽ

```
import threading

class SharedCounter:
    '''
    A counter object that can be shared by multiple threads.
    '''
    _lock = threading.RLock()
```

```
    def __init__(self, initial_value = 0):
        self._value = initial_value

    def incr(self,delta=1):
        '''
        Increment the counter with locking
        '''
        with SharedCounter._lock:
            self._value += delta

    def decr(self,delta=1):
        '''
        Decrement the counter with locking
        '''
        with SharedCounter._lock:
             self.incr(-delta)
```

åIJÍäÿŁè¿źè£Źäÿłä¿Ńå■Řäÿ■ïijŇæšąæIJL’åŕźæŕŔäÿĂäÿłåőđä¿Ńäÿ■çŽĎåŔŕåŔŸåŕźèśąåŁăéŤĄïijŇåŔŰè
decr æŰźæşŢãĂĆ è£Źçğ■åőđçŐřæŰźåijŔçŽĎäÿĂäÿłçL’źçĆźæŸŕïijŇæŮăèőžè£ŹäÿłçśżæIJL’åď’ŽåřŚäÿłåőđä¿
ä£ąåŔůéĞŔåŕźèśąæŸŕäÿĂäÿłåżžçńŃåIJÍåĚśäžńèőąæŢřåŹÍå§žçąĂäÿŁçŽĎåŘŇæ■ěåŐ§èŕ■ãĂĆåęĆæđIJèőąæŢ
èŕ■åŔěåřĘèőąæŢřåŹÍåĞŔ1ïijŇçž£çÍŃèćńåĚĄèőÿæL’ğèąŇãĂĆwith
èŕ■åŔěæL’ğèąŇçżŞæİ§åŘŐïijŇèőąæŢřåŹÍåŁăïijŚãĂĆåęĆæđIJèőąæŢřåŹÍäÿž0ïijŇçž£çÍŃåřĘèćńéŸżåąđïijŇçŽ

```
from threading import Semaphore
import urllib.request

# At most, five threads allowed to run at once
_fetch_url_sema = Semaphore(5)

def fetch_url(url):
    with _fetch_url_sema:
        return urllib.request.urlopen(url)
```

åęĆæđIJä¡ăåŕźçž£çÍŃåŘŇæ■ěåŐ§èŕ■çŽĎåžŢåśĆçŘĘèőžåŠŇåőđçŐřæĎ§åĚť’èűčïijŇåŔŕäżěåŔĆèĂČæŞ■

14.5 12.5 éŸšæ■ćæ■żéŤĄçŽĎåŁăéŤĄæIJžåĹű

éŮőéćŸ

ä¡ăæ■čåIJÍåĘŹäÿĂäÿłåď’Žçž£çÍŃçÍŃåžŔïijŇåĚűäÿ■çž£çÍŃéIJĂèęĄäÿĂæňąèŐůåŔŰåď’ŽäÿłéŤĄïijŇæ■d

èğčåĘşæŰźæąĹ

åIJÍåď’Žçž£çÍŃçÍŃåžŔäÿ■ïijŇæ■żéŤĄéŮőéćŸå¿Ĺåď’ğäÿĂéČÍåĹĘæŸŕçŤśäžŐçž£çÍŃåŘŇæŮűèŐůåŔŰå
æŮűåĂŹåŔŚçŤ§éŸżåąđïijŇéĆčäźĹè£Źäÿłçž£çÍŃåřśåŔŕèČ¡éŸżåąđåĚűäżŰçž£çÍŃçŽĎæL’ğèąŇïijŇäżŐèĂŇå
èğčåĘşæ■żéŤĄéŮőéćŸçŽĎäÿĂçğ■æŰźæąĹæŸŕäÿžçÍŃåžŔäÿ■çŽĎæŕŔäÿĂäÿłéŤĄåĹĘéĚ■äÿĂäÿłåŤŕäÿĂçŽ
æŸŕéİđåÿÿåőźæŸŞåőđçŐřçŽĎïijŇçď’žä¿ŃåęĆäÿŃïijŽ

```
import threading
from contextlib import contextmanager

# Thread-local state to stored information on locks already acquired
_local = threading.local()

@contextmanager
def acquire(*locks):
    # Sort locks by object identifier
    locks = sorted(locks, key=lambda x: id(x))

    # Make sure lock order of previously acquired locks is not␣
↪violated
    acquired = getattr(_local,'acquired',[])
    if acquired and max(id(lock) for lock in acquired) >=␣
↪id(locks[0]):
        raise RuntimeError('Lock Order Violation')

    # Acquire all of the locks
    acquired.extend(locks)
    _local.acquired = acquired

    try:
        for lock in locks:
            lock.acquire()
        yield
    finally:
        # Release locks in reverse order of acquisition
        for lock in reversed(locks):
            lock.release()
        del acquired[-len(locks):]
```

åęĆä¡Ţä¡£çŤĺè£Źäÿłäÿ£äÿŃæŰĞçőąçŘĘåŹĺåŚćïij§ä¡ääŔŕäžěæŇL’çĚğæ■čåÿÿéĂŤå¿ĎåĹŻåžžäÿĂäÿłéŤ
`acquire()` åĞ¡æŢřæİěçŤşèŕůéŤĄïijŇ çd’žä¿ŃåęĆäÿŃïijŽ

```
import threading
x_lock = threading.Lock()
y_lock = threading.Lock()

def thread_1():
    while True:
        with acquire(x_lock, y_lock):
            print('Thread-1')

def thread_2():
    while True:
        with acquire(y_lock, x_lock):
            print('Thread-2')

t1 = threading.Thread(target=thread_1)
```

```
t1.daemon = True
t1.start()

t2 = threading.Thread(target=thread_2)
t2.daemon = True
t2.start()
```

åęĆædIJä¡ăæL'ğèąŇè£Źæőţäžčçă Ąïij Ňä¡ăäijŽåŔŚçŐřåőČå■şä¡£åIJĺäÿ■åŘŇçŽĎåĞ¡æŢřäÿ■äžěäÿ■åŘŇç
åĚűåĚşéŤőåIJĺäžŐïijŇåIJĺçňňäÿĂæőţäžčçăĄäÿ■ïijŇæĹŚäžňářźè£ŹäžŽéŤĄè£ŻèąŇäžĘæŐŠåžŔãĆéĂŽè£Ğ
åęĆædIJæIJL'åď'Žäÿł `acquire()` æŞ■ä¡IJèćńåţŇåěŮèřČçŤĺïijŇåŔŕäžěéĂŽè£ĞçžçĺŃæIJňåIJřå■ŸåĆĺïijĹT
åĄĞèő¿ä¡ăçŽĎäžčçăĄæŸŕè£ŹæăůåĘŹçŽĎïijŽ

```
import threading
x_lock = threading.Lock()
y_lock = threading.Lock()

def thread_1():

    while True:
        with acquire(x_lock):
            with acquire(y_lock):
                print('Thread-1')

def thread_2():
    while True:
        with acquire(y_lock):
            with acquire(x_lock):
                print('Thread-2')

t1 = threading.Thread(target=thread_1)
t1.daemon = True
t1.start()

t2 = threading.Thread(target=thread_2)
t2.daemon = True
t2.start()
```

åęĆædIJä¡ăè£ŘèąŇè£ŹäÿłçL'ĹæIJňçŽĎäžčçăĄïijŇå£ĚåőŽäijŽæIJL'äÿĂäÿłçž£çĺŃåŔŚçŢ§åť'ľæžČïijŇåi

```
Exception in thread Thread-1:
Traceback (most recent call last):
  File "/usr/local/lib/python3.3/threading.py", line 639, in _
↪bootstrap_inner
    self.run()
  File "/usr/local/lib/python3.3/threading.py", line 596, in run
    self._target(*self._args, **self._kwargs)
  File "deadlock.py", line 49, in thread_1
    with acquire(y_lock):
  File "/usr/local/lib/python3.3/contextlib.py", line 48, in __
↪enter__
```

```
    return next(self.gen)
  File "deadlock.py", line 15, in acquire
    raise RuntimeError("Lock Order Violation")
RuntimeError: Lock Order Violation
>>>
```

åŔŠçŤ§åťľæžČçŽĎåŐ§åŻäåIJĺäžŐïijŇæŕŔäÿłçž£çĺŃéČ¡èőřå¡ŢçİĂèĞłåůšåůšçżŔèŐůåŔŰåĹřçŽĎéŤĄã
acquire()　　åĞ¡æŢřäijŽæčĂæ§ěäźŃåĽ■åůšçżŔèŐůåŔŰçŽĎéŤĄåĹÙèąĺïijŇ
çŤšäžŐéŤĄæŸŕæŇĽçĚğå■ĞåžŔæŐŠåĹÙèŐůåŔŰçŽĎïijŇæĽĂäžěåĞ¡æŢřäijŽèőď äÿžäźŃåĽ■åůšèŐůåŔŰç

èőĺèőž

æ■żéŤĄæŸŕæŕŔäÿĂäÿłåď Žçž£çĺŃçĺŃåžŔéČ¡äijŽéİcäÿť çŽĎäÿĂäÿłéŮőécŸïijĹåřśåČŔåőČæŸŕæŕŔäÿĂ
çž£çĺŃåŔłèČ¡åŘŇæŮűä£ĺæŇĄäÿĂäÿłéŤĄïijŇè£ŹæăůçĺŃåžŔåřśäÿ■äijŽècńæ■żéŤĄéŮőécŸæĽĂåŻřæĽřãĂĆ

æ■żéŤĄçŽĎæčĂæţŇäÿŐæĄćåď■æŸŕäÿĂäÿłåĞăäźŐæšąæIJĽäijŸéŽĚçŽĎèğčåĘşæŰźæąĹçŽĎæĽľåśŢ
è£ŘèąŇçŽĎæŮűåĂŹäijŽæŕŔéŽŤäÿĂæőţæŮűéŮťéĞ■ç¡őèőąæŢřåŹĺïijŇåIJĺæšąæIJĽåŘŠçŤ§æ■żéŤĄçŽĎæČ
èűĚæŮűïijŇè£ŹæŮűçĺŃåžŔäijŽéĂŽè£ĞéĞ■åŘŕèĞłèžńæĄćåď■åĹřæ■čåÿÿçŁűæĂĄãĂĆ

éĄ£åĚ■æ■żéŤĄæŸŕåŖęåď ŰäÿĂçğ■èğčåĘşæ■żéŤĄéŮőécŸçŽĎæŰźåijŔïijŇåIJĺè£ŽçĺŃèŐůåŔŰéŤĄçŽ
æ■żéŤĄçŁűæĂĄãĂĆèŕĄæŸŐářśçŢŹçżŹèŕżèĂĚä¡IJäÿžçżČäźăäžĘãĂĆéĄ£åĚ■æ■żéŤĄçŽĎäÿżèęĄæĂİæČşa
æ■żéŤĄçŽĎäÿĂäÿłå£ĚèęĄæİąäžűïijŇäžŐèĂŇéĄ£åĚ■çĺŃåžŔè£ŻåĚěæ■żéŤĄçŁűæĂĄãĂĆ

äÿŃéİcäżěäÿĂäÿłåĚşäžŐçž£çĺŃæ■żéŤĄçŽĎçżŔåĚÿéŮőécŸïijŽâĂIJåŞšå■ęåőűåřśéď ŘéŮőécŸâĂİïijŇä
éİcåĽ■æIJĽäÿĂçćŮéě■åŠŇäÿĂåŔłç■ůå■ŘãĂĆåIJĺè£ŹéĞŇæŕŔäÿłåŞšå■ęåőűåŔŕäžěçIJŃåĄŽæŸŕäÿĂäÿłçŃı
æĂİèĂČãĂĄåŘČéě■äÿĽçğ■çŁűæĂĄäÿ■çŽĎäÿĂäÿłãĂĆéIJĂèęĄæşĺæĎŔçŽĎæŸŕïijŇæŕŔäÿłåŞšå■ęåőűåŘČ
éĆčäźĹäžŮäžňäžŤäÿłéČ¡åŘłèČ¡æŃ£çİĂäÿĂåŘłç■ůå■ŘåİŘåIJĺéĆčåĎ£ïijŇçŽť åĹřéě£æ■żãĂĆæ■ď æŮűäžŮä
äÿŃéİcæŸŕäÿĂäÿłçőĂå■ŢçŽĎä¡£çŤĺæ■żéŤĄéĄ£åĚ■æIJžåĹůèğčåĘşâĂIJåŞšå■ęåőűåřśéď ŘéŮőécŸâĂİçŽĎå

```
import threading

# The philosopher thread
def philosopher(left, right):
    while True:
        with acquire(left,right):
             print(threading.currentThread(), 'eating')

# The chopsticks (represented by locks)
NSTICKS = 5
chopsticks = [threading.Lock() for n in range(NSTICKS)]

# Create all of the philosophers
for n in range(NSTICKS):
    t = threading.Thread(target=philosopher,
                         args=(chopsticks[n],chopsticks[(n+1) %␣
↪NSTICKS]))
    t.start()
```

æIJĂåŘŐïijŇèęĄçĽźåĹńæşĺæĎŔåĹřïijŇäÿžäźĘéĄ£åĚ■æ■żéŤĄïijŇæĽĂæIJĽçŽĎåŁäéŤĄæŞ■ä¡IJå£Ěé
acquire()　　åĞ¡æŢřäĂĆåęĆæđIJäżčçăĄäÿ■çŽĎæ§ŘéČĺåĹĘçżŢè£Ğacquire

åĞ¡æŢřçŻťæŐěçŤşèřůéŤĄïijŇéĆčäźĹæŢťäÿłæ■żéŤĄéĄ£åĚ■æIJžåĹűářšäÿ■ètůä¡IJçŤĺäžEãĂĆ

14.6 12.6 ä£ìå■Ÿçž£çÍÑçŽĎçŁűæĂĄä£qæĄŕ

éŮőécŸ

ä¡ăéIJĂèęĄä£Ìå■Ÿæ■čåIJÍè£ŘèąÑçž£çÍÑçŽĎçŁűæĂĄïijÑè£ŹäÿłçŁűæĂĄåŕźäžŐåĚűäżŰçŽĎçž£çÍÑæŸ

èğčåĘşæŰźæąĹ

æIJL'æŮűåIJÍåď'Žçž£çÍÑçijŰçÍÑäÿ■ïijÑä¡ăéIJĂèęĄåŘłä£Ìå■Ÿå¡ŞåL'■è£ŘèąÑçž£çÍÑçŽĎçŁűæĂĄãĂĆ
èęĄè£ŹäźĹåĄŽïijÑåŘŕä¡£çŤÍ `thread.local()` åĹŻåžžäÿĂäÿłæIJňåIJřçž£çÍÑå■ŸåĆÍåŕźèśąãĂĆ
åŕźè£ŹäÿłåŕźèśąçŽĎåśđæĂğçŽĎä£Ìå■ŸåŠÑèŕźåŘŰæŞ■ä¡IJéČ¡åŘłäijŽåŕźæL'ğèąÑçž£çÍÑåŘŕèğĄïijÑèĂÑåĚ

ä¡IJäÿžä¡£çŤÍæIJňåIJřå■ŸåĆÍçŽĎäÿĂäÿłæIJL'èűčçŽĎåőđéŹĚä¿Ńå■ŘïijÑ
èĂČèŹŚåIJÍ8.3åřŘèŁĆåőŽäźL'è£ĞçŽĎ `LazyConnection` äÿŁäÿŃæŰĞçőąçŘĘåŹÍçśżãĂĆ
äÿŃéÍćæĹŚäžñåŕźåőČè£ŻèąÑäÿĂäžŻåřŘçŽĎä£őæŤźä¡£å¿ŮåőČåŘŕäžěéĂĆçŤĺäžŐåď'Žçž£çÍÑïijŽ

```
from socket import socket, AF_INET, SOCK_STREAM
import threading

class LazyConnection:
    def __init__(self, address, family=AF_INET, type=SOCK_STREAM):
        self.address = address
        self.family = AF_INET
        self.type = SOCK_STREAM
        self.local = threading.local()

    def __enter__(self):
        if hasattr(self.local, 'sock'):
            raise RuntimeError('Already connected')
        self.local.sock = socket(self.family, self.type)
        self.local.sock.connect(self.address)
        return self.local.sock

    def __exit__(self, exc_ty, exc_val, tb):
        self.local.sock.close()
        del self.local.sock
```

äžčçăĄäÿ■ïijÑèĞłåůśèğĆår§ářźäžŐ `self.local` åśđæĂğçŽĎä¡£çŤÍãĂĆ
åőČèćńåĹİåğÑåŇŰår¿äÿĂäÿł `threading.local()` åőđä¿ÑãĂĆ
åĚűäžŰæŰźæşŢæŞ■ä¡IJèćńå■ŸåĆÍäÿž `self.local.sock` çŽĎåěŮæŐěå■ŮåŕźèśąãĂĆ
æIJL'äžĘè£ŹäžŻåŕśåŘŕäžěåIJÍåď'Žçž£çÍÑäÿ■åőL'åĚÍçŽĎä¡£çŤÍ `LazyConnection`
åőđä¿ÑäžĘãĂĆä¿ÑåęĆïijŽ

```
from functools import partial
def test(conn):
    with conn as s:
```

```
        s.send(b'GET /index.html HTTP/1.0\r\n')
        s.send(b'Host: www.python.org\r\n')

        s.send(b'\r\n')
        resp = b''.join(iter(partial(s.recv, 8192), b''))

    print('Got {} bytes'.format(len(resp)))

if __name__ == '__main__':
    conn = LazyConnection(('www.python.org', 80))

    t1 = threading.Thread(target=test, args=(conn,))
    t2 = threading.Thread(target=test, args=(conn,))
    t1.start()
    t2.start()
    t1.join()
    t2.join()
```

åőČäźŃæĽĂäžěèąŇå¿ŮéĂŽçŽĎđŐ§åŽäæŸŕæŕŔäÿłçž£çÍŃäijŽåĹŽåžžäÿĂäÿłèĞłåůśäÿŞåśđçŽĎđěŮæŐ
åŻäæ■d'ïijŇå¡Şäÿ■åŘŇçŽĎçž£çÍŃæĽ'ğèąŇåěŮæŐěå■ŮæŞ■ä¡IJæŮűïijŇçŤśäžŐæŞ■ä¡IJçŽĎæŸŕäÿ■åŘŇçŽ

èőÍèőž

åIJÍåď'ğéČÍåĹEçÍŃåžŔäÿ■åĹŽåžžåŠŇæŞ■ä¡IJçž£çÍŃçL'źåőŽçŁűæĂĄåźűäÿ■äijŽæIJL'äžĂäźĹéŮőécŸãĹ
äÿ■è£ĞïijŇå¡ŞåĞžäžEéŮőécŸçŽĎæŮűåĂŹïijŇéĂŽåÿÿæŸŕåŻääÿžæ§Řäÿłåŕźèśąèćńåď'Žäÿłçž£çÍŃä¡£çŤÍåĹ
æŕŤåęĆäÿĂäÿłåěŮæŐěå■ŮæĹŰæŰĞäžűãĂĆä¡ääÿ■èČ¡èől'æL'ĂæIJL'çž£çÍŃèt'ąçŇőäÿĂäÿłå■ŢçŃňåŕźèśąïijĺ
åŻääÿžåď'Žäÿłçž£çÍŃåŘŇæŮűèŕžåŠŇåĘŹçŽĎæŮűåĂŹäijŽäžğçŤ§æůůäźśãĂĆ
æIJňåIJřçž£çÍŃå■ŸåĆÍéĂŽè£Ğèől'è£ŹäžŻètĎæžŘåŔłèČ¡åIJÍècńä¡£çŤÍçŽĎçž£çÍŃäÿ■åŔŕèğĄæİèèğčåĘşè£Ź

æIJňèŁĆäÿ■ïijŇä¡£çŤÍ `thread.local()` åŔŕäžěèől' `LazyConnection` çśżæŤŕæŇĄäÿĂäÿłçž£çÍŃäÿĂäÿłè£đæŐěïijŇ
èĂŇäÿ■æŸŕåŕźäžŐæL'ĂæIJL'çŽĎè£ŻçÍŃéČ¡åŔłæIJL'äÿĂäÿłè£đæŐěãĂĆ

åĚűåŐ§çŘEæŸŕïijŇæŕŔäÿł `threading.local()` åődä¿ŃäÿžæŕŔäÿłçž£çÍŃçżt'æŁd'çİĂäÿĂäÿłå■Ţç
æL'ĂæIJL'æŹőéĂŽåődä¿ŃæŞ■ä¡IJæŕŤåęĆèŐůåŔŰãĂĄä£őæŤźåŠŇåĹăéŹd'åĂijäžĚäžĚæŞ■ä¡IJè£Źäÿłå■Ůå
æŕŔäÿłçž£çÍŃä¡£çŤÍäÿĂäÿłçŃňçńŇçŽĎå■ŮåĚÿåŕśåŔŕäžěä£ÌèŕĄæŢřæ■őçŽĎéŽŤçęžäžĘãĂĆ

14.7 12.7 åĹŽåžžäÿĂäÿłçž£çÍŃæśă

éŮőécŸ

ä¡ăăĹŽåžžäÿĂäÿłåůěä¡IJèĂĚçž£çÍŃæśăïijŇçŤÍæİěçŻÿåžŤåőćæĹůçńŕèŕůæśĆæĹŰæL'ğèąŇåĚűäžŰçŽĎå

èğčåEşæŰźæąĹ

`concurrent.futures` åĞ¡æŢřåžŞæIJL'äÿĂäÿł `ThreadPoolExecutor` çśżåŔŕäžěèćńçŤÍæİěåőŇæĹŘè£ŹäÿłäžžåŁąãĂĆ äÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎTCPæIJ■åŁąåŹĺijŇä¡£çŤÍäžE

```
from socket import AF_INET, SOCK_STREAM, socket
from concurrent.futures import ThreadPoolExecutor

def echo_client(sock, client_addr):
    '''
    Handle a client connection
    '''
    print('Got connection from', client_addr)
    while True:
        msg = sock.recv(65536)
        if not msg:
            break
        sock.sendall(msg)
    print('Client closed connection')
    sock.close()

def echo_server(addr):
    pool = ThreadPoolExecutor(128)
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(addr)
    sock.listen(5)
    while True:
        client_sock, client_addr = sock.accept()
        pool.submit(echo_client, client_sock, client_addr)

echo_server(('',15000))
```

åęĆædIJä¡ăæČşæL’ŃåŁĺåĹŻåżžä¡ăèĞłåůśçŽĎçž£çĺŃæśäïijŇ
éĂŽåÿÿåŘŕäžěä¡£çŤĺäÿĂäÿłQueueæİěè¡żæİ¿åőđçŐřãĂĆäÿŃéİćæŸŕäÿĂäÿłçĺ■å¿őäÿ■åŘŇä¡ĘæŸŕæL’ŃåŁĺåő

```
from socket import socket, AF_INET, SOCK_STREAM
from threading import Thread
from queue import Queue

def echo_client(q):
    '''
    Handle a client connection
    '''
    sock, client_addr = q.get()
    print('Got connection from', client_addr)
    while True:
        msg = sock.recv(65536)
        if not msg:
            break
        sock.sendall(msg)
    print('Client closed connection')

    sock.close()

def echo_server(addr, nworkers):
```

```
    # Launch the client workers
    q = Queue()
    for n in range(nworkers):
        t = Thread(target=echo_client, args=(q,))
        t.daemon = True
        t.start()

    # Run the server
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(addr)
    sock.listen(5)
    while True:
        client_sock, client_addr = sock.accept()
        q.put((client_sock, client_addr))

echo_server(('',15000), 128)
```

ä¡£çŤÍ ThreadPoolExecutor çŻÿåŕźäžŐæĽ'ŃåŁÍåőđçŐřçŽĎäÿĂäÿłåě¡åď'ĎåIJÍäžŐåőČä¡£å¿Ů äżżåŁąæŔŘäžđ'èĂĚæŻt'æŰźä¿£çŽĎäżŐèćńèřČçŤÍåĞ¡æŢřäÿ■èŐůåŔŰè£ŤåŻđåĂijãĂĆä¿ŃåęĆïijŇä¡ăåŔŕèČ

```
from concurrent.futures import ThreadPoolExecutor
import urllib.request

def fetch_url(url):
    u = urllib.request.urlopen(url)
    data = u.read()
    return data

pool = ThreadPoolExecutor(10)
# Submit work to the pool
a = pool.submit(fetch_url, 'http://www.python.org')
b = pool.submit(fetch_url, 'http://www.pypy.org')

# Get the results back
x = a.result()
y = b.result()
```

ä¿Ńå■Řäÿ■è£ŤåŻđçŽĎhandleåŕźèśąäijŽåÿőä¡ăåď'ĎçŘĘæĽ'ĂæIJĽçŽĎéŸżåąđäÿŐå■Řä¡IJïijŇçĎűåŘŐä çĽźåĹńçŽĎïijŇa.result() æŞ■ä¡IJäijŽéŸżåąđè£ŻçÍŃçŻt'åĹřåŕźåžŤçŽĎåĞ¡æŢřæĽ'ğèąŇåőŇæĹŘåźűè£Ť

èőÍèőž

éĂŽåÿÿæÌěèőšïijŇä¡ăåžŤèŕěéĄ£åĚ■çijŰåĘŹçž£çÍŃæŢřéĞŘåŔŕäžěæŮäéŹŘåĹűåćđéŢ£çŽĎçÍŃåžŘãĂĆ

```
from threading import Thread
from socket import socket, AF_INET, SOCK_STREAM

def echo_client(sock, client_addr):
    '''
```

```
    Handle a client connection
    '''
    print('Got connection from', client_addr)
    while True:
        msg = sock.recv(65536)
        if not msg:
            break
        sock.sendall(msg)
    print('Client closed connection')
    sock.close()

def echo_server(addr, nworkers):
    # Run the server
    sock = socket(AF_INET, SOCK_STREAM)
    sock.bind(addr)
    sock.listen(5)
    while True:
        client_sock, client_addr = sock.accept()
        t = Thread(target=echo_client, args=(client_sock, client_
↪addr))
        t.daemon = True
        t.start()

echo_server(('',15000))
```

åř¡çőąè£ŹäÿłäźšåŘŕäžěåůěä¡IJïijŇ ä¡ĘæŸŕåőČäÿ■èČ¡æŁţå¿ąæIJL’äžžèŕŢåŻ¿éĂŽè£ĞåĹŻåžžåď ğéĞŔçž
éĂŽè£Ğä¡£çŤĺécĎåĚĹåĹİåğŇåŇŰçŽĎçž£çĺŃæśäïijŇä¡ååŘŕäžěèő¿ç¡őåŘŇæŮűè£ŘèąŇçž£çĺŃçŽĎäÿŁéŹŘa

ä¡ăåŘŕèČ¡äijŽåĚşå£ČåĹŻåžžåď ğéĞŔçž£çĺŃäijŽæIJL’äžĂäźĹåŘŐæđIJãĂĆ
çŐřäžčæŞ■ä¡IJçşżç§åŘŕäžěå¿Ĺè¡żæİ¿çŽĎåĹŻåžžåĞää■Čäÿłçž£çĺŃçŽĎçž£çĺŃæśãäĂĆ
çŤŽèĞşïijŇåŘŇæŮűåĞää■Čäÿłçž£çĺŃç■L’å¿Ěåůěä¡IJåźűäÿ■äijŽåŕźåĚűäžŰäžčçăĄäžğçŤ§æĂğèČ¡å¡śåŞ■ãĂ
å¡ŞçĎűäžĘïijŇåęĆæđIJæL’ĂæIJL’çž£çĺŃåŘŇæŮűèćńåŤd’éĘŠåźűçńŇå■şåIJĺCPUäÿŁæL’ğèąŇïijŇéĆčåřśäÿ■
éĂŽåÿÿïijŇä¡ääžŤèŕěåŘłåIJĺI/Oåď ĎçŘĘçŻÿåĚşäžčçăĄäÿ■ä¡£çŤĺçž£çĺŃæśãäĂĆ

åĹŻåžžåď ğçŽĎçž£çĺŃæśäçŽĎäÿĂäÿłåŘŕèČ¡éIJĂèęĄåĚşæşĺçŽĎéŮőécŸæŸŕåĘĚå■ŸçŽĎä¡£çŤĺãĂĆ
ä¿ŇåęĆïijŇåęĆæđIJä¡ääIJĺOS Xçşżç§äÿŁéİćåĹŻåžž2000äÿłçž£çĺŃïijŇçşżç§æŸ¿çd’žPythonè£ŻçĺŃä¡£çŤĺã
äÿ■è£Ğïij Ňè£ŹäÿłèőąçőŮéĂŽåÿÿæŸŕæIJL’èŕŕåůőçŽĎãĂĆå¡ŞåĹŻåžžäÿĂäÿłçž£çĺŃæŮűïijŇæŞ■ä¡IJçşżç§äij
æŤ¿ç¡őçž£çĺŃçŽĎæL’ğèąŇæĂĹïijĹéĂŽåÿÿæŸŕ8MBåď ğåřŘïijL’ãĂĆä¡ĘæŸŕè£ŹäÿłåĘĚå■ŸåŘłæIJL’äÿĂåřŔ
åŻăæ■d’ïijŇPythonè£ŻçĺŃä¡£çŤĺåĹřçŽĎçIJ§åőđåĘĚå■ŸåĚűåőđå¿ĹåřŔ
ïijĹæŕŤåęĆïijŇåŕźäžŐ2000äÿłçž£çĺŃæİèèőšïijŇåŘłä¡£çŤĺåĹřäžĘ70MBçŽĎçIJ§åőđåĘĚå■ŸïijŇèĂŇäÿ■æŸŕ
åęĆæđIJä¡ääŃĚå£ČèŹŻæŃ§åĘĚå■Ÿåď ğåřŘïijŇåŘŕäžěä¡£çŤĺ threading.
stack_size() åĞ¡æŢřæİěéŹ■ä¡ŐåőČãĂĆä¿ŃåęĆïijŽ

```
import threading
threading.stack_size(65536)
```

åęĆæđIJä¡ääŁääÿŁè£ŹæIąèŕ■åŘěåźűåĘ■æňąè£ŘèąŇåL’■éİćçŽĎåĹŻåžž2000äÿłçž£çĺŃèŕŢéłŇïijŇ
ä¡ääijŽåŔŠçŐřPythonè£ŻçĺŃåŘłä¡£çŤĺåĹřäžĘåď ğæęĆ210MBçŽĎèŹŽæŃ§åĘĚå■ŸïijŇèĂŇçIJ§åőđåĘĚå■Ÿ
æşĺæĎŘçž£çĺŃæĂĹåď ğåřŘå£ĚéąžèĞşåřŠäÿž32768å■ŮèŁĆïijŇéĂŽåÿÿæŸŕçşżç§åĘĚå■Ÿéąţåď ğåřŘïijĹ409

14.8 12.8 çőÄå■ŢçŽĎåźűèąŇçijŰçíŃ

éŮőécŸ

ä¡ăæIJL'äÿłçÍŃåžŔèęĄæL'ğèąŇCPUåŕĘéŻĘåđŃåůěä¡IJïijŇä¡ăæČşèől'äżŰåĹl'çŤĺåd'Žæäÿ CPUçŽĎäijŸå

èğčåĘşæŰźæąĹ

concurrent.futures åžŞæŔŘä¿ŻäžĘäÿĂäÿł ProcessPoolExecutor çśżïijŇ åŔŕèćńçŤĺæÌěåIJĺäÿĂäÿłå■ŢçŃňçŽĎPythonèğčéĞŁåŹĺäÿ■æL'ğèąŇèőąçőŮåŕĘéŻĘåđŃçŽĎäżżåŁąãĂĆ äÿ■è£ĞïijŇèęĄä¡£çŤĺåőČïijŇä¡ăéęŰåĚĹèęĄæIJL'äÿĂäžŻèőąçőŮåŕĘéŻĘåđŃçŽĎäżżåŁąãĂĆ æĹŚäžňéĂŽè£ĞäÿĂäÿłçőÄå■ŢèĂŇåőđéŹĚçŽĎä¿Ńå■ŘæÌěæijŤçd'žåőČãĂĆåĄĞåőŽä¡ăæIJL'äÿłApache webæIJ■åŁąåŹĺæŮěå£ŮçŻőå¡ŢçŽĎgzipåŐŃçijl'åŇĚïijŽ

```
logs/
   20120701.log.gz
   20120702.log.gz
   20120703.log.gz
   20120704.log.gz
   20120705.log.gz
   20120706.log.gz
   ...
```

è£ŻäÿĂæ■ěåĄĞèő¿æŕŔäÿłæŮěå£ŮæŰĞäžűåĘĚåőźçśżäijijäÿŃéÍćè£Źæăůïijž

```
124.115.6.12 - - [10/Jul/2012:00:18:50 -0500] "GET /robots.txt ..."␣
↪200 71
210.212.209.67 - - [10/Jul/2012:00:18:51 -0500] "GET /ply/ ..." 200␣
↪11875
210.212.209.67 - - [10/Jul/2012:00:18:51 -0500] "GET /favicon.ico ..
↪." 404 369
61.135.216.105 - - [10/Jul/2012:00:20:04 -0500] "GET /blog/atom.xml␣
↪..." 304 -
...
```

äÿŃéÍćæŸŕäÿĂäÿłèĎŽæIJňïijŇåIJÍè£ŹäžŻæŮěå£ŮæŰĞäžűäÿ■æ§ěæL'¿åĞžæL'ĂæIJL'èő£éŮőè£Ğrobot

```
# findrobots.py

import gzip
import io
import glob

def find_robots(filename):
    '''
    Find all of the hosts that access robots.txt in a single log␣
↪file
    '''
    robots = set()
```

```
    with gzip.open(filename) as f:
        for line in io.TextIOWrapper(f,encoding='ascii'):
            fields = line.split()
            if fields[6] == '/robots.txt':
                robots.add(fields[0])
    return robots

def find_all_robots(logdir):
    '''
    Find all hosts across and entire sequence of files
    '''
    files = glob.glob(logdir+'/*.log.gz')
    all_robots = set()
    for robots in map(find_robots, files):
        all_robots.update(robots)
    return all_robots

if __name__ == '__main__':
    robots = find_all_robots('logs')
    for ipaddr in robots:
        print(ipaddr)
```

åL’■éÍćçŽĎçÍŃåžŘä¡£çŤÍäžĘéĂŽåÿÿçŽĎmap-reduceéčŐæäijæİěçijŰåĘŹãĂĆ åĞ¡æŢř `find_robots()` åIJÍäÿĂäÿłæŰĞäżűåŘ■éŻĘåŘĹäÿŁåĄŽmapæŞ■ä¡IJïijŇåźűåřĘçżŞæđIJæśĞæĂżäÿžäÿĂääź§åřśæŸŕ `find_all_robots()` åĞ¡æŢřäÿ■çŽĎ `all_robots` éŻĘåŘĹãĂĆçŐřåIJÍïijŇåĄĞèő¿ä¡ăæČşèęĄä£őæŤźè£ŹäÿłçÍŃåžŘèőľåőČä¡£çŤÍåd’ŽæăÿCPUãĂĆå¿ĹçőĂå■ŢâĂŤâĂŤåŘłéIJĂèęĄåřĘmap()æŞ■ä¡IJæŻ£æ■ćäÿžäÿĂäÿł `concurrent.futures` åžŞäÿ■çŤ§æĹŘçŽĎçśżäijijæŞ■ä¡IJå■şåŘŕãĂĆäÿŃéİćæŸŕäÿĂäÿłçőĂå■Ţä£őæŤźçL’ĹæIJñïijŽ

```
# findrobots.py

import gzip
import io
import glob
from concurrent import futures

def find_robots(filename):
    '''
    Find all of the hosts that access robots.txt in a single log␣
 ↪file

    '''
    robots = set()
    with gzip.open(filename) as f:
        for line in io.TextIOWrapper(f,encoding='ascii'):
            fields = line.split()
            if fields[6] == '/robots.txt':
                robots.add(fields[0])
    return robots
```

```
def find_all_robots(logdir):
    '''
    Find all hosts across and entire sequence of files
    '''
    files = glob.glob(logdir+'/*.log.gz')
    all_robots = set()
    with futures.ProcessPoolExecutor() as pool:
        for robots in pool.map(find_robots, files):
            all_robots.update(robots)
    return all_robots

if __name__ == '__main__':
    robots = find_all_robots('logs')
    for ipaddr in robots:
        print(ipaddr)
```

éĂŽè£Ğè£Źäÿłä£őæŤźåŘŐïijŇè£ŘèąŇè£ŹäÿłèĎŽæIJňäžğçŤ§åŘŇæăůçŽĎçżŞæđIJïijŇä¡ĘæŸŕåIJÍåŻŻ
åőđéŹĚçŽĎæĂğèČ¡äijŸåŇŰæŢĹæđIJæăźæ■ő䡚çŽĎæIJžåŹÍCPUæŢřéĞŔçŽĎäÿ■åŘŇèĂŇäÿ■åŘŇãĂĆ

èőÍèőž

`ProcessPoolExecutor` çŽĎåĚÿåđŃçŤÍæşŢåęĆäÿŃïijŽ

```
from concurrent.futures import ProcessPoolExecutor

with ProcessPoolExecutor() as pool:
    ...
    do work in parallel using pool
    ...
```

åĚűåŐ§çŘĘæŸŕïijŇäÿĂäÿł `ProcessPoolExecutor`
åĹŻåžžNäÿłçŃňçńŃçŽĎPythonèğčéĞŁåŹÍïijŇ NæŸŕçşżçż§äÿŁéİćåŔŕçŤÍCPUçŽĎäÿłæŢřãĂĆä¡ååŔŕäżěéĂŽ
`ProcessPoolExecutor(N)` æİëä£őæŤź åď ĎçŘĘåŹÍæŢřéĞŔãĂĆè£Źäÿłåď ĎçŘĘæśääijŽäÿĂçŻť è£ŘèąĘ
çĎűåŘŐåď ĎçŘĘæśäèćńåĚşéŮ■ãĂĆäÿ■è£ĞïijŇçÍŃåžŔäijŽäÿĂçŻť ç■L’å¿ĚçŻť åĹřæL’ĂæIJL’æŔŘäžď çŽĎå

èćńæŔŘäžď åĹřæśääÿ■çŽĎåůěä¡IJå£ĚéążèćńåőŽäźL’äÿžäÿĂäÿłåĞ¡æŢřãĂĆæIJL’äÿď çğ■æŰźæşŢåŐżæ
åęĆæđIJä¡ăæČşèőľäÿĂäÿłåĹŮèąÍæŐÍårijæĹŰäÿĂäÿł `map()`
æŞ■ä¡IJåźűèąŇæL’ğèąŇçŽĎèŕİïijŇåŔŕä¡£çŤÍ `pool.map()` :

```
# A function that performs a lot of work
def work(x):
    ...
    return result

# Nonparallel code
results = map(work, data)

# Parallel implementation
```

```
with ProcessPoolExecutor() as pool:
    results = pool.map(work, data)
```

åŘęåď ŰïijŇä¡ååŔřäžěä¡£çŤÍ `pool.submit()` æÌěæL'ŃåŁÍçŽĎæŔŘäžď'å■Ţäÿłäżžå Łąïij Ž

```
# Some function
def work(x):
    ...
    return result

with ProcessPoolExecutor() as pool:
    ...
    # Example of submitting work to the pool
    future_result = pool.submit(work, arg)

    # Obtaining the result (blocks until done)
    r = future_result.result()
    ...
```

åęĆæđIJä¡ăæL'ŃåŁÍæŔŘäžď'äÿĂäÿłäżžåŁąïijŇçżŞæđIJæŸřäÿĂäÿł Future åőđä¿ŃãĂĆ èęĄèŐůåŔŰæIJĂçżÍçżŞæđIJïijŇä¡ăéIJĂèęĄèřČçŤÍåőČçŽĎ result() æŰźæşŢãĂĆ åőČäijŽéŸżåąđè£ŽçÍŇçŻť'åÍřçżŞæđIJèćńè£ŤåŻđæÌěãĂĆ

åęĆæđIJäÿ■æČşéŸżåąđïijŇä¡ăè£ŸåŔřäžěä¡£çŤÍäÿĂäÿłåŻđèřČåĞ¡æŢřïijŇä¿ŃåęĆïijŽ

```
def when_done(r):
    print('Got:', r.result())

with ProcessPoolExecutor() as pool:
     future_result = pool.submit(work, arg)
     future_result.add_done_callback(when_done)
```

åŻđèřČåĞ¡æŢřæŐěåŔŮäÿĂäÿł Future åőđä¿ŃïijŇèćńçŤÍæÌěèŐůåŔŰæIJĂçźÍçŽĎçżŞæđIJïijÍæŕŤåęĆ åř¡çőąåď'ĎçŘĘæśăå¿Łåőźæ Ÿ Şä¡£çŤÍïijŇåIJÍèő¿èőąåď'ğçÍŇåžŔçŽĎæŮűåĂŹè£ŸæŸŕæIJL'å¿Łåď'ŽéIJĂèęĄæ

- è£Źçğ■ázűèąŇåď'ĎçŘĘæŁĂæIJŕåŔłéĂĆçŤÍäžŐéĆčäžŻåŔŕäžěèćńåĹĘèğčäÿžäžŠçŻÿçŃňçńŃéČÍåĹĘçŽ
- èćńæŔŘäžď'çŽĎäżżåŁąå£Ěéążæ ŸŕçőĂå■ŢåĞ¡æŢřå¡ćåijŔãĂĆåŕźäžŐæŰźæşŢãĂĄéŮ■åŇĚåŠŇåĚűäżŮ
- åĞ¡æŢřåŔĆæŢřåŠŇè£ŤåŻđåĂijå£ĚéążåĕijåőźpickleïijŇåŻääÿžèęĄä¡£çŤÍåĹřè£ŻçÍŃéŮť'çŽĎéĂŽä£ąï
- èćńæŔŘäžď'çŽĎäżżåŁąåĞ¡æŢřäÿ■åžŤä£ÌçŢŹçŁűæĂĄæĹŰæIJL'åL'ŕä¡IJçŤÍãĂĆéŹď'äžĘæL'Şå■řæŮěå

  äÿĂæŮęåŘŕåŁÍä¡ääÿ■èČ¡æŐğåĹűå■Řè£ŻçÍŃçŽĎäżżä¡Ţèą Ňäÿžïij ŇåŽäæ■d'æIJĂåě¡ä£ÌæŇĄçőĂå■ŢåŠ
- åIJÍUnixäÿŁè£ŻçÍŃæśăéĂŽè£ĞěČçŤÍ `fork()` çşżçż§èřČçŤÍèćńåĹŽåžžïijŇ

åőČäijŽåĚŃéŽĘPythonèğčéĞŁåŹÍïijŇåŇĚæŃňforkæŮűçŽĎæL'ĂæIJL'çÍŃåžŘçŁűæĂĄãĂĆ èĂŇåIJÍWindowsäÿŁïijŇåĚŃéŽĘèğčéĞŁåŹÍæŮűäÿ■äijŽåĚŃéŽĘçŁűæĂĄãĂĆ åőđéŹĚçŽĎ-forkæŞ■ä¡IJäijŽåIJÍçňňäÿĂæňąèřČçŤÍ `pool.map()` æĹŰ `pool.submit()` åŘŐåŔŚçŤ§ãĂĆ

- å¡Şä¡ăæůůåŘĹä¡£çŤÍè£ŻçÍŃæśăåŠŇåď'Žçž£çÍŃçŽĎæŮűåŹèęĄçL'źåĹńářŔå£ČãĂĆ

ä¡ăåžŤèŕěåIJĺåĹŻåžžäžžä¡Ţçž£çĺŃäźŃåL’■åĚĹåĹŻåžžåźűæ£Äæť’žè£ŻçĺŃæśăïijĹæŕŤåęĆåIJĺçĺŃåžŔåŘŕ

14.9 12.9 PythonçŽĎåĚĺåśĂéŤĄéŮőécŸ

éŮőécŸ

ä¡ăåůšçżŔåŘňèŕť’è£ĞåĚĺåśĂèğčéĞŁåŹĺéŤĄGILïijŇæŃĚå£ČåőČäijŽå¡śåŞ■åĹřåď’Žçž£çĺŃçĺŃåžŔçŽĎæ

èğčåĘşæŰźæąĹ

åř¡çőąPythonåőŇåĚĺæŤŕæŇĄåď’Žçž£çĺŃçijŰçĺŃïijŇ ä¡ĘæŸŕèğčéĞŁåŹĺçŽĎCèŕ■èĺĂåőđçŐŕéČĺåĹĘåIJĺ
åőđéŹĚäÿŁïijŇèğčéĞŁåŹĺèćńäÿĂäÿłåĚĺåśĂèğčéĞŁåŹĺéŤĄä£İæŁď’çİĂïijŇåőČçąőä£İäžžä¡ŢæŮűåĂŹéČ¡åŔł
GILæIJĂåď’ğçŽĎéŰőécŸåřśæŸŕPythonçŽĎåď’Žçž£çĺŃçĺŃåžŔåźűäÿ■èČ¡åĹl’çŤĺåď’ŽæäÿCPUçŽĎäijŸåŁ£
ïijĹæŕŤåęĆäÿĂäÿłä¡£çŤĺäžĘåď’Žäÿłçž£çĺŃçŽĎèőąçőŮåŕĘéŻĘåđŃçĺŃåžŔåŘłäijŽåIJĺäÿĂäÿłå■ŢCPUäÿŁéĺć

åIJĺèőĺèőžæŹőéĂŽçŽĎGILäźŃåL’■ïijŇæIJL’äÿĂçĆźèęĄåijžěřČçŽĎæŸŕGILåŘłäijŽå¡śåŞ■åĹřéĆčäžŽäÿ
åęĆæđIJä¡ăçŽĎçĺŃåžŔåď’ğéČĺåĹĘåŘłäijŽæűL’åŘŁåĹřI/OïijŇæŕŤåęĆç¡ŚçżIJäžď’äžŠïijŇéĆčäźĹä¡£çŤĺåď’Žç
åŻääÿžåőČäžňåď’ğéČĺåĹĘæŮűéŮťéČ¡åIJĺç■L’å¿ĚãĂĆåőđéŹĚäÿŁïijŇä¡ăåőŇåĚĺåŘŕäžěæŤ¿å£ČçŽĎåĹŻåžžå
çŐŕäžčæŞ■ä¡IJçşżçż§è£ŘèąŇè£ŹäźĹåď’Žçž£çĺŃæšąæIJL’äžžä¡ŢåŐŃåŁŻïijŇæšąåŢěåŘŕæŃĚå£ČçŽĎãĂĆ

èĂŇåŕźäžŐä¿İèţŰCPUçŽĎçĺŃåžŔïijŇä¡ăéIJĂèęĄåijĐæÿĚæěŽæL’ğèąŇçŽĎèőąçőŮçŽĎçL’źçĆźãĂĆ
ä¿ŃåęĆïijŇäijŸåŇŰåžŢåśĆçőŮæşŢèęĄæŕŤä¡£çŤĺåď’Žçž£çĺŃè£ŘèąŇå£ńå¿Ůåď’ŽãĂĆ
çśżäijijçŽĎïijŇçŤśäžŐPythonæŸŕèğčéĞŁæL’ğèąŇçŽĎïijŇåęĆæđIJä¡ăåřĘéČčäžŽæĂğèČ¡çŞűécĹäżčçăĄçğżå
éĂ§åžęäź§äijŽæŔŘå■ĞçŽĎå¿Ĺå£ńãĂĆåęĆæđIJä¡ăèęĄæŞ■ä¡IJæŢřçżĎïijŇéĆčäźĹä¡£çŤĺNumPyè£ŹæăůçŽĎ
æIJĂåŘŐïijŇä¡ăè£ŸåŕźäžèèĂČèŹŠäÿŇåĚűäžŮåŕéĂL’ăődçŐŕæŰźæąĹïijŇæŕŤåęĆPyPyïijŇăőČéĂŽè£ĞäÿA
ïijĹäÿ■è£ĞåIJĺåĘŹè£ŹæIJňäźęçŽĎæŮűåĂŹåőČè£Ÿäÿ■èČ¡æŤŕæŇĄPython 3ïijL’ãĂĆ

è£ŸæIJL’äÿĂçĆźèęĄæşĺæĐŔçŽĎæŸŕïijŇçž£çĺŃäÿ■æŸŕäÿŞéŮĺçŤĺæİěäijŸåŇŰæĂğèČ¡çŽĎãĂĆ
äÿĂäÿłCPUä¿İèţŰåđŃçĺŃåžŔåŘŕèČ¡äijŽä¡£çŤĺçž£çĺŃæİěçőąçŘEäÿĂäÿłåŻ¿å¡ćçŤĺæĹůçŢŇéİćãĂĄäÿĂäÿłç
è£ŹæŮűåĂŹïijŇGILäijŽäžğçŤ§äÿĂäžŻéŮőécŸïijŇåŽääÿžåęĆæđIJäÿĂäÿłçž£çĺŃéŢ£æIJ§æŇĄæIJL’GILçŽĐ
äžŃåődäÿŁïijŇäÿĂäÿłåĘŹçŽĎäÿ■åě¡çŽĎCèŕ■èĺĂæL’l’åśŢäijŽåŕijèĞt’è£ŹäÿłéŮőécŸæŻt’åŁääÿěéĞ■ïijŇ
åř¡çőąäžčçăĄçŽĎèőąçőŮéČĺåĹĘäijŽæŕŤäźŃåL’■è£ŘèąŇçŽĎæŻt’å£ńäžŽãĂĆ

èŕť’äžĘè£ŹäźĹåď’ŽïijŇçŐŕåIJĺæČşèŕť’çŽĎæŸŕæĹŚäžňæIJL’äÿd’çğ■ç■ŰçŢěæİèèğčåĘşGILçŽĎçijžçĆźãĂ
éęŰåĚĹïijŇåęĆæđIJä¡ăåőŇåĚĺåůěä¡IJäžŐPythonçŐŕåćČäÿ■ïijŇä¡ăåŘŕäžěä¡£çŤĺ
`multiprocessing` æĺąåİŮæİěåĹŻåžžäÿĂäÿłè£ŽçĺŃæśăïijŇ
åźűåČŔå■ŔåŘŇåď’ĎçŘĘåŹĺäÿĂæăůçŽĎä¡£çŤĺåőČãĂĆä¿ŃåęĆïijŇåĄĞåęĆä¡ăæIJL’åęĆäÿŃçŽĎçž£çĺŃäžčç

```
# Performs a large calculation (CPU bound)
def some_work(args):
    ...
    return result

# A thread that calls the above function
def some_thread():
    while True:
        ...
        r = some_work(args)
    ...
```

ä£őæŤźäżčçăĄïijŇä¡£çŤĺè£ŻçĺŃæśăïijŽ

```
# Processing pool (see below for initiazation)
pool = None

# Performs a large calculation (CPU bound)
def some_work(args):
    ...
    return result

# A thread that calls the above function
def some_thread():
    while True:
        ...
        r = pool.apply(some_work, (args))
        ...

# Initiaze the pool
if __name__ == '__main__':
    import multiprocessing
    pool = multiprocessing.Pool()
```

è£ŹäÿłéĂŽè£Ğä¡£çŤĺäÿĂäÿłæŁĂåůğåĹľçŤĺè£ŻçĺŃæśăèğčåĘşäžĘGILçŽĎéŮőéćŸãĂĆ å¡ŞäÿĂäÿłçž£çĺŃæČşèęĄæĽğèąŇCPUåŕĘéŻĘåđŃåůěä¡IJæŮűïijŇäijŽåřĘäżżåŁąåŔŚçżŹè£ŻçĺŃæśăãĂĆ çĎűåŘŐè£ŻçĺŃæśăäijŽåIJĺåŔęåď ŰäÿĂäÿłè£ŻçĺŃäÿ■åŘŕåŁĺäÿĂäÿłå■ŢçŃňçŽĎPythonèğčéĞŁåŹĺæİěåůěä¡I å¡Şçž£çĺŃç■Ľå¿ĚçżŞæđIJçŽĎæŮűåĂŹäijŽéĞŁæŤ¿GILãĂĆ åźűäÿŤïijŇçŤśäžŐèőąçőŮäżżåŁąåIJĺå■ŢçŃňèğ åIJĺäÿĂäÿłåď Žæăÿçşżçż§äÿŁéİćïijŇä¡ăäijŽåŔŚçŐřè£ŹäÿłæŁĂæIJŕåŔŕäżěèőľä¡ăå¿Ĺåě¡çŽĎåĹľçŤĺåď ŽCPU

åŔęåď ŰäÿĂäÿłèğčåĘşGILçŽĎç■ŰçŢěæŸŕä¡£çŤĺCæĽľåśŢçijŰçĺŃæŁĂæIJŕãĂĆ äÿżèęĄæĂİæČşæŸŕåřĘèőąçőŮåŕĘéŻĘåđŃäżżåŁąè¡ňçğżçżŹCïijŇèů§PythonçŃňçńŃïijŇåIJĺåůěä¡IJçŽĎæŮű è£ŹåŔŕäżěéĂŽè£ĞåIJĺCäżčçăĄäÿ■æŔŠåĚěäÿŃéİćè£ŹæăůçŽĎçĽźæőŁåőŔæİěåőŇæĹŘïijŽ

```
#include "Python.h"
...

PyObject *pyfunc(PyObject *self, PyObject *args) {
   ...
   Py_BEGIN_ALLOW_THREADS
   // Threaded C code
   ...
   Py_END_ALLOW_THREADS
   ...
}
```

åęĆæđIJä¡ăä¡£çŤĺåĚűäżŰåůěåĚůèő£éŮőCèŕ■èĺĂïijŇæŕŤåęĆåŕźäžŐCythonçŽĎctypesåžŞïijŇä¡ăäÿ■éIJĂ ä¿ŃåęĆïijŇctypesåIJĺèřČçŤĺCæŮűäijŽèĞłåŁĺéĞŁæŤ¿GILãĂĆ

èőÍèőž

èőÿåď ŽçÍŃåžŔåŚŸåIJÍéÌćåŕźçž£çÍŃæĂğèČ¡éŮőéćŸçŽĎæŮűåĂŹïijŇél ňäÿŁåřśäijŽæĂłç¡łGILïijŇäžĂ
åĚűåőđè£Źæăůå■Řåď łäÿ■åŐŽéĄŞäź§åď łåď ľçIJ§äžĘçĆźãĂĆ
ä¡IJäÿžäÿĂäÿłçIJ§åőđçŽĎä¿Ńå■ŘïijŇåIJÍåď Žçž£çÍŃçŽĎç¡ŚçżIJçijŰçÍŃäÿ■çěđçğŸçŽĎ
stalls åŔŕèČ¡æŸŕåŽääÿžåĚűäžŰåŐ§åŽäæŕŤåęĆäÿĂäÿłDNSæ§ěæL’¿åžűæŮűïijŇèĂŇèů§GILæŕńæŮăåĚş
æIJĂåŘŐä¡ăçIJ§çŽĎéIJĂèęĄåĚĹåŐżæŘđæĞĆä¡ăçŽĎäžčçăĄæŸŕåŘęçIJ§çŽĎèćńGILå¡śåŞ■åĹřãĂĆ
åŘŇæŮűè£ŸèęĄæŸŐçŹ¡GILåď ğéČĺåĹĘéČ¡åžŤèŕěåŔłåĚşæşÍCPUçŽĎåď ĎçŘĘèĂŇäÿ■æŸŕI/O.

åęĆæđIJä¡ăåĞĘåď Ğä¡£çŤÍäÿĂäÿłåď ĎçŘĘåŹÍæśăïijŇæşÍæĎŔçŽĎæŸŕè£Źæăůå ĄŽæűL’åŘŁåĹřæŢřæ■ć
èćńæL’ğèąŇçŽĎæŞ■ä¡IJéIJĂèęĄæŤ¿åIJÍäÿĂäÿłéĂŽè£Ğdefèŕ■åŘěåőŽäźL’çŽĎPythonåĞ¡æŢřäÿ■ïijŇäÿ■èČ¡
åźűäÿŤåĞ¡æŢřåŔĆæŢřåŠŇè£ŤåŻđåĂijå£ĚéążèęĄåĚijåőźpickleãĂĆ
åŘŇæăůïijŇèęĄæL’ğèąŇçŽĎäżżåŁąéĞŘå£Ěéążèűşåď §åď ğäżěäijèèąéécİåď ŰçŽĎéĂŽä£ąåijĂéŤĂãĂĆ

åŘęåď ŰäÿĂäÿłéŽ¿çĆźæŸŕå¡Şæůůåŕĺä¡£çŤÍçž£çÍŃåŠŇè£ŻçÍŃæśăçŽĎæŮűåĂŹäijŽèőľä¡ăå¿Ĺåď ťçŰi
åęĆæđIJä¡ăèęĄåŘŇæŮűä¡£çŤÍäÿd’èĂĚïijŇæIJĂåě¡åIJÍçÍŃåžŔåŘŕåŁÍæŰűïijŇåĹŻåżžäżżä¡Ţçž£çÍŃäźŃåL’■å
çĎűåŘŐçž£çÍŃä¡£çŤÍåŘŇæăůçŽĎè£ŻçÍŃæśăæİèè£ŻèąŇåőĆäżnçŽĎèőąçőŮåŕĘéŻĘđŃåůëä¡IJãĂĆ

CæL’l’åśŢæIJĂéĞ■èęĄçŽĎçL’źå¿ĄæŸŕåőĆäżnåŠŇPythonèğčéĞŁåŹÍæŸŕä£ÍæŇĄçŃňçńŃçŽĎãĂĆ
äź§åřśæŸŕèŕťïijŇåęĆæđIJä¡ăåĞĘåď ĞåřĘPythonäÿ■çŽĎäżżåŁąåĹĘéĚ■åĹřCäÿ■åŐżæL’ğèąŇïijŇ
ä¡ăéIJĂèęĄçąőä£ÍCäżčçăĄçŽĎæŞ■ä¡IJèů§Pythonä£ÍæŇĄçŃňçńŃïijŇ
è£Źåřśæ ĎŔåŚşçİĂäÿ■èęĄä¡£çŤÍPythonæŢřæ■őçżŞæđĎäżěåŔŁäÿ■èęĄěřČçŤÍPythonçŽĎC
APIãĂĆ åŘęåď ŰäÿĂäÿłåřśæŸŕä¡ăèęĄçąőä£ÍCæL’l’åśŢæL’ĂåĄŽçŽĎåůěä¡IJæŸŕèűşåď §çŽĎïijŇåĂijå¿Ůä¡ăè
äź§åřśæŸŕèŕťCæL’l’åśŢæŃĚèť’§ètůäžĘåď ğéĞŘçŽĎèőąçőŮäżżåŁąïijŇèĂŇäÿ■æŸŕåřŚæŢřåĞăäÿłèőąçőŮãĂĆ

è£ŹäžŻèğčåĘşGILçŽĎæŰźæąĹåźűäÿ■èČ¡éĂĆçŤÍäžŐæL’ĂæIJL’éŮőéćŸãĂĆ
ä¿ŃåęĆïijŇæ§ŘäžŻçśżåđŃçŽĎåžŤçŤÍçÍŃåžŔåęĆæđIJèćńåĹĘèğčäÿžåď Žäÿłè£ŻçÍŃåď ĎçŘĘçŽĎèŕĺåźűäÿ■è
äź§äÿ■èČ¡åřĘåőĆçŽĎéČĺåĹĘäžčçăĄæŤźæĹŘCèŕ■èÍĂæL’ğèąŇãĂĆ
åŕźäžŐè£ŹäžŻåžŤçŤÍçÍŃåžŔïijŇä¡ăåřśèęĄèĞłåůśéIJĂæśĆèğčåĘşæŰźæąĹäžĘ
ïijĹæŕŤåęĆåď Žè£ŻçÍŃèő£éŮőåĚśäžńåĘĚå■ŸåŃžïijŇåď Žèğčæđ ŘåŹÍè£ŘèąŇäžŐåŘŇäÿĂäÿłè£ŻçÍŃç■L’ïijL
æĹŰèĂĚïijŇä¡ăè£ŸåŔŕäżěèĂČèŹŚäÿŃåĚűäžŰçŽĎèğčéĞŁåŹÍåőđçŐřïijŇæŕŤåęĆPyPyãĂĆ

äžĘèğčæŻť åď ŽåĚşäžŐåIJÍCæL’l’åśŢäÿ■éĞŁæŤ¿GILïijŇèŕůåŔĆèĂČ15.7åŠŇ15.10åŕŘèŁĆãĂĆ

14.10 12.10 åőŽäźĽäÿĂäÿłActoräżżåŁą

éŮőéćŸ

ä¡ăæČşåőŽäźL’èů§actoræĺąåijŔäÿ■çśżäijijâĂIJactorsâĂİèğŠèL’šçŽĎäżżåŁą

èğčåĘşæŰźæąĹ

actoræĺąåijŔæŸŕäÿĂçğ■æIJĂåŔd’èĂĄçŽĎäź§æŸŕæIJĂçőĂå■ŢçŽĎåźűèąŇåŠŇåĹĘåÿČåijŔèőąçőŮèğčä
äžŃåőđäÿŁïijŇåőČåď ľçŤ§çŽĎçőĂå■ŢæĂğæŸŕåőČąęĆæ■d’åŘŮæńćè£ŐçŽĎéĞ■èęĄåŐ§åŽääźŃäÿĂãĂĆ
çőĂå■ŢæİěëőšïijŇäÿĂäÿłactoråřśæŸŕäÿĂäÿłåźűåŘŚæL’ğèąŇçŽĎäżżåŁąïijŇåŘłæŸŕçőĂå■ŢçŽĎæL’ğèąŇåŘ
åŞ■åžŤè£ŹäžŻæűĹæĄŕæŮűïijŇåőČåŔŕèČ¡è£ŸäijŽçżŹåĚűäžŰactoråŘŚéĂĄæŽť è£ŻäÿĂæ■ěçŽĎæűĹæĄŕãĂ
actoräźŃéŮť çŽĎéĂŽä£ąæŸŕå■ŢåŘŚåŠŇåijĆæ■ěçŽĎãĂĆåŽăæ■d’ïijŇæűĹæĄŕåŘŚéĂĄèĂĚäÿ■ç§ěéĄŞæűĹ
äź§äÿ■äijŽæŐěæŤűåĹřäÿĂäÿłæűĹæĄŕåůšècńåď ĎçŘĘçŽĎåŽđåžŤæĹŰéĂŽç§ěãĂĆ

çżŞåŘĹä¡£çŤĺäÿĂäÿłçž£çĺŃåŠŇäÿĂäÿłéŸ§åĹŮåŔŕäżěå¿Ĺåőźæ ŸŞçŽĎåőŽäźĽ’actorïijŇä¿ŃåęĆïijŽ

```
from queue import Queue
from threading import Thread, Event

# Sentinel used for shutdown
class ActorExit(Exception):
    pass

class Actor:
    def __init__(self):
        self._mailbox = Queue()

    def send(self, msg):
        '''
        Send a message to the actor
        '''
        self._mailbox.put(msg)

    def recv(self):
        '''
        Receive an incoming message
        '''
        msg = self._mailbox.get()
        if msg is ActorExit:
            raise ActorExit()
        return msg

    def close(self):
        '''
        Close the actor, thus shutting it down
        '''
        self.send(ActorExit)

    def start(self):
        '''
        Start concurrent execution
        '''
        self._terminated = Event()
        t = Thread(target=self._bootstrap)

        t.daemon = True
        t.start()

    def _bootstrap(self):
        try:
            self.run()
        except ActorExit:
            pass
        finally:
            self._terminated.set()
```

```
    def join(self):
        self._terminated.wait()

    def run(self):
        '''
        Run method to be implemented by the user
        '''
        while True:
            msg = self.recv()

# Sample ActorTask
class PrintActor(Actor):
    def run(self):
        while True:
            msg = self.recv()
            print('Got:', msg)

# Sample use
p = PrintActor()
p.start()
p.send('Hello')
p.send('World')
p.close()
p.join()
```

è£Źäÿłä¿Ńå■Řäÿ■ïijŇä¡ää¡£çŤĺactoråőđä¿ŃçŽĎ send() æŰźæşŢåŔŠéĂĄæúĹæĄŕçżŹåőČäžñãĂĆ åĚűæIJžåĹűæŸŕïijŇè£ŹäÿłæŰźæşŢäijŽåřĘæúĹæĄŕæŤ¿åĚěäÿĂäÿçĎűåŘŐåřĘåĚűè¡ñäžd'çżŹåd'ĎçŘĘèćńæŐěåŔŮæűĹæĄŕçŽĎäÿĂäÿłåĘĚéČĺçž£çĺŇãĂĆ close() æŰźæşŢéĂŽè£ĞåIJĺéŸ§åĹŮäÿ■æŤ¿åĚěäÿĂäÿłçL'źæőŁçŽĎđąŞĺåĚţåĂijïijĹActorExitïijL'æİěåĚşéÚçŤĺæĹůåŔŕäžěéĂŽè£Ğçżğæl'£ActoråźűåőŽäźL'åőđçŐřèĞłåůšåd'ĎçŘĘéĂżè¿Śrun()æŰźæşŢæİěåőŽäźL'æŰřActorExit åijĆåÿÿçŽĎä¡£çŤĺåřśæŸŕçŤĺæĹůèĞłåőŽäźL'äžčçăĄåŔŕäžěåIJĺéIJĂèęĄçŽĎæŮűåĂŹæİěæ■ŢèĆïijĹåijĆåÿÿèćńget()æŰźæşŢæŁŻåĞžåźűäijäæŠ■åĞžåŐżïijL'ãĂĆ

åęĆædIJä¡ăæŤ¿åő¡åŕźäžŐåŘŇæ■ěåŠŇåijĆæ■ěæűĹæĄŕåŔŠéĂĄçŽĎèęĄæśĆïijŇ çśżactorårźèśąè£ŸåŔŕäžěéĂŽè£ĞçŤ§æĹŘåŹĺæİěçőĂåŇŰåőŽäźL'ãĂĆä¿ŃåęĆïijŽ

```
def print_actor():
    while True:

        try:
            msg = yield       # Get a message
            print('Got:', msg)
        except GeneratorExit:
            print('Actor terminating')

# Sample use
p = print_actor()
next(p)      # Advance to the yield (ready to receive)
p.send('Hello')
```

```
p.send('World')
p.close()
```

èőÍèőž

actoræÍąåijŘçŽĎé■ĚåŁŻåřśåIJÍäžŐåőČçŽĎçőÄå■ŢæĂğãĂĆ
åőđéŹĚäÿŁïijŇè£ŹéĞŇäżĚäżĚåŘłæIJL’äÿĂäÿłæăÿå£ČæŞ■ä¡IJ send() .
çŤŽèĞşïijŇåŕźäžŐåIJÍå§žäžŐactorçşżçż§äÿ■çŽĎâĂIJæűŁæĄŕâĂİçŽĎæşŻåŇŰæęĆå£ţåŘŕäżěåůšåd’Žçğ■æŰ
ä¿ŇåęĆïijŇä¡ăåŘŕäżěäżěåĚČçżĎå¡ćåijŘäijăéĂŠæăĞç■¿æűŁæĄŕïijŇèől’actoræL’ğèąŇäÿ■åŘŇçŽĎæŞ■ä¡IJïij

```
class TaggedActor(Actor):
    def run(self):
        while True:
            tag, *payload = self.recv()
            getattr(self,'do_'+tag)(*payload)

    # Methods correponding to different message tags
    def do_A(self, x):
        print('Running A', x)

    def do_B(self, x, y):
        print('Running B', x, y)

# Example
a = TaggedActor()
a.start()
a.send(('A', 1))       # Invokes do_A(1)
a.send(('B', 2, 3))    # Invokes do_B(2,3)
```

ä¡IJäÿžåŘęåd’ŰäÿĂäÿłä¿Ńå■ŘïijŇäÿŃéÍćçŽĎactoråĚĄèőÿåIJÍäÿĂäÿłåůěä¡IJèĂĚäÿ■è£ŘèąŇäżżæĎŘçŽ
åźűäÿŤéĂŽè£ĞäÿĂäÿłçL’źæőŁçŽĎResultåŕźèśąè£ŤåŻđçżŞæđIJïijŽ

```
from threading import Event
class Result:
    def __init__(self):
        self._evt = Event()
        self._result = None

    def set_result(self, value):
        self._result = value

        self._evt.set()

    def result(self):
        self._evt.wait()
        return self._result

class Worker(Actor):
    def submit(self, func, *args, **kwargs):
```

```
        r = Result()
        self.send((func, args, kwargs, r))
        return r

    def run(self):
        while True:
            func, args, kwargs, r = self.recv()
            r.set_result(func(*args, **kwargs))

# Example use
worker = Worker()
worker.start()
r = worker.submit(pow, 2, 3)
print(r.result())
```

æIJĂåŘŐïijŇâĂIJåŔŠéĂĄâĂİäÿĂäÿłäżżåŁąæűĹæĄŕçŽĎæęĆå£ţåŔŕäżěèćńæL'ľåśŢåĹřåď'Žè£ŻçÍŃçŤŽ
ä¿ŇåęĆïijŇäÿĂäÿłçśżactorårźŕèśąçŽĎ `send()` æŰźæşŢåŔŕäżěèćńçijŰçÍŃèőľåőČèČ¡åIJĺäÿĂäÿłåěŮæŐěå■Ů
æĹŰéĂŽè£Ğæ§ŘäžŽæűĹæĄŕäÿ■éŮt'äżűïijĹæŕŤåęĆAMQPãĂĄZMQç■L'ïijL'æİěåŔŠéĂĄãĂĆ

14.11 12.11 åőđçŐřæűĹæĄŕåŔŠåÿČ/èőćéŸĚæĺąåđŃ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłå§žäžŐçž£çÍŃéĂŽä£ąçŽĎçÍŃåžŔïijŇæČşèőľåőČäžňåőđçŐřåŔŠåÿČ/èőćéŸĚæĺąåijŔçŽİ

èğčåĘşæŰźæąĹ

èęĄåőđçŐřåŔŠåÿČ/èőćéŸĚçŽĎæűĹæĄŕéĂŽä£ąæĺąåijŔïijŇ
ä¡ăéĂŽåÿÿèęĄåijŢåĚěäÿĂäÿłå■ŢçŃňçŽĎâĂIJäžđ'æ■ćæIJžâĂİæĹŰâĂIJç¡ŚåĚşâĂİåŕźèśąä¡IJäÿžæL'ĂæIJL'æ
äź§åřśæŸŕèŕt'ïijŇäÿ■çŽt'æŐěåřĘæűĹæĄŕäżŐäÿĂäÿłäżżåŁąåŔŠéĂĄåĹřåŘęäÿĂäÿłïijŇèĂŇæŸŕåřĘåĚűåŔŠé
çĎűåŘŐçŤśäžđ'æ■ćæIJžåřĘåőČåŔŠéĂĄçźŹäÿĂäÿłæĹŰåđ'ŽäÿłèćńåĕşèĄŤäżżåŁąãĂĆäÿŃéİćæŸŕäÿĂäÿłéİđ

```
from collections import defaultdict

class Exchange:
    def __init__(self):
        self._subscribers = set()

    def attach(self, task):
        self._subscribers.add(task)

    def detach(self, task):
        self._subscribers.remove(task)

    def send(self, msg):
        for subscriber in self._subscribers:
            subscriber.send(msg)
```

```
# Dictionary of all created exchanges
_exchanges = defaultdict(Exchange)

# Return the Exchange instance associated with a given name
def get_exchange(name):
    return _exchanges[name]
```

äÿĂäÿłäžď'æ■ćæIJžåřśæŸŕäÿĂäÿłæŹőéĂŽåŕźèśąïijŇèť§èť'čçžť'æŁď'äÿĂäÿłæť'żèůČçŽĎèőćéŸĚèĂĚéŻł
æŕŔäÿłäžď'æ■ćæIJžéĂŽè£ĞäÿĂäÿłåŘ■çğřåőŽä¡■ïijŇget_exchange()
éĂŽè£ĞçżŹåőŽäÿĂäÿłåŘ■çğřè£ŤåŻđçŻÿåžŤçŽĎ Exchange åőđä¿ŃãĂĆ

äÿŃéİćæŸŕäÿĂäÿłçőĂå■Ţä¿Ńå■ŘïijŇæijŤçď'žäžĘåęĆä¡Ţä¡£çŤĺäÿĂäÿłäžď'æ■ćæIJžïijŽ

```
# Example of a task.  Any object with a send() method

class Task:
    ...
    def send(self, msg):
        ...

task_a = Task()
task_b = Task()

# Example of getting an exchange
exc = get_exchange('name')

# Examples of subscribing tasks to it
exc.attach(task_a)
exc.attach(task_b)

# Example of sending messages
exc.send('msg1')
exc.send('msg2')

# Example of unsubscribing
exc.detach(task_a)
exc.detach(task_b)
```

åř¡çőąåŕźäžŐè£ŹäÿłéŮőécŸæIJL'å¿Ĺåď'ŽçŽĎåŔŸçğ■ïijŇäÿ■è£ĞäÿĞåŔŸäÿ■çężåĚűåőŮãĂĆ
æűĹæĄŕäijŽèćńåŔŠéĂĄçżŹäÿĂäÿłäžď'æ■ćæIJžïijŇçĎűåŘŐäžď'æ■ćæIJžäijŽåřĘåőČäžňåŔŠéĂĄçżŹèćńçżŠå

èőĺèőž

éĂŽè£ĞéŸ§åĹŮåŔŠéĂĄæűĹæĄçŽĎäżżåŁąæĹŰçž£çĺŃçŽĎæĺąåijŔå¿Ĺåőźæ ŸŞèćńåőđçŐřåźűäÿŤäź§
äÿ■è£ĞïijŇä¡£çŤĺåŔŠåÿČ/èőćéŸĚæĺąåijŔçŽĎåě¡åď'ĎæŻť'åŁäæŸŐæŸ¿ãĂĆ

éęŰåĚĹïijŇä¡£çŤĺäÿĂäÿłäžď'æ■ćæIJžåŔŕäżěçőĂåŇŰåď'ğéČĺåĹĘæűL'åŔŁåĹřçž£çĺŃéĂŽä£ąçŽĎåůěä¡I
æŮäéIJĂåŐżåĘŹéĂŽè£Ğåď'Žè£ŻçĺŃæĺąåİŮæİěæŞ■ä¡IJåď'Žäÿłçž£çĺŃïijŇä¡ääŔłéIJĂèęĄä¡£çŤĺè£Źäÿłäžď'æ

æ§Řçğ■çÍŃåžęäÿŁïijŃè£Źäÿłåřśèů§æŮěå£ŮæÍąåİŮçŽĎåůěä¡IJåÓ§çŘĘçśżäijijãĂĆ
åőđéŹĚäÿŁïijŃåőČåŘŕäżěè¡żæİ¿çŽĎèğčèĂęçÍŃåžŔäÿ■åď'ŽäÿłäżżåŀąãĂĆ

åĚűæňąïijŃäžď'æ■ćæIJžåź£æŠ■æűĹæĄŕçżŹåď'Žäÿłèőćé ŸĚèĂĚçŽĎèČ¡åŁŻåÿęæİěäžĘäÿĂäÿłåĚÍæŰřçŽ
ä¿ŃåęĆïijŃä¡ääŘŕäżěä¡£çŤÍåď'Žäżżåŀąçşżçż§ãĂĄåź£æŠ■æĹŰæĽĞåĞžãĂĆ
ä¡ăè£ŸåŘŕäżěéĂŽè£ĞäżěæŹőéĂŽèőćé ŸĚèĂĚèžńäż¡çżŚåőŽæİěædĎåżžèřČèŕŢåŠŃèŕŁæŰ■ůåěåĚůãĂĆ
ä¿ŃåęĆïijŃäÿŃéİćæŸŕäÿĂäÿłçőĂå■ŢçŽĎèŕŁæŰ■çśżïijŃåŘŕäżěæŸ¿çď'žèćńåŔŚéĂĄçŽĎæűĹæĄŕïijŽ

```
class DisplayMessages:
    def __init__(self):
        self.count = 0
    def send(self, msg):
        self.count += 1
        print('msg[{}]: {!r}'.format(self.count, msg))

exc = get_exchange('name')
d = DisplayMessages()
exc.attach(d)
```

æIJĂåŘŐïijŃèŕěåőđçŐŕçŽĎäÿĂäÿłéĞ■èęĄçL'źçĆźæŸŕåőČèČ¡åĚijåőźåď'Žäÿłâ ĂIJtask-likeâĂİåŕźèśąãĂĆ ä¿ŃåęĆïijŃæűĹæĄŕæŐěåŘŮèĂĚåŘŕäżěæŸŕactorïijĹ12.10åřŔèŁĆäżŃçż■ïijL'ãĂĄå■ŔçÍŃ `send()` æŰźæşŢçŽĎäÿIJèě£ãĂĆ

åĚşäžŐäžď'æ■ćæIJžçŽĎäÿĂäÿłåŘŕèČ¡éŮőécŸæŸŕåŕźäžŐèőćéŸĚèĂĚçŽĎæ■čçąőçżŚåőŽåŠŃèğčçżŚãĂ
äÿžäžĘæ■čçąőçŽĎçőąçŘĘètĎæžŘïijŃæŕŘäÿĂäÿłçżŚåőŽçŽĎèőćéŸĚèĂĚå£ĚéążæIJĂçżĹèęĄèğčçżŚãĂĆ
åIJĺäżčçăĄäÿ■éĂŽåÿÿäijŽæŸŕåČŔäÿŃéİćè£ŹæăůçŽĎæĺąåijŔïijŽ

```
exc = get_exchange('name')
exc.attach(some_task)
try:
    ...
finally:
    exc.detach(some_task)
```

æ§Řçğ■æĎŘäźL'äÿŁïijŃè£ŹäÿłåŠŃä¡£çŤÍæŰĞäżűãĂĄéŤĄåŠŃçśżäijijåŕźèśąå¿ĹåČŔãĂĆ éĂŽåÿÿå¿Ĺåőźæ ŸŞäijŽå£ŸèőřæIJĂåŘŐçŽĎ `detach()` æ■ěéłď'ãĂĆ äÿžäžĘçőĂåŃŰè£ŹäÿłïijŃä¡ääŘŕäżěèĂČèŹŚä¡£çŤÍäÿŁäÿŃæŰĞçőąçŘĘåŹĺå■ŔèőőãĂĆ ä¿ŃåęĆïijŃåIJĺäžď'æ■ćæIJžåŕźèśąäÿŁåćđåŁääÿĂäÿł `subscribe()` æŰźæşŢïijŃåęĆäÿŃïijŽ

```
from contextlib import contextmanager
from collections import defaultdict

class Exchange:
    def __init__(self):
        self._subscribers = set()

    def attach(self, task):
        self._subscribers.add(task)

    def detach(self, task):
```

```
        self._subscribers.remove(task)

    @contextmanager
    def subscribe(self, *tasks):
        for task in tasks:
            self.attach(task)
        try:
            yield
        finally:
            for task in tasks:
                self.detach(task)

    def send(self, msg):
        for subscriber in self._subscribers:
            subscriber.send(msg)

# Dictionary of all created exchanges
_exchanges = defaultdict(Exchange)

# Return the Exchange instance associated with a given name
def get_exchange(name):
    return _exchanges[name]

# Example of using the subscribe() method
exc = get_exchange('name')
with exc.subscribe(task_a, task_b):
     ...
     exc.send('msg1')
     exc.send('msg2')
     ...

# task_a and task_b detached here
```

æIJĂåŘŐè£ŸåžŤèŕěæşĺæĎŘçŽĎæŸŕåĚşäžŐäžďæ■ćæIJžçŽĎæĂİæČşæIJL'å¿Ĺåď'Žçğ■çŽĎæL'l'åśŢåőđä¿ŃåęĆïijŇäžď'æ■ćæIJžåŘŕäżěåőđçŐřäÿĂæŢt'äÿłæűĹæĄŕéĂŽéĄŞéŻĘåŘĹæĹŰæŔŘä¿Żäžď'æ■ćæIJžåŘ■çğäžď'æ■ćæIJžè£ŸåŘŕäżěèćńæL'l'åśŢåĹřåĹĘåÿČåijŔèőąçőŮçĺŃåžŔäÿ■ïijĹæŕŤåęĆïijŇářĘæűĹæĄŕèůŕçŤśåĹřäj

14.12 12.12 ä¡£çŤĺçŤ§æĹŘåŹĺäžčæŻ£çž£çĺŃ

éŮőécŸ

ä¡ăæČşä¡£çŤĺçŤ§æĹŘåŹĺïijĹå■ŔçĺŃïijL'æŻ£äžčçşżçż§çž£çĺŃæİěåőđçŐřåźűåŔŚãĂĆè£ŹäÿłæIJL'æŮűåŀ

èğčåĘşæŰźæąĹ

èęĄä¡£çŤĺçŤ§æĹŘåŹĺåőđçŐřèĞłåůśçŽĎåźűåŔŚïijŇä¡ăéęŰåĚĹèęĄåŕźçŤ§æĹŘåŹĺåĞ¡æŢřåŠŇ `yield` èŕ■åŘěæIJL'æůśåĹżçŘĘèğčãĂĆ `yield` èŕ■åŘěäijŽèőľäÿĂäÿłçŤ§æĹŘåŹĺæŇĆèţůåőČçŽĎæL'ğèąŇïi

åřĘçŤ§æĹŘåŹĺå¡ŞåĄŽæ§Řçğ■âĂIJäżżåŁąâĂİåźűä¡£çŤĺäżżåŁąå■Ŕä¡IJåĹĞæ■ćæİěæŻ£æ■ćåőČäžňçŽĎæĽğ
èęĄæijŤçď žè£Źçğ■æĂİæČşïijŇèĂČèŹŚäÿŃéİćäÿď äÿłä¡£çŤĺçőĂå■ŢçŽĎ yield
èŕ■åŔěçŽĎçŤ§æĹŘåŹĺåĞ¡æŢřïijŽ

```
# Two simple generator functions
def countdown(n):
    while n > 0:
        print('T-minus', n)
        yield
        n -= 1
    print('Blastoff!')

def countup(n):
    x = 0
    while x < n:
        print('Counting up', x)
        yield
        x += 1
```

è£ŹäžŻåĞ¡æŢřåIJĺåĘĚéČĺä¡£çŤĺyieldèŕ■åŔěïijŇäÿŃéİćæŸŕäÿĂäÿłåőđçŐřäžEçőĂå■ŢäżżåŁąèřČåžęåŹĺ

```
from collections import deque

class TaskScheduler:
    def __init__(self):
        self._task_queue = deque()

    def new_task(self, task):
        '''
        Admit a newly started task to the scheduler

        '''
        self._task_queue.append(task)

    def run(self):
        '''
        Run until there are no more tasks
        '''
        while self._task_queue:
            task = self._task_queue.popleft()
            try:
                # Run until the next yield statement
                next(task)
                self._task_queue.append(task)
            except StopIteration:
                # Generator is no longer executing
                pass

# Example use
sched = TaskScheduler()
sched.new_task(countdown(10))
```

```
sched.new_task(countdown(5))
sched.new_task(countup(15))
sched.run()
```

`TaskScheduler` çśżåIJÍäÿĂäÿłå¿łçŐŕäÿ■è£ŘèąŇçŤ§æĹŘåŹĺéŻĘåŘĹâĂŤâĂŤæŕŔäÿłéČ¡è£ŘèąŇèŁŘèąŇè£Źäÿłä¿Ńå■ŘïijŇè¿ŞåĞžåęĆäÿŃïijŽ

```
T-minus 10
T-minus 5
Counting up 0
T-minus 9
T-minus 4
Counting up 1
T-minus 8
T-minus 3
Counting up 2
T-minus 7
T-minus 2
...
```

åĹřæ■d'äÿžæ■ćïijŇæĹŚäżňåŏđéŹĚäÿŁåůšçżŔåŏđçŐřäžĘäÿĂäÿłâĂIJæŞ■ä¡IJçşżçż§âĂİçŽĎæIJĂåřŔæă
çŤ§æĹŘåŹĺåĞ¡æŢřåřśæŸŕèŏd'äÿžïijŇèĂŇyieldèŕ■åŔěæŸŕäżżåŁąæŇĆèţůçŽĎä£ąåŔůãĂĆ
èřČåžęåŹĺå¿łçŐŕæčĂæ§ěäżżåŁąåĹŮèąÍçŻt'åĹřæšąæIJL'äżżåŁąèęĄæL'ğèąŇäÿžæ■ćãĂĆ

åŏđéŹĚäÿŁïijŇä¡ăåŘŕèČ¡æČşèęĄä¡£çŤÍçŤ§æĹŘåŹĺæİěåŏđçŐřçőĂå■ŢçŽĎåźűåŔŚãĂĆ
éĆčäźĹïijŇåIJÍåŏđçŐřactoræĹŰç¡ŚçżIJæIJ■åŁąåŹÍçŽĎæŮűåĂŹä¡ăåŘŕäżěä¡£çŤÍçŤ§æĹŘåŹĺæİěæŻ£äżčçž£ç

äÿŃéİćçŽĎäżčçăĄæijŤçd'žäžĘä¡£çŤÍçŤ§æĹŘåŹĺæİěåŏđçŐřäÿĂäÿłäÿ■ä¿ÌètŰçž£çÍŃçŽĎactorïijŽ

```
from collections import deque

class ActorScheduler:
    def __init__(self):
        self._actors = { }          # Mapping of names to actors
        self._msg_queue = deque()   # Message queue

    def new_actor(self, name, actor):
        '''
        Admit a newly started actor to the scheduler and give it a␣
↪name
        '''
        self._msg_queue.append((actor,None))
        self._actors[name] = actor

    def send(self, name, msg):
        '''
        Send a message to a named actor
        '''
        actor = self._actors.get(name)
        if actor:
            self._msg_queue.append((actor,msg))
```

```
    def run(self):
        '''
        Run as long as there are pending messages.
        '''
        while self._msg_queue:
            actor, msg = self._msg_queue.popleft()
            try:
                 actor.send(msg)
            except StopIteration:
                 pass

# Example use
if __name__ == '__main__':
    def printer():
        while True:
            msg = yield
            print('Got:', msg)

    def counter(sched):
        while True:
            # Receive the current count
            n = yield
            if n == 0:
                break
            # Send to the printer task
            sched.send('printer', n)
            # Send the next count to the counter task (recursive)

            sched.send('counter', n-1)

    sched = ActorScheduler()
    # Create the initial actors
    sched.new_actor('printer', printer())
    sched.new_actor('counter', counter(sched))

    # Send an initial message to the counter to initiate
    sched.send('counter', 10000)
    sched.run()
```

åőŇåĚĺåijĎæĞĆè£ŹæőţäżčçăĄéIJĂèęĄæŻťæůśåĚěçŽĎå■ęäźăïijŇä¡ĘæŸŕåĚşéŤőçĆźåIJĺäžŐæŤűéŻĘæ
æIJňèťÍäÿŁïijŇèřČåžęåŹĺåIJÍæIJL'éIJĂèęĄåŔŚéĂĄçŽĎæűĹæĄŕæŮűäijŽäÿĂçŻťè£ŘèąŇçİĂãĂĆ
èőąæŢřçŤ§æĹŘåŹĺäijŽçżŹèĞłåůśåŔŚéĂĄæűĹæĄŕåźűåIJĺäÿĂäÿłéĂŠå¡Šå¿łçŐŕäÿ■çżŞæİ§ãĂĆ

äÿŃéİćæŸŕäÿĂäÿłæŻťåŁăéńŸçžğçŽĎä¿Ńå■ŘïijŇæijŤçď žäžĘä¡£çŤĺçŤ§æĹŘåŹĺæİěåőđçŐŕäÿĂäÿłåźűå

```
from collections import deque
from select import select

# This class represents a generic yield event in the scheduler
class YieldEvent:
```

```
    def handle_yield(self, sched, task):
        pass
    def handle_resume(self, sched, task):
        pass

# Task Scheduler
class Scheduler:
    def __init__(self):
        self._numtasks = 0        # Total num of tasks
        self._ready = deque()     # Tasks ready to run
        self._read_waiting = {}   # Tasks waiting to read
        self._write_waiting = {} # Tasks waiting to write

    # Poll for I/O events and restart waiting tasks
    def _iopoll(self):
        rset,wset,eset = select(self._read_waiting,
                                self._write_waiting,[])
        for r in rset:
            evt, task = self._read_waiting.pop(r)
            evt.handle_resume(self, task)
        for w in wset:
            evt, task = self._write_waiting.pop(w)
            evt.handle_resume(self, task)

    def new(self,task):
        '''
        Add a newly started task to the scheduler
        '''

        self._ready.append((task, None))
        self._numtasks += 1

    def add_ready(self, task, msg=None):
        '''
        Append an already started task to the ready queue.
        msg is what to send into the task when it resumes.
        '''
        self._ready.append((task, msg))

    # Add a task to the reading set
    def _read_wait(self, fileno, evt, task):
        self._read_waiting[fileno] = (evt, task)

    # Add a task to the write set
    def _write_wait(self, fileno, evt, task):
        self._write_waiting[fileno] = (evt, task)

    def run(self):
        '''
        Run the task scheduler until there are no tasks
```

```
        '''
        while self._numtasks:
             if not self._ready:
                  self._iopoll()
             task, msg = self._ready.popleft()
             try:
                 # Run the coroutine to the next yield
                 r = task.send(msg)
                 if isinstance(r, YieldEvent):
                     r.handle_yield(self, task)
                 else:
                     raise RuntimeError('unrecognized yield event')
             except StopIteration:
                 self._numtasks -= 1

# Example implementation of coroutine-based socket I/O
class ReadSocket(YieldEvent):
    def __init__(self, sock, nbytes):
        self.sock = sock
        self.nbytes = nbytes
    def handle_yield(self, sched, task):
        sched._read_wait(self.sock.fileno(), self, task)
    def handle_resume(self, sched, task):
        data = self.sock.recv(self.nbytes)
        sched.add_ready(task, data)

class WriteSocket(YieldEvent):
    def __init__(self, sock, data):
        self.sock = sock
        self.data = data
    def handle_yield(self, sched, task):

        sched._write_wait(self.sock.fileno(), self, task)
    def handle_resume(self, sched, task):
        nsent = self.sock.send(self.data)
        sched.add_ready(task, nsent)

class AcceptSocket(YieldEvent):
    def __init__(self, sock):
        self.sock = sock
    def handle_yield(self, sched, task):
        sched._read_wait(self.sock.fileno(), self, task)
    def handle_resume(self, sched, task):
        r = self.sock.accept()
        sched.add_ready(task, r)

# Wrapper around a socket object for use with yield
class Socket(object):
    def __init__(self, sock):
        self._sock = sock
```

```
    def recv(self, maxbytes):
        return ReadSocket(self._sock, maxbytes)
    def send(self, data):
        return WriteSocket(self._sock, data)
    def accept(self):
        return AcceptSocket(self._sock)
    def __getattr__(self, name):
        return getattr(self._sock, name)

if __name__ == '__main__':
    from socket import socket, AF_INET, SOCK_STREAM
    import time

    # Example of a function involving generators.  This should
    # be called using line = yield from readline(sock)
    def readline(sock):
        chars = []
        while True:
            c = yield sock.recv(1)
            if not c:
                break
            chars.append(c)
            if c == b'\n':
                break
        return b''.join(chars)

    # Echo server using generators
    class EchoServer:
        def __init__(self,addr,sched):
            self.sched = sched
            sched.new(self.server_loop(addr))

        def server_loop(self,addr):
            s = Socket(socket(AF_INET,SOCK_STREAM))

            s.bind(addr)
            s.listen(5)
            while True:
                c,a = yield s.accept()
                print('Got connection from ', a)
                self.sched.new(self.client_handler(Socket(c)))

        def client_handler(self,client):
            while True:
                line = yield from readline(client)
                if not line:
                    break
                line = b'GOT:' + line
                while line:
                    nsent = yield client.send(line)
```

```
                    line = line[nsent:]
            client.close()
            print('Client closed')

    sched = Scheduler()
    EchoServer(('',16000),sched)
    sched.run()
```

è£ŹæőţäžčçăæIJL’çĆźåď■æİĆãĂĆäÿ■è£ĞïijŇåőČåőđçŐřäžĘäÿĂäÿłářŔåđŃçŽĎæŞ■ä¡IJçşžçż§ãĂĆ
æIJL’äÿĂäÿłářšçżłçŽĎäżżåŁąéŸ§åĹŮïijŇåźűäÿŤè£ŸæIJL’åŻăI/OäijŚçIJăçŽĎäżżåŁąç■L’å¿ĚåŇžå§§ãĂĆ
è£ŸæIJL’å¿Ĺåď’ŽèřČåžęåŹÍèť§èť čåIJÍářšçżłéŸ§åĹŮåŠŇI/Oç■L’å¿ĚåŇžå§§äźŃéŮť’çğżåŁÍäżżåŁąãĂĆ

èőÍèőž

åIJÍæđĎåżžå§žäžŐçŤ§æĹŘåŹÍçŽĎåźűåŔŚæąĘæđűæŮűïijŇéĂŽåÿÿäijŽä¡£çŤÍæŻť’åÿÿèğĄçŽĎyieldå¡ćå

```
def some_generator():
    ...
    result = yield data
    ...
```

ä¡£çŤÍè£Źçğ■å¡ćåijŔçŽĎyieldèŕ■åŔěçŽĎåĞ¡æŢřéĂŽåÿÿèćńçğřäÿžâĂIJå■ŔçÍŃâĂİãĂĆ
éĂŽè£ĞèřČåžęåŹÍïijŇyieldèŕ■åŔěåIJÍäÿĂäÿłå¿łçŐŕäÿ■èćńåď’ĎçŘĘïijŇåęĆäÿŃïijŽ

```
f = some_generator()

# Initial result. Is None to start since nothing has been computed
result = None
while True:
    try:
        data = f.send(result)
        result = ... do some calculation ...
    except StopIteration:
        break
```

è£ŹéĞŇçŽĎéĂżè¿ŚçÍ■å¿őæIJL’çĆźåď’■æİĆãĂĆäÿ■è£ĞïijŇèćńäijăçżŹ
`send()` çŽĎåĂijåőŽäźL’äžĘåIJÍyieldèŕ■åŔěéĘŠæİěæŮűçŽĎè£ŤåŻđåĂijãĂĆ
åŻăæ■d’ïijŇåęĆæđIJäÿĂäÿłyieldåĞĘåď’ĞåIJÍářźäźŃåL’■yield-
æŢřæ■őçŽĎåŻđåžŤäÿ■è£ŤåŻđçżŞæđIJæŮűïijŇäijŽåIJÍäÿŃäÿĂæňą `send()`
æŞ■ä¡IJè£ŤåŻđãĂĆ åęĆæđIJäÿĂäÿłçŤ§æĹŘåŹÍåĞ¡æŢřåĹŽåijĂåğŃè£ŘèąŇïijŇåŔŚéĂĄäÿĂäÿłNoneåĂijäij

éŹd’äžĘåŔŚéĂĄåĂijåd’ŰïijŇè£ŸåŔřäžěåIJÍäÿĂäÿłçŤ§æĹŘåŹÍäÿŁéİćæL’ğèąŇäÿĂäÿł
`close()` æŰźæşŢãĂĆ åőČäijŽářijèĞť’åIJÍæL’ğèąŇyieldèŕ■åŔěæŮűæŁŽåĞžäÿĂäÿł
`GeneratorExit` åijĆåÿÿïijŇäżŐèĂŇçžĹæ■ćæL’ğèąŇãĂĆ
åęĆæđIJè£ŽäÿĂæ■ěèő¿èőąïijŇäÿĂäÿłçŤ§æĹŘåŹÍåŔřäžěæ■ŢèŐůè£ŹäÿłåijĆåÿÿåźűæL’ğèąŇæÿĚçŘĘæŞ■ä¡I
åŘŇæăůè£ŸåŔřäžěä¡£çŤÍçŤ§æĹŘåŹÍçŽĎ `throw()` æŰźæşŢåIJÍyield-
èŕ■åŔěæL’ğèąŇæŮűçŤ§æĹŘäÿĂäÿłäżżæĎŔçŽĎæL’ğèąŇæŇĞäžd’ãĂĆ
äÿĂäÿłäżżåŁąěřČåžęåŹÍåŔřåĹl’çŤÍåőČæİěåIJÍè£ŘèąŇçŽĎçŤ§æĹŘåŹÍäÿ■åď’ĎçŘĘéŤŹèŕřãĂĆ

æIJĂåŘŐäÿĂäÿłä¿Ńå■Řäÿ■ä¡£çŤĺçŽĎ `yield from` èŕ■åŔěèćńçŤĺæİěåőđçŐřå■ŔçĺŃïijŇåŔŕäžěèćńåĚ
æIJńèťĺäÿŁåřśæŸŕåřĘæŐğåĹűæİČéĂŔæŸŐçŽĎäijåè¿ŞçżŹæŰřçŽĎåĞ¡æŢřãĂĆ
äÿ■åČŔæŹőéĂŽçŽĎçŤ§æĹŘåŹĺïijŇäÿĂäÿłä¡£çŤĺ `yield from`
èćńèřČçŤĺçŽĎåĞ¡æŢřåŔŕäžěè£ŤåŻđäÿĂäÿłä¡IJäÿž `yield from`
èŕ■åŔěçżŞæđIJçŽĎåĂijãĂĆ åĚşäžŐ `yield from` çŽĎæŻťåď̌Žä£ąæĄŕåŔŕäžěåIJĺ PEP
380 äÿ■æĽ¿åĹřãĂĆ

æIJĂåŘŐïijŇåęĆæđIJä¡£çŤĺçŤ§æĹŘåŹĺçijŰçĺŃïijŇèęĄæŔŘéĘšä¡ăçŽĎæŸŕåőČè£ŸæŸŕæIJĽå¿Łåď̌Žç
çĽźåĹńæŸŕïijŇä¡ăå¿Ůäÿ■åĹřäžżä¡Ţçž£çĺŃåŔŕäžěæŔŘä¿ŻçŽĎåě¡åď̌ĎãĂĆä¿ŃåęĆïijŇåęĆæđIJä¡ăæĽğèąŇĆ
åőČäijŽåřĘæŢťäÿłäżżåŁąæŇĆèţůç§ěéĄŞæŞ■ä¡IJåőŇæĹŘãĂĆäÿžäžĘèğčåĘşè£ŹäÿłéŮőéćŸïijŇ
ä¡ăåŔłèČ¡éĂĽæŃĺ̌åřĘæŞ■ä¡IJåğŤæť¿çżŹåŘęåď̌ŰäÿĂäÿłåŔŕäžěçŃňçńŃè£ŘèąŇçŽĎçž£çĺŃæĹŰè£ŻçĺŃãĂ
åŔęåď̌ŰäÿĂäÿłéŹŘåĹűæŸŕåď̌ğéČĺåĹĘPythonåžŞåźűäÿ■èČ¡å¿Łåě¡çŽĎåĚijåőźå§žäžŐçŤ§æĹŘåŹĺçŽĎçž£çĺ
åęĆæđIJä¡ăéĂĽæŃĺ̌è£ŹäÿłæŰźæąĹïijŇä¡ăäijŽåŔŚçŐŕä¡ăéIJĂèęĄèĞłåůśæŤźåĘŹå¿Łåď̌ŽæăĞåĞĘåžŞåĞ¡æ
ä¡IJäÿžæIJńèŁĆæŔŘåĹřçŽĎå■ŔçĺŃåŠŇçŻÿåĚşæŁĂæIJŕçŽĎäÿĂäÿłå§žçąĂèČŇæŹřïijŇåŔŕäžěæ§ěçIJŃ
PEP 342 åŠŇ âĂIJå■ŔçĺŃåŠŇåźűåŔŚçŽĎäÿĂéŮĺæIJĽèűčèŕ¿çĺŃâĂİ

PEP 3156 åŘŇæăůæIJĽäÿĂäÿłåĚşäžŐä¡£çŤĺå■ŔçĺŃçŽĎåijĆæ■ěI/OæĺąåđŃãĂĆ
çĽźåĹńçŽĎïijŇä¡ăäÿ■åŔŕèČ¡èĞłåůśåŐžåőđçŐřäÿĂäÿłåžŢåśĆçŽĎå■ŔçĺŃèřČåžęåŹĺãĂĆ
äÿ■è£ĞïijŇåĚşäžŐå■ŔçĺŃçŽĎæĂİæČşæŸŕå¿Łåď̌ŽæţĄèąŇåžŞçŽĎå§žçąĂïijŇ åŇĚæŃň
gevent, greenlet, Stackless Python äžěåŔŁåĚűäžŰçśżäijijåůěçĺŃãĂĆ

14.13 12.13 åď̌Žäÿłçž£çĺŃéŸ§åĹŮè¡őèŕć

éŮőécŸ

ä¡ăæIJĽäÿĂäÿłçž£çĺŃéŸ§åĹŮéŻĘåŘĹïijŇæČşäÿžåĹřæİěçŽĎåĚČçť ăè¡őèŕćåőČäžňïijŇ
åŕśèů§ä¡ăäÿžäÿĂäÿłåőćæĹůçńŕèŕůæśĆåŐżè¡őèŕćäÿĂäÿłç¡ŞçżIJè£đæŐěéŻĘåŘĹçŽĎæŰźåijŔäÿĂæăůãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐè¡őèŕćéŮőécŸçŽĎäÿĂäÿłåÿÿèğĄèğčåĘşæŰźæąĹäÿ■æIJĽäÿłå¿ĹåřŚæIJĽäžžç§ěéĄŞçŽĎæŁĂåů
æIJńèťĺäÿŁèőšåĚűæĂİæČşåŕśæŸŕïijŽåŕźäžŐæŕŔäÿłä¡ăæČşèęĄè¡őèŕćçŽĎéŸ§åĹŮïijŇä¡ăåĹŻåžžäÿĂåŕźè£đæ
çĎűåŘŐä¡ăåIJĺåĚűäÿ■äÿĂäÿłåěŮæŐěå■ŮäÿŁéİćçijŰåĘŹäžčçăĄæİěæăĞèŕĘå■ŸåIJĺçŽĎæŢřæ■őïijŇ
åŔęåď̌ŰäÿĂäÿłåěŮæŐěå■Ůèćńäijăçžź `select()` æĹŰçśżäijijçŽĎäÿĂäÿłè¡őèŕćæŢřæ■őåĹřè¿¿çŽĎåĞ¡æŢř

```
import queue
import socket
import os

class PollableQueue(queue.Queue):
    def __init__(self):
        super().__init__()
        # Create a pair of connected sockets
        if os.name == 'posix':
            self._putsocket, self._getsocket = socket.socketpair()
        else:
            # Compatibility on non-POSIX systems
            server = socket.socket(socket.AF_INET, socket.SOCK_
↪STREAM)
```

```
            server.bind(('127.0.0.1', 0))
            server.listen(1)
            self._putsocket = socket.socket(socket.AF_INET, socket.
↪SOCK_STREAM)
            self._putsocket.connect(server.getsockname())
            self._getsocket, _ = server.accept()
            server.close()

    def fileno(self):
        return self._getsocket.fileno()

    def put(self, item):
        super().put(item)
        self._putsocket.send(b'x')

    def get(self):
        self._getsocket.recv(1)
        return super().get()
```

åIJÍè£ŹäÿłäżčçăĄäÿ■ïijŇäÿĂäÿłæŰřçŽĎ Queue åőđä¿ŃçśżåđŃèćńåőŽäźĽïijŇåžŢåśĆæŸŕäÿĂäÿłèćńè£
åIJÍUnixæIJžåŹÍäÿŁçŽĎ socketpair() åĞ¡æŢřèČ¡è¡żæİ¿çŽĎåĹŻåżžè£ŹæăůçŽĎåěŮæŐěå■ŮãĂĆ
åIJÍWindowsäÿŁéİćïijŇä¡ăå£Ěéążä¡£çŤÍçśżäijijäżčçăĄæİěæÍąæŃ§åőČãĂĆ
çĎűåŘŐåőŽäźĽæŹőéĂŽçŽĎ get() åŠŇ put() æŰźæşŢåIJÍè£ŹäžŻåěŮæŐěå■ŮäÿŁéİćæİěæĽğèąŇI/OæŞ
put() æŰźæşŢåĘ■åřĘæŢřæ■őæŤ¿åĚěéŸ§åĹŮåŘŐäijŽåĘŹäÿĂäÿłå■Ţå■ŮèŁĆåĹřæ§ŘäÿłåěŮæŐěå■Ůäÿ■å
èĂŇ get() æŰźæşŢåIJÍäżŐéŸ§åĹŮäÿ■çğżéŹďäÿĂäÿłåĚČçťăæŮűäijŽäżŐåŔęåďŰäÿĂäÿłåěŮæŐěå■Ůäÿ■

fileno() æŰźæşŢä¡£çŤÍäÿĂäÿłåĞ¡æŢřæŕŤåęĆ select() æİěèőľè£ŹäÿłéŸ§åĹŮåŔŕäżěèćńè¡őèŕćãĂĆ åőČäżĚäżĚåŔłæŸŕæŽťéIJšäžĘåžŢåśĆèćń get() åĞ¡æŢřä¡£çŤÍåĹřçŽĎsocketçŽĎæŰĞäżűæŔŔè£řçňęèĂŇåůšãĂĆ

äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇåőŽäźĽäžĘäÿĂäÿłäÿžåĹřæİěçŽĎåĚČçťăçŻŚæŐğåďŽäÿłéŸ§åĹŮçŽĎæű

```
import select
import threading

def consumer(queues):
    '''
    Consumer that reads data on multiple queues simultaneously
    '''
    while True:
        can_read, _, _ = select.select(queues,[],[])
        for r in can_read:
            item = r.get()
            print('Got:', item)

q1 = PollableQueue()
q2 = PollableQueue()
q3 = PollableQueue()
t = threading.Thread(target=consumer, args=([q1,q2,q3],))
t.daemon = True
t.start()
```

```
# Feed data to the queues
q1.put(1)
q2.put(10)
q3.put('hello')
q2.put(15)
...
```

åęĆæđIJä¡ăèŕŢçİÄè£ŘèąŇåőČïijŇä¡ăäijŽåŔŠçŐřè£ŹäÿłæűĹèť źèĂĚäijŽæŐěåŔŮåĹřæL’ĂæIJL’çŽĎèćń

èőĺèőž

åŕźäžŐè¡őèŕćéİđçśżæŰĞäżűåŕźèśąïijŇæŕŤåęĆéŸ§åĹŮéĂŽåÿÿéČ¡æŸŕæŕŤè¿ČæčŸæL’ŃçŽĎéŮőécŸãĂ
ä¿ŇåęĆïijŇåęĆæđIJä¡ăäÿ■ä¡£çŤĺäÿŁéİćçŽĎåěŮæŐěå■ŮæŁĂæIJŕïijŇ
ä¡ăåŤŕäÿĂçŽĎéĂL’æŃl’åřśæŸŕçijŰåĘŹäžčçăĄæİěå¿łçŐŕéĄ■åŐĘè£ŹäžŻéŸ§åĹŮåźűä¡£çŤĺäÿĂäÿłåőŽæŮűå

```
import time
def consumer(queues):
    while True:
        for q in queues:
            if not q.empty():
                item = q.get()
                print('Got:', item)

        # Sleep briefly to avoid 100% CPU
        time.sleep(0.01)
```

è£ŹæăůåĄŽåĚűåőđäÿ■åŘĹçŘĘïijŇè£ŸäijŽåijŢåĚěåĚűäżŰçŽĎæĂğèČ¡éŮőécŸãĂĆ
ä¿ŇåęĆïijŇåęĆæđIJæŰřçŽĎæŢřæ■őèćńåŁăåĚěåĹřäÿĂäÿłéŸ§åĹŮäÿ■ïijŇèĞşåřŚèęĄèŁś10æńçğŠæL’■èČ¡è
åęĆæđIJä¡ăäźŃåL’■çŽĎè¡őèŕćè£ŸèęĄåŐżè¡őèŕćåĚűäżŰåŕźèśąïijŇæŕŤåęĆç¡ŚçżIJåěŮæŐěå■ŮéĆčè£ŸäijŽæ
ä¿ŇåęĆïijŇåęĆæđIJä¡ăæČşåŘŇæŮűè¡őèŕćåěŮæŐěå■ŮåŠŇéŸ§åĹŰïijŇä¡ăåŔŕèČ¡èęĄåČŔäÿŃéİćè£Źæăůä¡£

```
import select

def event_loop(sockets, queues):
    while True:
        # polling with a timeout
        can_read, _, _ = select.select(sockets, [], [], 0.01)
        for r in can_read:
            handle_read(r)
        for q in queues:
            if not q.empty():
                item = q.get()
                print('Got:', item)
```

è£ŹäÿłæŰźæąĹéĂŽè£ĞåřĘéŸ§åĹŮåŠŇåěŮæŐěå■Ůç■L’åŘŇåŕźå¿ĚæİěèğčåĘşäžĘåd’ğéČĺåĹĘçŽĎéŮőe
äÿĂäÿłå■ŢçŃňçŽĎ `select()` èřČçŤĺåŔŕèćńåŘŇæŮűçŤĺæİěè¡őèŕćĂĆ
ä¡£çŤĺèűĚæŮűæĹŮåĚűäžŰå§žäžŐæŮűéŮt’çŽĎæIJžåĹűæİěæL’ğèąŇåŠĺæIJ§æĂğæčĂæ§ěåźűæšąæIJL’å£Ěèę

çŤŽèĞşïijŇåęĆæđIJæŢřæ■őèćńåŁääĚěåĹřäÿĂäÿłéŸ§åĹŮïijŇæűĹèť źèĂĚåĞääźŐåŔŕäżěåőđæŮűçŽĎèćńéĂ
ář¡çőąäijŽæIJL’äÿĂçĆźçĆźåžŢåśĆçŽĎI/Oæ■§èĂŮïijŇä¡£çŤĺåőČéĂŽåÿÿäijŽèŐůå¿ŮæŻť’åě¡çŽĎåŞ■åžŤæŮ

14.14 12.14 åIJÍUnixçşżçż§äÿŁéİćåŘŕåŁĺåőĹæŁd’è£ŻçÍŃ

éŮőécŸ

ä¡ăæČşçijŰåĘŹäÿĂäÿłä¡IJäÿžäÿĂäÿłåIJÍUnixæĹŰçśżUnixçşżçż§äÿŁéİćè£ŘèąŇçŽĎåőĹæŁd’è£ŻçÍŃè£

èğčåĘşæŰźæąĹ

åĹŻåżžäÿĂäÿłæ■čçąőçŽĎåőĹæŁd’è£ŻçÍŃéIJĂèęĄäÿĂäÿłçš¿çąőçŽĎçşżçż§èřČçŤĺåžŔåĹŮäżěåŔŁåŕźäž
äÿŃéİćçŽĎäžčçăĄåśŢçd’žäžĘæĂŐæăůåőŽäźĽäÿĂäÿłåőĹæŁd’è£ŻçÍŃïijŇåŔŕäżěåŘŕåŁĺåŘŐå¿Ĺåőźæ ŸŞçŽĎ

```
#!/usr/bin/env python3
# daemon.py

import os
import sys

import atexit
import signal

def daemonize(pidfile, *, stdin='/dev/null',
                          stdout='/dev/null',
                          stderr='/dev/null'):

    if os.path.exists(pidfile):
        raise RuntimeError('Already running')

    # First fork (detaches from parent)
    try:
        if os.fork() > 0:
            raise SystemExit(0)    # Parent exit
    except OSError as e:
        raise RuntimeError('fork #1 failed.')

    os.chdir('/')
    os.umask(0)
    os.setsid()
    # Second fork (relinquish session leadership)
    try:
        if os.fork() > 0:
            raise SystemExit(0)
    except OSError as e:
        raise RuntimeError('fork #2 failed.')

    # Flush I/O buffers
```

```
    sys.stdout.flush()
    sys.stderr.flush()

    # Replace file descriptors for stdin, stdout, and stderr
    with open(stdin, 'rb', 0) as f:
        os.dup2(f.fileno(), sys.stdin.fileno())
    with open(stdout, 'ab', 0) as f:
        os.dup2(f.fileno(), sys.stdout.fileno())
    with open(stderr, 'ab', 0) as f:
        os.dup2(f.fileno(), sys.stderr.fileno())

    # Write the PID file
    with open(pidfile,'w') as f:
        print(os.getpid(),file=f)

    # Arrange to have the PID file removed on exit/signal
    atexit.register(lambda: os.remove(pidfile))

    # Signal handler for termination (required)
    def sigterm_handler(signo, frame):
        raise SystemExit(1)

    signal.signal(signal.SIGTERM, sigterm_handler)

def main():
    import time
    sys.stdout.write('Daemon started with pid {}\n'.format(os.
↪getpid()))
    while True:
        sys.stdout.write('Daemon Alive! {}\n'.format(time.ctime()))
        time.sleep(10)

if __name__ == '__main__':
    PIDFILE = '/tmp/daemon.pid'

    if len(sys.argv) != 2:
        print('Usage: {} [start|stop]'.format(sys.argv[0]),␣
↪file=sys.stderr)
        raise SystemExit(1)

    if sys.argv[1] == 'start':
        try:
            daemonize(PIDFILE,
                      stdout='/tmp/daemon.log',
                      stderr='/tmp/dameon.log')
        except RuntimeError as e:
            print(e, file=sys.stderr)
            raise SystemExit(1)

        main()
```

```
    elif sys.argv[1] == 'stop':
        if os.path.exists(PIDFILE):
            with open(PIDFILE) as f:
                os.kill(int(f.read()), signal.SIGTERM)
        else:
            print('Not running', file=sys.stderr)
            raise SystemExit(1)

    else:
        print('Unknown command {!r}'.format(sys.argv[1]), file=sys.
↪stderr)
        raise SystemExit(1)
```

èęĄåŘŕåŁĺè£ŹäÿłåőĹæŁd'è£ŻçĺŃïijŇçŤĺæĹůéIJĂèęĄä¡£çŤĺąęĆäÿŃçŽĎåŠ¡äžd'ïijŽ

```
bash % daemon.py start
bash % cat /tmp/daemon.pid
2882
bash % tail -f /tmp/daemon.log
Daemon started with pid 2882
Daemon Alive! Fri Oct 12 13:45:37 2012
Daemon Alive! Fri Oct 12 13:45:47 2012
...
```

åőĹæŁd'è£ŻçĺŃåŔŕäžěåőŇåĚĺåIJĺåŘŐåŔřè£ŘèąŇïijŇåŻäæ■d'è£ŹäÿłåŠ¡äžd'äijŽçńŇå■şè£ŤåŻđãĂĆ
äÿ■è£ĞïijŇä¡ăåŔŕäžěåČŔäÿŁéİćéĆčæăůæ§ěçIJŃäÿŐåőĆçŻÿåĚşçŽĎpidæŰĞäžűåŠŇæŮěå£ŮãĂĆèęĄåĄIJæ

```
bash % daemon.py stop
bash %
```

èőĺèőž

æIJňèŁĆåőŽäźL'äžĘäÿĂäÿłåĞ¡æŢř `daemonize()` ïijŇåIJĺçĺŃåžŔåŘŕåŁĺæŮűèćńèřČçŤĺä¡£å¿ŮçĺŃåžŔ
`daemonize()` åĞ¡æŢřåŔłæŐěåŔŮåĚşéŤőå■ŮåŔĆæŢřïijŇè£ŹæăůçŽĎèŕĺåŔŕéĂĽ'åŔĆæŢřåIJĺèćńä¡£çŤĺæŮ
åőČäijŽåijžåĹűçŤĺæĹůåČŔäÿŃéİćè£ŹæăůäŔ¡£çŤĺåőČïijŽ

```
daemonize('daemon.pid',
          stdin='/dev/null,
          stdout='/tmp/daemon.log',
          stderr='/tmp/daemon.log')
```

èĂŇäÿ■æŸŕåČŔäÿŃéİćè£ŹæăůåŘńçşŁäÿ■æÿĚçŽĎèřČçŤĺïijŽ

```
# Illegal. Must use keyword arguments
daemonize('daemon.pid',
          '/dev/null', '/tmp/daemon.log','/tmp/daemon.log')
```

åĹŻåžžäÿĂäÿłåőĹæŁd'è£ŻçĺŃçŽĎæ■ěéłd'çIJŃäÿŁåŐżäÿ■æŸŕå¿ĹæŸŞæĞĆïijŇä¡ĘæŸŕåd'ğä¡ŞæĂİæČ

éęŰåĚĹïijŇäÿĂäÿłåőĹæŁď'è£ŻçÍŃå£Ěéążèę Ąäż ŐçĹűè£ŻçÍŃäÿ■èĎśçężãĂĆ è£ŹæŸŕçŤś
`os.fork()` æŞ■ä¡IJæİěåőŇæĹŘçŽĎïijŇåźűçńŃå■şèćńçĹűè£ŻçÍŃçżĹæ■ćãĂĆ

åIJÍå■Řè£ŻçÍŃåŔŸæĹŘå■ď'åĎ£åŘŐïijŇèřČçŤÍ `os.setsid()`
åĹŻåżžäžĘäÿĂäÿłåĚÍæŰřçŽĎè£ŻçÍŃäijŻèŕİïijŇåźűèő¿ç¡őå■Řè£ŻçÍŃäÿžéęŰéćĘãĂĆ
åőČäijŽèő¿ç¡őè£Źäÿłå■Řè£ŻçÍŃäÿżæŰřçŽĎè£ŻçÍŃçżĎçŽĎéęŰéćĘïijŇåźűçąőä£İäÿ■äijŽåĘ■æIJL'æŐğåĹűç
åęĆæđIJè£ŹäžŻåŘňäÿŁåŐžåď'łé■Ťåźźïij ŇåŻääÿžåőČéIJĂèęĄåřĘåőĹæŁď'è£ŻçÍŃåŘŇçżĹçńŕåĹĘçęžåijĂåźű
èřČçŤÍ `os.chdir()` åŠŇ `os.umask(0)` æŤźåŔŸäžĘå¡ŞåL'■åůëä¡IJçŽőå¡Ţåźűé Ğ■ç¡őæŰĞäżűæİČéŹŘæ
ä£őæŤźçŽőå¡ŢéĂŽåÿÿæŸŕäÿłåě¡äÿżæĎŔïijŇåŻääÿžè£Źæăůå Ŕŕäžëä¡£å¿ŮåőČäÿ■åĘ■åůëä¡IJåIJÍècńåŘŕåŁÍæ

åŘęåď'ŰäÿĂäÿłèřČçŤÍ `os.fork()` åIJÍè£ŹéĞŇæŻť'åŁăçěđçğŸçĆźãĂĆ
è£ŹäÿĂæ■ëä¡£å¿ŮåőĹæŁď'è£ŻçÍŃåď'śåŐżäžĘèŐůåŔŰæŰřçŽĎæŐğåĹűçżĹçńŕçŽĎèČ¡åŁŻåźűäÿŤèőľåőČæ
ïijĹæIJňèť'ĺäÿŁïijŇèŕědaemonæŤ¿åijČäžĘåőČçŽĎäijŽèŕİéęŰéćĘä¡Öä¡■ïijŇåŻăæ■ď'åĘ■äź§æšąæIJL'æİČéŹŘ
år¡çőąä¡ääŔŕäžěå£¡çŢèè£ŹäÿĂæ■ëïijŇä¡ĘæŸŕæIJĂåě¡äÿ■èęĄè£ŹäźĹåĄŽãĂĆ

äÿĂæŮęåőĹæŁď'è£ŻçÍŃècńæ■čçąőçŽĎåĹĘçężïijŇåőČäijŽéĞ■æŰřåĹİåğŃåŇŰæăĞåĞĘI/OæţĄæŇĞåŘ
è£ŹäÿĂéČĺåĹĘæIJL'çĆźéŽ¿æĞĆãĂĆèů§æăĞåĞĘI/OæţĄçŻÿåĚşçŽĎæŰĞäżűåŕźèšąçŽĎäijŢçŤĺåIJÍèğčéĞŁå
ïijĹsys.stdout, sys.__stdout__ç■L'ïijL'ãĂĆ äžĚäžĚçőĂå■ŢçŽĎåĚşéŮ■
`sys.stdout` åźűéĞ■æŰřæŇĞåőŽåőČæŸŕèąŇäÿ■éĂŽçŽĎïijŇ
åŽääÿžæšąåŁđæşŢç§ěéĄŞåőČæŸŕåŘęåĚĺéČĺéČ¡æŸŕçŤÍçŽĎæŸŕ `sys.stdout` ãĂĆ
è£ŹéĞŇïijŇæĹŚäžňæL'ŞäijĂäžĘäÿĂäÿłå■ŢçŃňçŽĎæŰĞäżűåŕźèšąïijŇåźűěřČçŤÍ `os.`
`dup2()` ïijŇ çŤĺåőČæİěäžčæŻ£èćń `sys.stdout` ä¡£çŤÍçŽĎæŰĞäżűæŔŘè£řçňęãĂĆ
è£Źæăůïij Ň`sys.stdout` ä¡£çŤÍçŽĎåŐ§åĞŇæŰĞäżűäijŽèćńåĚşéŮ■åźűçŤśæŰřçŽĎæİèæŻ£æ■ćãĂĆ
è£ŸèęĄäijžěřČçŽĎæŸŕäżżä¡ŢçŤĺäžŐæŰĞäżűçijŰçăĄæĹŰæŰĞæIJňåď'ĎçŘĘçŽĎæăĞåĞĘI/OæţĄè£ŸäijŽä

åőĹæŁď'è£ŻçÍŃçŽĎäÿĂäÿłéĂŽåÿÿåőđèůţæŸŕåIJÍäÿĂäÿłæŰĞäżűäÿ■åĘŹåĚěè£ŻçÍŃIDïijŇåŔŕäžěèćńåĚ
`daemonize()` åĞ¡æŢřçŽĎæIJĂåŘŐéČĺåĹĘåĘŹäžĘè£ŹäÿłæŰĞäżűïijŇä¡ĘæŸŕåIJÍçÍŃäžŔçżĹæ■ćæŮűåĹăe
`atexit.register()` åĞ¡æŢřæşĺåĘŇäžĘäÿĂäÿłåĞ¡æŢřåIJÍPythonèğčéĞŁåŹÍçżĹæ■ćæŮűæL'ğèąŇãĂĆ
äÿĂäÿłårźäžŐSIGTERMçŽĎä£ąåŘůåď'ĎçŘĘåŹÍçŽĎåőŽäźL'åŘŇæăůéIJĂèęĄèćńäijŸéŻĚçŽĎåĚşĹ■ãĂĆ
ä£ąåŘůåď'ĎçŘĘåŹÍçőĂå■ŢçŽĎæŁŻåĞžäžĘ `SystemExit()` åijĆåÿÿãĂĆ
æĹŰèőÿè£ŹäÿĂæ■ęçIJŇäÿŁåŐżæšąå£ĚèęĄïijŇä¡ĘæŸŕæšąæIJL'åőČïijŇ
çżĹæ■ćä£ąåŘůäijŽä¡£å¿ŮäÿĂæL'ğèąŇ `atexit.register()`
æşĺåĘŇçŽĎæÿĚçŘĘæŞ■ä¡IJçŽĎæŰůåĂŹåřśæİĂæŐL'äžĘèğčéĞŁåŹÍãĂĆ
äÿĂäÿłæİĂæŐL'è£ŻçÍŃçŽĎä¿Ńå■ŘäžĉçăĄåŔŕäžěåIJÍçÍŃåžŔæIJĂåŘŐçŽĎ `stop`
åŚ¡äžď'çŽĎæŞ■ä¡IJäÿ■çIJŇåĹřãĂĆ

æŻť'åď'ŽåĚşäžŐçijŰåĘŹåőĹæŁď'è£ŻçÍŃçŽĎä£ąæĄŕåŔŕäžěæ§ěçIJŃãĂŁUNIX
çŐŕåćČéńŸçžğçijŰçÍŃãĂŃ, çňňäžŇçL'Ĺ by W. Richard
Stevens and Stephen A. Rago (Addison-Wesley, 2005)ãĂĆ
år¡çőąåőČæŸŕåĚşæşĺäÿŐCèŕ■èÍĂçijŰçÍŃïijŇä¡ĘæŸŕæL'ĂæIJL'çŽĎåĘĚåőźéČ¡éĂĆçŤÍäžŐPythonïijŇ
åŽääÿžæL'ĂæIJL'éIJĂèęĄçŽĎPOSIXåĞ¡æŢřéČ¡åŔŕäžěåIJÍæăĞåĞĘåžŞäÿ■æL'¿åĹřãĂĆ

15 çňňå■ĄäÿL'çńäïijŽèĎŽæIJňçijŰçÍŃäÿŐçşżçż§çőąçŘĘ

èőÿåď'Žäžžä¡£çŤÍPythonä¡IJäÿžäÿĂäÿłshellèĎŽæIJňçŽĎæŻ£äžčïijŇçŤÍæİěåőđçŐŕåÿÿçŤÍçşżçż§äżżåĹą

Contents:

15.1 13.1 éĂŽè£ĞéĞ■åőŽåŘŚ/çőąéĄŞ/æŰĞäžűæŐěåŔŮè¿ŞåĚě

éŮőécŸ

ä¡ăåÿŇæIJŻä¡ăçŽĎèĎŽæIJňæŐěåŔŮäżżä¡ŢçŤĺæĹůèőď'äÿžæIJĂçőĂå■ŢçŽĎè¿ŞåĚěæŰźåijŔãĂĆåŇĚæ
éĞ■åőŽåŘŚæŰĞäžűåĹřèŕěèĎŽæIJňïijŇæĹŰåIJĺåŚ¡äžď'èąŇäÿ■äijăéĂŠäÿĂäÿłæŰĞäžűåŘ■æĹŰæŰĞäžűåŘ■

èğčåĘşæŰźæąĹ

PythonåĘĚç¡őçŽĎ `fileinput` æĺąåİŮèől'è£ŹäÿłåŔŸå¿ŮçőĂå■ŢãĂĆåęĆæđIJä¡ăæIJL'äÿĂäÿłäÿŃéİćè

```
#!/usr/bin/env python3
import fileinput

with fileinput.input() as f_input:
    for line in f_input:
        print(line, end='')
```

éĆčäźĹä¡ăåřśèČ¡äžěåL'■éİćæŔŘåĹřçŽĎæL'ĂæIJL'æŰźåijŔæİěäÿžæ■ď'èĎŽæIJňæŔŘä¿Żè¿ŞåĚěãĂĆåĄ
`filein.py` åźűåřĘåĚűåŔŸäÿžåŔŕæL'ğèąŇæŰĞäžűïijŇ éĆčäźĹä¡ăåŔŕäžěåČŔäÿŃéİćè£ŹæăůèřČçŤĺåőČïijŇ

```
$ ls | ./filein.py          # Prints a directory listing to stdout.
$ ./filein.py /etc/passwd   # Reads /etc/passwd to stdout.
$ ./filein.py < /etc/passwd # Reads /etc/passwd to stdout.
```

èőÍèőž

`fileinput.input()` åĹŻåžžåźűè£ŤåŻđäÿĂäÿł `FileInput` çśžçŽĎåőđä¿ŃãĂĆ
èŕěåőđä¿ŃéŹď'äžĘæŃěæIJL'äÿĂäžŻæIJL'çŤÍçŽĎåÿőåŁl'æŰźæşŢåď'ŰïijŇåőČè£ŸåŔŕèćńå¡ŞåĄŽäÿĂäÿłäÿŁă
åŻăæ■ď'ïijŇæŢť'åŘĹèţůæİěïijŇåęĆæđIJæĹŚäžňèęĄåĘŹäÿĂäÿłæL'Şå■řåď'Žäÿłæ ŰĞäžűè¿ŞåĞžçŽĎèĎŽæIJň

```
>>> import fileinput
>>> with fileinput.input('/etc/passwd') as f:
>>>     for line in f:
...         print(f.filename(), f.lineno(), line, end='')
...
/etc/passwd 1 ##
/etc/passwd 2 # User Database
/etc/passwd 3 #

<other output omitted>
```

éĂŽè£ĞářĘåőČä¡IJäÿžäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺä¡£çŤĺïijŇåŔŕäžěçąőä£ĺåőČäÿ■åĘ■ä¡£çŤĺæŮűæÚ
èĂŇäÿŤæĹŚäžňåIJĺäźŃåŘŐè£ŸæijŤçď'žäžĘ `FileInput` çŽĎäÿĂäžŻæIJL'çŤÍçŽĎåÿőåŁl'æŰźæşŢæİěèŐů

15.2 13.2 çžĹæ■ćçÍŃåžŔåźűçżŹåĞžéŤŹèŕŕä£ąæĄŕ

éŮőécŸ

ä¡ăæČşåŘŚæăĞåĞĘéŤŹèŕŕæĽŞå■řäÿĂæİąæűĹæĄŕåźűè£ŤåŻđæ§ŘäÿłéİđéŽűçŁűæĂĄçăĄæİěçżĹæ■ćçÍ

èğčåĘşæŰźæąĹ

ä¡ăæIJĽäÿĂäÿłçÍŃåžŔåČŔäÿŃéİćè£ŹæăůçżĹæ■ćïijŇæŁŻåĞžäÿĂäÿł `SystemExit` åijĆåÿÿïijŇä¡£çŤÍéŤŹèŕŕæűĹæĄŕä¡IJäÿžåŔĆæŢřãĂĆä¿ŃåęĆïijŽ

```
raise SystemExit('It failed!')
```

åőČäijŽářĘæűĹæĄŕåIJÍ `sys.stderr` äÿ■æĽŞå■řïijŇçĎűåŘŐçÍŃåžŔäžěçŁűæĂĄçăĄ1éĂĂåĞžãĂĆ

èőÍèőž

æIJňèŁĆèŹ¡çĎűå¿Łç§■ářŔïijŇä¡ĘæŸŕåőČèČ¡èğčåĘşåIJÍåĘŹèĎŽæIJňæŮűçŽĎäÿĂäÿłåÿÿèğĄéŮőécŸãäź§ářśæŸŕèŕťïijŇå¡Şä¡ăæČşèęĄçżĹæ■ćæ§ŘäÿłçÍŃåžŔæŮűïijŇä¡ăåŔŕèČ¡äijŽåČŔäÿŃéİćè£ŹæăůåĘŹïijŽ

```
import sys
sys.stderr.write('It failed!\n')
raise SystemExit(1)
```

åęĆædIJä¡ăçŻť'æŐěářĘæűĹæĄŕä¡IJäÿžåŔĆæŢřäijăçżŹ `SystemExit()` ïijŇéĆčäźĹä¡ăåŔŕäžěçIJAçŢěåĚűäžŰæ■ěélď'ïijŇ æŕŤåęĆimportèŕ■åŘěæĹŰářĘéŤŹèŕŕæűĹæĄŕåĘŹåĚě `sys.stderr`

15.3 13.3 èğčædŘåŚ¡äžď'èąŇéĂĽéąź

éŮőécŸ

ä¡ăçŽĎçÍŃåžŔåęĆä¡ŢèČ¡åď§èğčædŘåŚ¡äžď'èąŇéĂĽéąźïijĹä¡■äžŐsys.argväÿ■ïijĽ

èğčåĘşæŰźæąĹ

`argparse` æĺąåİŮåŘŕèćńçŤĺæİèèğčædŘåŚ¡äžď'èąŇéĂĽéąźãĂĆäÿŃéİćäÿĂäÿłçőĂå■Ţä¿Ńå■ŘæijŢçď'ž

```
# search.py
'''
Hypothetical command-line tool for searching a collection of
files for one or more text patterns.
'''
import argparse
parser = argparse.ArgumentParser(description='Search some files')
```

```
parser.add_argument(dest='filenames',metavar='filename', nargs='*')

parser.add_argument('-p', '--pat',metavar='pattern', required=True,
                    dest='patterns', action='append',
                    help='text pattern to search for')

parser.add_argument('-v', dest='verbose', action='store_true',
                    help='verbose mode')

parser.add_argument('-o', dest='outfile', action='store',
                    help='output file')

parser.add_argument('--speed', dest='speed', action='store',
                    choices={'slow','fast'}, default='slow',
                    help='search speed')

args = parser.parse_args()

# Output the collected arguments
print(args.filenames)
print(args.patterns)
print(args.verbose)
print(args.outfile)
print(args.speed)
```

èŕěçÍŃåžŔåőŽäźL’äžĘäÿĂäÿłåęĆäÿŃä¡£çŤÍçŽĎåŚ¡äžd’èąŇèğčæđŘåŹÍïijŽ

```
bash % python3 search.py -h
usage: search.py [-h] [-p pattern] [-v] [-o OUTFILE] [--speed {slow,
 ↪fast}]
                 [filename [filename ...]]

Search some files

positional arguments:
  filename

optional arguments:
  -h, --help            show this help message and exit
  -p pattern, --pat pattern
                        text pattern to search for
  -v                    verbose mode
  -o OUTFILE            output file
  --speed {slow,fast}   search speed
```

äÿŃéIćçŽĎéČĺåĹĘæijŤçd’žäžĘçÍŃåžŔäÿ■çŽĎæŢřæ■őéČĺåĹĘãĂĆäżŤçżĘèğĆår§print()èí■åŘěçŽĎæL’Ş

```
bash % python3 search.py foo.txt bar.txt
usage: search.py [-h] -p pattern [-v] [-o OUTFILE] [--speed {fast,
 ↪slow}]
```

```
                [filename [filename ...]]
search.py: error: the following arguments are required: -p/--pat

bash % python3 search.py -v -p spam --pat=eggs foo.txt bar.txt
filenames = ['foo.txt', 'bar.txt']
patterns  = ['spam', 'eggs']
verbose   = True
outfile   = None
speed     = slow

bash % python3 search.py -v -p spam --pat=eggs foo.txt bar.txt -o␣
↪results
filenames = ['foo.txt', 'bar.txt']
patterns  = ['spam', 'eggs']
verbose   = True
outfile   = results
speed     = slow

bash % python3 search.py -v -p spam --pat=eggs foo.txt bar.txt -o␣
↪results \
             --speed=fast
filenames = ['foo.txt', 'bar.txt']
patterns  = ['spam', 'eggs']
verbose   = True
outfile   = results
speed     = fast
```

åŕźäžŐéĂL’éąźåĂijçŽĎè£ŻäÿĂæ■ěåď’ĎçŘĘçŤśçÍŃåžŔæÌěåĘşåőŽïijŇçŤÍä¡äèĞłåůśçŽĎéĂżè¿ŚæÌěæŻ£ `print()` åĞ¡æŢřãĂĆ

èőÍèőž

`argparse` æÍąåİŮæŸŕæăĞåĞĘåžŞäÿ■æIJĂåď’ğçŽĎæÍąåİŮäźŃäÿĂïijŇæŃěæIJL’åď’ğéĞŔçŽĎéĚ■ç¡őé æIJňèŁĆåŘłæŸŕæijŤçď’žäžĘåĚűäÿ■æIJĂå§žçąĂçŽĎäÿĂäžŻçL’źæĂğïijŇåÿőåŁl’ä¡ääĚěéŮÍãĂĆ

äÿžäžĘèğčædŘåŚ¡äžď’èąŇéĂL’éąźïijŇä¡äéęŰåĚÍèęĄåĹŻåżžäÿĂäÿł `ArgumentParser` åőđä¿ŃïijŇ åźűä¡£çŤÍ `add_argument()` æŰźæşŢåčřæŸŐä¡äæČşèęĄæŤŕæŇĄçŽĎéĂL’éąźãĂĆ åIJÍæŕŔäÿł `add_argument()` èřČçŤÍäÿ■ïijŇ`dest` åŔĆæŢřæŇĞåőŽèğčædŘçżŞædIJèćńæŇĞæť¿çżŹåśđæĂğçŽĎåŘ■å■ŮãĂĆ `metavar` åŔĆæŢřèćńçŤÍæİěçŤ§æĹŘåÿőåŁl’ä£ąæĄŕãĂĆ`action` åŔĆæŢřæŇĞåőŽèů§åśđæĂğåŕźåžŤçŽĎåď’ĎçŘĘéĂżè¿ŚïijŇ éĂŽåÿÿçŽĎåĂijäÿž `store` ,èćńçŤÍæİěå■ŸåĆÍæ§ŘäÿłåĂijæĹŮèőšåď’ŽäÿłåŔĆæŢřåĂijæŤűéŽĘåĹřäÿĂäÿłåĹŮèąÍäÿ■ãĂĆ äÿŃéİćçŽĎåŔĆæŢřæŤűéŽĘæL’ĂæIJL’åL’l’ä¡ŹçŽĎåŚ¡äžď’èąŇåŔĆæŢřåĹřäÿĂäÿłåĹŮèąÍäÿ■ãĂĆåIJÍæIJňä¿Ń

```
parser.add_argument(dest='filenames',metavar='filename', nargs='*')
```

äÿŃéİćçŽĎåŔĆæŢřæăźæ■őåŔĆæŢřæŸŕåŘęå■ŸåIJÍæİèèő¿ç¡őäÿĂäÿł `Boolean` æăĞå£ŮïijŽ

```
parser.add_argument('-v', dest='verbose', action='store_true',
                    help='verbose mode')
```

äÿŃéİćçŽĎåŔĆæŢřæŐěåŔŮäÿĂäÿłå■ŢçŃňåĂijåźűåřĘåĚűå■ŸåĆĺäÿžäÿĂäÿłå■ŮçňęäÿšïijŽ

```
parser.add_argument('-o', dest='outfile', action='store',
                    help='output file')
```

äÿŃéİćçŽĎåŔĆæŢřèŕť æŸŐåĚĄèőÿæ§ŘäÿłåŔĆæŢřéĞ■åď■åĞžçŐřåď ŽæňąïijŇåźűåřĘåőČäżňè£¡åŁäåĺ required æăĞå£Ůèąĺçď žèŕěåŔĆæŢřèĞşåřŚèęĄæIJĽäÿĂäÿłãĂĆ-p åŠŇ --pat èąĺçď žäÿď äÿłåŔĆæŢřåŘ■å¡ćåijŔéČ¡åŔŕä¡£çŤĺãĂĆ

```
parser.add_argument('-p', '--pat',metavar='pattern', required=True,
                    dest='patterns', action='append',
                    help='text pattern to search for')
```

æIJĂåŘŐïijŇäÿŃéİćçŽĎåŔĆæŢřèŕť æŸŐæŐěåŔŮäÿĂäÿłåĂijïijŇä¡ĘæŸŕäijŽåřĘåĚűåŠŇåŔŕèČ¡çŽĎéĂ

```
parser.add_argument('--speed', dest='speed', action='store',
                    choices={'slow','fast'}, default='slow',
                    help='search speed')
```

äÿĂæŮęåŔĆæŢřéĂĽéąźèćńæŇĞåőŽïijŇä¡ăåřśåŔŕäżěæĽğèąŇ parser.parse() æŰźæşŢäžĘãĂĆ åőČäijŽåď ĎçŘĘ sys.argv çŽĎåĂijåźűè£ŤåŻđäÿĂäÿłçżŞæđIJåőđä¿ŃãĂĆ æŕŔäÿłåŔĆæŢřåĂijäijŽèćńèő¿ç¡őæĹŘèŕěåőđä¿Ńäÿ■ add_argument() æŰźæşŢçŽĎ dest åŔĆæŢřæŇĞåőŽçŽĎåśđæĂğåĂijãĂĆ

è£Ÿå¿Ĺåď Žçğ■åĚűäżŰæŰźæşŢèğčæđŘåŚ¡äżď èąŇéĂĽéąźãĂĆ ä¿ŃåęĆïijŇä¡ăåŔŕèČ¡äijŽæĽŃåŁĺçŽĎåď ĎçŘĘ sys.argv æĹŰèĂĚä¡£çŤĺ getopt æĺąåİŮãĂĆ ä¡ĘæŸŕïijŇåęĆæđIJä¡ăéĞĞçŤĺæIJňèŁĆçŽĎæŰźåijŔïijŇåřĘäijŽåĞŔåřŚå¿Ĺåď ŽåĘŮä¡ŹäżčçăĄï argparse æĺąåİŮåůšçżŔåÿőä¡ăåď ĎçŘĘäžĘãĂĆ ä¡ăåŔŕèČ¡è£ŸäijŽçćřåĹřä¡£çŤĺ optparse åžŞèğčæđŘéĂĽéąźçŽĎäżčçăĄãĂĆ åř¡çőą optparse åŠŇ argparse å¿ĹåČŔïijŇä¡ĘæŸŕåŘŐèĂĚæŻť åĚĹè£ŻïijŇåŻăæ■ď åIJĺæŰřçŽĎçĺŃåžŔäÿ■ä¡ăåžŤèŕěä¡£çŤĺåőČãĂĆ

15.4 13.4 è£ŘèąŇæŮűåijźåĞžåŕĘçăĄè¿ŞåĚěæŔŘçď ž

éŮőéćŸ

ä¡ăåĘŹäžĘäÿłèĎŽæIJňïijŇè£ŘèąŇæŮűéIJĂèęĄäÿĂäÿłåŕĘçăĄãĂĆæ■ď èĎŽæIJňæŸŕäžď äžŠåijŔçŽĎïij èĂŇæŸŕéIJĂèęĄåijźåĞžäÿĂäÿłåŕĘçăĄè¿ŞåĚěæŔŘçď žïijŇèőľ çŤĺæĹůèĞłåůśè¿ŞåĚěãĂĆ

èğčåĘşæŰźæąĹ

è£ŹæŮűåĂŹPythonçŽĎ getpass æĺąåİŮæ■čæŸŕä¡ăæĽĂéIJĂèęĄçŽĎãĂĆä¡ăåŔŕäżěèőľ ä¡ăå¿Ĺè¡żæİ¿ åźűäÿŤäÿ■äijŽåIJĺçŤĺæĹůçżĹçńŕåŻđæŸ¿åŕĘçăĄãĂĆäÿŃéİćæŸŕåĚůä¡ŞäżčçăĄïijŽ

```
import getpass

user = getpass.getuser()
passwd = getpass.getpass()

if svc_login(user, passwd):     # You must write svc_login()
   print('Yay!')
else:
   print('Boo!')
```

åIJÍæ■d'äžčçăĄäÿ■ïijŇsvc_login() æŸŕä¡ăèęĄåőđçŐřçŽĎåd'ĎçŘĘářEçăĄçŽĎåĞ¡æŢřïijŇåĚůä¡Şç

èőÍèőž

æşÍæĎŔåIJÍåL'■éİćäžčçăĄäÿ■ getpass.getuser() äÿ■äijŽåijźåĞžçŤÍæĹůåŘ■çŽĎè¿ŞåĚěæŔŘçd'žãĂĆ åőČäijŽæăźæ■őèŕěçŤÍæĹůçŽĎshellçŐŕåćČæĹŰèĂĚäijŽä¿Íæ■őæIJňåIJřçşżçż§çŽĎårEçăĄåžŞïijĹæŤŕæŇĄ *pwd* æÍąåİŮçŽĎåźşåŔřïijL'æİěä¡£çŤÍå¡ŞåL'■çŤÍæĹůçŽĎçŹżå¡ŢåŘ■ïijŇ

åęĆæđIJä¡ăæČşæŸ¿çd'žçŽĎåijźåĞžçŤÍæĹůåŘ■è¿ŞåĚěæŔŘçd'žïijŇä¡£çŤÍåĘĚç¡őçŽĎ input åĞ¡æŢřïijŽ

```
user = input('Enter your username: ')
```

è£ŸæIJL'äÿĂçĆźå¿ĹéĞ■èęĄïijŇæIJL'äžŻçşżçż§åŔŕèČ¡äÿ■æŤŕæŇĄ getpass() æŰźæşŢéŽŘèŮŔè¿ŞåĚěåŕEçăĄąãĂĆ è£Źçğ■æČĚåĘţäÿŃïijŇPythonäijŽæŔŘåL'■è■ęåŚĹä¡ăè£ŹäžŻéŮőécŸïi

15.5 13.5 èŐůåŔŰçżĹçńŕçŽĎåd'ğářŔ

éŮőécŸ

ä¡ăéIJĂèęĄç§ěéĄŞå¡ŞåL'■çżĹçńŕçŽĎåd'ğářŔäžěä¿£æ■čçąőçŽĎæäijåijŔåŇŰè¿ŞåĞžãĂĆ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ os.get_terminal_size() åĞ¡æŢřæİěåĄŽåĹřè£ŹäÿĂçĆźãĂĆ

äžčçăĄçd'žä¿ŃïijŽ

```
>>> import os
>>> sz = os.get_terminal_size()
>>> sz
os.terminal_size(columns=80, lines=24)
>>> sz.columns
80
>>> sz.lines
```

```
24
>>>
```

èőÍèőž

æIJL'åd'łåd'ŽæŰźåijŔæİěå¿Ůç§ěçżĹçńŕåd'ğářŔäžĘïijŇäžŐèřżåŔŰçŐŕåćČåŔŸéĞŔåĹřæL'ğèąŇåžŢåšĆç
`ioctl()` åĞ¡æŢřç■L'ç■L'ãĂĆ äÿ■è£ĞïijŇäÿžäżĂäźĹèęĄåŐżçăŤçl'űè£ŹäžŻåd'■æİĆçŽĎåŁđæşŢèĂŇäÿ■æ

15.6 13.6 æL'ğèąŇåd'ŰéČĺåŚ¡äžd'åźűèŐůåŔŰåőČçŽĎè¿ŞåĞž

éŮőécŸ

ä¡ăæČşæL'ğèąŇäÿĂäÿłåd'ŰéČĺåŚ¡äžd'åźűäžěPythonå■ŮçňęäÿšçŽĎå¡ćåijŔèŐůåŔŰæL'ğèąŇçżŞæđIJãĂ

èğčåĘşæŰźæąĹ

ä¡£çŤÍ `subprocess.check_output()` åĞ¡æŢřãĂĆä¿ŃåęĆïijŽ

```
import subprocess
out_bytes = subprocess.check_output(['netstat','-a'])
```

è£ŹæőţäžčçăĄæL'ğèąŇäÿĂäÿłæŇĞåőŽçŽĎåŚ¡äžd'åźűåřĘæL'ğèąŇçżŞæđIJäžěäÿĂäÿłå■ŮèŁĆå■Ůçňęäÿ
åęĆæđIJä¡ăéIJĂèęĄæŰĞæIJňå¡ćåijŔè£ŤåŻđïijŇåŁääÿĂäÿłèğčçăĄæ■ěéłd'å■şåŔřãĂĆä¿ŃåęĆïijŽ

```
out_text = out_bytes.decode('utf-8')
```

åęĆæđIJèćńæL'ğèąŇçŽĎåŚ¡äžd'äžěéIđéŻűçăĄè£ŤåŻđïijŇåřśäijŽæŁŻåĞžåijĆåÿÿãĂĆ
äÿŃéIćçŽĎä¿Ńå■Řæ■ŢèŐůåĹřéŤŹèřŕåźűèŐůåŔŰè£ŤåŻđçăĄïijŽ

```
try:
    out_bytes = subprocess.check_output(['cmd','arg1','arg2'])
except subprocess.CalledProcessError as e:
    out_bytes = e.output       # Output generated before error
    code      = e.returncode   # Return code
```

éžŸèőd'æČĚåĘţäÿŃïijŇcheck_output() äžĚäžĚè£ŤåŻđè¿ŞåĚěåĹřæăĞåĞĘè¿ŞåĞžçŽĎåĂijãĂĆ
åęĆæđIJä¡ăéIJĂèęĄåŘŇæŮűæŤűéŻĘæăĞåĞĘè¿ŞåĞžåŠŇéŤŹèřŕè¿ŞåĞžïijŇä¡£çŤÍ `stderr`
åŔĆæŢřïijŽ

```
out_bytes = subprocess.check_output(['cmd','arg1','arg2'],
                                    stderr=subprocess.STDOUT)
```

åęĆæđIJä¡ăéIJĂèęĄçŤĺäÿĂäÿłèűĚæŮűæIJžåĹűæIđæL'ğèąŇåŚ¡äžd'ïijŇä¡£çŤÍ `timeout`
åŔĆæŢřïijŽ

```
try:
    out_bytes = subprocess.check_output(['cmd','arg1','arg2'],␣
↪timeout=5)
except subprocess.TimeoutExpired as e:
    ...
```

éĂŽåÿÿæÌèèőšïijŇåŚ¡äžd'çŽĎæĽğèąŇäÿ■éIJĂèęĄä¡£çŤĺåĹřåžŢåśĆshellçŐřåćČïijĹæŕŤåęĆshãĂĄbash äÿĂäÿłå■Ůçňęäÿšå ĹŮèąĺäijŽèćńäijåéĂŠçżŹäÿĂäÿłä¡Őçžğçşżçż§åŚ¡äžd'ïijŇæŕŤåęĆ os. execve() ãĂĆ åęĆæđIJä¡ăæČşèől'åŚ¡äžd'èćńäÿĂäÿłshellæĽğèąŇïijŇäijåéĂŠäÿĂäÿłå■Ůçňęäÿšå ŔĆæŢřïij shell=True. æIJL'æŮűåĂŹä¡ăæČşèęĄPythonåŐżæĽğèąŇäÿĂäÿłåd'■æÍĆçŽĎshellåŚ¡äžd'çŽĎæŮűåĂŹè

```
out_bytes = subprocess.check_output('grep python | wc > out',␣
↪shell=True)
```

éIJĂèęĄæşĺæĎŔçŽĎæŸŕåIJĺshelläÿ■æĽğèąŇåŚ¡äžd'äijŽå■ŸåIJĺäÿĂåőŽçŽĎåőL'åĚĺéčŐéŹl'ïijŇçL'źåĹ è£ŹæŮűåĂŹåŔŕäžěä¡£çŤĺ shlex.quote() åĞ¡æŢřæÌèèőšåŔĆæŢřæ■čçąőçŽĎçŤĺåŔŇåijŢçŤĺåijŢètůæÌěã

èőĺèőž

ä¡£çŤĺ check_output() åĞ¡æŢřæŸŕæĽğèąŇåd'ŰéČĺåŚ¡äžd'åźűèŐůåŔŰåĚűè£ŤåŻđåĂijçŽĎæIJĂçĉ ä¡ĘæŸŕïijŇåęĆæđIJä¡ăéIJĂèęĄåŕźå■Řè£ŻçĺŇåĄŽæŻt'åd'■æÍĆçŽĎäžd'äžŠïijŇæŕŤåęĆçżŹåőČåŔŠéĂĄè¿Şå è£ŹæŮűåĂŹåŔŕçŻt'æŐěä¡£çŤĺ subprocess.Popen çśżãĂĆä¿ŇåęĆïijŽ

```
import subprocess

# Some text to send
text = b'''
hello world
this is a test
goodbye
'''

# Launch a command with pipes
p = subprocess.Popen(['wc'],
          stdout = subprocess.PIPE,
          stdin = subprocess.PIPE)

# Send the data and get the output
stdout, stderr = p.communicate(text)

# To interpret as text, decode
out = stdout.decode('utf-8')
err = stderr.decode('utf-8')
```

subprocess æĺąåİŮåŕźäžŐä¿İèţŰTTYçŽĎåd'ŰéČĺåŚ¡äžd'äÿ■åŘĹéĂĆçŤĺãĂĆ ä¿ŇåęĆïijŇä¡ăäÿ■èČ¡ä¡£çŤĺåőČæÌèèĞłåŁĺåŇŰäÿĂäÿłçŤĺæĹůè¿ŞåĚěåŕĘçăĄçŽĎäżżåŁąïijĹæŕŤåęĆäÿĂäÿłss è£ŹæŮűåĂŹïijŇä¡ăéIJĂèęĄä¡£çŤĺåĹřçňňäÿL'æŰźæĺąåİŮäžĘïijŇæŕŤåęĆå§žäžŐèŚŮåŘ■çŽĎ expect åőűæŮŔçŽĎåůěåĚůïijĹpexpectæĹŰçśżäijijçŽĎïijL'

15.7 13.7 åď■åĹűæĹŰèĂĚçğżåŁĺæŰĞäžűåŠŇçŻőå¡Ţ

éŮőécŸ

ä¡ăæČşèęĄåď■åĹűæĹŰçğżåŁĺæŰĞäžűåŠŇçŻőå¡ŢïijŇä¡ĘæŸŕåŔĹäÿ■æČşèřČçŤĺshellåŚ¡äžď˜ãĂĆ

èğčåĘşæŰźæąĹ

shutil æĺąåİŮæIJL'å¿Ĺåď'Žä¿£æ■ůçŽĎåĞ¡æŢřåŔŕäžěåď■åĹűæŰĞäžűåŠŇçŻőå¡ŢãĂĆä¡£çŤĺèţůæİěé

```
import shutil

# Copy src to dst. (cp src dst)
shutil.copy(src, dst)

# Copy files, but preserve metadata (cp -p src dst)
shutil.copy2(src, dst)

# Copy directory tree (cp -R src dst)
shutil.copytree(src, dst)

# Move src to dst (mv src dst)
shutil.move(src, dst)
```

è£ŹäžŻåĞ¡æŢřçŽĎåŔĆæŢřéČ¡æŸŕå■Ůçňęäÿšå¡ćåijŔçŽĎæŰĞäžűæĹŰçŻőå¡ŢåŘ■ãĂĆ
åžŢåśĆèŕ■äźĽ'æĺąæŃ§äžĘçśżäijijçŽĎUnixåŚ¡äžď˜ïijŇåęĆäÿŁéİćçŽĎæşĺéĞŁéČĺåĹĘãĂĆ

éżŸèŏď'æČĚåĘţäÿŃïijŇåŕźäžŐçňęåŘůéŞ¿æŐëèĂŇåůšè£ŹäžŻåŚ¡äžď'åď'ĎçŘĘçŽĎæŸŕåőČæŇĞåŘŚçŽ
ä¿ŃåęĆïijŇåęĆæđIJæžŘæŰĞäžűæŸŕäÿĂäÿłçňęåŘůéŞ¿æŐëïijŇéĆčäźĹçŻőæăĞæŰĞäžűåřĘäijŽæŸŕçňęåŘůé
åęĆæđIJä¡ååŔłæČşåď'■åĹűçňęåŘůéŞ¿æŐëæIJňèžńïijŇéĆčäźĹéIJĂèęĄæŇĞåőŽåĚşéŤőå■ŮåŔĆæŢř
follow_symlinks ,åęĆäÿŃïijŽ

åęĆæđIJä¡ăæČşä£ÍçŢŹèćńåď'■åĹűçŻőå¡Ţäÿ■çŽĎçňęåŘůéŞ¿æŐëïijŇåČŔè£ŹæăůåĄŽïijŽ

```
shutil.copytree(src, dst, symlinks=True)
```

copytree() åŔŕäžěèőľ'ä¡ăåIJĺåď'■åĹűè£ĞçĺŃäÿ■éĂĽæŃľæĂğçŽĎå£¡çŢěæ§ŘäžŻæŰĞäžűæĹŰçŻőå
ä¡ăåŔŕäžěæŔŘä¿ŻäÿĂäÿłå£¡çŢěåĞ¡æŢřijŇæŐěåŘŮäÿĂäÿłçŻőå¡ŢåŘ■åŠŇæŰĞäžűåŘ■åĹŮèąĺä¡IJäÿžè¿ŞåŘ

```
def ignore_pyc_files(dirname, filenames):
    return [name in filenames if name.endswith('.pyc')]

shutil.copytree(src, dst, ignore=ignore_pyc_files)
```

çŤśäžŐå£¡çŢěæ§Řçğ■æĺąåijŔçŽĎæŰĞäžűåŘ■æŸŕå¿Ĺåÿÿè ğĄçŽĎïijŇåŻăæ■đ'äÿĂäÿłä¿£æ■ůçŽĎåĞ¡æ
ignore_patterns() åůšçżŘåŇĚåŘńåIJĺéĞŇéİćäžĘãĂĆä¿ŃåęĆïijŽ

```
shutil.copytree(src, dst, ignore=shutil.ignore_patterns('*~', '*.pyc
↪'))
```

èőÍèőž

ä¡£çŤÍ `shutil` åď■åĹűæŰĞäžűåŠŇçŻőå¡Ţäź§å£ŠçőĂå■ŢäžEçĆźåŘğãĂĆ
äÿ■è£ĞïijŇåŕźäžŐæŰĞäžűåĚČæŢřæ■őä£qæĄŕïijŇ`copy2()` è£ŹæăůçŽĎåĞ¡æŢřåŘłèČ¡ař¡èĞłåůśæIJĂåď ğè
èő£éŰőæŮűéŮť ãĂĄåĹŻåžžæŮűéŮť åŠŇæİČéŹŘè£ŹäžŻå§žæIJňä£qæĄŕäijŽèćńä£İçŢŹïijŇ
ä¡ĘæŸŕåŕźäžŐæĽĂæIJL'èĂĚãĂĄACLsãĂĄètĎæžŘforkåŠŇåĚűäžŰæŻť æůśåśĆæňąçŽĎæŰĞäžűåĚČä£qæA
è£Źäÿłè£Ÿå¿Ůä¿ÌètŰäžŐåžŢåśĆæŞ■ä¡IJçşżçż§çśžåđŃåŠŇçŤÍæĹůæĽĂæŃěæIJL'çŽĎè£éŰőæİČéŹŘãĂĆ
ä¡ăéĂŽåÿÿäÿ■äijŽåŐżä¡£çŤÍ `shutil.copytree()` åĞ¡æŢřæİěæĽ'ğèąŇçşżçż§åď'Ğäž¡ãĂĆ
å¡Şåď'ĎçŘĘæŰĞäžűåŘ■çŽĎæŮűåĂŹïijŇæIJĂåě¡ä¡£çŤÍ `os.path`
äÿ■çŽĎåĞ¡æŢřæİěçąőä£İæIJĂåď ğçŽĎåŔfçğżæď■æĂğïijĹçL'źåĹńæŸŕåŘŇæŮűèęĄéĂĆçŤÍäžŐUnixåŠŇWi
ä¿ŃåęĆïijŽ

```
>>> filename = '/Users/guido/programs/spam.py'
>>> import os.path
>>> os.path.basename(filename)
'spam.py'
>>> os.path.dirname(filename)
'/Users/guido/programs'
>>> os.path.split(filename)
('/Users/guido/programs', 'spam.py')
>>> os.path.join('/new/dir', os.path.basename(filename))
'/new/dir/spam.py'
>>> os.path.expanduser('~/guido/programs/spam.py')
'/Users/guido/programs/spam.py'
>>>
```

ä¡£çŤÍ `copytree()` åď■åĹűæŰĞäžűåď źçŽĎäÿĂäÿłæčŸæĽŃçŽĎéŮőécŸæŸŕåŕźäžŐéŤŹèŕŕçŽĎåď Ď
ä¿ŃåęĆïijŇåIJĺåď■åĹűè£ĞçĺŃäÿ■ïijŇåĞ¡æŢřåŘŕèČ¡äijŽçćřåĹřæ■§åİŔçŽĎçňęåŘůéŞ¿æŐěïijŇåŽääÿžæİČéŹ
äÿžäžĘèğčåĘşè£ŹäÿłéŮőécŸïijŇæĽĂæIJL'çćřåĹřçŽĎéŮőécŸäijŽèćńæŤűéŽĘåĹřäÿĂäÿłåĹŮèąĺäÿ■åźűæĽŞ
äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŽ

```
try:
    shutil.copytree(src, dst)
except shutil.Error as e:
    for src, dst, msg in e.args[0]:
         # src is source name
         # dst is destination name
         # msg is error message from exception
         print(dst, src, msg)
```

åęĆædIJä¡ăæŔŘä¿ŻåĚşéŤőå■ŮåŔĆæŢř `ignore_dangling_symlinks=True` ïijŇ
è£ŹæŮűåĂŹ `copytree()` äijŽå£¡çŢěæŐĽæŮăæŢĹçňęåŘůéŞ¿æŐěãĂĆ

æIJňèŁĆæijŤçď žçŽĎè£ŹäžŻåĞ¡æŢřéČ¡æŸŕæIJĂåÿÿèğĄçŽĎãĂĆäÿ■è£ĞïijŇ`shutil`
è£ŸæIJĽæŻť åď ŽçŽĎåŠŇåď■åĹűæŢřæ■őçŻÿåĚşçŽĎæŞ■ä¡IJãĂĆ
åőĈçŽĎæŰĞæąčå¿ĹåĂijå¿ŮäÿĂçIJŃïijŇåŘĆèĂČ Python documentation

15.8 13.8 åĹŻåżžåŠŇèğčåŐŃå¡Šæąčæ ŰĞäżű

éŮőécŸ

ä¡ăéIJĂèęĄåĹŻåżžæĹŰèğčåŐŃåÿÿèğĄæăijåijŔçŽĎå¡Šæąčæ ŰĞäżűïijĹæŕŤåęĆ.tar, .tgzæĹŰ.zipïijL'

èğčåĘşæŰźæąĹ

shutil æĺąåİŮæŃěæIJL'äÿd'äÿłåĞ¡æŢřâĂŤâĂŤ make_archive() åŠŇ unpack_archive() åŔŕæt'¿äÿŁçŤĺåIJžãĂĆ ä¿ŃåęĆïijŽ

```
>>> import shutil
>>> shutil.unpack_archive('Python-3.3.0.tgz')

>>> shutil.make_archive('py33','zip','Python-3.3.0')
'/Users/beazley/Downloads/py33.zip'
>>>
```

make_archive() çŽĎçňňäžŇäÿłåŔĆæŢřæŸŕæIJ§æIJŻçŽĎè¿ŞåĞžæăijåijŔãĂĆ åŔŕäžěä¡£çŤĺ get_archive_formats() èŐůåŔŰæL'ĂæIJL'æŤŕæŇAçŽĎå¡ŠæąčæăijåijŔåĹŰèąĺãĂĆä¿Ň

```
>>> shutil.get_archive_formats()
[('bztar', "bzip2'ed tar-file"), ('gztar', "gzip'ed tar-file"),
 ('tar', 'uncompressed tar file'), ('zip', 'ZIP file')]
>>>
```

èőÍèőž

Pythonè£ŸæIJL'åĚűäżŰçŽĎæĺąåİŮåŔŕçŤĺæİěåd'ĎçŘĘåd'Žçğ■å¡ŠæąčæăijåijŔïijĹæŕŤåęĆtarfile, zipfile, gzip, bz2ïijL'çŽĎåžŢåśĆçżĘèŁĆãĂĆ äÿ■è£ĞïijŇåęĆædIJä¡ääžĚäžĚåŔłæŸŕèęĄåĹŻåżžæĹŰæŔŘåŔŰ åŔŕäžěçŻť'æŐěä¡£çŤĺ shutil äÿ■çŽĎè£ŹäžŻéńŸåśĆåĞ¡æŢřãĂĆ

è£ŹäžŻåĞ¡æŢřè£ŸæIJL'å¿Ĺåd'ŽåĚűäżŰéĂL'éąźïijŇçŤĺäžŐæŮěå£ŮæL'Şå■řãĂĄécĎæčĂãĂĄæŰĞäżűa åŔĆèĂČ shutilæŰĞæąč

15.9 13.9 éĂŽè£ĞæŰĞäżűåŘ■æ§ěæL'¿æŰĞäżű

éŮőécŸ

ä¡ăéIJĂèęĄåĘŹäÿĂäÿłæűL'åŔŁåĹřæŰĞäżűæ§ěæL'¿æŞ■ä¡IJçŽĎèĎŽæIJñïijŇæŕŤåęĆåŕźæŮěå£Ůå¡Šæą ä¡ăäÿ■æČşåIJĺPythonèĎŽæIJñäÿ■èřĆçŤĺshellïijŇæĹŰèĂĚä¡ăèęĄåőđçŐřäÿĂäžŻshelläÿ■èČ¡åĄŽçŽĎåŁ§èČ

èğčåĘşæŰźæąĹ

æ§ěæĽ¿æŰĞäżűïijŇåŔŕä¡£çŤĺ os.walk() åĞ¡æŢřïijŇäijääÿĂäÿłéąűçžğçŻőå¡ŢåŘ■çżŹåőČãĂĆ äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘïijŇæ§ěæĽ¿çĽźåőŽçŽĎæŰĞäżűåŘ■åźűç■ŤåžŤæĽĂæIJĽçňęåŘĹæİąäżűçŽĎæŰ

```
#!/usr/bin/env python3.3
import os

def findfile(start, name):
    for relpath, dirs, files in os.walk(start):
        if name in files:
            full_path = os.path.join(start, relpath, name)
            print(os.path.normpath(os.path.abspath(full_path)))

if __name__ == '__main__':
    findfile(sys.argv[1], sys.argv[2])
```

ä£İå■ŸèĎŽæIJňäÿžæŰĞäżűfindfile.pyïijŇçĎűåŘŐåIJĺåŚ¡äżď èąŇäÿ■æĽğèąŇåőČãĂĆ æŇĞåőŽåĹİåğŃæ§ěæĽ¿çŻőå¡ŢäżěåŔŁåŘ■å■Ůä¡IJäÿžä¡■ç¡őåŔĆæŢřïijŇåęĆäÿŃïijŽ

èőĺèőž

os.walk() æŰźæşŢäÿžæĹŚäżňéĄ■åŐĘçŻőå¡ŢæăŚïijŇ æŕŔæňąè£ŻåĚěäÿĂäÿłçŻőå¡ŢïijŇåőČäijŽè£ŤåŻđäÿĂäÿłäÿĽåĚČçżĎïijŇåŇĚåŘńçŻÿåŕźäžŐæ§ěæĽ¿çŻőå¡Ţ äżěåŔŁéĆčäÿłçŻőå¡ŢäÿŃéİćçŽĎæŰĞäżűåŘ■åĹŮèąĺãĂĆ

åŕźäžŐæŕŔäÿłåĚČçżĎïijŇåŔłéIJĂæčĂæţŃäÿĂäÿŃçŻőæăĞæŰĞäżűåŘ■æŸŕåŘęåIJĺæŰĞäżűåĹŮèąĺäÿ■ os.path.join() åŘĹåźűèůŕå¿ĎãĂĆ äÿžäžĘéĄ£åĚ■åěĞæĂłçŽĎèůŕå¿ĎåŘ■æŕŤåęĆ ./ ./foo//bar ïijŇä¡£çŤĺäžĘåŔęåď Űäÿď äÿłåĞ¡æŢřæİěä£őæ■čçżŞæđIJãĂĆ çňňäÿĂäÿłæŸŕ os.path.abspath() ,åőČæŐěåŔŮäÿĂäÿłèůŕå¿ĎïijŇåŔŕèČ¡æŸŕçŻÿåŕźèůŕå¿ĎïijŇæIJĂåŘŐè£ŤåŻđçżİå çňňäžŇäÿłæŸŕ os.path.normpath() ïijŇçŤĺæİěè£ŤåŻđæ■čåÿÿèůŕå¿ĎïijŇåŔŕäżěèğčåĘşåŔŇæŰIJæİĘã

åř¡çőąè£ŹäÿłèĎŽæIJňçŻÿåŕźäžŐUNIXåźşåŔřäÿŁéİćçŽĎå¿Ĺåď Žæ§ěæĽ¿æİěèőšèęĄçőĂå■Ţå¿Ĺåď Žïij åźűäÿŤïijŇè£ŸèČ¡å¿Ĺè¡żæİ¿çŽĎåŁäåĚěåĚűäżŰçŽĎåŁ§èČ¡ãĂĆ æĹŚäżňåĘ■æijŤçď žäÿĂäÿłä¿Ńå■ŘïijŇäÿŃéİćçŽĎåĞ¡æŢřæĽŞå■řæĽĂæIJĽæIJĂè£Śèćńä£őæŤźè£ĞçŽĎæŰ

```
#!/usr/bin/env python3.3

import os
import time

def modified_within(top, seconds):
    now = time.time()
    for path, dirs, files in os.walk(top):
        for name in files:
            fullpath = os.path.join(path, name)
            if os.path.exists(fullpath):
                mtime = os.path.getmtime(fullpath)
                if mtime > (now - seconds):
                    print(fullpath)
```

```
if __name__ == '__main__':
    import sys
    if len(sys.argv) != 3:
        print('Usage: {} dir seconds'.format(sys.argv[0]))
        raise SystemExit(1)

    modified_within(sys.argv[1], float(sys.argv[2]))
```

åIJÍæ■d'åĞ¡æŢřçŽĎå§žçąĂäźŃäÿŁïijŇä¡£çŤÍos,os.path,globç■L'çśżäijijæÍąåİŮïijŇä¡ääřśèČ¡åőđçŐřæŻt
åŔřåŔĆèĂČ5.11åřŔèŁĆåŠŇ5.13åřŔèŁĆç■L'çŻÿåĚşçńäèŁĆãĂĆ

15.10 13.10 èŕżåŔŰéĚ■ç¡őæŰĞäżű

éŮőécŸ

æĂŐæăůèŕżåŔŰæŹőéĂŽ.iniæäijåijŔçŽĎéĚ■ç¡őæŰĞäžűïij§

èğčåĘşæŰźæąĹ

configparser æÍąåİŮèČ¡èćńçŤÍæİěèŕżåŔŰéĚ■ç¡őæŰĞäżűãĂĆä¿ŇåęĆïijŇåĄĞèő¿ä¡ăæIJL'åęĆäÿŇç

```
; config.ini
; Sample configuration file

[installation]
library=%(prefix)s/lib
include=%(prefix)s/include
bin=%(prefix)s/bin
prefix=/usr/local

# Setting related to debug configuration
[debug]
log_errors=true
show_warnings=False

[server]
port: 8080
nworkers: 32
pid-file=/tmp/spam.pid
root=/www/root
signature:
    =================================
    Brought to you by the Python Cookbook
    =================================
```

äÿŇéİćæŸŕäÿĂäÿłèŕżåŔŰåŠŇæŘŘåŔŰåĚűäÿ■åĂijçŽĎä¿Ňå■ŘïijŽ

```
>>> from configparser import ConfigParser
>>> cfg = ConfigParser()
>>> cfg.read('config.ini')
['config.ini']
>>> cfg.sections()
['installation', 'debug', 'server']
>>> cfg.get('installation','library')
'/usr/local/lib'
>>> cfg.getboolean('debug','log_errors')

True
>>> cfg.getint('server','port')
8080
>>> cfg.getint('server','nworkers')
32
>>> print(cfg.get('server','signature'))

\==================================
Brought to you by the Python Cookbook
\==================================
>>>
```

åęĆæđIJæIJL'éIJĂèęĄïijŇä¡ăè£ŸèČ¡ä£őæŤźéĚ■ç¡őåźűä¡£çŤĺ `cfg.write()` æŰźæşŢåřĘåĚűåĘŹåŻđåĹřæŰĞäżűäÿ■ãĂĆä¿ŃåęĆïijŽ

```
>>> cfg.set('server','port','9000')
>>> cfg.set('debug','log_errors','False')
>>> import sys
>>> cfg.write(sys.stdout)
```

```
[installation]
library = %(prefix)s/lib
include = %(prefix)s/include
bin = %(prefix)s/bin
prefix = /usr/local

[debug]
log_errors = False
show_warnings = False

[server]
port = 9000
nworkers = 32
pid-file = /tmp/spam.pid
root = /www/root
signature =
          ==================================
          Brought to you by the Python Cookbook
          ==================================
>>>
```

èőĺèőž

éĚ■ç¡őæŰĞäżűä¡IJäÿžäÿĂçğ■åŔŕèŕżæĂğå¿Ĺåě¡çŽĎæăijåijŔïijŇéİđåÿÿéĂĆçŤĺäžŐå■ŸåĆĺçĺŃåžŔäÿ■çŽ
åIJĺæŕŔäÿłéĚ■ç¡őæŰĞäżűäÿ■ïijŇéĚ■ç¡őæŢřæ■őäijŽèćńåĹĘçżĎïijĹæŕŤåęĆä¿Ńå■Řäÿ■çŽĎâĂIJinstallationâ
âĂIJdebugâĂİ åŠŇ âĂIJserverâĂİïijĽãĂĆ æŕŔäÿłåĹĘçżĎåIJĺåĚűäÿ■æŇĞåőŽåŕźåžŤçŽĎåŘĎäÿłåŔŸéĞŔåĂ

åŕźäžŐåŔŕåőđçŐřåŘŇæăůåŁ§èČ¡çŽĎéĚ■ç¡őæŰĞäżűåŠŇPythonæžŘæŰĞäżűæŸŕæIJĽå¿Ĺåď ğçŽĎäÿ■
éęŰåĚĹïijŇéĚ■ç¡őæŰĞäżűçŽĎèŕ■æşŢèęĄæŻť èĞłçŤśäžŻïijŇäÿŃéİćçŽĎèţŃåĂijèŕ■åŔěæŸŕç■Ľæ ŢĹçŽĎïij

```
prefix=/usr/local
prefix: /usr/local
```

éĚ■ç¡őæŰĞäżűäÿ■çŽĎåŘ■å■ŮæŸŕäÿ■åŇžåĹĘåď ğåřŔåĘŹçŽĎãĂĆä¿ŃåęĆïijŽ

```
>>> cfg.get('installation','PREFIX')
'/usr/local'
>>> cfg.get('installation','prefix')
'/usr/local'
>>>
```

åIJĺèğčæđŘåĂijçŽĎæŮűåĂŹïijŇgetboolean() æŰźæşŢæ§ěæĽ¿äżżä¡ŢåŔŕèąŇçŽĎåĂijãĂĆä¿ŃåęĆ

```
log_errors = true
log_errors = TRUE
log_errors = Yes
log_errors = 1
```

æĹŰèőÿéĚ■ç¡őæŰĞäżűåŠŇPythonäżčçăĄæIJĂåď ğçŽĎäÿ■åŘŇåIJĺäžŐïijŇåőČåźűäÿ■æŸŕäżŐäÿŁèĂŇ
æŰĞäżűæŸŕåőĽèčĚäÿĂäÿłæŢť ä¡ŞèćńèŕżåŔŰçŽĎãĂĆåęĆæđIJçćřåĹřäžĘåŔŸéĞŔæŻ£æ■ćïijŇåőČåőđéŹĚä
ä¿ŃåęĆïijŇåIJĺäÿŃéİćè£ŹäÿłéĚ■ç¡őäÿ■ïijŇprefix åŔŸéĞŔåIJĺä¡£çŤĺåőČçŽĎåŔŸéĞŔäźŃåĽ■æĹŰäźŃåŘ

```
[installation]
library=%(prefix)s/lib
include=%(prefix)s/include
bin=%(prefix)s/bin
prefix=/usr/local
```

ConfigParser æIJĽäÿłåőźæŸŞèćńå£¡èğĘçŽĎçĽźæĂğæŸŕåőČèČ¡äÿĂæňąèŕżåŔŰåď ŽäÿłéĚ■ç¡őæŰ
ä¿ŃåęĆïijŇåĄĞèő¿äÿĂäÿłçŤĺæĹůåČŔäÿŃéİćè£ŹæăůæđĎéĂăäžĘäżŰäżňçŽĎéĚ■ç¡őæŰĞäżűïijŽ

```
; ~/.config.ini
[installation]
prefix=/Users/beazley/test

[debug]
log_errors=False
```

èŕżåŔŰè£ŹäÿłæŰĞäżűïijŇåőČåŕśèČ¡èů§äźŃåĽ■çŽĎéĚ■ç¡őåŘĹåźűèţůæİěãĂĆåęĆïijŽ

```
>>> # Previously read configuration
>>> cfg.get('installation', 'prefix')
```

```
'/usr/local'

>>> # Merge in user-specific configuration
>>> import os
>>> cfg.read(os.path.expanduser('~/.config.ini'))
['/Users/beazley/.config.ini']

>>> cfg.get('installation', 'prefix')
'/Users/beazley/test'
>>> cfg.get('installation', 'library')
'/Users/beazley/test/lib'
>>> cfg.getboolean('debug', 'log_errors')
False
>>>
```

äżŤçżĘèğĆåŕ§äÿŃ prefix åŔŸéĞŔæŸŕæĂŐæăůèęĘçŻŰåĚűäżŰçŻÿåĚşåŔŸéĞŔçŽĎïijŇæŕŤåęĆ library çŽĎèő¿åőŽåĂijãĂĆ äžğçŤ§è£Źçğ■çżŞæđIJçŽĎåŐ§åŻăæŸŕåŔŸéĞŔçŽĎæŤźåĘŹéĞĞåŔŰçŽĎæ ä¡ăåŔŕäżěåČŔäÿŃéİćè£ŹæăůåĄŽèŕŢéłŇïijŽ

```
>>> cfg.get('installation','library')
'/Users/beazley/test/lib'
>>> cfg.set('installation','prefix','/tmp/dir')
>>> cfg.get('installation','library')
'/tmp/dir/lib'
>>>
```

æIJĂåŘŐè£ŸæIJL'å¿ĹéĞ■èęĄäÿĂçĆźèęĄæşĺæĎŔçŽĎæŸŕPythonåźűäÿ■èČ¡æŤŕæŇĄ.iniæŰĞäżűåIJĺåĺ çąőä£İä¡ăåůšçżŔåŔĆéŸĚäžĘconfigparseræŰĞæąčäÿ■çŽĎèŕ■æşŢèŕęæČĚäżěåŔŁæŤŕæŇĄçL'źæĂğãĂĆ

15.11 13.11 çżŹçőĂå■ŢèĎŽæIJňåćđåŁăæŮěå£ŮåŁ§èČ¡

éŮőéćŸ

ä¡ăåÿŇæIJŻåIJĺèĎŽæIJňåŠŇçĺŃåžŔäÿ■åřĘèŕŁæŰ■ä£ąæĄŕåĘŹåĚěæŮěå£ŮæŰĞäżűãĂĆ

èğčåĘşæŰźæąĹ

æL'Şå■řæŮěå£ŮæIJĂçőĂå■ŢæŰźåijŔæŸŕä¡£çŤĺ logging æĺąåİŮãĂĆä¿ŃåęĆïijŽ

```
import logging

def main():
    # Configure the logging system
    logging.basicConfig(
        filename='app.log',
        level=logging.ERROR
    )
```

```
    # Variables (to make the calls that follow work)
    hostname = 'www.python.org'
    item = 'spam'
    filename = 'data.csv'
    mode = 'r'

    # Example logging calls (insert into your program)
    logging.critical('Host %s unknown', hostname)
    logging.error("Couldn't find %r", item)
    logging.warning('Feature is deprecated')
    logging.info('Opening file %r, mode=%r', filename, mode)
    logging.debug('Got here')

if __name__ == '__main__':
    main()
```

äÿŁéİćäžŤäÿłæŮěå£ŮèřČçŤĺïijĹcritical(), error(), warning(), info(), debug()ïijĽäżěéŹ■åžŔæŰźåijŔèąĺçď žäÿ■åŘŇçŽĎäÿěéĞ■çžğåĹńãĂĆ basicConfig() çŽĎ level åŔĆæŢřæŸŕäÿĂäÿłè£Ğæżď åŹĺãĂĆ æĽĂæIJĽçžğåĹńä¡ŐäžŐæ■ď çžğåĹńçŽĎæŮěå£ŮæűĹæĄŕéČ¡äijŽèćńå£¡çŢěæŐĽãĂĆ æŕŔäÿłloggingæŞ■ä¡IJçŽĎåŔĆæŢřæŸŕäÿĂäÿłæűĹæĄŕå■ŮçňęäÿšïijŇåŘŐéİćåĘ■èů§äÿĂäÿłæĹŰåď ŽäÿłåŔĆ æđĎéĂăæIJĂçżĹçŽĎæŮěå£ŮæűĹæĄŕçŽĎæŮűåĂŹæĹŚäżňä¡£çŤĺäžĘ%æŞ■ä¡IJçňęæİěæăijåijŔåŇŰæűĹæĄ

è£ŘèąŇè£ŹäÿłçĺŃåžŔåŘŐïijŇåIJĺæŰĞäżű app.log äÿ■çŽĎåĘĚåőźåžŤèŕěæŸŕäÿŃéİćè£ŹæăůïijŽ

```
CRITICAL:root:Host www.python.org unknown
ERROR:root:Could not find 'spam'
```

åęĆæđIJä¡ăæČşæŤźåŔŸè¿ŞåĞžç■ĽçžğïijŇä¡ăåŔŕäżěä£őæŤź basicConfig() èřČçŤĺäÿ■çŽĎåŔĆæŢřãĂĆä¿ŃåęĆïijŽ

```
logging.basicConfig(
     filename='app.log',
     level=logging.WARNING,
     format='%(levelname)s:%(asctime)s:%(message)s')
```

æIJĂåŘŐè¿ŞåĞžåŔŸæĹŘåęĆäÿŃïijŽ

```
CRITICAL:2012-11-20 12:27:13,595:Host www.python.org unknown
ERROR:2012-11-20 12:27:13,595:Could not find 'spam'
WARNING:2012-11-20 12:27:13,595:Feature is deprecated
```

äÿŁéİćçŽĎæŮěå£ŮéĚ■ç¡őéČ¡æŸŕçąňçijŰçăĄåĹřçĺŃåžŔäÿ■çŽĎãĂĆåęĆæđIJä¡ăæČşä¡£çŤĺéĚ■ç¡őæŰČ åŔŕäżěåČŔäÿŃéİćè£Źæăůä£őæŤź basicConfig() èřČçŤĺïijŽ

```
import logging
import logging.config

def main():
    # Configure the logging system
```

```
    logging.config.fileConfig('logconfig.ini')
    ...
```

åĹŻåžžäÿĂäÿłäÿŃéÍćè£ŹæăůçŽĎæŰĞäžűïijŇåŘ■å■ŮåŔń logconfig.ini ïijŽ

```
[loggers]
keys=root

[handlers]
keys=defaultHandler

[formatters]
keys=defaultFormatter

[logger_root]
level=INFO
handlers=defaultHandler
qualname=root

[handler_defaultHandler]
class=FileHandler
formatter=defaultFormatter
args=('app.log', 'a')

[formatter_defaultFormatter]
format=%(levelname)s:%(name)s:%(message)s
```

åęĆæđIJä¡ăæČşä£őæŤźéĚ■ç¡őïijŇåŘŕäžěçŻť'æŐěçijŰè¿ŚæŰĞäžűlogconfig.iniå■şåŔŕãĂĆ

èőÍèőž

åř¡çőąåŕźäžŐ logging æÍąåİŮèĂŇåůšæIJL'å¿Ĺåđ'ŽæŻť'énŸçžğçŽĎéĚ■ç¡őéĂL'éąźïijŇäÿ■è£Ğè£ŹéĞŇçŽĎæŰźæąĹåŕźäžŐçőĂå■ŢçŽĎçÍŃåžŔåŠŇèĎŽæIJňåůšçżŔèűşåđ'§äžĘãĂĆåŘłæČşåIJÍěřČçŤÍæŮěå£ŮæŞ■ä¡IJåL'■åĚĹæL'ğèąŇäÿŃbasicConfig()åĞ¡æŢřæŰźæşŢïijŇä¡ăçŽĎçÍŃåžŔåřśè

åęĆæđIJä¡ăæČşèęĄä¡ăçŽĎæŮěå£ŮæűĹæĄŕåĘŹåĹřæăĞåĞĘéŤŹèŕŕäÿ■ïijŇèĂŇäÿ■æŸŕæŮěå£ŮæŰĞäž basicConfig() æŮűäÿ■äijăæŰĞäžűåŘ■åŔĆæŢřå■şåŔŕãĂĆä¿ŃåęĆïijŽ

```
logging.basicConfig(level=logging.INFO)
```

basicConfig() åIJÍçÍŃåžŔäÿ■åŘłèČ¡èćńæL'ğèąŇäÿĂæňąãĂĆåęĆæđIJä¡ăçÍ■åŘŐæČşæŤźåŔŸæŮěå åřśéIJĂèęĄåĚĹèŐůåŔŰ root logger ïijŇçĎűåŘŐçŻť'æŐëä£őæŤźåőČãĂĆä¿ŃåęĆïijŽ

```
logging.getLogger().level = logging.DEBUG
```

éIJĂèęĄåijžèřČçŽĎæŸŕæIJňèŁĆåŔłæŸŕæijŤçđ'žäžĘ logging æÍąåİŮçŽĎäÿĂäžŻå§žæIJňçŤÍæşŢãĂĆ åőČåŔŕäžěåĄŽæŻť'åđ'ŽæŻť'énŸçžğçŽĎåőŽåĹűãĂĆ åĚşäžŐæŮěå£ŮåőŽåĹűåŇŰäÿĂäÿłå¿Ĺåě¡çŽĎèţĎæžŘæŸŕ Logging Cookbook

15.12 13.12 çžŹåĞ¡æŢřåžŞåćđåŁäæŮěå£ŮåŁ§èČ¡

éŮőécŸ

ä¡äæČşçżŹæ§ŘäÿłåĞ¡æŢřåžŞåćđåŁäæŮěå£ŮåŁ§èČ¡ïijŇä¡ĘæŸŕåŔĹäÿ■èČ¡å¡śåŞ■åĹřéĆčäžŻäÿ■ä¡£çŤĺ

èğčåĘşæŰźæąĹ

åŕźäžŐæČşèęĄæĽğèąŇæŮěå£ŮæŞ■ä¡IJçŽĎåĞ¡æŢřåžŞèĂŇåůšïijŇä¡ăåžŤèŕěåĹŻåżžäÿĂäÿłäÿŞåśđçŽĎ `logger` åŕźèśąïijŇåźűäÿŤåČŔäÿŃéİćè£ŹæăůåĹİåğŃåŇŰéĚ■ç¡őïijŽ

```
# somelib.py

import logging
log = logging.getLogger(__name__)
log.addHandler(logging.NullHandler())

# Example function (for testing)
def func():
    log.critical('A Critical Error!')
    log.debug('A debug message')
```

ä¡£çŤĺè£ŹäÿłéĚ■ç¡őïijŃézŸèőd'æČĚåĘţäÿŃäÿ■äijŽæĽŞå■řæŮěå£ŮãĂĆä¿ŃåęĆïijŽ

```
>>> import somelib
>>> somelib.func()
>>>
```

äÿ■è£ĞïijŇåęĆæđIJéĚ■ç¡őè£ĞæŮěå£Ůçşżçż§ïijŇéĆčäźĹæŮěå£ŮæűĹæĄŕæĽŞå■řåřšåijĂåğŃçŤ§æŢĹï

```
>>> import logging
>>> logging.basicConfig()
>>> somelib.func()
CRITICAL:somelib:A Critical Error!
>>>
```

èőĺèőž

éĂŽåÿÿæİěèőšïijŇä¡ääÿ■åžŤèŕěåIJĺåĞ¡æŢřåžŞäžčçăĄäÿ■èĞłåůśéĚ■ç¡őæŮěå£Ůçşżçż§ïijŇæĹŰèĂĚæŸ

èřČçŤĺ `getLogger(__name__)` åĹŻåżžäÿĂäÿłåŠŃèřČçŤĺæĺąåİŮåŘŃåŘ■çŽĎlog-geræĺąåİŮãĂĆ çŤśäžŐæĺąåİŮéČ¡æŸŕåŤŕäÿĂçŽĎïijŇåŻăæ■d'åĹŻåżžçŽĎloggeräźšårĘæŸŕåŤŕäÿĂçŽĎãĂĆ

`log.addHandler(logging.NullHandler())` æŞ■ä¡IJåřĘäÿĂäÿłçľžåd'ĎçŘĘåŹĺçżŚåőŽåĹřåĹ äÿĂäÿłçľžåd'ĎçŘĘåŹĺéżŸèőd'äijŽå£¡çŢěřČçŤĺæĽ'ĂæIJL'çŽĎæŮěå£ŮæűĹæĄŕãĂĆ åŽăæ■d'ïijŇåęĆæđIJä¡£çŤĺèŕěåĞ¡æŢřåžŞçŽĎæŮűåĂŹè£ŸæšąæIJL'éĚ■ç¡őæŮěå£ŮïijŇéĆčäźĹåřĘäÿ■äijŽæ

è£ŸæIJL'äÿĂçĆźåřšæŸŕåŕźäžŐåŘĎäÿłåĞ¡æŢřåžŞçŽĎæŮěå£ŮéĚ■ç¡őåŘŕäžěæŸŕçŽÿäžŠçŃňçńŃçŽĎïij ä¿ŃåęĆïijŇåŕźäžŐåęĆäÿŃçŽĎäžčçăĄijŽ

```
>>> import logging
>>> logging.basicConfig(level=logging.ERROR)

>>> import somelib
>>> somelib.func()
CRITICAL:somelib:A Critical Error!

>>> # Change the logging level for 'somelib' only
>>> logging.getLogger('somelib').level=logging.DEBUG
>>> somelib.func()
CRITICAL:somelib:A Critical Error!
DEBUG:somelib:A debug message
>>>
```

åIJÍè£ŹéĞŇïijŇæăźæŮěå£Ůèćńéĕ■ç¡őæĹŘäżĔäżĔè¿ŞåĞžERRORæĹŰæŻťéńŸçžğåĹńçŽĎæűĹæĄŕãĂäÿ■è£Ğ ïijŇsomelib çŽĎæŮěå£ŮçžğåĹńèćńå■ŢçŃňéĔ■ç¡őæĹŘåŔŕäżěè¿ŞåĞždebugçžğåĹńçŽĎæűĹæĄŕïiåČŔè£ŹæăůæŻťæŤźå■ŢçŃňæĺąåİŮçŽĎæŮěå£ŮéĔ■ç¡őåŕźäžŐèřČèŕŢæİěèőšæŸŕå¿ĹæŰźä¿£çŽĎïijŇåŻääÿžä¡ăæŮăéIJĂåŐżæŻťæŤźäżżä¡ŢçŽĎåĚĺåśĂæŮěå£ŮéĔ■ç¡őâĂŤâĂŤåŔłéIJĂèęĄä£őæŤźä¡ăæČşèęĄæŻ

Logging HOWTO èŕęçżĘäżŃçż■äžĘåęĆä¡ŢéĔ■ç¡őæŮěå£ŮæĺąåİŮåŠŇåĔűäżŰæIJL'çŤĺæŁĂåůğïijŇåŔŕä

15.13 13.13 åőđçŐřäÿĂäÿłèőąæŮűåŹÍ

éŮőécŸ

ä¡ăæČşèőřå¡ŢçÍŃåžŔæL'ğèąŇåd'Žäÿłäżżå£ąæL'ĂèŁśèt'źçŽĎæŮűéŮt'

èğčåĘşæŰźæąĹ

time æĺąåİŮåŇĔåŘńå¿Ĺåd'ŽåĞ¡æŢřæİěæL'ğèąŇèů§æŮűéŮt'æIJL'åĔşçŽĎåĞ¡æŢřãĂĆåř¡çőąåęĆæ■d'ïijŇéĂŽåÿÿæĹŚäżñäijŽåIJÍæ■d'å§žçąĂäźŇäÿŁædĎéĂääÿĂäÿłæŻť'éńŸçžğçŽĎæŐěåŔčæİěæİ

```
import time

class Timer:
    def __init__(self, func=time.perf_counter):
        self.elapsed = 0.0
        self._func = func
        self._start = None

    def start(self):
        if self._start is not None:
            raise RuntimeError('Already started')
        self._start = self._func()

    def stop(self):
        if self._start is None:
            raise RuntimeError('Not started')
```

```
        end = self._func()
        self.elapsed += end - self._start
        self._start = None

    def reset(self):
        self.elapsed = 0.0

    @property
    def running(self):
        return self._start is not None

    def __enter__(self):
        self.start()
        return self

    def __exit__(self, *args):
        self.stop()
```

è£ŹäÿłçśżåőŽäźĽäžĘäÿĂäÿłåŔŕäżěèćńçŤĺæĹůæăźæ■őéIJĂèęĄåŘŕåŁĺãĂĄåĄIJæ■ćåŠŇéĞ■ç¡őçŽĎèőąæ
åőČäijŽåIJĺ elapsed åśđæĂğäÿ■èőřå¡ŢæŢť äÿłæűĹèĂŮæŮűéŮť ãĂĆ
äÿŃéİćæŸŕäÿĂäÿłä¿Ńå■ŘæİěæijŤçď žæĂŐæăůä¡£çŤĺåőČïijŽ

```
def countdown(n):
    while n > 0:
        n -= 1

# Use 1: Explicit start/stop
t = Timer()
t.start()
countdown(1000000)
t.stop()
print(t.elapsed)

# Use 2: As a context manager
with t:
    countdown(1000000)

print(t.elapsed)

with Timer() as t2:
    countdown(1000000)
print(t2.elapsed)
```

èőĺèőž

æIJňèŁĆæŔŘä¿ŻäžĘäÿĂäÿłçőĂå■ŢèĂŇåőđçŤĺçŽĎçśżæİěåőđçŐřæŮűéŮť èőřå¡ŢäżěåŔŁèĂŮæŮűèőąç
åŘŇæŮűäź§æŸŕåŕźä¡£çŤĺwithèŕ■åŔěäżěåŔŁäÿŁäÿŃæŰĞçőąçŘĘåŹĺå■ŔèőőçŽĎäÿĂäÿłå¿Ĺåě¡çŽĎæijŤçď ž

åIJĺèőąæŮűäÿ■èęĄèĂČèŹŚäÿĂäÿłåžŢåśĆçŽĎæŮűéŮť åĞ¡æŢřéŮőéćŸãĂĆäÿĂèĹňæİěèŕť ïijŇ

ä¡£çŤÍ `time.time()` æĹŰ `time.clock()` èőąçőŮçŽĎæŮűéŮť'çš¿åžęåŻăæŞ■ä¡IJçşżçż§çŽĎäÿ■åŘŇäijŹ
èĂŇä¡£çŤÍ `time.perf_counter()` åĞ¡æŢřåŔŕäžěçąőä£Ïä¡£çŤÍçşżçż§äÿŁéÍćæIJĂçš¿çąőçŽĎèőąæŮűåŹ

äÿŁè£řäžčçăĄäÿ■çŤś `Timer` çśżèőřå¡ŢçŽĎæŮűéŮť'æŸŕéŠ§èąĺæŮűéŮť'ïijŇåźűåŇĚåŘńäžĘæĽ'ĂæIJL'ä
åęĆæđIJä¡ăåŔłæČşèőąçőŮèřěè£ŻçÍŃæĽ'ĂèŁśèť'źçŽĎCPUæŮűéŮť'ïijŇåžŤèŕěä¡£çŤÍ `time.process_time()` æÌěäžčæŻ£ïijŽ

```
t = Timer(time.process_time)
with t:
    countdown(1000000)
print(t.elapsed)
```

`time.perf_counter()` åŠŇ `time.process_time()` éČ¡äijŽè£ŤåŽđářŔæŢřå¡ćåijŔçŽĎçğŠæŢřæŮűéŮť'ãĂĆ åőđéŹĚçŽĎæŮűéŮť'åĂijæšąæIJL'äżżä¡ŢæĎŔäźĽ'ïij

æŻť'åđ'ŽåĚşäžŐèőąæŮűåŠŇæĂğèČ¡åĹĘæđŘçŽĎä¿Ńå■ŘèŕůåŔĆèĂČ14.13åŕŔèŁĆãĂĆ

15.14 13.14 éŹŘåĹűåĘĚå■ŸåŠŇCPUçŽĎä¡£çŤÍéĞŔ

éŮőécŸ

ä¡ăæČşåŕźåIJÍUnixçşżçż§äÿŁéÍćè£ŘèąŇçŽĎçÍŃåžŔèő¿ç¡őåĘĚå■ŸæĹŰCPUçŽĎä¡£çŤÍéŹŘåĹűãĂĆ

èğčåĘşæŰźæąĹ

`resource` æÍąåİŮèČ¡åŘŇæŮűæĽ'ğèąŇè£ŹäÿđäÿłäżżåŁąãĂĆä¿ŃåęĆïijŇèęĄéŹŘåĹűCPUæŮűéŮť'ïijŀ

```
import signal
import resource
import os

def time_exceeded(signo, frame):
    print("Time's up!")
    raise SystemExit(1)

def set_max_runtime(seconds):
    # Install the signal handler and set a resource limit
    soft, hard = resource.getrlimit(resource.RLIMIT_CPU)
    resource.setrlimit(resource.RLIMIT_CPU, (seconds, hard))
    signal.signal(signal.SIGXCPU, time_exceeded)

if __name__ == '__main__':
    set_max_runtime(15)
    while True:
        pass
```

çÍŃåžŔè£ŘèąŇæŮűïijŇ`SIGXCPU` ä£ąåŔůåIJÍæŮűéŮť'è£ĞæIJ§æŮűèćńçŤ§æĹŘïijŇçĎűåŘŐæĽ'ğèąŇæÿ

èęĄéŹŘåĹűåĘĚå■Ÿä¡£çŤÍïijŇèő¿ç¡őåŔŕä¡£çŤÍçŽĎæĂżåĘĚå■ŸåĂijå■şåŔŕïijŇåęĆäÿŃïijŽ

```
import resource

def limit_memory(maxsize):
    soft, hard = resource.getrlimit(resource.RLIMIT_AS)
    resource.setrlimit(resource.RLIMIT_AS, (maxsize, hard))
```

åČŔè£Źæăůèő¿ç¡őäžĘåĘĚå■ŸéŹŘåĹűåŘŐïijŇçÍŃåžŔè£ŘèąŇåĹřæšąæIJL'åd'Žä¡ŹåĘĚå■ŸæŮűäijŽæŁŽ
`MemoryError` åijĆåÿÿãĂĆ

èőÍèőž

åIJÍæIJňèŁĆä¿Ńå■Řäÿ■ïijŇ`setrlimit()` åĞ¡æŢřèćńçŤÍæİěèő¿ç¡őçL'źåőŽètĎæžŘäÿŁéİćçŽĎè¡réŹŘ
è¡réŹŘåĹűæŸŕäÿĂäÿłåĂijïijŇå¡ŞèűĚè£Ğè£ŹäÿłåĂijçŽĎæŮűåĂŹæŞ■ä¡IJçşżçż§éĂŽåÿÿäijŽåŔŚéĂĄäÿĂäÿł
çąňéŹŘåĹűæŸŕçŤÍæİěæŇĞåőŽè¡réŹŘåĹűèČ¡èő¿åőŽçŽĎæIJĂåd'ğåĂijãĂĆéĂŽåÿÿæİèèőšïijŇè£ŹäÿłçŤśçşż
år¡çőąçąňéŹŘåĹűåŔŕäžěæŤźåřŘäÿĂçĆźïijŇä¡ĘæŸŕæIJĂåě¡äÿ■èęĄä¡£çŤÍçŤÍæĹůè£ŽçÍŃåÖżä£őæŤźãĂĆ

`setrlimit()` åĞ¡æŢřè£ŸèČ¡èćńçŤÍæİěèő¿ç¡őå■Řè£ŽçÍŃæŢřéĞŔãĂĄæL'ŞåijĂæŰĞäžűæŢřäžěåŘŁç
æŻť'åd'Žèŕęæ ČĚèŕůåŘĆèĂČ `resource` æÍąåİŮçŽĎæŰĞæąčãĂĆ

éIJĂèęĄæşÍæĎŔçŽĎæŸŕæIJňèŁĆåĘĚåőźåŘŁèČ¡éĂĆçŤÍäžŐUnixçşżçż§ïijŇåźűäÿŤäÿ■ä£İèŕĄæL'ĂæIJL
æŕŤåęĆæĹŚäžňåIJÍæţŃèŕŢçŽĎæŮűåĂŹïijŇåőĆèČ¡åIJÍLinuxäÿŁéİćæ■čåÿÿè£ŘèąŇïijŇä¡ĘæŸŕåIJÍOS
XäÿŁå■t'äÿ■èČ¡ãĂĆ

15.15 13.15 åŘŕåŁÍäÿĂäÿłWEBæţŔèğĹåŹÍ

éŮőécŸ

ä¡ăæČşéĂŽè£ĞèĎŽæIJňåŘŕåŁÍæţŔèğĹåŹÍåźűæL'ŞåijĂæŇĞåőŽçŽĎURLç¡Śéąţ

èğčåĘşæŰźæąĹ

`webbrowser` æÍąåİŮèČ¡èćńçŤÍæİěåŘŕåŁÍäÿĂäÿłæţŔèğĹåŹÍïijŇåźűäÿŤäÿŐåźşåŘřæŮăåĚşãĂĆä¿ŃåęĆ

```
>>> import webbrowser
>>> webbrowser.open('http://www.python.org')
True
>>>
```

åőČäijŽä¡£çŤÍéžŸèőd'æţŔèğĹåŹÍæL'ŞåijĂæŇĞåőŽç¡ŚéąţãĂĆåęĆæđIJä¡ăè£ŸæČşåŕźç¡ŚéąţæL'ŞåijĂæŰ

```
>>> # Open the page in a new browser window
>>> webbrowser.open_new('http://www.python.org')
True
>>>

>>> # Open the page in a new browser tab
>>> webbrowser.open_new_tab('http://www.python.org')
```

```
True
>>>
```

è£ŹæăůåřśåŕŕäżěæĽŞåijĂäÿĂäÿłæŰřçŽĎæţŕèğĹåŹĺçłŮåŕčæĹŰèĂĚæăĞç■¿ïijŇåŕłèęĄæţŕèğĹåŹĺæŤ

åęĆæđIJä¡ăæČşæŇĞåőŽæţŕèğĹåŹĺçśżåđŃïijŇåŕŕäżěä¡£çŤĺ `webbrowser.get()` åĞ¡æŢřæİěæŇĞåőŽæ§ŘäÿłçĽźåőŽæţŕèğĹåŹĺãĂĆä¿ŃåęĆïijŽ

```
>>> c = webbrowser.get('firefox')
>>> c.open('http://www.python.org')
True
>>> c.open_new_tab('http://docs.python.org')
True
>>>
```

åŕźäžŐæŤŕæŇĄçŽĎæţŕèğĹåŹĺåŘ■çğřåĹŮèąĺåŕŕæ§ěéŸĚ‘PythonæŰĞæąč <http://docs.python.org/3/library/webbrowser.html>‘_

èőĺèőž

åIJĺèĎŽæIJňäÿ■æĽŞåijĂæţŕèğĹåŹĺæIJĽæŮűåĂŹäijŽå¿ĹæIJĽçŤĺãĂĆä¿ŃåęĆïijŇæ§ŘäÿłèĎŽæIJňæĽ
ä¡ăæČşå£ńéĂ§æĽŞåijĂäÿĂäÿłæţŕèğĹåŹĺæİěçąőä£İåőČåůšçżŕæ■čåÿÿè£ŘèąŇäžĘãĂĆ
æĹŰèĂĚæŸŕæ§ŘäÿłçĺŃåžŕäżěHTMLç¡Śéąţæăijåijŕè¿ŞåĞžæŢřæ■őïijŇä¡ăæČşæĽŞåijĂæţŕèğĹåŹĺæ§ěçIJŃ
äÿ■çőąæŸŕäÿŁéİćåŞłçğ■æČĚåĘţïijŇä¡£çŤĺ `webbrowser` æĺąåİŮéČ¡æŸŕäÿĂäÿłçőĂå■ŢåőđçŤĺçŽĎèğčåĘşa

16 çňňå■ĄåŻŻçńăïijŽæţŃèŕŢãĂĄèřČèŕŢåŠŇåijĆåÿÿ

èŕŢéłŇè£ŸæŸŕå¿ĹæčŠçŽĎïijŇä¡ĘæŸŕèřČèŕŢïij§åřśæšąéĆčäźĹæIJĽèűčäžĘãĂĆäžŃåőđæŸŕïijŇåIJĺPyth

Contents:

16.1 14.1 æţŃèŕŢstdoutè¿ŞåĞž

éŮőéćŸ

ä¡ăçŽĎçĺŃåžŕäÿ■æIJĽäÿłæŰźæşŢäijŽè¿ŞåĞžåĹřæăĞåĞĘè¿ŞåĞžäÿ■ïijĹsys.stdoutïijĽãĂĆäź§åřśæŸŕè
ä¡ăæČşåĘźäÿłæţŃèŕŢæİěèŕĄæŸŐåőČïijŇçżŹåőŽäÿĂäÿłè¿ŞåĚěïijŇçŻÿåžŤçŽĎè¿ŞåĞžèČ¡æ■čåÿÿæŸ¿çď žå

èğčåĘşæŰźæąĹ

ä¡£çŤĺ `unittest.mock` æĺąåİŮäÿ■çŽĎ `patch()` åĞ¡æŢřïijŇ
ä¡£çŤĺèţůæİěéİđåÿÿçőĂå■ŢïijŇåŕŕäżěäÿžå■ŢäÿłæţŃèŕŢæĺąæŃ§ `sys.stdout`
çĎűåŘŐåŻđæżŽïijŇ åźűäÿŤäÿ■äžğçŤ§åď ğéĞŔçŽĎäÿť æŮűåŔŸéĞŔæĹŰåIJĺæţŃèŕŢçŤĺä¿ŃçŻť æŐěæŽť éIJ

ä¡IJäÿžäÿĂäÿłä¿Ńå■ŘïijŇæĹŚäżňåIJĺ `mymodule` æĺąåİŮäÿ■åőŽäźĽåęĆäÿŃäÿĂäÿłåĞ¡æŢřïijŽ

```
# mymodule.py

def urlprint(protocol, host, domain):
    url = '{}://{}.{}'.format(protocol, host, domain)
    print(url)
```

éżŸèőď æČĚåĘţäÿŃåĘĚç¡őçŽĎ print åĞ¡æŢřäijŽåřĘè¿ŞåĞžåŔŚéĂĄåĹř sys.stdout ãĂĆ äÿžäžĘæţŃèŕŢè¿ŞåĞžçIJ§çŽĎåIJĺéĆčéĞŇïijŇä¡ăåŔŕäżěä¡£çŤĺäÿĂäÿłæŻ£èžńåŕźèśąæİěæĺąæŃ ä¡£çŤĺ unittest.mock æĺąåİŮçŽĎ patch() æŰźæşŢåŔŕäżěå¿ĹæŰźä¿£çŽĎåIJĺæţŃèŕŢè£ŘèąŇçŽĎäÿŁ åźűäÿŤå¡ŞæţŃèŕŢåőŇæĹŘæŮűåĂŹèĞłåŁĺè£ŤåŻđåőČäżňçŽĎåŐ§æIJĽçŁűæĂĄãĂĆäÿŃéİćæŸŕåŕź mymodule æĺąåİŮçŽĎæţŃèŕŢäżčçăĄïijŽ

```
from io import StringIO
from unittest import TestCase
from unittest.mock import patch
import mymodule

class TestURLPrint(TestCase):
    def test_url_gets_to_stdout(self):
        protocol = 'http'
        host = 'www'
        domain = 'example.com'
        expected_url = '{}://{}.{}\n'.format(protocol, host, domain)

        with patch('sys.stdout', new=StringIO()) as fake_out:
            mymodule.urlprint(protocol, host, domain)
            self.assertEqual(fake_out.getvalue(), expected_url)
```

èőĺèőž

urlprint() åĞ¡æŢřæŐěåŔŮäÿĽäÿłåŔĆæŢřïijŇæţŃèŕŢæŰźæşŢåijĂåğŃäijŽåĚĹèő¿ç¡őæŕŔäÿĂäÿłåĺ expected_url åŔŸéĞŔèćńèő¿ç¡őæĹŘåŇĚåŘńæIJ§æIJŻçŽĎè¿ŞåĞžçŽĎå■ŮçňęäÿšãĂĆ

unittest.mock.patch() åĞ¡æŢřèćńçŤĺä¡IJäÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇä¡£çŤĺ StringIO åŕźèśąæİěäżčæŻ£ sys.stdout . fake_out åŔŸéĞŔæŸŕåIJĺèŕěè£ŻçĺŃäÿ■èćńåĹŻåżžçŽĎæĺąæŃ§åŕźèśąãĂĆ åIJĺwith-èŕ■åŔěäÿ■ä¡£çŤĺåőČåŔŕäżěæĽğèąŇåŘĎçğ■æčĂæ§ěãĂĆå¡Şwithèŕ■åŔěçżŞæİ§æŮűïijŇpatch äijŽåřĘæĽĂæIJĽäÿIJèě£æĄćåď■åĹřæţŃèŕŢåijĂåğŃåĽ■çŽĎçŁűæĂĄãĂĆ æIJĽäÿĂçĆźéIJĂèęĄæşĺæĎŔçŽĎæŸŕæ§ŘäžŻåŕźPythonçŽĎCæĽľåśŢåŔŕèČ¡äijŽå£¡çŢěæŐĽ sys.stdout çŽĎéĚ■ç¡őäžŇçŻťæŐěåĘŹåĚěåĹřæăĞåĞĘè¿ŞåĞžäÿ■ãĂĆ éŹŘäžŐçŕĞåźĚïijŇæIJňèŁĆäÿ■äijŽæűĽåŔŁåĹřè£ŹæŰźéİćçŽĎèőšèğčïijŇåőČéĂĆçŤĺäžŐçžŕPythonäżčçăĄ åęĆæđIJä¡ăçIJ§çŽĎéIJĂèęĄåIJĺCæĽľåśŢäÿ■æ■ŢèŐůI/OïijŇä¡ăåŔŕäżěåĚĹæĽŞåijĂäÿĂäÿłäÿťæŮűæŰĞäżű æŻťåďŽåĚşäžŐæ■ŢèŐůäżěå■Ůçňęäÿšå¡ćåijŔæ■ŢèŐůI/OåŠŇ StringIO åŕźèśąèŕůåŔĆéŸĚ5.6åřŔèŁĆãĂĆ

16.2 14.2 åIJĺå■ŢåĚČæţŃèŕŢäÿ■çżŹåŕźèśąæĽŞèąĕäÿĄ

éŮőécŸ

ä¡ăåĘŹçŽĎå■ŢåĚČæţŃèŕŢäÿ■éIJĂèęĄçżŹæŇĞåőŽçŽĎåŕźèśąæĽŞèąĕäÿĄïijŇ
çŤĺæİěæŰ■èĺĂåőČäžňåIJĺæţŃèŕŢäÿ■çŽĎæIJ§æIJŻèąŇäÿžïijĹæŕŤåęĆïijŇæŰ■èĺĂèćńèřČçŤĺæŮűçŽĎåŔĆæŢ

èğčåĘşæŰźæąĹ

unittest.mock.patch() åĞ¡æŢřåŔŕèćńçŤĺæİěèğčåĘşè£ŹäÿłéŮőécŸãĂĆ
patch() è£ŸåŔŕèćńçŤĺä¡IJäÿĂäÿłèčĚéěřåŹĺãĂĄäÿŁäŸŃæŰĞçőąçŘĘåŹĺæĹŰå■ŢçŃňä¡£çŤĺïijŇåř¡çőąåźűă
ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłářĘåőČå¡ŞåĄŽèčĚéěřåŹĺä¡£çŤĺçŽĎä¿Ńå■ŘïijŽ

```
from unittest.mock import patch
import example

@patch('example.func')
def test1(x, mock_func):
    example.func(x)       # Uses patched example.func
    mock_func.assert_called_with(x)
```

åőČè£ŸåŔŕäžěèćńå¡ŞåĄŽäÿĂäÿłäÿŁäŸŃæŰĞçőąçŘĘåŹĺïijŽ

```
with patch('example.func') as mock_func:
    example.func(x)      # Uses patched example.func
    mock_func.assert_called_with(x)
```

æIJĂåŘŐïijŇä¡ăè£ŸåŔŕäžěæĽŃåŁĺçŽĎä¡£çŤĺåőČæĽŞèąĕäÿĄïijŽ

```
p = patch('example.func')
mock_func = p.start()
example.func(x)
mock_func.assert_called_with(x)
p.stop()
```

åęĆæđIJåŔŕèČ¡çŽĎèŕİïijŇä¡ăèČ¡åd'§åŘăåŁăèčĚéěřåŹĺåŠŇäÿŁäŸŃæŰĞçőąçŘĘåŹĺæİěçżŹåd'Žäÿłářźèś

```
@patch('example.func1')
@patch('example.func2')
@patch('example.func3')
def test1(mock1, mock2, mock3):
    ...

def test2():
    with patch('example.patch1') as mock1, \
         patch('example.patch2') as mock2, \
         patch('example.patch3') as mock3:
    ...
```

èőĺèőž

patch() æŐěåŔŮäÿĂäÿłåůšå■ŸåIJĺåŕźèśąçŽĎåĚĺèůŕå¿ĎåŘ■ïijŇåřĘåĚűæŻ£æ■ćäÿžäÿĂäÿłæŰřçŽĎåĂ
åŐ§æİěçŽĎåĂijäijŽåIJĺèčĚéěřåŹĺåĞ¡æŢřæĹŰäÿŁäÿŃæŰĞçőąçŘĘåŹĺåőŇæĹŘåŘŐèĞłåŁĺæĄćåď■åŻđæİěã
éżŸèőď æČĚåĘţäÿŃïijŇæĽĂæIJĽåĂijäijŽèćń MagicMock åőđä¿ŃæŻ£äżčãĂĆä¿ŃåęĆïijŽ

```
>>> x = 42
>>> with patch('__main__.x'):
...     print(x)
...
<MagicMock name='x' id='4314230032'>
>>> x
42
>>>
```

äÿ■è£ĞïijŇä¡ăåŔŕäżěéĂŽè£ĞçżŹ patch() æŔŘä¿ŻçňňäžŇäÿłåŔĆæŢřæİěåřĘåĂijæŻ£æ■ćæĹŘäżżä¡Ţ

```
>>> x
42
>>> with patch('__main__.x', 'patched_value'):
...     print(x)
...
patched_value
>>> x
42
>>>
```

èćńçŤĺæİěä¡IJäÿžæŻ£æ■ćåĂijçŽĎ MagicMock åőđä¿ŃèČ¡åď§æĺąæŃ§åŔŕèřČçŤĺåŕźèśąåŠŇåőđä¿ŃãĂ
äżŰäżňèőřå¡ŢåŕźèśąçŽĎä¡£çŤĺä£ąæĄŕåźűåĚĄèőÿä¡ăæĽğèąŇæŰ■èĺĂæčĂæ§ěïijŇä¿ŃåęĆïijŽ

```
>>> from unittest.mock import MagicMock
>>> m = MagicMock(return_value = 10)
>>> m(1, 2, debug=True)
10
>>> m.assert_called_with(1, 2, debug=True)
>>> m.assert_called_with(1, 2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File ".../unittest/mock.py", line 726, in assert_called_with
    raise AssertionError(msg)
AssertionError: Expected call: mock(1, 2)
Actual call: mock(1, 2, debug=True)
>>>

>>> m.upper.return_value = 'HELLO'
>>> m.upper('hello')
'HELLO'
>>> assert m.upper.called

>>> m.split.return_value = ['hello', 'world']
>>> m.split('hello world')
```

```
['hello', 'world']
>>> m.split.assert_called_with('hello world')
>>>

>>> m['blah']
<MagicMock name='mock.__getitem__()' id='4314412048'>
>>> m.__getitem__.called
True
>>> m.__getitem__.assert_called_with('blah')
>>>
```

äÿĂèĹňæİěèőšïijŇè£ŹäžŻæŞ■ä¡IJäijŽåIJĺäÿĂäÿłå■ŢåĚČæţŃèŕŢäÿ■åőŇæĹŘãĂĆä¿ŃåęĆïijŇåĄĞèő¿ä¡ä

```
# example.py
from urllib.request import urlopen
import csv

def dowprices():
    u = urlopen('http://finance.yahoo.com/d/quotes.csv?s=@^DJI&f=sl1
↪')
    lines = (line.decode('utf-8') for line in u)
    rows = (row for row in csv.reader(lines) if len(row) == 2)
    prices = { name:float(price) for name, price in rows }
    return prices
```

æ■čåÿÿæİěèőšïijŇè£ŹäÿłåĞ¡æŢřäijŽä¡£çŤĺ urlopen() äžŐWe-
bäÿŁéİćèŐůåŔŰæŢřæ■őåźűèğčædŘåőČãĂĆ åIJĺå■ŢåĚČæţŃèŕŢäÿ■ïijŇä¡ääŕŕäžěçżŹåőČäÿĂäÿłécĎåĚĹåőŽ

```
import unittest
from unittest.mock import patch
import io
import example

sample_data = io.BytesIO(b'''\
"IBM",91.1\r
"AA",13.25\r
"MSFT",27.72\r
\r
''')

class Tests(unittest.TestCase):
    @patch('example.urlopen', return_value=sample_data)
    def test_dowprices(self, mock_urlopen):
        p = example.dowprices()
        self.assertTrue(mock_urlopen.called)
        self.assertEqual(p,
                         {'IBM': 91.1,
                          'AA': 13.25,
                          'MSFT' : 27.72})

if __name__ == '__main__':
```

```
    unittest.main()
```

æIJñä¿Ńäÿ■ïijÑä¡■äžŐ example æĺąåİŮäÿ■çŽĎ urlopen() åĞ¡æŢřèćńäÿĂäÿłæĺąæŃ§åŕźèśąæŻ£äžčïijÑ èŕěåŕźèśąäijŽè£ŤåŻđäÿĂäÿłåŇĚåŘńæţŃèŕŢæŢřæ■őçŽĎ ByteIO().

è£ŸæIJL'äÿĂçĆźïijÑåIJĺæL'Şèąěäÿ ĄæŮűæĹŚäžňä¡£çŤĺäžĘ example.urlopen æİěäžčæŻ£ urllib.request.urlopen ãĂĆ å¡Şä¡ääĹŻåžžèąěäÿĄçŽĎæŮűåĂŹïijÑä¡ää£ĚéążäĄ£çŤĺåőČäžňåIJĺæţŃèŕŢäžčçăĄäÿ■çŽĎåŘ■çğřãĂĆ çŤśäžŐæţŃèŕŢäžčçăĄä¡£çŤĺäžĘ from urllib.request import urlopen ,éĆčäźĹ dowprices() åĞ¡æŢř äÿ■ä¡£çŤĺçŽĎ urlopen() åĞ¡æŢřåőđéŹĚäÿŁåŕśä¡■äžŐ example æĺąåİŮäžĘãĂĆ

æIJňèŁĆåőđéŹĚäÿŁåŘłæŸŕåŕź unittest.mock æĺąåİŮçŽĎäÿĂæňąæţĚåŕİè¿Ďæ■ćãĂĆ æŻťåď'ŽæŻťéńŸçžğçŽĎçL'źæĂğïijÑèŕůåŘĆèĂČ åőŸæŰźæŰĞæąč

16.3 14.3 åIJĺå■ŢåĚČæţŃèŕŢäÿ■æţŃèŕŢåijĆåÿÿæČĚåĘţ

éŮőécŸ

ä¡ăæČşåĘŹäÿłæţŃèŕŢçŤĺä¿ŃæİěåĞĘçąőçŽĎåĹd'æŰ■æ§ŘäÿłåijĆåÿÿæŸŕåŘęècńæŁŻåĞžãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐåijĆåÿÿçŽĎæţŃèŕŢåŘŕä¡£çŤĺ assertRaises() æŰźæşŢãĂĆ ä¿ŃåęĆïijÑåęĆæđIJä¡ăæČşæţŃèŕŢæ§ŘäÿłåĞ¡æŢřæŁŻåĞžäžĘ ValueError åijĆåÿÿïijÑåČŘäÿŃéİćè£ŹæăůåĘŹïijŽ

```
import unittest

# A simple function to illustrate
def parse_int(s):
    return int(s)

class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        self.assertRaises(ValueError, parse_int, 'N/A')
```

åęĆæđIJä¡ăæČşæţŃèŕŢåijĆåÿÿçŽĎåĚůä¡ŞåĂijïijÑéIJĂèęĄçŤĺåĹřåŘęåď'ŰäÿĂçğ■æŰźæşŢïijŽ

```
import errno

class TestIO(unittest.TestCase):
    def test_file_not_found(self):
        try:
            f = open('/file/not/found')
        except IOError as e:
            self.assertEqual(e.errno, errno.ENOENT)
```

```
        else:
            self.fail('IOError not raised')
```

èőÍèőž

assertRaises() æŰźæşŢäÿžæţŃèŕŢåijĆåÿÿå■ŸåIJÍæĂğæŔŘä¿ŻäžĘäÿĂäÿłçőĂä¿£æŰźæşŢãĂĆ äÿĂäÿłåÿÿèğĄçŽĎéŹůéŸśæŸŕæL'ŃåŁĺåŐżè£ŻèąŇåijĆåÿÿæčĂæţÑãĂĆæŕŤåęĆïijŽ

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        try:
            r = parse_int('N/A')
        except ValueError as e:
            self.assertEqual(type(e), ValueError)
```

è£Źçğ■æŰźæşŢçŽĎéŮőécŸåIJĺäžŐåőČå¿Ĺåőźæ ŸŞéĄŮæijŔåĚűäżŰæČĚåĘţïijÑæŕŤåęĆæšąæIJL'äżżä¡T éĆčäźĹä¡ăè£Ÿå¿ŮéIJĂèęĄåćđåŁăåŔęåd'ŰçŽĎæčĂæţŃè£ĞçÍÑïijÑåęĆäÿÑéÍćè£Źæăůïijž

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        try:
            r = parse_int('N/A')
        except ValueError as e:
            self.assertEqual(type(e), ValueError)
        else:
            self.fail('ValueError not raised')
```

assertRaises() æŰźæşŢäijŽåd'ĎçŘĘæL'ĂæIJL'çżĘèŁĆïijŇåŻăæ■d'ä¡ăåžŤèŕëä¡£çŤĺåőČăĂĆ

assertRaises() çŽĎäÿĂäÿłçijžçĆźæŸŕåőČæţÑäÿ■äžĘåijĆåÿÿåĚůä¡ŞçŽĎåĂijæŸŕåd'ŽåŕŠãĂĆ äÿžäžĘæţÑèŕŢåijĆåÿÿåĂijïijÑåŔŕäżëä¡£çŤÍ assertRaisesRegex() æŰźæşŢïijÑ åőČåŔŕåŘÑæŮűæţÑèŕŢåijĆåÿÿçŽĎå■ŸåIJÍäžëåŔŁéĂŽè£Ğæ■čåĹŹåijŔåÑźéĚ■åijĆåÿÿçŽĎå■Ůçňęäÿšèąĺçd

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        self.assertRaisesRegex(ValueError, 'invalid literal .*',
                               parse_int, 'N/A')
```

assertRaises() åŠÑ assertRaisesRegex() è£ŸæIJL'äÿĂäÿłåőźæŸŞå£¡çŢęçŽĎåIJřæŰźářśæŸŕåőČäžñè£ŸèČ¡èćńå¡ŞåĄŽäÿŁäÿÑæŰĞçőąçŘĘåŹĺä¡£çŤĺ

```
class TestConversion(unittest.TestCase):
    def test_bad_int(self):
        with self.assertRaisesRegex(ValueError, 'invalid literal .*
↪'):
            r = parse_int('N/A')
```

ä¡Eä¡äçŽĎæţÑèŕŢæűL'åŔŁåĹřåd'Žäÿłæl'ğèąÑæ■ěélď'çŽĎæŮűåĂŹè£Źçğ■æŰźæşŢářšå¿ĹæIJL'çŤĺäžĘą

16.4 14.4 åřĘæţŃèŕŢè¿ŞåĞžçŤĺæŮěå£Ůèőřå¡ŢåĹřæŰĞäżűäÿ■

éŮőéćŸ

ä¡ăåÿŇæIJŻåřĘå■ŢåĚČæţŃèŕŢçŽĎè¿ŞåĞžåĘŹåĹřåĹřæ§ŘäÿłæŰĞäżűäÿ■åŐżïijŇèĂŇäÿ■æŸŕæĽŞå■řå

èğčåĘşæŰźæąĹ

è£ŘèąŇå■ŢåĚČæţŃèŕŢäÿĂäÿłåÿÿèğĄæŁĂæIJŕåřśæŸŕåIJĺæţŃèŕŢæŰĞäżűåžŢéČĺåŁăåĚěäÿŃéİćè£Źæőţ

```
import unittest

class MyTest(unittest.TestCase):
    pass

if __name__ == '__main__':
    unittest.main()
```

è£ŹæăůçŽĎèŕİæţŃèŕŢæŰĞäżűåřśæŸŕåŔŕæĽğèąŇçŽĎïijŇåźűäÿŤäijŽåřĘè£ŘèąŇæţŃèŕŢçŽĎçżŞæđIJæ
åęĆæđIJä¡ăæČşéĞ■åőŽåŘŚè¿ŞåĞžïijŇåřśéIJĂèęĄåČŔäÿŃéİćè£Źæăůä£őæŤź main()
åĞ¡æŢřïijŽ

```
import sys

def main(out=sys.stderr, verbosity=2):
    loader = unittest.TestLoader()
    suite = loader.loadTestsFromModule(sys.modules[__name__])
    unittest.TextTestRunner(out,verbosity=verbosity).run(suite)

if __name__ == '__main__':
    with open('testing.out', 'w') as f:
        main(f)
```

èőĺèőž

æIJňèŁĆæĎ§åĚť'èűčçŽĎéČĺåĹĘåźűäÿ■æŸŕåřĘæţŃèŕŢçżŞæđIJéĞ■åőŽåŘŚåĹřäÿĂäÿłæŰĞäżűäÿ■ïijŇ
èĂŇæŸŕéĂŽè£Ğè£ŹæăůåĄŽåŘŚä¡ăåśŢçď'žäžĘ unittest
æĺąåİŮäÿ■äÿĂäžŻåĂijå¿ŮåĚşæşĺçŽĎåĘĚéČĺåůěä¡IJåŐ§çŘĘãĂĆ

unittest æĺąåİŮéęŰåĚĹäijŽçżĎèčĚäÿĂäÿłæţŃèŕŢåěŮäżűãĂĆ
è£ŹäÿłæţŃèŕŢåěŮäżűåŇĚåŘńäžĘä¡ăåőŽäźĽçŽĎåŘĎçğ■æŰźæşŢãĂĆäÿĂæŮęåěŮäżűçżĎèčĚåőŇæĹŘïijŇå

è£Źäÿď'æ■ěæŸŕåĹĘåijĂçŽĎïijŇunittest.TestLoader
åőđä¿ŃèćńçŤĺæİěçżĎèčĚæţŃèŕŢåěŮäżűãĂĆ loadTestsFromModule()
æŸŕåőČåőŽäźĽçŽĎæŰźæşŢäźŃäÿĂïijŇçŤĺæİěæŤűéŻĘæţŃèŕŢçŤĺä¿ŃãĂĆ åőČäijŽäÿž
TestCase çśżæĽńæŔŔæ§ŘäÿłæĺąåİŮåźűåřĘåĚűäÿ■çŽĎæţŃèŕŢæŰźæşŢæŔŘåŔŰåĞžæİěãĂĆ
åęĆæđIJä¡ăæČşè£ŻèąŇçżĘçšŠåžęçŽĎæŐğåĹűïijŇ åŔŕäżěä¡£çŤĺ
loadTestsFromTestCase() æŰźæşŢæİěäżŐæ§ŘäÿłçżğæĽ£TestCaseçŽĎçśżäÿ■æŔŘåŔŰæţŃèŕŢæŰź

TextTestRunner çśżæŸŕäÿĂäÿłæţŃèŕŢè£ŘèąŇçśżçŽĎä¿Ńå■ŘïijŇ
è£ŹäÿłçśżçŽĎäÿżèęĄçŢĺéĂŤæŸŕæL'ğèąŇæ§ŘäÿłæţŃèŕŢåěŮäżűäÿ■åŇĚåŘńçŽĎæţŃèŕŢæŰźæşŢãĂĆ
è£Źäÿłçśżèů§æL'ğèąŇ unittest.main() åĞ¡æŢřæL'Ăä¡£çŢĺçŽĎæţŃèŕŢè£ŘèąŇåŹĺæŸŕäÿĂæăůçŽĎãĂ
äÿ■è£ĞïijŇæĹŚäżňåIJĺè£ŹéĞŇåŕźåőČè£ŻèąŇäžĘäÿĂäžŻåĹŮåžŢåśĆéĚ■ç¡őïijŇåŇĚæŃňè¿ŞåĞžæŰĞäżűåŠ
ař¡çőąæIJňèŁĆä¿Ńå■Řäžčçă Ąå¿ĹařŚïijŇä¡ĘæŸŕèČ¡æŇĞåŕijä¡ăåęĆä¡Ţåŕź
unittest æąĘæđűè£ŻèąŇæŻť'è£ŻäÿĂæ■ěçŽĎèĞłåőŽäźL'ãĂĆ
èęĄæČşèĞłåőŽäźL'æţŃèŕŢåěŮäżűçŽĎèčĚéĚ■æŰźåijŔïijŇä¡ăåŔŕäžěåŕź TestLoader
çśżæL'ğèąŇæŻť'åď'ŽçŽĎæŞ■ä¡IJãĂĆ äÿžäžĘèĞłåőŽäźL'æţŃèŕŢè£ŘèąŇïijŇä¡ăåŔŕäžěæđĎéĂăäÿĂäÿłèĞłåů
TextTestRunner çŽĎåŁ§èČ¡ãĂĆ èĂŇè£ŹäžŻåůšçżŔèűĚåĞžäžĘæIJňèŁĆçŽĎèŇĆåŻť'ãĂĆunittest
æĺąåİŮçŽĎæŰĞæąčåŕźåžŢåśĆåőđçŐŕåŐ§çŘĘæIJL'æŻť'æůśåĚěçŽĎèőšèğčïijŇåŔŕäžěåŐžçIJŃçIJŃãĂĆ

16.5 14.5 å£¡çŢěæĹŰæIJ§æIJŻæţŃèŕŢåď'śèť'ě

éŮőécŸ

ä¡ăæČşåIJĺå■ŢåĚČæţŃèŕŢäÿ■å£¡çŢěæĹŰæăĞèőŕæ§ŘäžŻæţŃèŕŢäijŽæŇL'çĚğécĎæIJ§è£ŘèąŇåď'śèť'ěã

èğčåĘşæŰźæąĹ

unittest æĺąåİŮæIJL'èčĚéěřåŹĺåŔŕçŤĺæİěæŐğåĹűåŕźæŇĞåőŽæţŃèŕŢæŰźæşŢçŽĎåď'ĎçŘĘïijŇä¿Ń

```
import unittest
import os
import platform

class Tests(unittest.TestCase):
    def test_0(self):
        self.assertTrue(True)

    @unittest.skip('skipped test')
    def test_1(self):
        self.fail('should have failed!')

    @unittest.skipIf(os.name=='posix', 'Not supported on Unix')
    def test_2(self):
        import winreg

    @unittest.skipUnless(platform.system() == 'Darwin', 'Mac␣
↪specific test')
    def test_3(self):
        self.assertTrue(True)

    @unittest.expectedFailure
    def test_4(self):
        self.assertEqual(2+2, 5)

if __name__ == '__main__':
    unittest.main()
```

åęĆæđIJä¡ăåIJÍMacäÿŁè£ŘèąŇè£Źæőţäžčçă Ąïij Ňä¡ăäijŽå¿ŮåĹřåęĆäÿŃè¿ŞåĞžïijŽ

```
bash % python3 testsample.py -v
test_0 (__main__.Tests) ... ok
test_1 (__main__.Tests) ... skipped 'skipped test'
test_2 (__main__.Tests) ... skipped 'Not supported on Unix'
test_3 (__main__.Tests) ... ok
test_4 (__main__.Tests) ... expected failure

----------------------------------------------------------------------
 ↪--
Ran 5 tests in 0.002s

OK (skipped=2, expected failures=1)
```

èőÍèőž

`skip()` èčĚéěřåŹÍèČ¡èćńçŤÍæİěå£¡çŢěæ§Řäÿłä¡ăäÿ■æČşè£ŘèąŇçŽĎæţŃèŕŢãĂĆ `skipIf()` åŠŇ `skipUnless()` åŕźäžŐä¡ăåŘłæČşåIJÍæ§Řäÿłç L'źåőŽåźşåŘřæĹŰPythonçL'ĹæIJňæĹŰåĚű ä¡£çŤÍ `@expected` çŽĎåď śèť ěèčĚéěřåŹÍæİěæăĞèőřéĆčäžŻçąőåőŽäijŽåď śèť ěçŽĎæţŃèŕŢïijŇåźűäÿŤåŕźè£

å£¡çŢěæŰźæşŢçŽĎèčĚéěřåŹÍè£ŸåŘŕäžěèćńçŤÍæİěčĚéěřæŢť äÿłæţŃèŕŢçśžïijŇæŕŤåęĆïijŽ

```
@unittest.skipUnless(platform.system() == 'Darwin', 'Mac specific␣
 ↪tests')
class DarwinTests(unittest.TestCase):
    pass
```

16.6 14.6 åď ĎçŘĘåď ŽäÿłåijĆåÿÿ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłäžčçăĄçL'Ğæőţå ŔŕèČ¡äijŽæŁŻåĞžåď Žäÿłäÿ■åŘŇçŽĎåijĆåÿÿïijŇæĂŐæăůæL'■èČ¡äÿ■

èğčåĘşæŰźæąĹ

åęĆæđIJä¡ăåŔŕäžěçŤĺå■ŢäÿłäžčçăĄåİŮåď ĎçŘĘäÿ■åŘŇçŽĎåijĆåÿÿïijŇåŔŕäžěårĘåőČäžňæŤ¿åĚěäÿĂä

```
try:
    client_obj.get_url(url)
except (URLError, ValueError, SocketTimeout):
    client_obj.remove_url(url)
```

åIJÍè£Źäÿłä¿Ńå■Řäÿ■ïijŇåĚČçěŰäÿ■äżżä¡ŢäÿĂäÿłåijĆåÿÿåŔŠçŤ§æŮűéČ¡äijŽæĽğèąŇ `remove_url()` æŰźæşŢãĂĆ åęĆæđIJä¡ăæČşåŕźåĚűäÿ■æ§Řäÿłåij Ćåÿÿè£ŻèąŇäÿ■åŘŇçŽĎåď ĎçŘĘïijŇåF `except` èŕ■åŔěäÿ■ïijŽ

```
try:
    client_obj.get_url(url)
except (URLError, ValueError):
    client_obj.remove_url(url)
except SocketTimeout:
    client_obj.handle_url_timeout(url)
```

å¿Ĺåď ŽçŽĎåijĆåÿÿäijŽæIJĽåśĆçžğåĚşçşżïijŇåŕźäžŐè£Źçğ■æČĚåĘţïijŇä¡ăåŔŕèČ¡ä¡£çŤĺåőČäżňçŽĎäÿ

```
try:
    f = open(filename)
except (FileNotFoundError, PermissionError):
    pass
```

åŔŕäżěèćńéĞ■åĘŹäÿžïijŽ

```
try:
    f = open(filename)
except OSError:
    pass
```

`OSError` æŸŕ `FileNotFoundError` åŠŇ `PermissionError` åijĆåÿÿçŽĎå§žçśżãĂĆ

èőĺèőž

åř¡çőąåď ĎçŘĘåď Žäÿłåij Ćåÿÿæ IJňèžńåźűæšąäżĂäźĹçĽźæőŁçŽĎïijŇäÿ■è£Ğä¡ăåŔŕäżěä¡£çŤĺ `as` åĚşéŤőå■ŮæİěèŐůå¿Ůèćńæ ŁŻåĞžåijĆåÿÿçŽĎåijŢçŤĺïijŽ

```
try:
    f = open(filename)
except OSError as e:
    if e.errno == errno.ENOENT:
        logger.error('File not found')
    elif e.errno == errno.EACCES:
        logger.error('Permission denied')
    else:
        logger.error('Unexpected error: %d', e.errno)
```

è£Źäÿłä¿Ńå■Řäÿ■ïijŇ `e` åŔŸéĞŔæŇĞåŘŚäÿĂäÿłèćńæŁŻåĞžçŽĎ `OSError` åijĆåÿÿåőđä¿ŃãĂĆ è£Źäÿłå IJĺä¡ăæČşæŻť è£ŻäÿĂæ■ěåĹĘæđŘè£Źäÿłåij Ćåÿÿç ŽĎæŮűåĂŹäijŽå¿ĹæIJĽçŤĺï

åŘŇæŮűè£ŸèęĄæşĺæĎŔçŽĎæŮűåĂŹ `except` èŕ■åŔěæŸŕéąžåžŔæčĂæ§ěçŽĎïijŇçńňäÿĂäÿłåŇźéĚ■ç ä¡ăåŔŕäżěå¿Ĺåőźæ ŸŞçŽĎæđĎéĂăåď Žäÿł `except` åŘŇæŮűåŇźéĚ■çŽĎæČĚå¡ćïijŇærŤåęĆïijŽ

```
>>> f = open('missing')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
FileNotFoundError: [Errno 2] No such file or directory: 'missing'
>>> try:
...     f = open('missing')
... except OSError:
...     print('It failed')
... except FileNotFoundError:
...     print('File not found')
...
It failed
>>>
```

è£ŹéĞŇçŽĎ `FileNotFoundError` èŕ■åŔěåźűæšąæIJL'æL'ğèąŇçŽĎåŐ§åŻăæŸŕ `OSError` æŻť'äÿĂèĹňïijŇåőČåŔŕåŇźéĚ■ `FileNotFoundError` åijĆåÿÿïijŇäžŐæŸŕåřśæŸŕçňňäÿĂäÿłåŇźéĚ■çŽĎãĂĆ åIJĺèřČèŕŢçŽĎæŮűåĂŹïijŇåęĆæđIJä¡ăářźæ§Řäÿłç L'źåőŽåijĆåÿÿä¡ăåŔŕäžěéĂŽè£ĞæšěçIJŃèŕěåijĆåÿÿçŽĎ `__mro__` åśđæĂğæİěå£ńéĂ§æţŔèğĹãĂĆæŕŤåęĆïijŽ

```
>>> FileNotFoundError.__mro__
(<class 'FileNotFoundError'>, <class 'OSError'>, <class 'Exception'>
 ↪,
 <class 'BaseException'>, <class 'object'>)
>>>
```

äÿŁéİćåĹŮèąĺäÿ■äżżä¡ŢäÿĂäÿłçŻť'åĹř `BaseException` çŽĎçśžéČ¡èČ¡èćńçŤĺäžŐ `except` èŕ■åŔěãĂĆ

16.7 14.7 æ■ŢèŐůæL'ĂæIJL'åijĆåÿÿ

éŮőécŸ

æĂŐæăůæ■ŢèŐůäžčçăĄäÿ■çŽĎæL'ĂæIJL'åijĆåÿÿïij§

èğčåĘşæŰźæąĹ

æČşèęĄæ■ŢèŐůæL'ĂæIJL'çŽĎåijĆåÿÿïijŇåŔŕäžěçŻť'æŐěæ■ŢèŐů `Exception` å■şåŔŕïijŽ

```
try:
    ...
except Exception as e:
    ...
    log('Reason:', e)       # Important!
```

è£ŹäÿłåřĘäijŽæ■ŢèŐůéŹď'äžĘ `SystemExit` ãĂĄ `KeyboardInterrupt` åŠŇ `GeneratorExit` äźŇåď'ŰçŽĎæL'ĂæIJL'åijĆåÿÿãĂĆ

åęĆæđIJä¡ăè£ŸæČşæ■ŢèŐůè£ŹäÿĽäÿłåijĆåÿÿïijŇářĘ `Exception` æŤźæĹŘ `BaseException` å■şåŔŕãĂĆ

èőÍèőž

æ■ŢèŐůæĽĂæIJL'åijĆåÿÿéĂŽåÿÿæŸŕçŤśäžŐçÍŇåžŔåŚŸåIJÍæ§ŘäžŻåď■æİĆæŞ■ä¡IJäÿ■åźűäÿ■èČ¡èőř
åęĆæđIJä¡ääÿ■æŸŕå¿ĹçżĘå£ČçŽĎäžžïijŇè£Źäź§æŸŕçijŰåĘŹäÿ■æŸŞèřČèŕŢäžčçăĄçŽĎäÿĂäÿłçőĂå■ŢæŰź

æ■čåŻăåęĆæ■ď'ïijŇåęĆæđIJä¡ăéĂĽ'æŃľæ■ŢèŐůæĽĂæIJL'åijĆåÿÿïijŇéĆčäźĹåIJÍæ§Řäÿłå IJřæŰźïijĹæ
åęĆæđIJä¡ăæšąæIJL'è£Źæăůå ĄŽïijŇæIJL'æŮűåĂŹä¡ăçIJŃåĹřåijĆåÿÿæĽŞå■řæŮűåŔŕèČ¡æŚÿäÿ■çİĂåď'èĎ

```
def parse_int(s):
    try:
        n = int(v)
    except Exception:
        print("Couldn't parse")
```

èŕŢçİĂè£ŘèąŇè£ŹäÿłåĞ¡æŢřïijŇçżŞæđIJåęĆäÿŃïijŽ

```
>>> parse_int('n/a')
Couldn't parse
>>> parse_int('42')
Couldn't parse
>>>
```

è£ŹæŮűåĂŹä¡ăåřśäijŽæŇăåď'æČşïijŽâĂIJè£ŹåŠŃåŻđăžŃåŢŁïij§âĂİ
åĄĞåęĆä¡ăåČŔäÿŃéİćè£ŹæăúéĞ■åĘŹè£ŹäÿłåĞ¡æŢřïijŽ

```
def parse_int(s):
    try:
        n = int(v)
    except Exception as e:
        print("Couldn't parse")
        print('Reason:', e)
```

è£ŹæŮűåĂŹä¡ăèČ¡èŐůåŔŰåęĆäÿŃè¿ŞåĞžïijŇæŇĞæŸŐäžĘæIJL'äÿłçijŰçÍŃéŤŹèŕřïijŽ

```
>>> parse_int('42')
Couldn't parse
Reason: global name 'v' is not defined
>>>
```

å¿ĹæŸŐæŸ¿ïijŇä¡ăåžŤèŕěår¡åŔŕèČ¡årĘåijĆåÿÿåď'ĎçŘĘåŹĺåőŽäźL'çŽĎçš¿åĞĘäÿĂäžŻãĂĆ
äÿ■è£ĞïijŇèęĄæŸŕä¡ăå£Ěéążæ■ŢèŐůæĽĂæIJL'åijĆåÿÿïijŇçąőä£ÍæL'Şå■řæ■čçąőçŽĎèŕŁæŰ■ä£ąæĄŕæĹŰå

16.8 14.8 åĹŻåžžèĞłåőŽäźĽåijĆåÿÿ

éŮőécŸ

åIJĺä¡ăæđĎåžžçŽĎåžŤçŤĺçĺŃåžŔäÿ■ïijŇä¡ăæČşářĘåžŢåśĆåijĆåÿÿåŇĚèčĚæĹŘèĞłåőŽäźĽçŽĎåijĆåÿÿ

èğčåĘşæŰźæąĹ

åĹŻåžžæŰřçŽĎåijĆåÿÿå¿ĹçőĂå■ŢâĂŤâĂŤåőŽäźĽæŰřçŽĎçśżïijŇèőľåőČçżğæL'£èĞł
`Exception` ïijĹæĹŰèĂĚæŸŕäžża¡ŢäÿĂäÿłåůšå■ŸåIJĺçŽĎåijĆåÿÿçśżåđŃïijL'ãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăçijŰåĘŹç¡ŚçżIJçŻÿåĚşçŽĎçĺŃåžŔïijŇä¡ăåŔŕèČ¡äijŽåőŽäźĽäÿĂäžŻçśżäijijåęĆäÿŃçŽ

```
class NetworkError(Exception):
    pass

class HostnameError(NetworkError):
    pass

class TimeoutError(NetworkError):
    pass

class ProtocolError(NetworkError):
    pass
```

çĎűåŘŐçŤĺæĹůářśåŔŕäžěåČŔéĂŽåÿÿéĆčæăůä¡£çŤĺè£ŹäžŻåijĆåÿÿäžĘïijŇä¿ŃåęĆïijŽ

```
try:
    msg = s.recv()
except TimeoutError as e:
    ...
except ProtocolError as e:
    ...
```

èŐĺèőž

èĞłåőŽäźĽåijĆåÿÿçśżåžŤèŕěæĂżæŸŕçżğæL'£èĞłåĘĚç¡őçŽĎ `Exception`
çśżïijŇ æĹŰèĂĚæŸŕçżğæL'£èĞłéĆčäžŻæIJňèžńářśæŸŕäžŐ `Exception`
çżğæL'£èĂŇæİěçŽĎçśżãĂĆ ář¡çőąæL'ĂæIJL'çśżåŘŇæŮűäź§çżğæL'£èĞł `BaseException`
ïijŇä¡Ęä¡ääÿ■åžŤèŕěä¡£çŤĺè£Źäÿłå§žçśżæİěåőŽäźĽæŰřçŽĎåijĆåÿÿãĂĆ `BaseException`
æŸŕäÿžçşżçż§éĂĂåĞžåijĆåÿÿèĂŇä£ĺçŢŹçŻĎïijŇæŕŤåęĆ `KeyboardInterrupt` æĹŰ
`SystemExit` äžěåŘŁåĚűäžŰéĆčäžŻäijŽçżŹåžŤçŤĺåŔŚéĂĄä£ąåŘůèĂŇéĂĂåĞžçŽĎåijĆåÿÿãĂĆ
åŽăæ■đ'ïijŇæ■ŢèŐůè£ŹäžŻåijĆåÿÿæIJňèžńæšąäžĂäźĹæĎŘäźĽ'ãĂĆ
è£ŹæăůçŽĎèŕİïijŇåĄĞåęĆä¡ăçżğæL'£`BaseException`åŕŕèČ¡äijŽåřijèĞť'ä¡ăçŽĎèĞłåőŽäźĽåijĆåÿÿäÿ■ä

åIJĺçĺŃåžŔäÿ■åijŢåĚěèĞłåőŽäźĽåijĆåÿÿåŘŕäžěä¡£å¿Ůä¡ăçŽĎäžčçăĄæŻť'åĚůåŘŕèŕźæĂğïijŇèČ¡æÿĚæ
è£ŸæIJL'äÿĂçğ■èő¿èőąæŸŕåřĘèĞłåőŽäźĽåijĆåÿÿéĂŽè£ĞçżğæL'£çżĎåŘĹèţůæİěãĂĆåIJĺåď■æİĆåžŢçŤĺçĺŃ
ä¡£çŤĺå§žçśżæİěåĹĘçżĎåŘĎçğ■åijĆåÿÿçśżäź§æŸŕå¿ĹæIJL'çŤĺçŽĎãĂĆåőČåŘŕäžěèőľçŤĺæĹůæ■ŢèŐůäÿĂä

```
try:
    s.send(msg)
except ProtocolError:
    ...
```

ä¡äè£ŸèČ¡æ■ŢèŐůæŻť'åď'ğèŇČåŻť'çŽĎåijĆåÿÿïijŇåřśåČŕäÿŃéİćè£ŹæăůïijŽ

```
try:
    s.send(msg)
except NetworkError:
    ...
```

åęĆæđIJä¡äæČşåőŽäźL'çŽĎæŰřåijĆåÿÿéĞ■åĘŹäžĘ `__init__()` æŰźæşŢïijŇ çąőä£İä¡ää¡£çŤĺæL'ĂæIJL'åŔĆæŢřèřČçŤĺ `Exception.__init__()` ïijŇä¿ŃåęĆïijŽ

```
class CustomError(Exception):
    def __init__(self, message, status):
        super().__init__(message, status)
        self.message = message
        self.status = status
```

çIJŃäÿŁåŐżæIJL'çĆźåěĞæĂłïijŇäÿ■è£ĞExceptionçŽĎéżŸèőď'èąŇäÿžæŸŕæŐěåŔŮæL'ĂæIJL'äijäéĂŠç `.args` åśđæĂğäÿ■. å¿Ĺåď'ŽåĚűäžŰåĞ¡æŢřåžŞåŠŇéČĺåĹĘPythonåžŞéżŸèőď'æL'ĂæIJL'åijĆåÿÿéČ¡å£Ěéąża `.args` åśđæĂğïijŇ åŻäæ■ď'åęĆæđIJä¡ää£¡çŢěäžĘè£ŹäÿĂæ■ěïijŇä¡ääijŽåŔŠçŐřæIJL'äžŻæŮűåĂŹä¡ääőŽäź äÿžäžĘæijŤçď'ž `.args` çŽĎä¡£çŤĺïijŇèĂČèŹŠäÿŃäÿŃéİćè£ŹäÿłäŁ¡£çŤĺåĘĚç¡őçŽĎ *Run-timeError*' åijĆåÿÿçŽĎäžď'äžŠäijŽèŕĺïijŇ æşĺæĎŔçIJŃraiseèŕ■åŔëäÿ■ä¡£çŤĺçŽĎåŔĆæŢřäÿłæŢřæŸŕæĂŐæă

```
>>> try:
...     raise RuntimeError('It failed')
... except RuntimeError as e:
...     print(e.args)
...
('It failed',)
>>> try:
...     raise RuntimeError('It failed', 42, 'spam')
... except RuntimeError as e:

...     print(e.args)
...
('It failed', 42, 'spam')
>>>
```

åĚşäžŐåĹŻåżżèĞłåőŽäźL'åijĆåÿÿçŽĎæŻť'åď'Žä£qæĄŕïijŇèŕůåŔĆèĂČ'PythonåőŸæŰźæŰĞæąč <https://docs.python.org/3/tutorial/errors.html>'_

16.9 14.9 æ■ŢèŐůåijĆåÿÿåŘŐæŁŻåĞžåŔęåď ŰçŽĎåijĆåÿÿ

éŮőéćŸ

ä¡ăæČşæ■ŢèŐůäÿĂäÿłåijĆåÿÿåŘŐæŁŻåĞžåŔęåď ŰäÿĂäÿłäÿ■åŘŇçŽĎåijĆåÿÿïijŇåŘŇæŮűè£Ÿå¿ŮåIJ

èğčåĘşæŰźæąĹ

äÿžäžĘéŞ¿æŐěåijĆåÿÿïijŇä¡£çŤĺ `raise from` èŕ■åŔěæİěäżčæŻ£çőĂå■ŢçŽĎ `raise` èŕ■åŔěãĂĆ åőČäijŽèőľä¡ăåŘŇæŮűä£İçŢŹäÿď äÿłåijĆåÿÿçŽĎä£ąæĄŕãĂĆä¿ŃåęĆïijŽ

```
>>> def example():
...     try:
...             int('N/A')
...     except ValueError as e:
...             raise RuntimeError('A parsing error occurred') from␣
↪e
...
>>> example()
Traceback (most recent call last):
  File "<stdin>", line 3, in example
ValueError: invalid literal for int() with base 10: 'N/A'
```

äÿŁéİćçŽĎåijĆåÿÿæŸŕäÿŃéİćçŽĎåijĆåÿÿäžğçŤ§çŽĎçŻť æŐěåŐ§åŻăïijŽ

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 5, in example
RuntimeError: A parsing error occurred
>>>
```

åIJĺåŻđæžŕäÿ■åŔŕäżěçIJŃåĹřïijŇäÿď äÿłåijĆåÿÿéČ¡èćńæ■ŢèŐůãĂĆ èęĄæČşæ■ŢèŐůè£ŹæăůçŽĎåijĆåÿÿïijŇä¡ăåŔŕäżěä¡£çŤĺäÿĂäÿłçőĂå■ŢçŽĎ `except` èŕ■åŔěãĂĆ äÿ■è£ĞïijŇä¡ăè£ŸåŔŕäżěéĂŽè£Ğæ§ěçIJŃåijĆåÿÿåŕźèśąçŽĎ `__cause__` åśđæĂğæİěèů§èÿłåijĆåÿÿéŞ¿ãĂĆä¿ŃåęĆïijŽ

```
try:
    example()
except RuntimeError as e:
    print("It didn't work:", e)

    if e.__cause__:
        print('Cause:', e.__cause__)
```

å¡ŞåIJĺ `except` åİŮäÿ■åŔĹæIJĽåŔęåď ŰçŽĎåijĆåÿÿèćńæŁŻåĞžæŮűäijŽåŕijèĞť äÿĂäÿłéŽŘèŮŔçŽĎå

```
>>> def example2():
...     try:
...             int('N/A')
```

```
...     except ValueError as e:
...             print("Couldn't parse:", err)
...
>>>
>>> example2()
Traceback (most recent call last):
  File "<stdin>", line 3, in example2
ValueError: invalid literal for int() with base 10: 'N/A'
```

åIJÍåď ĎçŘĘäýŁè£řåijĆåÿÿçŽĎæŮűåĂŹïijŇåŔęåď ŰäÿĂäýłåijĆåÿÿåŔŠçŤ§äžĘïijŽ

```
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 5, in example2
NameError: global name 'err' is not defined
>>>
```

è£ŹäýłäżŃå■Řäÿ■ïijŇä¡ääŘŇæŮűèŐůå¿ŮäžĘäÿď äýłåijĆåÿÿçŽĎä£ąæĄřïijŇä¡ĘæŸŕåŕźåijĆåÿÿçŽĎèğč è£ŹæŮűåĂŹïijŇNameError åijĆåÿÿèćńä¡IJäÿžçÍŇåžŔæIJĂçżĹåijĆåÿÿèćńæŁŻåĞžïijŇèĂŇäÿ■æŸŕä¡■äžŐ

åęĆæđIJïijŇä¡ăæČşå£¡çŢěæŐĽåijĆåÿÿéŞ¿ïijŇåŔŕä¡£çŤÍ raise from None:

```
>>> def example3():
...     try:
...             int('N/A')
...     except ValueError:
...             raise RuntimeError('A parsing error occurred') from␣
 ↪None
...
>>>
example3()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 5, in example3
RuntimeError: A parsing error occurred
>>>
```

èŐÍèŐž

åIJÍèő¿èőąäžčçăĄæŮűïijŇåIJÍåŔęåď ŰäÿĂäýł except äžčçăĄåİŮäÿ■ä¡£çŤÍ raise èŕ■åŔěçŽĎæŮűåĂŹä¡ăèęĄçĽźåĹńářŔå£ČäžĘãĂĆ åď ğåď ŽæŢřæČĚåĘţäÿŃïijŇè£Źçğ■ raise èŕ■åŔěéČ¡åžŤèŕěèćńæŤźæĹŘ raise from èŕ■åŔěãĂĆäź§åřśæŸŕèŕťä¡ăåžŤèŕěä¡£çŤÍäÿŃéİćè£Źçğ

```
try:
    ...
except SomeException as e:
    raise DifferentException() from e
```

è£Źæăůå ĄŽçŽĎåŐ§åŻăæŸŕä¡ăåžŤèŕěæŸ¿çď žçŽĎåřĘåŐ§åŻăéŞ¿æŐěęţůæİěãĂĆ

äź§åŕśæŸŕèŕť'ïijŇDifferentException æŸŕçŻť'æŐěäżŐ SomeException èą■çŤ§èĂŇæİěãĂĆ è£Źçğ■åĚşçşżåŔŕäżěäżŐåŻđæžŕçżŞæđIJäÿ■çIJŃåĞžæİěãĂĆ

åęĆæđIJä¡ăåČŔäÿŃéİćè£ŹæăůåĘŹäżčçăĄïijŇä¡ăäż■çĎűäijŽå¿ŮåĹřäÿĂäÿłéŞ¿æŐěåijĆåÿÿïijŇ äÿ■è£Ğè£ŹäÿłåźűæšąæIJL'å¿ĹæÿĚæŹřçŽĎèŕť'æŸŐè£ŹäÿłåijĆåÿÿéŞ¿åĹřåžŢæŸŕåĘĚéČĺåijĆåÿÿè£ŸæŸŕæ§

```
try:
   ...
except SomeException:
   raise DifferentException()
```

å¡Şä¡ăä¡£çŤĺ raise from èŕ■åŔěçŽĎèŕİïijŇåŕśå¿ĹæÿĚæěŽçŽĎèąĺæŸŐæŁŻåĞžçŽĎæŸŕçňňäžŇäÿłå

æIJĂåŘŐäÿĂäÿłä¿Ńå■Řäÿ■éŽŘèŮŔåijĆåÿÿéŞ¿ä£ąæĄŕãĂĆ åř¡çőąéŽŘèŮŔåijĆåÿÿéŞ¿ä£ąæĄŕäÿ■åĹl'äžŐåŻđæžŕïijŇåŘŇæŮűåőČäź§äÿćåď'śäžĘå¿Ĺåď'ŽæIJL'çŤĺçŽĎèřČ äÿ■è£ĞäÿĞäžŃçŽĘåźşç■L'ïijŇæIJL'æŮűåĂŹåŔłä£İçŢŹéĂĆå¡ŞçŽĎä£ąæĄŕäź§æŸŕå¿ĹæIJL'çŤĺçŽĎãĂĆ

16.10 14.10 éĞ■æŰřæŁŻåĞžèćńæ■ŢèŐůçŽĎåijĆåÿÿ

éŮőéćŸ

ä¡ăåIJĺäÿĂäÿł except åİŮäÿ■æ■ŢèŐůäžĘäÿĂäÿłåijĆåÿÿïijŇçŐřåIJĺæČşéĞ■æŰřæŁŻåĞžåőČãĂĆ

èğčåĘşæŰźæąĹ

çőĂå■ŢçŽĎä¡£çŤĺäÿĂäÿłå■ŢçŃňçŽĎ rasie èŕ■åŔěå■şåŔŕïijŇä¿ŃåęĆïijŽ

```
>>> def example():
...     try:
...             int('N/A')
...     except ValueError:
...             print("Didn't work")
...             raise
...

>>> example()
Didn't work
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 3, in example
ValueError: invalid literal for int() with base 10: 'N/A'
>>>
```

èőĺèőž

è£ŹäÿłéŮőéćŸéĂŽåÿÿæŸŕå¡Şä¡ăéIJĂèęĄåIJĺæ■ŢèŐůåijĆåÿÿåŘŐæL'ğèąŇæ§ŘäÿłæŞ■ä¡IJïijĹæŕŤåęĆèć äÿĂäÿłå¿ĹåÿÿèğĄçŽĎçŤĺæşŢæŸŕåIJĺæ■ŢèŐůæL'ĂæIJL'åijĆåÿÿçŽĎåď'ĎçŘĘåŹĺäÿ■ïijŽ

```
try:
    ...
except Exception as e:
    # Process exception information in some way
    ...

    # Propagate the exception
    raise
```

16.11 14.11 è¿ŞåĞžè■ęåŚLä£ąæĄŕ

éŮőécŸ

ä¡ăåÿŇæIJŻèĞłåůśçŽĎçÍŃåžŔèČ¡çŤ§æĹŘè■ęåŚLä£ąæĄŕïijĹærŤåęĆåž§åijČçL'źæĂğæĹŰä¡£çŤĺéŮőéć

èğčåĘşæŰźæąĹ

èęĄè¿ŞåĞžäÿĂäÿłè■ęåŚLæűĹæĄŕïijŇåŔŕä¡£çŤĺ warning.warn() åĞ¡æŢřãĂĆä¿ŃåęĆïijŽ

```
import warnings

def func(x, y, logfile=None, debug=False):
    if logfile is not None:
        warnings.warn('logfile argument deprecated',␣
↪DeprecationWarning)
    ...
```

warn() çŽĎåŔĆæŢřæŸŕäÿĂäÿłè■ęåŚLæűĹæĄŕåŠŇäÿĂäÿłè■ęåŚLçśżïijŇè■ęåŚLçśżæIJL'åęĆäÿŃåĞă DeprecationWarning, SyntaxWarning, RuntimeWarning, ResourceWarning, æĹŰ FutureWarning.

åŕźè■ęåŚLçŽĎåd'ĎçŘEåŔŰåEşäžŐä¡ăåęĆä¡Ţè£ŘèąŇèğčéĞŁåŹĺäżěåŔŁäÿĂäžŽåĚűäżŰéĚ■ç¡õãĂĆ ä¿ŃåęĆïijŇåęĆæđIJä¡ăä¡£çŤĺ -W all éĂL'éąźåŐżè£ŘèąŇPythonïijŇä¡ăäijŽå¿ŮåĹřåęĆäÿŃçŽĎè¿ŞåĞžïijŽ

```
bash % python3 -W all example.py
example.py:5: DeprecationWarning: logfile argument is deprecated
  warnings.warn('logfile argument is deprecated',␣
↪DeprecationWarning)
```

éĂŽåÿÿæİěèőšïijŇè■ęåŚLäijŽè¿ŞåĞžåĹřæăĞåĞEéŤŹèŕŕäÿŁãĂĆåęĆæđIJä¡ăæČşèőšè■ęåŚLè¡ñæ■ćäÿžå -W error éĂL'éąźïijŽ

```
bash % python3 -W error example.py
Traceback (most recent call last):
  File "example.py", line 10, in <module>
    func(2, 3, logfile='log.txt')
```

```
  File "example.py", line 5, in func
    warnings.warn('logfile argument is deprecated',␣
↪DeprecationWarning)
DeprecationWarning: logfile argument is deprecated
bash %
```

èőÍèőž

åIJÍä¡ăçžťæŁď'è¡ŕäżűĭijŇæŔŘçď'žçŤÍæÍůæ§ŘäžŻä£ąæĄŕĭijŇä¡ĘæŸŕåŔÍäÿ■éIJĂèęĄåřĘåĚűäÿŁå■ĞäÿÍ
ä¿ŇåęĆĭijŇåĄĞèő¿ä¡ăåĞĘåď'Ğä£őæŤźæ§ŘäÿłåĞ¡æŢřåžŞæÍŰæąĘæđűçŽĎåŁ§èČ¡ĭijŇä¡ăåŔŕäżěåĚÍäÿžä¡ăè
ä¡ăè£ŸåŔŕäżěè■ęåŚŁçŤÍæÍůäÿĂäžŻåŕźäżčçăĄæIJL'éŮőécŸçŽĎä¡£çŤÍæŰźåijŔãĂĆ

ä¡IJäÿžåŔęåď'ŰäÿĂäÿłåĘĚç¡őåĞ¡æŢřåžŞçŽĎè■ęåŚŁä¡£çŤÍä¿Ńå■ŘĭijŇäÿŃéÍćæijŤçď'žäžĘäÿĂäÿłæšąæ

```
>>> import warnings
>>> warnings.simplefilter('always')
>>> f = open('/etc/passwd')
>>> del f
__main__:1: ResourceWarning: unclosed file <_io.TextIOWrapper name=
↪'/etc/passwd'
 mode='r' encoding='UTF-8'>
>>>
```

éżŸèőď'æČĚåĘţäÿŃĭijŇåźűäÿ■æŸŕæL'ĂæIJL'è■ęåŚŁæűĹæĄŕéČ¡äijŽåĞžçŐřãĂĆ`-W` éĂL'éąźèČ¡æŐğåĹűè■ęåŚŁæűĹæĄŕçŽĎè¿ŞåĞžãĂĆ `-W all` äijŽè¿ŞåĞžæL'ĂæIJL'è■ęåŚŁæűĹæĄŕĭijŇ`-W ignore` å£¡çŢěæŐL'æL'ĂæIJL'è■ęåŚŁĭijŇ`-W error` åřĘè■ęåŚŁè¡ñæ■ćæĹŘåijĆåÿÿãĂĆ åŔęåď'ŰäÿĂçğ■éĂL'æŃl'ïijŇä¡ăè£Ÿåŕŕäżěä¡£çŤÍ `warnings.simplefilter()` åĞ¡æŢřæŐğåĹűè¿ŞåĞžãĂĆ `always` åŔĆæŢřäijŽèől'æL'ĂæIJL'è■ęåŚŁæűĹæĄŕåĞžçŐřĭijŇ`` `ignore`` å£¡çŢěèřČæL'ĂæIJL'çŽĎè■ęåŚŁĭijŇ`error` åřĘè■ęåŚŁè¡ñæ■ćæĹŘåijĆåÿÿãĂĆ

åŕźäžŐçőĂå■ŢçŽĎçŤ§æĹŘè■ęåŚŁæűĹæĄŕçŽĎæČĚåĘţè£ŹäžŻåůšçżŔèűşåď'§äžĘãĂĆ `warnings` æĺąåİŮåŕźè£Ğæżď'åŠŇè■ęåŚŁæűĹæĄŕåď'ĎçŘĘæŔŘä¿ŻäžĘåď'ğéĞŔçŽĎæŻť'énŸçžğçŽĎéĚ■ç¡æŻť'åď'Žä£ąæĄŕèřůåŔĆèĂČ PythonæŰĞæąč

16.12 14.12 èřČèŕŢå§žælJňçŽĎçÍŃåžŔåť'l'æžČéŤŹèŕŕ

éŮőécŸ

ä¡ăçŽĎçÍŃåžŔåť'l'æžČåŘŐèřěæĂŐæăůåŐżèřČèŕŢåőČïij§

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăçŽĎçÍŃåžŔåŻääÿžæ§ŘäÿłåijĆåÿÿèĂŇåť'l'æžČïijŇè£ŘèąŇ `python3 -i someprogram.py` åŔŕæL'ğèąŇçőĂå■ŢçŽĎèřČèŕŢãĂĆ

-i éĂL'éąźåŘŕèőľçÍŃåžŘçżŞæİ§åŘŐæL'Şåij Ăäÿ Ăäÿłäžd'äžŠåijŔshellãĂĆçĎűåŘŐä¡ăářśèČ¡æ§ěçIJŃçŐŕáćČïijŇä¿ŃåęĆïijŇåĄĞèő¿ä¡ăæIJL'äÿŃéİćçŽĎäžčçăĄïijŽ

```
# sample.py

def func(n):
    return n + 10

func('Hello')
```

è£ŘèąŇ `python3 -i sample.py` äijŽæIJL'çśżäijijåęĆäÿŃçŽĎè¿ŞåĞžïijŽ

```
bash % python3 -i sample.py
Traceback (most recent call last):
  File "sample.py", line 6, in <module>
    func('Hello')
  File "sample.py", line 4, in func
    return n + 10
TypeError: Can't convert 'int' object to str implicitly
>>> func(10)
20
>>>
```

åęĆædIJä¡ăçIJŃäÿ■åĹřäÿŁéİćè£ŹæăůçŽĎïijŇåŔŕäžěåIJÍçÍŃåžŔåťľ'æžČåŘŐæL'ŞåijĂPythonçŽĎèřČèŕĘ

```
>>> import pdb
>>> pdb.pm()
> sample.py(4)func()
-> return n + 10
(Pdb) w
  sample.py(6)<module>()
-> func('Hello')
> sample.py(4)func()
-> return n + 10
(Pdb) print n
'Hello'
(Pdb) q
>>>
```

åęĆædIJä¡ăçŽĎäžčçăĄæL'ĂåIJÍçŽĎçŐŕåćČå¿ĹéŽ¿èŐůåŘŰäžd'äžŠshellïijĹæŕŤåęĆåIJÍæ§Řäÿłæ IJ■åŁą éĂŽåÿÿåŘŕäžěæ■ŢèŐůåijĆåÿÿåŘŐèĞłåůśæL'Şå■řèů§èÿłä£ąæĄŕãĂĆä¿ŃåęĆïijŽ

```
import traceback
import sys

try:
    func(arg)
except:
    print('**** AN ERROR OCCURRED ****')
    traceback.print_exc(file=sys.stderr)
```

èęĄæŸŕä¡ăçŽĎçÍŃåžŔæšąæIJL'åť'ľ'æžČïijŇèĂŇåŘłæŸŕäžğçŢ§äžĘäÿĂäžŻä¡ăçIJŃäÿ■æĞĆçŽĎçżŞædI

ä¡ååIJÍæĎ§åĚt'èűčçŽĎåIJřæŰźæŔŠåĚěäÿĂäÿŃ `print()` èí■åŘěäź§æŸŕäÿłäÿ■éŤŹçŽĎéĂL'æŃl'ãĂĆ
äÿ■è£ĞïijŇèęĄæŸŕä¡ăæL'ŞçőŮè£Źæăůå₳ŽïijŇæIJL'äÿĂäžŽářŔæŁĂåůğåŘŕäžěåÿőåŁl'ä¡ããĂĆ
éęŰåĚĹïijŇ`traceback.print_stack()` åĞ¡æŢřäijŽä¡ăçÍŃåžŔè£ŘèąŇåĹřéĆčäÿłçĆźçŽĎæŮűåĂŹåĹŻ

```
>>> def sample(n):
...     if n > 0:
...             sample(n-1)
...     else:
...             traceback.print_stack(file=sys.stderr)
...
>>> sample(5)
  File "<stdin>", line 1, in <module>
  File "<stdin>", line 3, in sample
  File "<stdin>", line 3, in sample
  File "<stdin>", line 3, in sample
  File "<stdin>", line 3, in sample
  File "<stdin>", line 3, in sample
  File "<stdin>", line 5, in sample
>>>
```

åŘęåď'ŰïijŇä¡ãè£ŸåŘŕäžěåČŔäÿŃéÍćè£ŹæăůäĄ£çŢÍ `pdb.set_trace()` åIJÍäżžä¡ŢåIJřæŰźæL'ŃåŁÍçŽĎåŘŕåŁÍèřČèŕŢåŹÍïijŽ

```
import pdb

def func(arg):
    ...
    pdb.set_trace()
    ...
```

å¡ŞçÍŃåžŔæŕŤè¿Čåď'ğèĂŇä¡ăæČşèřČèŕŢæŐğåĹűæţĄçÍŃäžěåŔŁåĞ¡æŢřåŔĆæŢřçŽĎæŮűåĂŹè£Źäÿłåìä¿ŃåęĆïijŇäÿĂæŮęèřČèŕŢåŹÍåijĂåğŃè£ŘèąŇïijŇä¡ăåŕśèČ¡åď'§ä¡£çŢÍ `print` æÍèèğĆæţŃåŘŸéĞŘåĂijæĹŰæŢšåĞžæ§Řäÿłåŝ¡äžď'æŕŤåęĆ w æÍèèŐůåŘŰè£¡èÿłä£ąæĄŕãĂĆ

èőÍèőž

äÿ■èęĄåřĘèřČèŕŢåijĎçŽĎè£ĞäžŐåď'■æÍĆåŇŰãĂĆäÿĂäžŽçőĂå■ŢçŽĎéŤŹèŕŕåŘłéIJĂèęĄèğĆår§çÍŃåžåődéŹĚçŽĎéŤŹèŕŕäÿĂèĹňæŸŕåăĘæăĹçŽĎæIJĂåŘŐäÿĂèąŇãĂĆ
ä¡ååIJÍåijĂåŘŚçŽĎæŮűåĂŹïijŇäź§åŘŕäžěåIJÍä¡ăéIJĂèęĄèřČèŕŢçŽĎåIJřæŰźæŘŠåĚěäÿĂäÿŃ `print()` åĞ¡æŢræÍèèŕŁæŰ■ä£ąæĄŕïijĹåŘłéIJĂèęĄæIJĂåŘŐåŘŚåÿČçŽĎæŮűåĂŹåĹăéŹď'è£ŹäžŽæL'Şå■ř

èřČèŕŢåŹÍçŽĎäÿĂäÿłåÿÿèğĄçŤÍæşŢæŸŕèğĆæţŃæ§Řäÿłåůšçżŕåť'l'æžČçŽĎåĞ¡æŢřäÿ■çŽĎåŘŸéĞŘãĂçš€éĄŞæĂŐæăůåIJÍåĞ¡æŢřåť'l'æžČåŘŐè£ŽåĚèèřČèŕŢåŹÍæŸŕäÿĂäÿłå¿ĹæIJL'çŤÍçŽĎæŁĂèČ¡ãĂĆ

å¡Şä¡ăæČşèğčåL'ŰäÿĂäÿłéIđåÿÿåď'■æÍĆçŽĎçÍŃåžŔïijŇåžŢåśĆçŽĎæŐğåĹűéĂżè¿Śä¡ääÿ■æŸŕå¿ĹæÿĚæŘŠåĚě `pdb.set_trace()` è£ŹæăůçŽĎèŕ■åŘěářšå¿ĹæIJL'çŤÍäžĘãĂĆ

åőđéŹĚäÿŁïijŇçÍŃåžŔäijŽäÿĂçŻť'è£ŘèąŇåĹřçćřåĹř `set_trace()` èŕ■åŘěä¡■ç¡őïijŇçĎűåŘŐçńŃél'ňè£ŻåĚèèřČèŕŢåŹÍãĂĆ çĎűåŘŐä¡ăářšåŔŕäžěåĄŽæŻť'åď'ŽçŽĎäžŃäžĘãĂĆ

åęĆædIJä¡ăä¡£çŤÍIDEæİěåĄŽPythonåijĂåŔŠïijŇéĂŽåÿÿIDEéČ¡äijŽæŔŘä¿ŻèĞłåůśçŽĎèřČèřŢåŹÍæİěæŻťåďŽè£ŹæŰźéİćçŽĎä£ąæĄŕåŔŕäžěåŔĆèĂČä¡ăä¡£çŤÍçŽĎIDEæĽŃåĘÑãĂĆ

16.13 14.13 çżŹä¡äçŽĎçÍŃåžŔåĄŽæĂğèČ¡æţŃèŕŢ

éŮőécŸ

ä¡ăæČşæţŃèŕŢä¡ăçŽĎçÍŃåžŔè£ŘèąÑæĽĂèŁśèťźçŽĎæŮűéŮťåźűåĄŽæĂğèČ¡æţŃèŕŢãĂĆ

èğčåĘşæŰźæąĹ

åęĆædIJä¡ääŔłæŸŕçőĂå■ŢçŽĎæČşæţŃèŕŢäÿŃä¡äçŽĎçÍŃåžŔæŢťä¡ŞèŁśèťźçŽĎæŮűéŮťïijÑéĂŽåÿÿä¡£çŤÍUnixæŮűéŮťåĞ¡æŢřářśèąÑäžĘïijÑærŤåęĆïijŽ

```
bash % time python3 someprogram.py
real 0m13.937s
user 0m12.162s
sys  0m0.098s
bash %
```

åęĆædIJä¡äè£ŸéIJĂèęĄäÿĂäÿłçÍŃåžŔåŘĎäÿłçżĘèŁĆçŽĎèŕęçżĘæŁěåŚŁïijŇåŔŕäžěä¡£çŤÍ `cProfile` æĺąåİŮïijŽ

```
bash % python3 -m cProfile someprogram.py
         859647 function calls in 16.016 CPU seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall␣
↪filename:lineno(function)
   263169    0.080    0.000    0.080    0.000 someprogram.
↪py:16(frange)
      513    0.001    0.000    0.002    0.000 someprogram.
↪py:30(generate_mandel)
   262656    0.194    0.000   15.295    0.000 someprogram.py:32(
↪<genexpr>)
        1    0.036    0.036   16.077   16.077 someprogram.py:4(
↪<module>)
   262144   15.021    0.000   15.021    0.000 someprogram.py:4(in_
↪mandelbrot)
        1    0.000    0.000    0.000    0.000 os.py:746(urandom)
        1    0.000    0.000    0.000    0.000 png.py:1056(_readable)
        1    0.000    0.000    0.000    0.000 png.py:1073(Reader)
        1    0.227    0.227    0.438    0.438 png.py:163(<module>)
      512    0.010    0.000    0.010    0.000 png.py:200(group)
    ...
bash %
```

äÿ■è£ĞéĂŽåÿÿæČĚåĘţæŸŕäżŃäžŐè£ŹäÿďäÿłæđĄçńŕäźŃéŮťãĂĆæŕŤåęĆä¡ääůšçżŔç§ěéĄŞäżčçăĄè£Ł
åŕźäžŐè£ŹäžŻåĞ¡æŢřçŽĎæĂğèČ¡æţŃèŕŢïijŇåŔŕäżěä¡£çŤĺäÿĂäÿłçőĂå■ŢçŽĎèčĚéěřåŹĺïijŽ

```
# timethis.py

import time
from functools import wraps

def timethis(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        start = time.perf_counter()
        r = func(*args, **kwargs)
        end = time.perf_counter()
        print('{}.{} : {}'.format(func.__module__, func.__name__,␣
↪end - start))
        return r
    return wrapper
```

èęĄä¡£çŤĺè£ŹäÿłèčĚéěřåŹĺïijŇåŔłéIJĂèęĄåřĘåĚűæŤ¿ç¡őåIJĺä¡ăèęĄè£ŻèąŇæĂğèČ¡æţŃèŕŢçŽĎåĞ¡æŢ

```
>>> @timethis
... def countdown(n):
...     while n > 0:
...             n -= 1
...
>>> countdown(10000000)
__main__.countdown : 0.803001880645752
>>>
```

èęĄæţŃèŕŢæ§Řäÿłäżčçă Ąå İŮè£ŘèąŇæŮűéŮťïijŇä¡ääŔŕäżěåőŽäźL'äÿĂäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺïijŇ

```
from contextlib import contextmanager

@contextmanager
def timeblock(label):
    start = time.perf_counter()
    try:
        yield
    finally:
        end = time.perf_counter()
        print('{} : {}'.format(label, end - start))
```

äÿŃéİćæŸŕä¡£çŤĺè£ŹäÿłäÿŁäÿŃæŰĞçőąçŘĘåŹĺçŽĎä¿Ńå■ŘïijŽ

```
>>> with timeblock('counting'):
...     n = 10000000
...     while n > 0:
...             n -= 1
...
counting : 1.5551159381866455
```

```
>>>
```

åŕźäžŐæţŃèŕŢå¿ĹåřŔçŽĎäżčçăĄçL'Ğæőţè£ŘèąŇæĂğèČ¡ïijŇä¡£çŤÍ timeit æÍąåİŮäijŽå¿ĹæŰźä¿£ïijŇä¿ŃåęĆïijŽ

```
>>> from timeit import timeit
>>> timeit('math.sqrt(2)', 'import math')
0.1432319980012835
>>> timeit('sqrt(2)', 'from math import sqrt')
0.10836604500218527
>>>
```

timeit äijŽæL'ğèąŇçňňäÿĂäÿłåŔĆæŢřäÿ■èŕ■åŔě100äÿĞæňąåźűèőąçőŮè£ŘèąŇæŮűéŮť'ãĂĆ çňňäžŇäÿłåŔĆæŢřæŸŕè£ŘèąŇæţŃèŕŢäźŃåL'■éĚ■ç¡őçŐŕåćČãĂĆåęĆæđIJä¡äæČşæŤźåŔŸå¿łçŐŕæL'ğèąŇæň åŔŕäžěåČŔäÿŃéİćè£Źæăůèő¿ç¡ő number åŔĆæŢřçŽĎåĂijïijŽ

```
>>> timeit('math.sqrt(2)', 'import math', number=10000000)
1.434852126003534
>>> timeit('sqrt(2)', 'from math import sqrt', number=10000000)
1.0270336690009572
>>>
```

èőÍèőž

å¡ŞæL'ğèąŇæĂğèČ¡æţŃèŕŢçŽĎæŮűåĂŹïijŇéIJĂèęĄæşÍæĎŔçŽĎæŸŕä¡äèŐůåŔŰçŽĎçżŞæđIJéČ¡æŸŕè time.perf_counter() åĞ¡æŢřäijŽåIJÍçżŹåőŽåźşåŔŕäÿŁèŐůåŔŰæIJĂéńŸçš¿åžęçŽĎèőąæŮűåĂijãĂĆ äÿ■è£ĞïijŇåőČäż■çĎűè£ŸæŸŕå§žäžŐæŮűéŠ§æŮűéŮť'ïijŇå¿Ĺåď'ŽåŻäçť'ăäijŽå¡śåŞ■åĹřåőČçŽĎçš¿çąőåžęï åęĆæđIJä¡ååŕźäžŐæL'ğèąŇæŮűéŮť'æŻť'æĎ§åĚť'èűčïijŇä¡£çŤÍ time.process_time() æİěäžčæŻ£åőČãĂĆä¿ŃåęĆïijŽ

```
from functools import wraps
def timethis(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        start = time.process_time()
        r = func(*args, **kwargs)
        end = time.process_time()
        print('{}.{} : {}'.format(func.__module__, func.__name__,␣
↪end - start))
        return r
    return wrapper
```

æIJĂåŘŐïijŇåęĆæđIJä¡äæČşè£ŻèąŇæŻť'æůśåĚěçŽĎæĂğèČ¡åĹĘæđŘïijŇéĆčäźĹä¡äéIJĂèęĄèŕęçżĘéŸİ time ãĂĄtimeit åŠŇåĚűäžŰçŻÿåĚşæĺąåİŮçŽĎæŰĞæąčãĂĆ è£Źæăůä¡äåŔŕäžěçŘĘèğčåŠŇåźşåŔŕçŻÿåĚşçŽĎåůőåijĆäžěåŔŁäÿĂäžŻåĚűäžŰéŹůéŸśãĂĆ è£ŸåŔŕäžěåŔĆèĂČ13.13åŕŘèŁĆäÿ■çŻÿåĚşçŽĎäÿĂäÿłåĹŻåžžèőąæŮűåŹĺçśżçŽĎä¿Ńå■ŘãĂĆ

16.14 14.14 åŁăéĂ§çÍŃåžŔè£ŘèąŇ

éŮőécŸ

ä¡ăçŽĎçÍŃåžŔè£ŘèąŇåd'łæĚćïijŇä¡ăæČşåIJÍäÿ■ä¡£çŤÍåd'■æİĆæŁĂæIJŕæŕŤåęĆCæL'ľ'åśŢæĹŰJITçijŰ

èğčåĘşæŰźæąĹ

åĚşäžŐçÍŃåžŔäijŸåŇŰçŽĎçňňäÿĂäÿłåĞĘåĹŹæŸŕâĂIJäÿ■èęĄäijŸåŇŰâĂİïijŇçňňäžŇäÿłåĞĘåĹŹæŸŕ
åęĆæđIJä¡ăçŽĎçÍŃåžŔè£ŘèąŇçijŞæĚćïijŇéęŰåĚĹä¡ăå¿Ůä¡£çŤÍ14.13åřŔèŁĆçŽĎæŁĂæIJŕåĚĹåŕźåőČè£Żèă

éĂŽåÿÿæİěèőšä¡ăäijŽåŔŠçŐřä¡ăå¿ŮçÍŃåžŔåIJÍåřŠæŢřåĞääÿłçČ■çĆźåIJřæŰźèŁśèť źäžĘåd'ğéĞŔæŮűé
ærŤåęĆåĘĚå■ŸçŽĎæŢřæ■őåd'ĎçŘĘå¿łçŐŕãĂĆäÿĂæŮęä¡ăăőŽä¡■åĹrè£ŹäžŻçĆźïijŇä¡ăářśåŔŕäžěä¡£çŤĺäÿŇ

ä¡£çŤĺåĞ¡æŢř

å¿Ĺåd'ŽçÍŃåžŔåŚŸåĹŽåijĂåğŃäijŽä¡£çŤÍPythonèŕ■èÍĂåĘŹäÿĂäžŻçőĂå■ŢèĎŽæIJñãĂĆ
å¡ŞçijŰåĘŹèĎŽæIJňçŽĎæŮűåĂŹïijŇéĂŽåÿÿäźăæČŕäžĘåĘŹæŕńæŮăçżŞæđĎçŽĎäžčçăĄïijŇærŤåęĆïijŽ

```
# somescript.py

import sys
import csv

with open(sys.argv[1]) as f:
     for row in csv.reader(f):

          # Some kind of processing
          pass
```

å¿ĹåřŚæIJL'äžžç§ěéĄŞïijŇåČŔè£ŹæăůåőŽäźL'åIJĺåĚĺåśĂèŇČåŻť'çŽĎäžčçăĄè£ŘèąŇètůæİěèęĄæŕŤåőŽ
è£Źçğ■éĂ§åžęåůőåijĆæŸŕçŤśäžŐåśĂéČĺåŔŸéĞŔåŠŇåĚĺåśĂåŔŸéĞŔçŽĎåőđçŐřæŰźåijŔïijĹä¡£çŤĺåśĂéČĺ
åŻăæ■d'ïijŇåęĆæđIJä¡ăæČşèől'çÍŃåžŔè£ŘèąŇæŻť'å£ńäžŻïijŇåŔłéIJĂèęĄåřĘèĎŽæIJñèŕ■åŔěæŤ¿åĚěåĞ¡æŢ

```
# somescript.py
import sys
import csv

def main(filename):
    with open(filename) as f:
         for row in csv.reader(f):
             # Some kind of processing
             pass

main(sys.argv[1])
```

éĂ§åžęçŽĎåůőåijĆåŔŰåĘşäžŐåőđéŹĚè£ŘèąŇçŽĎçÍŃåžŔïijŇäÿ■è£Ğæăźæ■őçżŔéłŇïijŇä¡£çŤĺåĞ¡æŢï
30%çŽĎæĂğèČ¡æŔŘå■ĞæŸŕå¿Ĺåÿÿèğ Ąç ŽĎãĂĆ

ář¡åŔŕèČ¡åŐżæŐL'åśđæĂğèó£éŰő

æfŔäÿĂæňąä¡£çŤĺçĆź(.)æŞ■ä¡IJçňęæİèëő£éŮőåśđæĂğçŽĎæŮűåĂŹäijŽåÿęæİèéćİåď´ŰçŽĎåijĂéŤĂãĂ
åőČäijŽèğęåŔŚçL’źåőŽçŽĎæŰźæşŢïijŇærŤåęĆ `__getattribute__()` åŠŇ
`__getattr__()` ïijŇè£ŹäžŽæŰźæşŢäijŽè£ŻèąŇå■ŮåĚÿæŞ■ä¡IJæŞ■ä¡IJãĂĆ

éĂŽåÿÿä¡ääŔŕäžěä¡£çŤĺ `from module import name`
è£ŹæăůçŽĎåŕijåĚěå¡ćåijŔïijŇäžěåŔŁä¡£çŤĺçżŚåőŽçŽĎæŰźæşŢãĂĆ
åĄĞèő¿ä¡ăæIJL’åęĆäÿŃçŽĎäžčçăĄçL’Ğæőţïijž

```
import math

def compute_roots(nums):
    result = []
    for n in nums:
        result.append(math.sqrt(n))
    return result

# Test
nums = range(1000000)
for n in range(100):
    r = compute_roots(nums)
```

åIJĺæĹŚäžňæIJžåŹĺäÿŁéİćæţŃèŕŢçŽĎæŮűåĂŹïijŇè£ŹäÿłçĺŃåžŔèŁśèť´źäžĘåď´ğæęĆ40çğŠãĂĆçŐřåIJĺa
`compute_roots()` åĞ¡æŢřåęĆäÿŃïijž

```
from math import sqrt

def compute_roots(nums):

    result = []
    result_append = result.append
    for n in nums:
        result_append(sqrt(n))
    return result
```

ä£őæŤźåŘŐçŽĎçL’ĹæIJňè£ŘèąŇæŮűéŮť´åď´ğæęĆæŸŕ29çğŠãĂĆåŤŕäÿĂäÿ■åŘŇäźŃåď´ĎåřśæŸŕæűĹéź
çŤĺ `sqrt()` äžčæŻ£äžĘ `math.sqrt()` ãĂĆ `The result.append()` æŰźæşŢèćńètŃçżŹäÿĂäÿłåśĂéČĺåŔŸéĞŔ `result_append`
ïijŇçĎűåŘŐåIJĺåĘĚéČĺå¿łçŐŕäÿ■ä¡£çŤĺåőČãĂĆ

äÿ■è£ĞïijŇè£ŹäžŽæŤźåŔŸåŔłæIJL’åIJĺåď´ğéĞŔéĞ■åď´■äžčçăĄäÿ■æL’■æIJL’æĎŔäźL’ïijŇærŤåęĆå¿łçĆ
åŻăæ■ď´ïijŇè£ŹäžŽäijŸåŇŰäź§åŔłæŸŕåIJĺæ§ŘäžŽçL’źåőŽåIJřæŰźæL’■åžŤèŕĕèćńä¡£çŤĺãĂĆ

çŘĘèğčåśĂéČĺåŔŸéĞŔ

äźŃåL’■æŔŘè£ĞïijŇåśĂéČĺåŔŸéĞŔäijŽærŤåĚĺåśĂåŔŸéĞŔè£ŘèąŇéĂ§åžęå£ńãĂĆ
åŕźäžŐéćŚçźĄèő£éŮőçŽĎåŘ■çğřïijŇéĂŽè£ĞåřĘè£ŹäžŽåŘ■çğřåŔŸæĹŘåśĂéČĺåŔŸéĞŔåŕźäžěåŁäéĂ§çĺŃä
ä¿ŃåęĆïijŇçIJŃäÿŃäźŃåL’■åŕźäžŐ `compute_roots()` åĞ¡æŢřè£ŻèąŇä£őæŤźåŘŐçŽĎçL’ĹæIJňïijž

```
import math

def compute_roots(nums):
    sqrt = math.sqrt
```

```
    result = []
    result_append = result.append
    for n in nums:
        result_append(sqrt(n))
    return result
```

åIJÍè£Źäÿłçl'ĺæIJñäÿ■ïijÑsqrt äżŐ match æĺąåİÙèćńæŃ£åĞžåźűæŤ¿åĚëäžĘäÿĂäÿłåśĂéČĺåŔŸéĞŔ åęĆæđIJä¡ãè£ŘèąÑè£ŹäÿłäżčçăĄïijÑåď ğæęĆèŁśèť ź25çğŠïijĹåŕźäžŐäźŃåL'■29çğŠåŔĹæŸŕäÿĂäÿłæŤźè£Żè£Źäÿłécİåď ŰçŽĎåŁăéĂ§åŐ§åŻăæŸŕåŻăäÿžåŕźäžŐåśĂéČĺåŔŸéĞŔ sqrt çŽĎæ§ěæL'¿èęĄå£ńäžŐåĚĺåśĂåŔŸéĞŔ sqrt

åŕźäžŐçśżäÿ■çŽĎåśđæĂğèő£éŮőäź§åŘŇæăůéĂĆçŤĺäžŐè£ŹäÿłåŐ§çŘĘãĂĆ éĂŽåÿÿæİěèőšïijÑæ§ěæL'¿æ§Räÿłå Äijæŕ Ťåę Ć self.name äijŽæŕŤèő£éŮőäÿĂäÿłåśĂéČĺåŔŸéĞŔèęĄæĚćäÿĂäžŻãĂĆ åIJĺåĘĚéČĺå¿łçŐŕäÿ■ïijÑåŔŕäžěåŕĘæ§Räÿłéİ JĂè

```
# Slower
class SomeClass:
    ...
    def method(self):
         for x in s:
             op(self.value)

# Faster
class SomeClass:

    ...
    def method(self):
         value = self.value
         for x in s:
             op(value)
```

éĄ£åĚ■äÿ■å£ĚèęĄçŽĎæŁ¡èśą

äżżä¡ŢæŮűåĂŹå¡Şä¡ää¡£çŤĺécİåď ŰçŽĎåď ĎçŘĘåśĆïijĹæŕŤåęĆèčĚéěřåŹĺãĂĄåśđæĂğèő£éŮőãĂĄæŔ æŕŤåęĆçIJŃäÿŃåęĆäÿŃçŽĎè£ŹäÿłçśżïijŽ

```
class A:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    @property
    def y(self):
        return self._y
    @y.setter
    def y(self, value):
        self._y = value
```

çŐřåIJÍè£Żèą Ñäÿ Ääÿłçő Åå■ŢæţŃèŕŢïijŽ

```
>>> from timeit import timeit
>>> a = A(1,2)
```

```
>>> timeit('a.x', 'from __main__ import a')
0.07817923510447145
>>> timeit('a.y', 'from __main__ import a')
0.35766440676525235
>>>
```

åŔŕäžěçIJŃåĹřïijŇèő£éŮőåśđæĂğyçŻÿæŕŤåśđæĂğxèĂŇèĺĂæĚćçŽĎäÿ■æ■ćäÿĂçĆźçĆźïijŇåď˙ğæęĆæł
åęĆæđIJä¡ăåIJĺæĎŔæĂğèČ¡çŽĎèŕİïijŇéĆčäźĹåřśéIJĂèęĄéĞ■æŰřåőąèğĘäÿŃářźäžŐyçŽĎåśđæĂğèő£éŮőåŽ
åęĆæđIJæšąæIJL’å£ĚèęĄïijŇåřśä¡£çŤĺçőĂå■ŢåśđæĂğåŘğãĂĆ
åęĆæđIJäżĚäżĚæŸŕåŻääÿžåĚűäžŰçijŰçĺŃèŕ■èĺĂéIJĂèęĄä¡£çŤĺgetter/setteråĞ¡æŢřåřśåŐżä£őæŤźäžčçăĄéč

ä¡£çŤĺåĘĚç¡őçŽĎåőźåŹÍ

åĘĚç¡őçŽĎæŢřæ■őçśżåđŃærŤåęĆå■Ůçňęäÿšã ĂĄåĚČçżĎãĂĄåĹŮèąĺãĂĄéŻĘåŘĹåŠŇå■ŮåĚÿéČ¡æŸŕ
åęĆæđIJä¡ăæČşèĞłåůśåőđçŐřæŰřçŽĎæŢřæ■őçżŞæđĎïijĹærŤåęĆéŞ¿æŐěåĹŮèąĺãĂĄåźşèąąæăŞç■L’ïijL’ïijŇ
éĆčäźĹèęĄæČşåIJĺæĂğèČ¡äÿŁè¿¿åĹřåĘĚç¡őçŽĎéĂ§åžęåĞääźŐäÿ■åŘŕèČ¡ïijŇåŻăæ■d’ïijŇè£ŸæŸŕäźŰäźŰ

éĄ£åĚ■åĹŻåžžäÿ■å£ĚèęĄçŽĎæŢřæ■őçżŞæđĎæĹŰåd’■åĹű

æIJL’æŮűåĂŹçĺŃåžŘåŚŸæČşæŸ¿æŚĘäÿŃïijŇæđĎéĂääÿĂäžŻåźűæšąæIJL’å£ĚèęĄçŽĎæŢřæ■őçżŞæđ

```
values = [x for x in sequence]
squares = [x*x for x in values]
```

äź§èőÿè£ŹéĞŇçŽĎæČşæşŢæŸŕéęŰåĚĹåřĘäÿĂäžŻåĂijæŤűéŻĘåĹřäÿĂäÿłåĹŮèąĺäÿ■ïijŇçĎűåŘŐä¡£çŤ
äÿ■è£ĞïijŇçňňäÿĂäÿłåĹŮèąĺåőŇåĚĺæšąæIJL’å£ĚèęĄïijŇåŔŕäžěçőĂå■ŢçŽĎåČŘäÿŃéİćè£ŹæăůåĘŹïijŽ

```
squares = [x*x for x in sequence]
```

äÿŐæ■d’çŻÿåĚşïijŇè£ŸèęĄæşĺæĎŔäÿŃéĆčäžŻåŕźPythonçŽĎåĚśäžńæŢřæ■őæIJžåĹűè£ĞäžŐåĄŔæL’ğ
æIJL’äžŻäžžåźűæšąæIJL’å¿Ĺåě¡çŽĎçŘĘèğčæĹŰä£ąäżżPythonçŽĎåĘĚå■ŸæĺąåđŃïijŇæžěçŤĺ
`copy.deepcopy()` äźŃçśżçŽĎåĞ¡æŢřãĂĆ éĂŽåÿÿåIJĺè£ŹäžŻäžčçăĄäÿ■æŸŕåŔŕäžěåŐżæŐL’åd’■åĹűæŞ

èőĺèőž

åIJĺäijŸåŇŰäźŃåL’■ïijŇæIJL’å£ĚèęĄåĚĹçăŤçl’űäÿŃä¡£çŤĺçŽĎçőŮæşŢãĂĆ
éĂL’æŃl’äÿĂäÿłåd’■æİĆåžęäÿž O(n log n) çŽĎçőŮæşŢèęĄæŕŤä¡ăåŐżèŕČæŢt’äÿĂäÿłåd’■æİĆåžęäÿž
O(n**2) çŽĎçőŮæşŢæL’ĂåÿęæİěçŽĎæĂğèČ¡æŔŘå■ĞèęĄåd’ğå¿Ůåd’ŽãĂĆ

åęĆæđIJä¡ăèğL’å¿Ůä¡ăè£ŸæŸŕå¿Ůè£ŻèąŇäijŸåŇŰïijŇéĆčäźĹèŕůäžŐæŢt’ä¡ŞèĂČèŹŚãĂĆ
ä¡IJäÿžäÿĂèĹňåĞĘåĹŹïijŇäÿ■èęĄåŕźçĺŃåžŘçŽĎæŕŔäÿĂäÿłéČĺåĹĘéČ¡åŐżäijŸåŇŰ,åŻăäÿžè£ŹäžŻä£őæŤź
ä¡ăåžŤèŕěäÿŞæşĺäžŐäijŸåŇŰäžğçŤ§æĂğèČ¡çŞűécĹçŽĎåIJŕæŰźïijŇæŕŤåęĆåĘĚéČĺå¿łçŐŕãĂĆ

ä¡ăè£ŸèęĄæşĺæĎŔå¿őåŕŔäijŸåŇŰçŽĎçżŞæđIJãĂĆä¿ŃåęĆèĂČèŹŚäÿŃéİćåĹŻåžžäÿĂäÿłå■ŮåĚÿçŽĎă

```
a = {
    'name' : 'AAPL',
    'shares' : 100,
    'price' : 534.22
}
```

```
b = dict(name='AAPL', shares=100, price=534.22)
```

åŘŐéÍćäÿĂçğ■åĘŹæşŢæŻťçőĂæťĄäÿĂäžŻïijÍä¡ăäÿ■éIJĂèęĄåIJÍåĚşéŤőå■ŮäÿŁè¿ŞåĚěåijŢåŔůïijL'ãĂ
äÿ■è£ĞïijŇåęĆæđIJä¡ăářĘè£Źäÿd'äÿłäžčçăĄçL'Ğæőţè£ŻèąŇæĂğèČ¡æţŃèŕŢářźæŕŤæŮűïijŇäijŽåŔŞçŐřä¡£ç
dict() çŽĎæŰźåijŔäijŽæĚćäžĘ3åĂ■ãĂĆ çIJŃåĹřè£ŹäÿłïijŇä¡ăæŸŕäÿ■æŸŕæIJL'åĘšåŁÍæŁŁæL'ĂæIJL'ä¡£
dict() çŽĎäžčçăĄéČ¡æŻ£æ■ćæĹŘçňňäÿĂçğ■ãĂĆ äÿ■åd'§ïijŇèĄłæŸŐçŽĎçÍŃåžŔåŚŸåŔłäijŽåĘşæşÍäžŰ

åęĆæđIJä¡ăçŽĎäijŸåŇŰèęĄæśĆæŕŤè¿ČéńŸïijŇæIJňèŁĆçŽĎè£ŹäžŻçőĂå■ŢæŁĂæIJŕæżąèűşäÿ■äžĘïij
ä¿ŃåęĆïijŇPyPyåůěçÍŃæŸŕPythonèğčéĞŁåŹÍçŽĎåŔęåd'ŰäÿĂçğ■åőđçŐřïijŇåőČäijŽåĹĘæđŘä¡ăçŽĎçÍŃåžĺ
åőČæIJL'æŮűåĂŹèČ¡æđĄåd'ğçŽĎæŔŘå■ĞæĂğèČ¡ïijŇéĂŽåÿÿåŔŕäžěæŐěè£ŚCäžčçăĄçŽĎéĂ§åžęãĂĆ
äÿ■è£ĞåŔŕæČIJçŽĎæŸŕïijŇåĹřåĘŹè£ŹæIJňäźęä¡■ç¡őïijŇPyPyè£Ÿäÿ■èČ¡åőŇåĚÍæŤŕæŇĄPython3.
åŽăæ■d'ïijŇè£ŹäÿłæŸŕä¡ăářĘæÌěéIJĂèęĄåŐżçăŤçl'űçŽĎãĂĆä¡ăè£ŸåŔŕäžěèĂČèŹŚäÿŃNumbaåůěçÍŃïijŇ
NumbaæŸŕäÿĂäÿłåIJÍä¡ăä¡£çŤÍèčĚéěřåŹÍæÌěéĂL'æŃl'PythonåĞ¡æŢřè£ŻèąŇäijŸåŇŰæŮűçŽĎåŁÍæĂĄçijŰ
è£ŹäžŽåĞ¡æŢřäijŽä¡£çŤÍLLVMèćńçijŰèŕŚæĹŘæIJňåIJřæIJžåŹÍçăĄãĂĆåőČåŘŇæăůåŔŕäžěæđĄåd'ğçŽĎæŔ
ä¡ĘæŸŕïijŇèů§PyPyäÿĂæăůïijŇåőČåŕźäžŐPython 3çŽĎæŤŕæŇĄçŐřåIJÍè£ŸåĄIJçŢŹåIJÍåőđéłŇéŸűæőţãĂĆ

æIJĂåŘŐæĹŚåijŢçŤÍJohn Ousterhoutèŕť'è£ĞçŽĎèŕÍä¡IJäÿžçżŞår¿ïijŽâĂIJæIJĂåě¡çŽĎæĂğèČ¡äijŸåŇŰ
çŻť'åĹřä¡ăçIJ§çŽĎéIJĂèęĄäijŸåŇŰçŽĎæŮűåĂŹåĘ■åŐżèĂČèŹŚáőČãĂĆçąőä£Ìä¡ăçÍŃåžŔæ■čçąőçŽĎè£Řè

17 çňňå■ĄäžŤçńäïijŽCèŕ■èÍĂæĽľåśŢ

æIJňçńăçİĂçIJijäžŐäžŐPythonèő£éŮőCäžčçăĄçŽĎéŮőécŸãĂĆèőÿåď'ŽPythonåĘĚç¡őåžŞæŸŕçŤÍCåĘŹ
èő£éŮőCæŸŕèőľ'PythonçŽĎåŕźçŐřæIJL'åžŞè£ŻèąŇäžd'äžŠäÿĂäÿłéĞ■èęĄçŽĎçżĎæĹŘéČÍåĹĘãĂĆ
è£Źäź§æŸŕäÿĂäÿłå¡Şä¡ăéÌćäÿt'äžŐPython 2 åĹř Python 3æL'l'åśŢäžčçăĄçŽĎéŮőécŸãĂĆ
èŹ¡çĎűPythonæŘŘä¿ŻäžĘäÿĂäÿłåź£æşŻçŽĎçijŰçÍŃAPIïijŇåőđéŹĚäÿŁæIJL'å¿Ĺåd'ŽæŰźæşŢæÌěåd'ĎçŘĘ
çŻÿæŕŤèŕŢåŽ¿çżŹåĞžåŕźäžŐæŕŘäÿĂäÿłåŘŕèČ¡çŽĎåůěåĚůæĹŰæŁĂæIJŕçŽĎèŕęçżĘåŘĆèĂČïijŇ
æĹŚäźĹéĞĞçŤÍçŽĎæŸŕæŸŕéŽĘäÿ■åIJÍäÿĂäÿłåřŘçL'Ğæőţç ŽĎC++äžčçăĄijŇäžěåŘŁäÿĂäžŻæIJL'äžčèąÍæ
è£ŹäÿłçŻőæăĞæŸŕæŘŘä¿ŻäÿĂçşžåĹŰçŽĎçijŰçÍŃæÍąæÌ£ïijŇæIJL'çżŔéłŇçŽĎçÍŃåžŔåŚŸåŔŕäžěæL'l'åśŢè

è£ŹéĞŇæŸŕæĹŚäžňářĘåIJÍåd'ğéČÍåĹĘçğŸçś■äÿ■åůëä¡IJçŽĎäžčçăĄijŽ

```
/* sample.c */_method
#include <math.h>

/* Compute the greatest common divisor */
int gcd(int x, int y) {
    int g = y;
    while (x > 0) {
        g = x;
        x = y % x;
        y = g;
    }
    return g;
}

/* Test if (x0,y0) is in the Mandelbrot set or not */
int in_mandel(double x0, double y0, int n) {
```

```
    double x=0,y=0,xtemp;
    while (n > 0) {
        xtemp = x*x - y*y + x0;
        y = 2*x*y + y0;
        x = xtemp;
        n -= 1;
        if (x*x + y*y > 4) return 0;
    }
    return 1;
}

/* Divide two numbers */
int divide(int a, int b, int *remainder) {
    int quot = a / b;
    *remainder = a % b;
    return quot;
}

/* Average values in an array */
double avg(double *a, int n) {
    int i;
    double total = 0.0;
    for (i = 0; i < n; i++) {
        total += a[i];
    }
    return total / n;
}

/* A C data structure */
typedef struct Point {
    double x,y;
} Point;

/* Function involving a C data structure */
double distance(Point *p1, Point *p2) {
    return hypot(p1->x - p2->x, p1->y - p2->y);
}
```

è£Źæőţäžčçă Ąå ŇĚåŘńäžĘåď Žçğ■äÿ■åŘŇçŽĎCèŕ■èÍĂçijŰçÍŃçĽźæĂğãĂĆ éęŰåĚĹïijŇè£ŹéĞŇæIJĽå¿Ĺåď ŽåĞ¡æŢřæŕŤåęĆ `gcd()` åŠŇ `is_mandel()` ãĂĆ `divide()` åĞ¡æŢřæŸŕäÿĂäÿłè£ŤåŻđåď ŽäÿłåĂijçŽĎCåĞ¡æŢřä¿Ńå■ŘïijŇåĚűäÿ■æIJĽäÿĂäÿłæŸŕéĂŽè£Č `avg()` åĞ¡æŢřéĂŽè£ĞäÿĂäÿłCæŢřçżĎæĽğèąŇæŢřæ■őèĄŽéŻĘæŞ■ä¡IJãĂĆ`Point` åŠŇ `distance()` åĞ¡æŢřæűĽåŔŁåĹřäžĘCçżŞæđĎä¡ŞãĂĆ

åŕźäžŐæŐěäÿŃæİěçŽĎæĽĂæIJĽåřŔèŁĆïijŇåĚĹåĄĞåőŽäÿŁéİćçŽĎäžčçăĄåůšçżŔèćńåĘŹåĚěäžĘąäÿĂ çĎűåŘŐåőČäžńçŽĎåőŽäźĽèćńåĘŹåĚěäÿĂäÿłåŘ■åŔńâĂIJsample.hâĂİçŽĎåď ť æŰĞäżűäÿ■ïijŇ åźűäÿŤèćńçijŰèŕŚäÿžäÿĂäÿłåžŞåŔńâĂIJlibsampleâĂİïijŇèČ¡èćńéŞ¿æŐěåĹřåĚűäžŰCèŕ■èÍĂäžčçăĄäÿ■ãĂĆ çijŰèŕŚåŠŇéŞ¿æŐěçŽĎçżĘèŁĆä¿İæ■őçşżçż§çŽĎäÿ■åŘŇèĂŇäÿ■åŘŇïijŇä¡ĘæŸŕè£Źäÿłäÿ■æŸŕæĹŚäžňåĚ åęĆæđIJä¡äèęĄåď ĎçŘĘCäžčçăĄïijŇæĹŚäžňåĄĞåőŽè£ŹäžŽå§žçąĂçŽĎäÿIJèě£ä¡äéČ¡æŐŇæŔąäžĘãĂĆ

Contents:

17.1 15.1 ä¡£çŤÍctypesèő£éŮőCäžčçăĄ

éŮőécŸ

ä¡ăæIJL'äÿĂäžŻCåĞ¡æŢřåůšçżŔèćńçijŰèŕŠåĹřåĚśäžńåžŞæĹŰDLLäÿ■ãĂCä¡ååÿŇæIJŻåŔŕäžěä¡£çŤÍçžńèĂŇäÿ■çŤÍçijŰåĘŹéćÍåd'ŰçŽĎCäžčçăĄæĹŰä¡£çŤÍçňňäÿL'æŰźæL'l'åśŢåůěåĚůãĂĆ

èğčåĘşæŰźæąĹ

åŕźäžŐéIJĂèęĄèŕČçŤÍCäžčçăĄçŽĎäÿĂäžŻåřŔçŽĎéŮőécŸïijŇéĂŽåÿÿä¡£çŤÍPythonæăĞåĞĘåžŞäÿ■çŽctypes æÍąåİŮåřśèűşåd'§äžĘãĂĆ èęĄä¡£çŤÍ ctypes ïijŇä¡ăéęŰåĚĹèęĄçąőä£İä¡ăèęĄèő£éŮőçŽĎCäžčçăĄïijĹåŘŇæăůçŽĎæđűædĐãĂĄå■Ůåd'ğåřŔãĂĄçijŰèŕŠåŹÍç■L'ïijL'çŽĎæ§ŘäÿłåĚśäžńåžŞäÿ■äžĘãĂĆäÿžäžĘè£Żèą ŇæIJňèŁĆçŽĎæijŢçd'žïijŇåĄĞèő¿ä¡ăæIJL'äÿĂäÿłåĚśäžńåžŞåŘ■å■ŮåŔńlibsample.so ïijŇéĞŇéİćçŽĎåĘĚåőźåřśæŸŕ15çńääżŇçż■éČĺåĹĘéĆčæăůãĂĆåŔęåd'Úè£ŸåĄĞèő¿è£Źäÿłlibsample.so æŰĞäžűèćnæŢ¿ç¡őåĹřä¡■äžŐ sample.py æŰĞäžűçŻÿåŘŇçŽĎçŻőå¡Ţäÿ■äžĘãĂĆ

èęĄèő£éŮőè£ŹäÿłåĞ¡æŢřåžŞïijŇä¡ăèęĄåĚĹæđĎåžžäÿĂäÿłåŇĚèčĚåőČçŽĎPythonæÍąåİŮïijŇåęĆäÿŇè

```
# sample.py
import ctypes
import os

# Try to locate the .so file in the same directory as this file
_file = 'libsample.so'
_path = os.path.join(*(os.path.split(__file__)[:-1] + (_file,)))
_mod = ctypes.cdll.LoadLibrary(_path)

# int gcd(int, int)
gcd = _mod.gcd
gcd.argtypes = (ctypes.c_int, ctypes.c_int)
gcd.restype = ctypes.c_int

# int in_mandel(double, double, int)
in_mandel = _mod.in_mandel
in_mandel.argtypes = (ctypes.c_double, ctypes.c_double, ctypes.c_
↪int)
in_mandel.restype = ctypes.c_int

# int divide(int, int, int *)
_divide = _mod.divide
_divide.argtypes = (ctypes.c_int, ctypes.c_int, ctypes.
↪POINTER(ctypes.c_int))
_divide.restype = ctypes.c_int

def divide(x, y):
```

```
    rem = ctypes.c_int()
    quot = _divide(x, y, rem)

    return quot,rem.value

# void avg(double *, int n)
# Define a special type for the 'double *' argument
class DoubleArrayType:
    def from_param(self, param):
        typename = type(param).__name__
        if hasattr(self, 'from_' + typename):
            return getattr(self, 'from_' + typename)(param)
        elif isinstance(param, ctypes.Array):
            return param
        else:
            raise TypeError("Can't convert %s" % typename)

    # Cast from array.array objects
    def from_array(self, param):
        if param.typecode != 'd':
            raise TypeError('must be an array of doubles')
        ptr, _ = param.buffer_info()
        return ctypes.cast(ptr, ctypes.POINTER(ctypes.c_double))

    # Cast from lists/tuples
    def from_list(self, param):
        val = ((ctypes.c_double)*len(param))(*param)
        return val

    from_tuple = from_list

    # Cast from a numpy array
    def from_ndarray(self, param):
        return param.ctypes.data_as(ctypes.POINTER(ctypes.c_double))

DoubleArray = DoubleArrayType()
_avg = _mod.avg
_avg.argtypes = (DoubleArray, ctypes.c_int)
_avg.restype = ctypes.c_double

def avg(values):
    return _avg(values, len(values))

# struct Point { }
class Point(ctypes.Structure):
    _fields_ = [('x', ctypes.c_double),
                ('y', ctypes.c_double)]

# double distance(Point *, Point *)
distance = _mod.distance
```

```
distance.argtypes = (ctypes.POINTER(Point), ctypes.POINTER(Point))
distance.restype = ctypes.c_double
```

åęĆæđIJäÿĂåĹĞæ■čåÿÿïijŇä¡ăåŕśåŔŕäżěåŁăè¡¡åźűä¡£çŤĺéĞŇéİćåőŽäźĽçŽĎCåĞ¡æŢřäžĘãĂĆä¿ŃåęĆï

```
>>> import sample
>>> sample.gcd(35,42)
7
>>> sample.in_mandel(0,0,500)
1
>>> sample.in_mandel(2.0,1.0,500)
0
>>> sample.divide(42,8)
(5, 2)
>>> sample.avg([1,2,3])
2.0
>>> p1 = sample.Point(1,2)
>>> p2 = sample.Point(4,5)
>>> sample.distance(p1,p2)
4.242640687119285
>>>
```

èőĺèőž

æIJňåřŔèŁĆæIJĽå¿Ĺåď ŽåĂijå¿ŮæĹŚäżňèŕęçżĘèőĺèőžçŽĎåIJřæŰźãĂĆ éęŰåĚĹæŸŕåŕźäžŐCåŠŇPythonäżčçăĄäÿĂèţůæĽŞåŇĚçŽĎéŮőéćŸïijŇåęĆæđIJä¡ăåIJĺä¡£çŤĺ ctypes æİěèő£éŮőçijŰèŕŚåŘŐçŽĎCäżčçăĄïijŇ éĆčäźĹéIJĂèęĄçąőä£İè£ŹäÿłåĚśäžńåžŞæŤ¿åIJĺ sample.py æĺąåİŮåŘŇäÿĂäÿłåIJřæŰźãĂĆ äÿĂçğ■åŔŕèČ¡æŸŕåřĘçŤ§æĹŘçŽĎ . so æŰĞäżűæŤ¿ç¡őåIJĺèęĄä¡£çŤĺåőČçŽĎPythonäżčçăĄåŘŇäÿĂäÿłçŻőå¡ŢäÿŃãĂĆ æĹŚäżňåIJĺ recipeâĂŤsample.py äÿ■ä¡£çŤĺ __file__ åŔŸéĞŔæİěæ§ěçIJŃåőČèćńåőĽèčĚçŽĎä¡■ç¡őïijŇ çĎűåŘŐæđĎéĂăäÿĂäÿłæŇĞåŘŚåŘŇäÿĂäÿłçŻőå¡Ţäÿ■ç libsample.so æŰĞäżűçŽĎèůŕå¿ĎãĂĆ

åęĆæđIJCåĞ¡æŢřåžŞèćńåőĽèčĚåĹřåĚűäżŰåIJřæŰźïijŇéĆčäźĹä¡ăåŕśèęĄä£őæŤźçŻÿåžŤçŽĎèůŕå¿ĎãĂ åęĆæđIJCåĞ¡æŢřåžŞåIJĺä¡ăæIJžåŹĺäÿŁèćńåőĽèčĚäÿžäÿĂäÿłæăĞåĞĘåžŞäžĘïijŇ éĆčäźĹåŔŕäżěä¡£çŤĺ ctypes.util.find_library() åĞ¡æŢřæİěæ§ěæĽ¿ïijŽ

```
>>> from ctypes.util import find_library
>>> find_library('m')
'/usr/lib/libm.dylib'
>>> find_library('pthread')
'/usr/lib/libpthread.dylib'
>>> find_library('sample')
'/usr/local/lib/libsample.so'
>>>
```

äÿĂæŮęä¡ăç§ěéĄŞäžĘCåĞ¡æŢřåžŞçŽĎä¡■ç¡őïijŇéĆčäźĹåŕśåŔŕäżěåČŔäÿŃéİćè£Źæăůä¡£çŤĺ ctypes.cdll.LoadLibrary() æİěåŁăè¡¡åőČïijŇ åĚűäÿ■ _path æŸŕæăĞåĞĘåžŞçŽĎåĚĺèůŕå¿ĎïijŽ

```
_mod = ctypes.cdll.LoadLibrary(_path)
```

åĞ¡æŢřåžŞèćńåŁäè¡¡åŘŐïijŇä¡äéIJĂèęĄçijŰåĘŹåĞääÿłèŕ■åŔěæİěæŔŘåŔŰçĽźåőŽçŽĎçňęåŔůåźűæŇ
åřśåČŔäÿŃéİćè£ŹäÿłäżčçăĄçĽĞæőţäÿĂæăůïijŽ

```
# int in_mandel(double, double, int)
in_mandel = _mod.in_mandel
in_mandel.argtypes = (ctypes.c_double, ctypes.c_double, ctypes.c_
↪int)
in_mandel.restype = ctypes.c_int
```

åIJÍè£ŹæőţäżčçăĄäÿ■ïijŇ.argtypes åśđæĂğæŸŕäÿĂäÿłåĚČçżĎïijŇåŇĚåŘńäžĘæ§ŘäÿłåĞ¡æŢřçŽĎè
èĂŇ .restype åřśæŸŕçŻÿåžŤçŽĎè£ŤåŻđçśżåđŃãĂĆ ctypes
åőŽäźĽäžĘåď ğéĞŔçŽĎçśżåđŃåŕźèśąïijĹæŕŤåęĆc_double, c_int, c_short, c_floatç■ĽïijĽïijŇ
äżčèąĺäžĘåŕźåžŤçŽĎCæŢřæ■őçśżåđŃãĂĆåęĆæđIJä¡ăæČşèőľPythonèČ¡åď §äijäéĂŠæ■čçąőçŽĎåŔĆæŢřçśż
éĆčäźĹè£ŹäžŻçśżåđŃç■¿åŘ■çŽĎçżŚåőŽæŸŕå¿ĹéĞ■èęĄçŽĎäÿĂæ■ěãĂĆåęĆæđIJä¡ăæšąæIJĽè£ŹäźĹåĄŽïi
è£ŸåŔŕèČ¡äijŽåŕijèĞťæŢťäÿłèğčéĞŁåŹĺè£ŻçĺŃæŇĆæŐĽãĂĆ
ä¡£çŤĺctypesæIJĽäÿĂäÿłéžżçČęçĆźçŽĎåIJřæŰźæŸŕåŐ§çŤ§çŽĎCäżčçăĄä¡£çŤĺçŽĎæIJŕèŕ■åŔŕèČ¡èů§Pytho
divide() åĞ¡æŢřæŸŕäÿĂäÿłå¿Ĺåě¡çŽĎä¿Ńå■ŘïijŇåőČéĂŽè£ĞäÿĂäÿłåŔĆæŢřéŹďäżěåŔęäÿĂäÿłåŔĆæŢ
åř¡çőąè£ŹæŸŕäÿĂäÿłå¿ĹåÿÿèğĄçŽĎCæŁĂæIJŕïijŇä¡ĘæŸŕåIJĺPythonäÿ■å■ťäÿ■ç§ěéĄŞæĂŐæăůæÿĚæŹřçŽ
ä¿ŃåęĆïijŇä¡ăäÿ■èČ¡åČŔäÿŃéİćè£ŹæăůçőĂå■ŢçŽĎåĄŽïijŽ

```
>>> divide = _mod.divide
>>> divide.argtypes = (ctypes.c_int, ctypes.c_int, ctypes.
↪POINTER(ctypes.c_int))
>>> x = 0
>>> divide(10, 3, x)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ctypes.ArgumentError: argument 3: <class 'TypeError'>: expected LP_
↪c_int
instance instead of int
>>>
```

åřśçőŮè£ŹäÿłèČ¡æ■čçąőçŽĎåůěä¡IJïijŇåőČäijŽè£İåŔ■PythonåŕźäžŐæŢťæŢřçŽĎäÿ■åŔŕæŻťæŤźåŐ§å
åŕźäžŐæűĽåŔŁåĹřæŇĞéŠĹçŽĎåŔĆæŢřïijŇä¡äéĂŽåÿÿéIJĂèęĄåĚĹæđĎåżžäÿĂäÿłçŻÿåžŤçŽĎctypesåŕźèśą

```
>>> x = ctypes.c_int()
>>> divide(10, 3, x)
3
>>> x.value
1
>>>
```

åIJÍè£ŹéĞŇïijŇäÿĂäÿł ctypes.c_int åőđä¿ŃèćńåĹŻåżžåźűä¡IJäÿžäÿĂäÿłæŇĞéŠĹèćńäijäè£ŻåŐżã
èů§æŹőéĂŽPythonæŢťå¡ćäÿ■åŘŇçŽĎæŸŕïijŇäÿĂäÿł c_int
åŕźèśąæŸŕåŔŕäżěèćńä£őæŤźçŽĎãĂĆ .value åśđæĂğåŔŕèćńçŤĺæİěèŐůåŔŰæĹŰæŻťæŤźè£ŹäÿłåĂijãĂĆ

åŕźäžŐéĆčäžŻäÿ■åČŔPythonçŽĎCèřČçŤĺïijŇéĂŽåÿÿåŔŕäżěåĘŹäÿĂäÿłåřŔçŽĎåŇĚèčĚåĞ¡æŢřãĂĆ
è£ŹéĞŇïijŇæĹŚäżňèőľ divide() åĞ¡æŢřéĂŽè£ĞåĚČçżĎæİěè£ŤåŻđäÿďäÿłçżŞæđIJïijŽ

```
# int divide(int, int, int *)
_divide = _mod.divide
_divide.argtypes = (ctypes.c_int, ctypes.c_int, ctypes.
↪POINTER(ctypes.c_int))
_divide.restype = ctypes.c_int

def divide(x, y):
    rem = ctypes.c_int()
    quot = _divide(x,y,rem)
    return quot, rem.value
```

avg() åĞ¡æŢřåŔĹæŸŕäÿĂäÿłæŰřçŽĎæŇŚæĹŸãĂĆCäžčçăĄæIJ§æIJŻæŐěåŔŮåĹřäÿĂäÿłæŇĞéŠĹåŠ
ä¡ĘæŸŕïijŇåIJĺPythonäÿ■ïijŇæĹŚäžňå£Ěéążè ĂČèŹŚè£ŹäÿłéŮőécŸïijŽæŢřçžĎæŸŕåŢěïij§åőČæŸŕäÿĂäÿłåĹ
è£ŸæŸŕ array æĺąåĺŮäÿ■çŽĎäÿĂäÿłæŢřçžĎïij§è£ŸæŸŕäÿĂäÿł numpy
æŢřçžĎïij§è£ŸæŸŕèŕťæĽĂæIJL'éČ¡æŸŕïij§ åőđéŹĚäÿŁïijŇäÿĂäÿłPythonâĂIJæŢřçžĎâĂİæIJL'åđ'Žçğ■å¡ćå

DoubleArrayType æijŤçđ'žäžĘæĂŐæăůåđ'ĎçŘĘè£Źçğ■æČĚåĘţãĂĆ
åIJĺè£Źäÿłçśžäÿ■åőŽäźL'äžĘäÿĂäÿłå■ŢäÿłæŰźæşŢ from_param() ãĂĆ
è£ŹäÿłæŰźæşŢçŽĎèğŠèL'šæŸŕæŐěåŔŮäÿĂäÿłå■ŢäÿłåŔĆæŢřçĎůåŘŐářĘåĚűåŘŚäÿŃè¡ňæ■ćäÿžäÿĂäÿłåŘĺ
ïijĹæIJňä¿Ńäÿ■æŸŕäÿĂäÿł ctypes.c_double çŽĎæŇĞéŠĹïijL'ãĂĆ
åIJĺ from_param() äÿ■ïijŇä¡ăåŔŕäžěåĄŽäżżä¡Ţä¡ăæČşåĄŽçŽĎäžŃãĂĆ
åŔĆæŢřçŽĎçśžåđŃåŘ■èćńæŔŘåŔŮåĞžæİěåźűèćńçŤĺäžŐåĹĘåŘŚåĹřäÿĂäÿłæŻť'åĚůä¡ŞçŽĎæŰźæşŢäÿ■åŐ
ä¿ŃåęĆïijŇåęĆæđIJäÿĂäÿłåĹŮèąĺèćńäijăéĂŠè£ĞæİěïijŇéĆčäźĹ typename åřśæŸŕ list
ïijŇ çĎűåŘŐ from_list æŰźæşŢèćńèřČçŤĺãĂĆ

åŕźäžŐåĹŮèąĺåŠŇåĚČçżĎïijŇfrom_list æŰźæşŢářĘåĚűè¡ňæ■ćäÿžäÿĂäÿł ctypes
çŽĎæŢřçžĎåŕźèśąãĂĆ è£ŹäÿłçIJŃäÿŁåŐżæIJL'çĆźåěĞæĂłïijŇäÿŃéİćæĹŚäžňä¡£çŤĺäÿĂäÿłäžd'äžŠåijŔä¿Ń
ctypes æŢřçžĎïijŽ

```
>>> nums = [1, 2, 3]
>>> a = (ctypes.c_double * len(nums))(*nums)
>>> a
<__main__.c_double_Array_3 object at 0x10069cd40>
>>> a[0]
1.0
>>> a[1]
2.0
>>> a[2]
3.0
>>>
```

åŕźäžŐæŢřçžĎåŕźèśąïijŇfrom_array() æŔŘåŔŮåžŢåśĆçŽĎåĘĚå■ŸæŇĞéŠĹåźűåřĘåĚűè¡ňæ■ćäÿž
ctypes æŇĞéŠĹåŕźèśąãĂĆä¿ŃåęĆïijŽ

```
>>> import array
>>> a = array.array('d',[1,2,3])
>>> a
array('d', [1.0, 2.0, 3.0])
>>> ptr_ = a.buffer_info()
>>> ptr
4298687200
```

```
>>> ctypes.cast(ptr, ctypes.POINTER(ctypes.c_double))
<__main__.LP_c_double object at 0x10069cd40>
>>>
```

from_ndarray() æijŤçd'žäžĘåŕźäžŐ numpy æŢřçżĎçŽĎè¡ňæ■ćæŞ■ä¡IJãĂĆ éĂŽè£ĞåőŽäźL' DoubleArrayType çśžåźűåIJÍ avg() çśžåđŃç■¿åŘ■äÿ■ä¡£çŤÍåőČïijŇ éĆčäźĹè£ŹäÿłåĞ¡æŢřåřśèČ¡æŐěåŔŮåd'Žäÿłäÿ■åŘŇçŽĎçśżæŢřçżĎè¿ŞåĚěäžĘïijŽ

```
>>> import sample
>>> sample.avg([1,2,3])
2.0
>>> sample.avg((1,2,3))
2.0
>>> import array
>>> sample.avg(array.array('d',[1,2,3]))
2.0
>>> import numpy
>>> sample.avg(numpy.array([1.0,2.0,3.0]))
2.0
>>>
```

æIJňèŁĆæIJĂåŘŐäÿĂéČÍåĹĘåŘŚä¡ăæijŤçd'žäžĘæĂŐæăůåd'ĎçŘĘäÿĂäÿłçőĂå■ŢçŽĎCçżŞæđĎãĂĆ åŕźäžŐçżŞæđĎä¡ŞïijŇä¡ääŔłéIJĂèęĄåČŔäÿŃéÍćè£ŹæăůçőĂå■ŢçŽĎåőŽäźL'äÿĂäÿłçśžïijŇåŇĚåŘńçŻÿåžŤç

```
class Point(ctypes.Structure):
    _fields_ = [('x', ctypes.c_double),
                ('y', ctypes.c_double)]
```

äÿĂæŮęçśżèćńåőŽäźL'åŘŐïijŇä¡ääŕśåŔŕäžěåIJÍçśżåđŃç■¿åŘ■äÿ■æĹŰèĂĚæŸŕéIJĂèęĄåőđä¿ŃåŇŰçżŞ

```
>>> p1 = sample.Point(1,2)
>>> p2 = sample.Point(4,5)
>>> p1.x
1.0
>>> p1.y
2.0
>>> sample.distance(p1,p2)
4.242640687119285
>>>
```

æIJĂåŘŐäÿĂäžŽåŕŔçŽĎæŔŘçd'žïijŽåęĆæđIJä¡ăæČşåIJÍPythonäÿ■èő£éŮőäÿĂäžŽåŕŔçŽĎCåĞ¡æŢřïijŇ ctypes æŸŕäÿĂäÿłå¿ĹæIJL'çŤÍçŽĎåĞ¡æŢřåžŞãĂĆ år¡çőąåęĆæ■d'ïijŇåęĆæđIJä¡ăæČşèęĄåŐżèő£éŮőäÿĂä Swig (15.9èŁĆäijŽèőšåĹř) æĹŰ CythonïijĹ15.10èŁĆïijL'ãĂĆ

åŕźäžŐåd'ğåđŃåžŞçŽĎèő£éŮőæIJL'äÿłäÿżèęĄéŮőécŸïijŇçŤšäžŐctypesåźűäÿ■æŸŕåőŇåĚÍèĞłåŁÍåŇŰï éĆčäźĹä¡ääŕšå£ĚéążèŁśèt'źåd'ğéĞŔæŮűéŮt'æÍěçijŰåĘŹæL'ĂæIJL'çŽĎçśżåđŃç¿åŘ■ïijŇåŕšåČŔä¿Ńå■Řäÿ åęĆæđIJåĞ¡æŢřåžŞåd'§åd'■æÍĆïijŇä¡ăè£Ÿå¿ŮåŐżçijŰåĘŹå¿Ĺåd'ŽåřŔçŽĎåŇĚèčĚåĞ¡æŢřåŠŇæŤŕæŇĄçśż åŔęåd'ŰïijŇéŹd'éÍđä¡ääůšçżŔåőŇåĚÍçš¿éĂŽäžĘæL'ĂæIJL'åžŢåśĆçŽĎCæŐěåŔčçżĘèŁĆïijŇåŇĚæŃňåĘĚå éĂŽåÿÿäÿĂäÿłå¿ĹåřŔçŽĎäžčçăĄçijžéŹůãĂĄèő£éŮőèűŁçŢŇæĹŰåĚűäžŰçśżäijijéŤŹèŕŕåřśèČ¡èől'PythonçÍŇ

ä¡IJäÿž ctypes çŽĎäÿĂäÿłæŻ£äžčïijŇä¡ăè£ŸåŔŕäžěèĂČèŹŚäÿŃCFFIãĂĆCFFIæŔŘä¿ŻäžĘå¿Ĺåd'Žç

ä¡ĘæŸŕä¡£çŤÍCèŕ■æşŢåźűæŤŕæŇĄæŻtˇådˇŽéńŸçžğçŽĎCäžčçăĄçśżåđŃãĂĆ
åĹřåĘŹè£ŹæIJñäźęäÿžæ■ćïijŇCFFIè£ŸæŸŕäÿĂäÿłçŽÿåŕźè¿ČæŰřçŽĎåůěçÍŃïijŇ
ä¡ĘæŸŕåőČçŽĎæţĄèąŇåžęæ■čåIJĺå£néĂ§äÿŁå■ĞãĂĆ çŤŽèĞşè£ŸæIJL’åIJĺèőÍèőžåIJÍPythonåřĘæİěçŽĎçL’Í

17.2 15.2 çőĂå■ŢçŽĎCæĽľåśŢæÍąåİŮ

éŮőécŸ

ä¡ăæČşäÿ■ä¿İéİăåĚűäžŰåůěåĚůïijŇçŻtˇæŐěä¡£çŤÍPythonçŽĎæĽľåśŢAPIæİěçijŰåĘŹäÿĂäžŽçőĂå■ŢçZ

èğčåĘşæŰźæąĹ

åŕźäžŐçőĂå■ŢçŽĎCäžčçăĄïijŇæđĎåżžäÿĂäÿłèĞłåőŽäźĽæĽľåśŢæÍąåİŮæŸŕå¿Ĺåőźæŷ§çŽĎãĂĆ
ä¡IJäÿžçňňäÿĂæ■ěïijŇä¡ăéIJĂèęĄçąőä£İä¡ăçŽĎCäžčçăĄæIJL’äÿĂäÿłæ■čçąőçŽĎåď’æŰĞäžűãĂĆä¿ŃåęĆïij

```
/* sample.h */

#include <math.h>

extern int gcd(int, int);
extern int in_mandel(double x0, double y0, int n);
extern int divide(int a, int b, int *remainder);
extern double avg(double *a, int n);

typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

éĂŽåÿÿæİěèőšïijŇè£ŹäÿłåďˇæŰĞäžűèęĄåŕźåžŤäÿĂäÿłåůšçżŔèćńå■ŢçŃňçijŰèŕŚè£ĞçŽĎåžŞãĂĆ
æIJL’äžĘè£ŹäžŻïijŇäÿŃéİćæĹŚäžňæijŤçďžäÿŃçijŰåĘŹæĽľåśŢåĞ¡æŢřçŽĎäÿĂäÿłçőĂå■Ţä¿Ńå■ŘïijŽ

```
#include "Python.h"
#include "sample.h"

/* int gcd(int, int) */
static PyObject *py_gcd(PyObject *self, PyObject *args) {
  int x, y, result;

  if (!PyArg_ParseTuple(args,"ii", &x, &y)) {
    return NULL;
  }
  result = gcd(x,y);
  return Py_BuildValue("i", result);
}

/* int in_mandel(double, double, int) */
```

```
static PyObject *py_in_mandel(PyObject *self, PyObject *args) {
  double x0, y0;
  int n;
  int result;

  if (!PyArg_ParseTuple(args, "ddi", &x0, &y0, &n)) {
    return NULL;
  }
  result = in_mandel(x0,y0,n);
  return Py_BuildValue("i", result);
}

/* int divide(int, int, int *) */
static PyObject *py_divide(PyObject *self, PyObject *args) {
  int a, b, quotient, remainder;
  if (!PyArg_ParseTuple(args, "ii", &a, &b)) {
    return NULL;
  }
  quotient = divide(a,b, &remainder);
  return Py_BuildValue("(ii)", quotient, remainder);
}

/* Module method table */
static PyMethodDef SampleMethods[] = {
  {"gcd",  py_gcd, METH_VARARGS, "Greatest common divisor"},
  {"in_mandel", py_in_mandel, METH_VARARGS, "Mandelbrot test"},
  {"divide", py_divide, METH_VARARGS, "Integer division"},
  { NULL, NULL, 0, NULL}
};

/* Module structure */
static struct PyModuleDef samplemodule = {
  PyModuleDef_HEAD_INIT,

  "sample",           /* name of module */
  "A sample module",  /* Doc string (may be NULL) */
  -1,                 /* Size of per-interpreter state or -1 */
  SampleMethods       /* Method table */
};

/* Module initialization function */
PyMODINIT_FUNC
PyInit_sample(void) {
  return PyModule_Create(&samplemodule);
}
```

èęĄçżŚåőŽè£ŹäÿłæĽľåśŢæĺąåİŮïijŇåČŔäÿŃéİćè£ŹæăůåĹŻåżžäÿĂäÿł `setup.py` æŰĞäżűïijŽ

```
# setup.py
from distutils.core import setup, Extension

setup(name='sample',
      ext_modules=[
        Extension('sample',
                  ['pysample.c'],
                  include_dirs = ['/some/dir'],
                  define_macros = [('FOO','1')],
                  undef_macros = ['BAR'],
                  library_dirs = ['/usr/local/lib'],
                  libraries = ['sample']
                  )
        ]
)
```

äÿžäžĘæđĎåžžæIJÄçżĹçŽĎåĞ¡æŢřåžŞïijŇåŔłéIJÄçőÄå■ŢçŽĎä¡£çŤÍ `python3 buildlib.py build_ext --inplace` åŚ¡äžď'å■şåŔřïijŽ

```
bash % python3 setup.py build_ext --inplace
running build_ext
building 'sample' extension
gcc -fno-strict-aliasing -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-
 ↪prototypes
 -I/usr/local/include/python3.3m -c pysample.c
 -o build/temp.macosx-10.6-x86_64-3.3/pysample.o
gcc -bundle -undefined dynamic_lookup
build/temp.macosx-10.6-x86_64-3.3/pysample.o \
 -L/usr/local/lib -lsample -o sample.so
bash %
```

åęĆäÿŁæĽĂçd'žïijŇåőČäijŽåĹŻåžžäÿĂäÿłåŘ■å■ŮåŔń `sample.so` çŽĎåĚśäžńåžŞãĂĆå¡Şèćńçij Űèŕ ŚåŘŐïijŇä¡ääřśèČ¡åřĘåőČä¡IJäÿžäÿĂäÿłæĺąåİŮåŕijåĚěè£ŻæİěäžĘïijŽ

```
>>> import sample
>>> sample.gcd(35, 42)
7
>>> sample.in_mandel(0, 0, 500)
1
>>> sample.in_mandel(2.0, 1.0, 500)

0
>>> sample.divide(42, 8)
(5, 2)
>>>
```

åęĆæđIJä¡ăæŸŕåIJĺWindowsæIJžåŹĺäÿŁéĺćåřİèŕŢè£ŹäžŻæ■ěéłd'ïijŇåŔŕèČ¡äijŽéĄĞåĹřåŘĎçğ■çŐŕåćČ PythonçŽĎäžŇè£ŻåĹűåĹĘåŔŚéĂŽåÿÿä¡£çŤĺäžĘMicrosoft Visual StudioæİěæđĎåžžãĂĆ äÿžäžĘèől'è£ŹäžŻæĽ'l'åśŢèČ¡æ■čåÿÿåůěä¡IJïijŇä¡ăéIJÄèęĄä¡£çŤĺåŘŇæăůæĹŰåĚijåőźçŽĎåůěåĚůæİěçijŰè åŔĆèĂČçŻÿäžŤçŽĎ PythonæŰĞæąč

èőÍèőž

åIJÍåřÌèřŢäżżä¡ŢæĽ’ŃåĘŹæĽ’ľ åśŢäźŃåĽ’■ïijŇæIJĂåě¡èČ¡åĚĹåŔĆèĂČäÿŃPythonæŰĞæąčäÿ■çŽĎ æĽ’ľ åśŢåŠŇåţŇåĚěPythonèğčéĞŁåŹÍ . PythonçŽĎCæĽ’ľ åśŢAPIå¿Ĺåď ğïijŇåIJÍè£ŹéĞŇæŢť äÿłåŐżèőšè£řåč äÿ■è£Ğåŕźäž ŐæIJĂæäÿå£ČçŽĎéČĺåĹĘè£ŸæŸŕåŔŕäżěèőÍèőžäÿŃçŽĎãĂĆ

éęŰåĚĹïijŇåIJÍæĽ’ľ åśŢæĺąåİŮäÿ■ïijŇä¡ăåĘŹçŽĎåĞ¡æŢřéČ¡æŸŕåČŔäÿŃéİćè£ŹæăůçŽĎäÿĂäÿłæŹőéĂ

```
static PyObject *py_func(PyObject *self, PyObject *args) {
  ...
}
```

`PyObject` æŸŕäÿĂäÿłèČ¡èąĺçď žäżżä¡ŢPythonåŕźèśąçŽĎCæŢřæ■őçśžåđŃãĂĆ åIJÍäÿĂäÿłéńŸçžğåśĆéİćïijŇäÿĂäÿłæĽ’ľ åśŢåĞ¡æŢřåřśæŸŕäÿĂäÿłæŐěåŔŮäÿĂäÿłPythonåŕźèśą ïijĹåIJÍ PyObject *argsäÿ■ïijĽåĚČçžĎåźűè£ŤåŽđäÿĂäÿłæŰřPythonåŕźèśąçŽĎCåĞ¡æŢřãĂĆ åĞ¡æŢřçŽĎ `self` åŔĆæŢřåŕźäžŐçőĂå■ŢçŽĎæĽ’ľ åśŢåĞ¡æŢřæšąæIJL’èćńä¡£çŤĺåĹřïijŇ äÿ■è£ĞåęĆæđIJä¡ăæČşåőŽäźĽæŰřçŽĎçśżæĹŮèĂĚæŸŕĆäÿ■çŽĎåŕźèśąçśżåđŃçŽĎèŕİåřśèČ¡æť¿äÿŁçŤĺåIJž éĆčäźĹ `self` åřśèČ¡åijŢçŤĺéĆčäÿłåőđä¿ŃäžĘãĂĆ

`PyArg_ParseTuple()` åĞ¡æŢřèćńçŤĺæİěåřĘPythonäÿ■çŽĎåĂijè¡ñæ■ćæĹŘCäÿ■åŕźåžŤèąĺçď žãĂĆ åőČæŐěåŔŮäÿĂäÿłæŇĞåőŽè¿ŞåĚěæăijåijåijŔçŽĎæăijåijŔåŇŰå■Ůçňęäÿšä¡IJäÿžè¿ŞåĚěïijŇæŕŤåęĆâĂIJiâĂİ åŘŇæăůè£ŸæIJL’å■ŸæŤ¿è¡ñæ■ćåŘŐçżŞæđIJçŽĎCåŔŸéĞŔçŽĎåIJřåİĂãĂĆ åęĆæđIJè¿ŞåĚěçŽĎåĂijäÿ■åŇźéĚ■è£ŹäÿłæăijåijŔåŇŰå■ŮçňęäÿšïijŇåřśäijŽæŁŻåĞžäÿĂäÿłåijĆåÿÿåźűè£Ťá éĂŽè£ĞæčĂæ§ěåźűè£ŤåŽđNULLïijŇäÿĂäÿłåŘĹéĂĆçŽĎåijĆåÿÿäijŽåIJÍèŕĈçŤĺäžčçăĄäÿ■èćńæŁŻåĞžãĂĆ

`Py_BuildValue()` åĞ¡æŢřèćńçŤĺæİěæăźæ■őCæŢřæ■őçśžåđŃåĹŻåżžPythonåŕźèśąãĂĆ åőČåŘŇæăůæŐěåŔŮäÿĂäÿłæăijåijŔåŇŰå■ŮçňęäÿšæİěæŇĞåőŽæIJ§æIJŻçśžåđŃãĂĆ åIJÍæĽ’ľ åśŢåĞ¡æŢřäÿ■ïijŇåőČèćńçŤĺæİěè£ŤåŽđçżŞæđIJçżŹPythonãĂĆ `Py_BuildValue()` çŽĎäÿĂäÿłçĽźæĂğæŸŕåőČèČ¡æđĎåżžæŻť åŁăåď ■æİĆçŽĎåŕźèśąçśžåđŃïijŇæŕŤåęĆ åIJÍ `py_divide()` äžčçăĄäÿ■ïijŇäÿĂäÿłä¿Ńå■ŘæijŢçď žäžĘæĂŐæăůè£ŤåŽđäÿĂäÿłåĚČçżĎãĂĆäÿ■è£Ğïï

```
return Py_BuildValue("i", 34);       // Return an integer
return Py_BuildValue("d", 3.4);      // Return a double
return Py_BuildValue("s", "Hello"); // Null-terminated UTF-8 string
return Py_BuildValue("(ii)", 3, 4); // Tuple (3, 4)
```

åIJÍæĽ’ľ åśŢæĺąåİŮåžŢéČĺïijŇä¡ăäijŽåŔŚçŐřäÿĂäÿłåĞ¡æŢřèąĺïijŇæŕŤåęĆæIJńèŁĆäÿ■çŽĎ `SampleMethods` èąĺãĂĆ è£ŹäÿłèąĺåŔŕäžěåĹŮåĞžCåĞ¡æŢřãĂĄPythonäÿ■ä¡£çŤĺçŽĎåŘ■å■ŮãĂĄæŰĞæa æĽĂæIJL’æĺąåİŮéČ¡éIJĂèęĄæŇĞåőŽè£ŹäÿłèąĺïijŇåŽăäÿžåőČåIJÍæĺąåİŮåĹİåğŃåŇŰæŮűèęĄèćńä¡£çŤĺåĹř

æIJĂåŘŐçŽĎåĞ¡æŢř `PyInit_sample()` æŸŕæĺąåİŮåĹİåğŃåŇŰåĞ¡æŢřïijŇä¡ĘèŕěæĺąåİŮçňňäÿĂæì è£ŹäÿłåĞ¡æŢřçŽĎäÿżèęĄåůěä¡IJæŸŕåIJÍèğčéĞŁåŹĺäÿ■æşĺåĘŇæĺąåİŮåŕźèśąãĂĆ

æIJĂåŘŐäÿĂäÿłèęĄçĆźéIJĂèęĄæŔŘåĞžæİěïijŇä¡£çŤĺCåĞ¡æŢřæİěæĽ’ľ åśŢPythonèęĄèĂČèŹŚçŽĎäžl ïijĹåőđéŹĚäÿŁïijŇC APIåŇĚåŘńäžĘèűĚè£Ğ500äÿłåĞ¡æŢřïijL’ãĂĆä¡ăåžŤèŕěårĘæIJňèŁĆå¡ŞåĄŽæŸŕäÿĂäÿ æŻť åď Žéńŷçžğå ĘĚåőźïijŇåŔŕäžěçIJŃçIJŃ `PyArg_ParseTuple()` åŠŇ `Py_BuildValue()` åĞ¡æŢřçŽĎæŰĞæąčïijŇ çĎűåŘŐè£ŻäÿĂæ■ěæĽ’ľ åśŢåijÃãĂĆ

17.3 15.3 çijŰåĘŹæĽľåśŢåĞ¡æŢřæŞ■ä¡IJæŢřçżĎ

éŮőéćŸ

ä¡ăæČşçijŰåĘŹäÿĂäÿłCæĽľåśŢåĞ¡æŢřæİěæŞ■ä¡IJæŢřçżĎïijŇåŔŕèČ¡æŸŕèćńarrayæĺąåİŮæĹŰçśżäijijN
äÿ■è£ĞïijŇä¡ăæČşèőľä¡ăçŽĎåĞ¡æŢřæŻťåŁăéĂŽçŤĺïijŇèĂŇäÿ■æŸŕéŠĹåŕźæ§ŘäÿłçĽźåőŽçŽĎåžŞæĽĂçŤ

èğčåĘşæŰźæąĹ

äÿžäžĘèČ¡èőľæŐěåŔŮåŠŇåďĎçŘĘæŢřçżĎåĚůæIJĽåŔŕçğżæď■æĂğïijŇä¡ăéIJĂèęĄä¡£çŤĺåĹř *Buffer Protocol* . äÿŃéİćæŸŕäÿĂäÿłæĽŃåĘŹçŽĎCæĽľåśŢåĞ¡æŢřä¿Ńå■ŘïijŇ çŤĺæİěæŐěåŔŮæŢřçżĎæŢřæ■őåźűèřČçŤĺæIJňçńăåijĂçŕĞéČĺåĹĘçŽĎ `avg(double *buf, int len)` åĞ¡æŢřïijŽ

```
/* Call double avg(double *, int) */
static PyObject *py_avg(PyObject *self, PyObject *args) {
  PyObject *bufobj;
  Py_buffer view;
  double result;
  /* Get the passed Python object */
  if (!PyArg_ParseTuple(args, "O", &bufobj)) {
    return NULL;
  }

  /* Attempt to extract buffer information from it */

  if (PyObject_GetBuffer(bufobj, &view,
      PyBUF_ANY_CONTIGUOUS | PyBUF_FORMAT) == -1) {
    return NULL;
  }

  if (view.ndim != 1) {
    PyErr_SetString(PyExc_TypeError, "Expected a 1-dimensional array
↪");
    PyBuffer_Release(&view);
    return NULL;
  }

  /* Check the type of items in the array */
  if (strcmp(view.format,"d") != 0) {
    PyErr_SetString(PyExc_TypeError, "Expected an array of doubles
↪");
    PyBuffer_Release(&view);
    return NULL;
  }

  /* Pass the raw buffer and size to the C function */
  result = avg(view.buf, view.shape[0]);
```

```
  /* Indicate we're done working with the buffer */
  PyBuffer_Release(&view);
  return Py_BuildValue("d", result);
}
```

äÿŃéİćæĹŚäžňæijŤçď'žäÿŃè£ŹäÿłæL'ľåśŢåĞ¡æŢřæŸŕåęĆä¡Ţåůěä¡IJçŽĎïijŽ

```
>>> import array
>>> avg(array.array('d',[1,2,3]))
2.0
>>> import numpy
>>> avg(numpy.array([1.0,2.0,3.0]))
2.0
>>> avg([1,2,3])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'list' does not support the buffer interface
>>> avg(b'Hello')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Expected an array of doubles
>>> a = numpy.array([[1.,2.,3.],[4.,5.,6.]])
>>> avg(a[:,2])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: ndarray is not contiguous
>>> sample.avg(a)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: Expected a 1-dimensional array
>>> sample.avg(a[0])

2.0
>>>
```

èőÍèőž

åřĘäÿĂäÿłæŢřçżĎåŕźèśąäijąçżŹCåĞ¡æŢřåŔŕèČ¡æŸŕäÿĂäÿłæL'ľåśŢåĞ¡æŢřåĄŽçŽĎæIJÅåÿÿèğĄçŽĎäž
å¿Ĺåď'ŽPythonåžŤçŤĺçĺŃåžŔïijŇäżŐåŻ¿åČŔåď'ĎçŘĘåĹřçğŚå■ęèőąçőŮïijŇéČ¡æŸŕå§žäžŐéńŸæĂğèČ¡çŽĎ
éĂŽè£ĞçijŰåĘŹèČ¡æŐěåŔŮåźűæŞ■ä¡IJæŢřçżĎçŽĎäžčçăĄïijŇä¡ăåŘŕäžěçijŰåĘŹå¿Ĺåě¡çŽĎåĚijåőźè£ŹäžŽ
èĂŇäÿ■æŸŕåŔłèČ¡åĚijåőźä¡äèĞłåůśçŽĎäžčçăĄąãĂĆ

äžčçăĄçŽĎåĚşéŤőçĆźåIJĺäžŐ `PyBuffer_GetBuffer()` åĞ¡æŢřãĂĆ
çżŹåőŽäÿĂäÿłäżżæĎŔçŽĎPythonåŕźèśąïijŇåőČäijŽèŕŢçİĂåŐżèŐůåŔŮåžŢåśĆåĘĚå■Ÿä£ąæĄŕïijŇåőČçőĂå
1. äijăçżŹ `PyBuffer_GetBuffer()` çŽĎçL'źæőŁæăĞå£ŮçżŹåĞžäžĘæL'ĂéIJĂçŽĎåĘĚå■ŸçijŞåĘšçśžåď
ä¿ŃåęĆïijŇ`PyBUF_ANY_CONTIGUOUS` èąĺçď'žæŸŕäÿĂäÿłèĄŢçşżçŽĎåĘĚå■ŸåŇžå§§ãĂĆ

åŕźäžŐæŢřçżĎãĂĄå■ŮèŁĆå■Ůçňęäÿšå§ŇåĚűäžŰçśżäijijåŕźèśąèĂŇèĺĂïijŇäÿĂäÿł
`Py_buffer` çżŞædĎä¡ŞåŇĚåŘńäžĘæL'ĂæIJL'åžŢåśĆåĘĚå■ŸçŽĎä£ąæĄŕãĂĆ

åőČåŇĚåŘńäÿĂäÿłæŇĞåŘŚåĘĚå■ŸåIJřåİĂãĂĄåď ğářŔãĂĄåĚČçť ăåď ğářŔãĂĄæăijåijŔåŠŇåĚűäžŰçżĘèŁ

```
typedef struct bufferinfo {
    void *buf;              /* Pointer to buffer memory */
    PyObject *obj;          /* Python object that is the owner */
    Py_ssize_t len;         /* Total size in bytes */
    Py_ssize_t itemsize;    /* Size in bytes of a single item */
    int readonly;           /* Read-only access flag */
    int ndim;               /* Number of dimensions */
    char *format;           /* struct code of a single item */
    Py_ssize_t *shape;      /* Array containing dimensions */
    Py_ssize_t *strides;    /* Array containing strides */
    Py_ssize_t *suboffsets; /* Array containing suboffsets */
} Py_buffer;
```

æIJňèŁĆäÿ■ïijŇæĹŚäžňåŔłåĚşæşĺæŐĕåŔŮäÿĂäÿłåŔŇçš¿åžęæţőçĆźæŢřæŢřçżĎä¡IJäÿžåŔĆæŢřãĂĆ
èęĄæčĂæ§ĕåĚČçť ăæŸŕåŘęæŸŕäÿĂäÿłåŔŇçš¿åžęæţőçĆźæŢřïijŇåŔłéIJĂéłŇèŕĄ
`format` åšđæĂğæŸŕäÿ■æŸŕå■ŮçňęäÿšâİdâĂİ. è£Źäÿłäź§æŸŕ `struct` æĺąåİŮçŤĺæİěçijŰçăĄäžŇè£ŻåĹűæŢřæ■őçŽĎãĂĆ éĂŽåÿÿæİěèőšïijŇ`format` åŘŕäžĕæŸŕäżżä¡ŢåĚijåőź `struct` æĺąåİŮçŽĎæäijåijŔåŇŰå■ŮçňęäÿšïijŇ
åźűäÿŤåęĆæđIJæŢřçżĎåŇĚåŘńäžĘĆçżŞæđĎçŽĎèŕİåőĆåŔŕäžĕåŇĚåŘńåď ŽäÿłåĂijãĂĆ
äÿĂæŮęæĹŚäžňåůšçżŔçąőåőŽäžĘåžŢåśĆçŽĎçijŞå■ŸåŇžä£ąæĄŕïijŇéĆčåŔłéIJĂèęĄçőĂå■ŢçŽĎårĘåőČäij
åőđéŹĚäÿŁïijŇæĹŚäžňäÿ■å£ĔæŃĚå£ČæŸŕæĂŐæăůçŽĎæŢřçżĎçśżåđŃæĹŰèĂĚåőČæŸŕèćńäżĂäźĹåžŞåĹŽ
è£Źäź§æŸŕäÿžäżĂäźĹè£Źäÿłåĝ¡æŢřèČ¡åĚijåőź `array` æĺąåİŮäź§èČ¡åĚijåőź `numpy` æĺąåİŮäÿ■çŽĎæŢřçżĎäžĘãĂĆ

åIJĺè£ŤåŻđæIJĂçżĹçżŞæđIJäźŃåL’■ïijŇåžŢåśĆçŽĎçijŞåĘšåŇžèğĘåŻ¿å£Ěéążä¡£çŤĺ
`PyBuffer_Release()` éĞŁæŤ¿æŐL’ãĂĆ äźŃæL’ĂäżěèęĄè£ŹäÿĂæ■ěæŸŕäÿžäžĘèČ¡æ■čçąőçŽĎçőąçŘĚ

åŘŇæăůïijŇæIJňèŁĆäź§äżĔäžĔåŔłæŸŕæijŤçď žäžĘąæŐěåŔŮæŢřçżĎçŽĎäÿĂäÿłåřĹçŽĎäżčçăĄçL’Ğæő
åęĆæđIJä¡ăçIJ§çŽĎèęĄåď ĎçŘĘæŢřçżĎïijŇä¡ăåŔŕèČ¡äijŽçćřåĹřåď Žçżť æŢřæ■őãĂĄåď ğæŢřæ■őãĂĄäÿ■å
éĆčäźĹåřšå¿ŮåŐżå■ęæŽť énŸçžğçŽĎäÿIJèè£äžĘãĂĆä¡åéIJĂèęĄåŔĆèĂČåőŸæŰźæŰĞæąčæİěèŐůåŔŰæŽť

åęĆæđIJä¡ăéIJĂèęĄçijŰåĘŹæűL’åŔŁåĹřæŢřçżĎåď ĎçŘĘçŽĎåď Žäÿłæ L’ľåśŢïijŇéĆčäźĹéĂŽè£ĞCytho

17.4 15.4 åIJĺCæL’ľåśŢæĺąåİŮäÿ■æŞ■ä¡IJéŽŘå¡ćæŇĞéŠĹ

éŮőécŸ

ä¡ăæIJL’äÿĂäÿłæL’ľåśŢæĺąåİŮéIJĂèęĄåď ĎçŘĘCçżŞæđĎä¡Şäÿ■çŽĎæŇĞéŠĹïijŇ
ä¡ĘæŸŕä¡ååŔĹäÿ■æČşæŽť éIJšçżŞæđĎä¡Şäÿ■äżżä¡ŢåĘĚéČĺçżĘèŁĆçżŹPythonãĂĆ

èğčåĘşæŰźæąĹ

éŽŘå¡ćçżŞæđĎä¡ŞåŔŕäžĕå¿ĹåőźæŸ§çŽĎéĂŽè£ĞårĘåőČäžňåŇĚèčĚåIJĺèČűåŻŁårźèśąäÿ■æİěåď ĎçŘĘ
èĂČèŹŚæĹŚäžňä¿Ńå■ŘäžčçăĄäÿ■çŽĎäÿŇåĹŮCäžčçăĄçL’Ğæőţïijž

```
typedef struct Point {
    double x,y;
```

```
} Point;

extern double distance(Point *p1, Point *p2);
```

äÿÑéÌćæŸŕäÿĂäÿłä¡£çŤÍèČűåŻŁåŇĚèčĚPointçżŞæđĎä¡ŞåŠŇ distance() åĞ¡æŢřçŽĎæĽľåśŢäżčçăĄåőđä¿ŃïijŽ

```
/* Destructor function for points */
static void del_Point(PyObject *obj) {
  free(PyCapsule_GetPointer(obj,"Point"));
}

/* Utility functions */
static Point *PyPoint_AsPoint(PyObject *obj) {
  return (Point *) PyCapsule_GetPointer(obj, "Point");
}

static PyObject *PyPoint_FromPoint(Point *p, int must_free) {
  return PyCapsule_New(p, "Point", must_free ? del_Point : NULL);
}

/* Create a new Point object */
static PyObject *py_Point(PyObject *self, PyObject *args) {

  Point *p;
  double x,y;
  if (!PyArg_ParseTuple(args,"dd",&x,&y)) {
    return NULL;
  }
  p = (Point *) malloc(sizeof(Point));
  p->x = x;
  p->y = y;
  return PyPoint_FromPoint(p, 1);
}

static PyObject *py_distance(PyObject *self, PyObject *args) {
  Point *p1, *p2;
  PyObject *py_p1, *py_p2;
  double result;

  if (!PyArg_ParseTuple(args,"OO",&py_p1, &py_p2)) {
    return NULL;
  }
  if (!(p1 = PyPoint_AsPoint(py_p1))) {
    return NULL;
  }
  if (!(p2 = PyPoint_AsPoint(py_p2))) {
    return NULL;
  }
  result = distance(p1,p2);
```

```
  return Py_BuildValue("d", result);
}
```

åIJÍPythonäÿ■åŔŕäżěåČŔäÿŃéİćè£Źæăůæİěä¡£çŤÍè£ŹäžŻåĞ¡æŢřïijŽ

```
>>> import sample
>>> p1 = sample.Point(2,3)
>>> p2 = sample.Point(4,5)
>>> p1
<capsule object "Point" at 0x1004ea330>
>>> p2
<capsule object "Point" at 0x1005d1db0>
>>> sample.distance(p1,p2)
2.8284271247461903
>>>
```

èőÍèőž

èČűåŻŁåŠŇCæŇĞéŠĹçśżäijijãĂĆåIJÍåĘĚéČÍïijŇåőČäżňèŐůåŔŰäÿĂäÿłéĂŽçŤÍæŇĞéŠĹåŠŇäÿĂäÿłåŘ
PyCapsule_New() åĞ¡æŢřå¿ĹåőźæŸŞçŽĎèćńåĹŻåżžãĂĆ
åŔęåď ŰïijŇäÿĂäÿłåŔŕéĂĽçŽĎæđŘæđĎåĞ¡æŢřèČ¡èćńçżŚåőŽåĹřèČűåŻŁäÿŁïijŇçŤÍæİěåIJÍèČűåŻŁåŕźèśąè

èęĄæŔŘåŔŰèČűåŻŁäÿ■çŽĎæŇĞéŠĹïijŇåŔŕä¡£çŤÍ PyCapsule_GetPointer()
åĞ¡æŢřåźűæŇĞåőŽåŘ■çğřãĂĆ åęĆæđIJæŔŘä¿ŻçŽĎåŘ■çğřåŠŇèČűåŻŁäÿ■åŇźéĚ■æĹŰåĚűäżŰéŤŹèŕŕåĞž

æIJňèŁĆäÿ■ïijŇäÿĂåŕźåůěåĚůåĞ¡æŢřâĂŤâĂŤ PyPoint_FromPoint()
åŠŇ PyPoint_AsPoint() èćńçŤÍæİěåĹŻåżžåŠŇäżŐèČűåŻŁåŕźèśąäÿ■æŔŘåŔŰ-
Pointåőđä¿ŃãĂĆ åIJÍäżżä¡ŢæĽľåśŢåĞ¡æŢřäÿ■ïijŇæĹŚäżňäijŽä¡£çŤÍè£ŹäžŻåĞ¡æŢřèĂŇäÿ■æŸŕçŻťæŐěä¡
è£Źçğ■èő¿èőąä¡£å¿ŮæĹŚäżňåŔŕäżěå¿ĹåőźæŸŞçŽĎåžŤåŕźåřĘæİěåŕźPointåžŢäÿŃçŽĎåŇĚèčĚçŽĎæŻťæŤź
ä¿ŃåęĆïijŇåęĆæđIJä¡ăåĘşåőŽä¡£çŤÍåŔęåď ŰäÿĂäÿłèČűåŻŁäžĘïijŇéĆčäźĹåŔłéIJĂèęĄæŻťæŤźè£Źäÿď äÿłå

åŕźäžŐèČűåŻŁåŕźèśąäÿĂäÿłéŽ¿çĆźåIJÍäžŐåđČåIJ¿åŻđæŤűåŠŇåĘĚå■ŸçőąçŘĘãĂĆ
PyPoint_FromPoint() åĞ¡æŢřæŐěåŔŮäÿĂäÿł must_free
åŔĆæŢřïijŇ çŤÍæİěæŇĞåőŽå¡ŞèČűåŻŁèćńéŤĂæŕĄæŮűåžŢåśĆPoint *
çżŞæđĎä¡ŞæŸŕåŘęåžŤèŕěèćńåŻđæŤűãĂĆ åIJÍæ§ŘäžŻCäżčçăĄäÿ■ïijŇå¡ŠåśđéŮőéćŸéĂŽåÿÿå¿ĹéŽ¿èćńåď Ì
çÍŃåžŔåŚŸåŔŕäżěä¡£çŤÍ extra åŔĆæŢřæİěæŐğåĹűïijŇèĂŇäÿ■æŸŕå■ŢæŰźéİćçŽĎåĘşåőŽåđČåIJ¿åŻđæŤ
èęĄæşÍæĎŔçŽĎæŸŕåŠŇçŐřæIJĽèČűåŻŁæIJĽåĚşçŽĎæđŘæđĎåŹÍèČ¡ä¡£çŤÍ
PyCapsule_SetDestructor() åĞ¡æŢřæİěæŻťæŤźãĂĆ

åŕźäžŐæűĽåŔŁåĹřçżŞæđĎä¡ŞçŽĎCäżčçăĄèĂŇèÍĂïijŇä¡£çŤÍèČűåŻŁæŸŕäÿĂäÿłæŕŤè¿ČåŘĹçŘĘçŽĎè
ä¿ŃåęĆïijŇæIJĽæŮűåĂŹä¡ăåźűäÿ■åĚşå£ČæŽťéIJšçżŞæđĎä¡ŞçŽĎåĘĚéČÍä£ąæĄŕæĹŰèĂĚåřĘåĚűè¡ňæ■ćæ
éĂŽè£Ğä¡£çŤÍèČűåŻŁïijŇä¡ăåŔŕäżěåIJÍåőČäÿŁéİćæŤ¿äÿĂäÿłè¡żéĞŔçžğçŽĎåŇĚèčĚåŹÍïijŇçĎűåŘŐåřĘåőČ

17.5 15.5 äžŐæĽľåśŢæÍąåİŮäÿ■åőŽäźĽåŠŇåŕijåĞžCçŽĎAPI

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłCæL'ľåśŢæÍąåİŮïijŇåIJÍåĘĚéČÍåőŽäźL'äžĘå¿Ĺåď'ŽæIJL'çŤÍçŽĎåĞ¡æŢřïijŇä¡ăæČşåřĘå APIä¿ŻåĚűäžŰåIJřæŰźä¡£çŤÍãĂĆ ä¡ăæČşåIJÍåĚűäžŰæL'ľåśŢæÍąåİŮäÿ■ä¡£çŤÍè£ŹäžŻåĞ¡æŢřïijŇä¡ĘæŸŕäÿ åźűäÿŤéĂŽè£ĞCçijŰèŕŚåŹÍ/éŞ¿æŐěåŹÍæİěåĄŽçIJŃäÿŁåŐżçL'źåĹńåď'■æİĆïijĹæĹŰèĂĚäÿ■åŔŕèČ¡åĄŽåĹř

èğčåĘşæŰźæąĹ

æIJňèŁĆäÿżèęĄéŮőécŸæŸŕåęĆä¡Ţåď'ĎçŘĘ15.4åřŔèŁĆäÿ■æŔŘåĹřçŽĎPointåŕźèśąãĂĆäżŤçżĘåŻđäÿ

```
/* Destructor function for points */
static void del_Point(PyObject *obj) {

  free(PyCapsule_GetPointer(obj,"Point"));
}

/* Utility functions */
static Point *PyPoint_AsPoint(PyObject *obj) {
  return (Point *) PyCapsule_GetPointer(obj, "Point");
}

static PyObject *PyPoint_FromPoint(Point *p, int must_free) {
  return PyCapsule_New(p, "Point", must_free ? del_Point : NULL);
}
```

çŐřåIJÍçŽĎéŮőécŸæŸŕæĂŐæăůåřĘ `PyPoint_AsPoint()` åŠŇ `Point_FromPoint()` åĞ¡æŢřä¡IJäÿžAPIåŕijåĞžïijŇ è£ŹæăůåĚűäžŰæL'ľåśŢæÍąåİŮèČ¡ä¡£çŤÍåźűéŞ¿æŐěåőČäżñïijŇærŤåęĆåęĆædIJä¡ăæIJL'åĚűäžŰæL'ľåśŢäź§

èęĄèğčåĘşè£ŹäÿłéŮőécŸïijŇéęŰåĚÍèęĄäÿž `sample` æL'ľåśŢåĘŹäÿłæŰřçŽĎåď't'æŰĞäžűåŘ■åŔń `pysample.h` ïijŇåęĆäÿŃïijŽ

```
/* pysample.h */
#include "Python.h"
#include "sample.h"
#ifdef __cplusplus
extern "C" {
#endif

/* Public API Table */
typedef struct {
  Point *(*aspoint)(PyObject *);
  PyObject *(*frompoint)(Point *, int);
} _PointAPIMethods;

#ifndef PYSAMPLE_MODULE
/* Method table in external module */
```

```
static _PointAPIMethods *_point_api = 0;

/* Import the API table from sample */
static int import_sample(void) {
  _point_api = (_PointAPIMethods *) PyCapsule_Import("sample._point_
↪api",0);
  return (_point_api != NULL) ? 1 : 0;
}

/* Macros to implement the programming interface */
#define PyPoint_AsPoint(obj) (_point_api->aspoint)(obj)
#define PyPoint_FromPoint(obj) (_point_api->frompoint)(obj)
#endif

#ifdef __cplusplus
}
#endif
```

è£ŹéĞŇæIJĂéĞ■èęĄçŽĎéČĺåĹĘæŸŕåĞ¡æŢřæŇĞéŠĹèąĺ `_PointAPIMethods` . åőČäijŽåIJĺåŕijåĞžæĺąåİŮæŮűčéćńåĹĺåğŃåŇŰïijŇçĎűåŘŐåŕijåĚěæĺąåİŮæŮűčéćńæ§ěæL'¿åĹřãĂĆä£őæŤźåŐ§åğŃçŽĎæL'ľåśŢæĺąåİŮæİěåąńåĚĚèąĺæäijåźűåřĘåőČåČŔäÿŃéİćè£ŹæăůåŕijåĞžïijŽ

```
/* pysample.c */

#include "Python.h"
#define PYSAMPLE_MODULE
#include "pysample.h"

...
/* Destructor function for points */
static void del_Point(PyObject *obj) {
  printf("Deleting point\n");
  free(PyCapsule_GetPointer(obj,"Point"));
}

/* Utility functions */
static Point *PyPoint_AsPoint(PyObject *obj) {
  return (Point *) PyCapsule_GetPointer(obj, "Point");
}

static PyObject *PyPoint_FromPoint(Point *p, int free) {
  return PyCapsule_New(p, "Point", free ? del_Point : NULL);
}

static _PointAPIMethods _point_api = {
  PyPoint_AsPoint,
  PyPoint_FromPoint
};
...
```

```
/* Module initialization function */
PyMODINIT_FUNC
PyInit_sample(void) {
  PyObject *m;
  PyObject *py_point_api;

  m = PyModule_Create(&samplemodule);
  if (m == NULL)
    return NULL;

  /* Add the Point C API functions */
  py_point_api = PyCapsule_New((void *) &_point_api, "sample._point_
↪api", NULL);
  if (py_point_api) {
    PyModule_AddObject(m, "_point_api", py_point_api);
  }
  return m;
}
```

æIJĂåŘŐïijŇäÿŃéİćæŸŕäÿĂäÿłæŰřçŽĎæĽľ'åśŢæĺąåİŮä¿Ńå■ŘïijŇçŤĺæİěåŁăè¡¡åźűä¡£çŤĺè£ŹäžŻAPIå

```
/* ptexample.c */

/* Include the header associated with the other module */
#include "pysample.h"

/* An extension function that uses the exported API */
static PyObject *print_point(PyObject *self, PyObject *args) {
  PyObject *obj;
  Point *p;
  if (!PyArg_ParseTuple(args,"O", &obj)) {
    return NULL;
  }

  /* Note: This is defined in a different module */
  p = PyPoint_AsPoint(obj);
  if (!p) {
    return NULL;
  }
  printf("%f %f\n", p->x, p->y);
  return Py_BuildValue("");
}

static PyMethodDef PtExampleMethods[] = {
  {"print_point", print_point, METH_VARARGS, "output a point"},
  { NULL, NULL, 0, NULL}
};

static struct PyModuleDef ptexamplemodule = {
  PyModuleDef_HEAD_INIT,
```

```
  "ptexample",           /* name of module */
  "A module that imports an API",  /* Doc string (may be NULL) */
  -1,                 /* Size of per-interpreter state or -1 */
  PtExampleMethods       /* Method table */
};

/* Module initialization function */
PyMODINIT_FUNC
PyInit_ptexample(void) {
  PyObject *m;

  m = PyModule_Create(&ptexamplemodule);
  if (m == NULL)
    return NULL;

  /* Import sample, loading its API functions */
  if (!import_sample()) {
    return NULL;
  }

  return m;
}
```

çijŰèŕŚè£ŹäÿłæŰřæÍąåİŮæŮűïijŇä¡ăçŤŽèĞşäÿ■éIJĂèęĄåŐżèĂČèŹŚæĂŐæăůåřĘåĞ¡æŢřåžŞæĹŰäżčçà
ä¿ŇåęĆïijŇä¡ăåŔŕäżěåČŔäÿŇéİćè£ŹæăůåĹŻåžžäÿĂäÿłçőĂå■ŢçŽĎ `setup.py` æŰĞäżűïijŽ

```
# setup.py
from distutils.core import setup, Extension

setup(name='ptexample',
      ext_modules=[
        Extension('ptexample',
                  ['ptexample.c'],
                  include_dirs = [],   # May need pysample.h␣
↪directory
                  )
        ]
)
```

åęĆædIJäÿĂåĹĞæ■čåÿÿïijŇä¡ääijŽåŔŚçŐřä¡ăçŽĎæŰřæL'l'åśŢåĞ¡æŢřèČ¡åŠŇåőŽäźL'åIJÍåĚűäżŰæÍąåİ
APIåĞ¡æŢřäÿĂèţůè£ŘèąŇçŽĎå¿Ĺåě¡ãĂĆ

```
>>> import sample
>>> p1 = sample.Point(2,3)
>>> p1
<capsule object "Point *" at 0x1004ea330>
>>> import ptexample
>>> ptexample.print_point(p1)
2.000000 3.000000
>>>
```

èőÍèőž

æIJňèŁĆå§žäžŐäÿĂäÿłåL’■æŔŘåřśæŸŕřijŇèČűåŻŁåŕźèśąèČ¡èŐůåŔŰäżżä¡Ţä¡ăæČşèęĄçŽĎåŕźèśąçŽĎ
è£Źæăůç ŽĎèŕÏřijŇåőŽäźĽ’æÍąåİŮäijŽåąńåĚĚäÿĂäÿłåĞ¡æŢřæŇĞéŠĹçŽĎçżŞæđĎä¡ŞřijŇåĹŻåżžäÿĂäÿłæŇČ
ä¿ŃåęĆ `sample._point_api`.

åĚűäžŰæÍąåİŮèČ¡åď§åIJÍåŕijåĚěæŮűèŐůåŔŰåĹřè£ŹäÿłåśđæĂğåźűæŔŘåŔŰåžŢåśĆçŽĎæŇĞéŠĹãĂĆ
äžŃåőđäÿŁïijŇPythonæŔŘä¿ŻäžĘ `PyCapsule_Import()`
åůěåĚůåĞ¡æŢřijŇäÿžäžĘåőŇæĹŘæĽ’ĂæIJL’çŽĎæ■ěéłď ãĂĆ
ä¡ääŔłéIJĂæŔŘä¿Żåśđæ ĂğçŽĎåŘ■å■Ůå■şåŔŕijĹærŤåęĆsample._point_apiïijL’ïijŇçĎűåŘŐäžŰåŕśäijŽäÿĂ

åIJÍåŕĘèćńåŕijåĞžåĞ¡æŢřåŔŸäÿžåĚűäžŰæÍąåİŮäÿ■æŹőéĂŽåĞ¡æŢřæŮűïijŇæIJL’äÿĂäžŻCçijŰçÍŃéŹů
åIJÍ `pysample.h` æŰĞäžűäÿ■ïijŇäÿĂäÿł `_point_api`
æŇĞéŠĹèćńçŤÍæİěæŇĞåŘŚåIJÍåŕijåĞžæÍąåİŮäÿ■èćńåĹİåğŃåŇŰçŽĎæŰźæşŢèąÍãĂĆ
äÿĂäÿłçŻÿåĚşçŽĎåĞ¡æŢř `import_sample()` èćńçŤÍæİěæŇĞåŘŚèČűåŽŁåŕijåĚěźűåĹİåğŃåŇŰè£Źäÿłæ
è£ŹäÿłåĞ¡æŢřå£Ěéążå IJÍäżżä¡ŢåĞ¡æŢřèćńä¡£çŤÍäźŃåĽ■èćńèŕČçŤÍãĂĆéĂŽåÿÿæİèèőšïijŇåőČäijŽåIJÍæÍąåİ
æIJĂåŘŐïijŇCçŽĎéćĎåď ĎçŘĘåőŔèćńåőŽäźĽ’ïijŇèćńçŤÍæİěéĂŽè£ĞæŰźæşŢèąÍåŐżåĹĘåŔŚè£ŹäžŻAPIåĞ
çŤÍæĹůåŘłéIJĂèęĄä¡£çŤÍè£ŹäžŻåŐ§ågŃåĞ¡æŢřåŘ■çğřå■şåŔŕijŇäÿ■éIJĂèęĄéĂŽè£ĞåőŔåŐżäžĘèğčåĚűä

æIJĂåŘŐïijŇè£ŸæIJL’äÿĂäÿłéĞ■èęĄçŽĎåŐ§åŻäèőľä¡ăåŐżä¡£çŤÍè£ŹäÿłæŁĂæIJŕæİěéŞ¿æŐěæÍąåİŮâ
åęĆæđIJä¡ăäÿ■æČşä¡£çŤÍæIJňæIJžçŽĎæŁĂæIJŕijŇéĆčä¡åářšå£Ěéążä¡£çŤÍåĚśäžńåžŞçŽĎéńŸçžğçL’źæĂğå
ä¿ŃåęĆïijŇářĘäÿĂäÿłæŹőéĂŽçŽĎAPIåĞ¡æŢřæŤ¿åĚěäÿĂäÿłåĚśäžńåžŞåźűçąőä£ÍæL’ĂæIJL’æL’ľåśŢæÍąåİŮ
è£ŹçğæŰźæşŢçąőåőđåŘŕèąŇïijŇä¡ĘæŸŕåőČçŽÿářźçźĄçŘŘïijŇçL’źåĹńæŸŕåIJÍåď ğåđŃçşžçž§äÿ■ãĂĆ
æIJňèŁĆæijŤçď žäžĘåęĆä¡ŢéĂŽè£ĞPythonçŽĎæŹőéĂŽåŕijåĚěæIJžåĹűåŠŇäžĚäžĚåĞääÿłèČűåŻŁèřČçŤÍæİ
ářźäžŐæÍąåİŮçŽĎçijŰèŕŚïijŇä¡ăåŘłéIJĂèęĄåőŽäźĽ’åď ť’æŰĞäžűïijŇèĂŇäÿ■éIJĂèęĄèĂČèŹŚåĞ¡æŢřåžŞçŽ

æŻť’åď ŽåĚşäžŐåĹl’çŤÍC APIæİěædĎéĂäæL’l’åśŢæÍąåİŮçŽĎä£qæĄŕåŘŕäžěåŘĆèĂČ
PythonçŽĎæŰĞæąč

17.6 15.6 äžŐCèŕ■èÍĂäÿ■èřČçŤÍPythonäžčçăĄ

éŮőécŸ

ä¡ăæČşåIJÍCäÿ■åőL’åĚÍçŽĎæL’ğèąŇæ§ŘäÿłPythonèřČçŤÍåźűè£ŤåŽđçżŞæđIJçżŹCãĂĆ
ä¿ŃåęĆïijŇä¡ăæČşåIJÍCèŕ■èÍĂäÿ■ä¡£çŤÍæ§ŘäÿłPythonåĞ¡æŢřä¡IJäÿžäÿĂäÿłåŽđèřČãĂĆ

èğčåĘşæŰźæąĹ

åIJÍCèŕ■èÍĂäÿ■èřČçŤÍPythonéİđåÿÿçőĂå■ŢïijŇäÿ■è£Ğèő¿èőąåĹřäÿĂäžŽåřŔçł■éŮÍãĂĆ
äÿŃéİćçŽĎCäžčçăĄåŚŁèŕL’ä¡ăæĂŐæăůåőL’åĚÍçŽĎèřČçŤÍïijŽ

```
#include <Python.h>

/* Execute func(x,y) in the Python interpreter.  The
   arguments and return result of the function must
   be Python floats */

double call_func(PyObject *func, double x, double y) {
```

```
  PyObject *args;
  PyObject *kwargs;
  PyObject *result = 0;
  double retval;

  /* Make sure we own the GIL */
  PyGILState_STATE state = PyGILState_Ensure();

  /* Verify that func is a proper callable */
  if (!PyCallable_Check(func)) {
    fprintf(stderr,"call_func: expected a callable\n");
    goto fail;
  }
  /* Build arguments */
  args = Py_BuildValue("(dd)", x, y);
  kwargs = NULL;

  /* Call the function */
  result = PyObject_Call(func, args, kwargs);
  Py_DECREF(args);
  Py_XDECREF(kwargs);

  /* Check for Python exceptions (if any) */
  if (PyErr_Occurred()) {
    PyErr_Print();
    goto fail;
  }

  /* Verify the result is a float object */
  if (!PyFloat_Check(result)) {
    fprintf(stderr,"call_func: callable didn't return a float\n");
    goto fail;
  }

  /* Create the return value */
  retval = PyFloat_AsDouble(result);
  Py_DECREF(result);

  /* Restore previous GIL state and return */
  PyGILState_Release(state);
  return retval;

fail:
  Py_XDECREF(result);
  PyGILState_Release(state);
  abort();   // Change to something more appropriate
}
```

èęĄä¡£çŤĺè£ŹäÿłåĞ¡æŢřïijŇä¡ăéIJĂèęĄèŐůåŔŰäijăéĂŠè£ĞæİěçŽĎæ§Řäÿłåůšå■ŸåIJÍPythonèřČçŤĺçŽ
æIJL'å¿Ĺåď'Žçğ■æŰźæşŢåŔŕäżěèől'ä¡ăè£ŹæăůåĄŽïijŇ æŕŤåęĆåřĘäÿĂäÿłåŔŕèřČçŤĺåŕźèśąäijăçżŹäÿĂäÿłæĽ

äÿŃéÍćæŸŕäÿĂäÿłçŏĂå■Ţä¿Ńå■ŘçŤÍæÍěæŐľéěŕäżŐäÿĂäÿłåţŇåĚěçŽĎPythonèğčéĞŁåŹÍäÿ■èřČçŤÍäÿ

```
#include <Python.h>

/* Definition of call_func() same as above */
...

/* Load a symbol from a module */
PyObject *import_name(const char *modname, const char *symbol) {
  PyObject *u_name, *module;
  u_name = PyUnicode_FromString(modname);
  module = PyImport_Import(u_name);
  Py_DECREF(u_name);
  return PyObject_GetAttrString(module, symbol);
}

/* Simple embedding example */
int main() {
  PyObject *pow_func;
  double x;

  Py_Initialize();
  /* Get a reference to the math.pow function */
  pow_func = import_name("math","pow");

  /* Call it using our call_func() code */
  for (x = 0.0; x < 10.0; x += 0.1) {
    printf("%0.2f %0.2f\n", x, call_func(pow_func,x,2.0));
  }
  /* Done */
  Py_DECREF(pow_func);
  Py_Finalize();
  return 0;
}
```

èęĄæđĎåżžä¿Ńå■ŘäżčçăĄïijŇä¡ăéIJĂèęĄçijŰèŕŚCåźűåřĘåőČéŞ¿æŐěåĹřPythonèğčéĞŁåŹÍãĂĆ
äÿŃéÍćçŽĎMakefileåŔŕäžěæŢŹä¡ăæĂŐæăůåĄŽïijĹäÿ■è£ĞåIJÍä¡ăæIJžåŹÍäÿŁéÍćéIJĂèęĄäÿĂäžŽéĚ■ç¡őïijĽ

```
all::
        cc -g embed.c -I/usr/local/include/python3.3m \
          -L/usr/local/lib/python3.3/config-3.3m -lpython3.3m
```

çijŰèŕŚåźűè£ŘèąŇäijŽäžğçŤ§çśżäijijäÿŃéÍćçŽĎè¿ŞåĞžïijŽ

```
0.00 0.00
0.10 0.01
0.20 0.04
0.30 0.09
0.40 0.16
...
```

äÿŃéİćæŸŕäÿĂäÿłçĺ■å¿őäÿ■åŘŇçŽĎä¿Ńå■ŘïijŇåśŢçď žäžĘäÿĂäÿłæĽľåśŢåĞ¡æŢřïijŇ åőČæŐěåŔŮäÿĂäÿłåŔŕèřČçŤĺåŕźèśąåŠŇåĚűäżŰåŔĆæŢřïijŇåźűåřĘåőČäżňäijäéĂŠçżŹ `call_func()` æİěåĄŽæţŃèŕŢïijŽ

```
/* Extension function for testing the C-Python callback */
PyObject *py_call_func(PyObject *self, PyObject *args) {
  PyObject *func;

  double x, y, result;
  if (!PyArg_ParseTuple(args,"Odd", &func,&x,&y)) {
    return NULL;
  }
  result = call_func(func, x, y);
  return Py_BuildValue("d", result);
}
```

ä¡£çŤĺè£ŹäÿłæĽľåśŢåĞ¡æŢřïijŇä¡ăèęĄåČŔäÿŃéİćè£ŹæăůæţŃèŕŢåőČïijŽ

```
>>> import sample
>>> def add(x,y):
...     return x+y
...
>>> sample.call_func(add,3,4)
7.0
>>>
```

èőĺèőž

åęĆæđIJä¡ăåIJĺCèŕ■èĺĂäÿ■èřČçŤĺPythonïijŇèęĄèőřä¡ŔæIJĂéĞ■èęĄçŽĎæŸŕCèŕ■èĺĂäijŽæŸŕäÿżä¡ŞãĂ äź§åřśæŸŕèŕťïijŇCèŕ■èĺĂèť§èť čæđĎéĂăåŔĆæŢřãĂĄèřČçŤĺPythonåĞ¡æŢřãĂĄæčĂæ§ěåijĆåÿÿãĂĄæčĂæ

ä¡IJäÿžçňňäÿĂæ■ěïijŇä¡ăå£ĚéążåĚĹæIJĽäÿĂäÿłèąĺçď žä¡ăåřĘèęĄèřČçŤĺçŽĎPythonåŔŕèřČçŤĺåŕźèśąãĂ è£ŹåŔŕäżěæŸŕäÿĂäÿłåĞ¡æŢřãĂĄçśżãĂĄæŰźæşŢãĂĄåĘĚç¡őæŰźæşŢæĹŰåĚűäżŰäżżæĎŔåőđçŐřäžĘ `__call__()` æŞ■ä¡IJçŽĎäÿIJèě£ãĂĆ äÿžäžĘçąőä£İæŸŕåŔŕèřČçŤĺçŽĎïijŇåŔŕäżěåČŔäÿŃéİćçŽĎäżčçăĄè `PyCallable_Check()` åĄŽæčĂæ§ěïijŽ

```
double call_func(PyObject *func, double x, double y) {
  ...
  /* Verify that func is a proper callable */
  if (!PyCallable_Check(func)) {
    fprintf(stderr,"call_func: expected a callable\n");
    goto fail;
  }
  ...
```

åIJĺCäżčçăĄéĞŇåď ĎçŘĘéŤŹèŕŕä¡ăéIJĂèęĄæăijåď ŰçŽĎåřŔå£ČãĂĆäÿĂèĹňæİěèőšïijŇä¡ăäÿ■èČ¡äżĚä éŤŹèŕŕåžŤèŕěä¡£çŤĺCäżčçăĄæŰźåijŔæİěèćńåď ĎçŘĘãĂĆåIJĺè£ŹéĞŇïijŇæĹŚäżňæĽŞçőŮåřĘåŕźéŤŹèŕŕçŽĎ `abort()` çŽĎéŤŹèŕŕåď ĎçŘĘåŹĺãĂĆ åőČäijŽçżŞæİ§æŐĽæŢťäÿłçĺŃåžŔïijŇåIJĺçIJ§åőđçŐŕåćČäÿŃéİćä¡ăå ä¡ăèęĄèőřä¡ŔçŽĎæŸŕåIJĺè£ŹéĞŇCæŸŕäÿżèğŠïijŇåŻăæ■ď åźűæšąæIJĽèů§æŁŻåĞžåijĆåÿÿçŻÿåŕźåžŤçŽĎæ éŤŹèŕŕåď ĎçŘĘæŸŕä¡ăåIJĺçijŰçĺŃæŮűå£ĚéążèęĄèĂČèŹŚçŽĎäžŃæČĚãĂĆ

èřČçŤÍäÿĂäÿłåĞ¡æŢřçŻÿåŕźæİěèőšå¿ĹçőĂå■ŢâĂŤâĂŤåŔŧéIJĂèęĄä¡£çŤÍ
`PyObject_Call()` ïijŇ äijăäÿĂäÿłåŔŕèřČçŤÍåŕźèśąçżŹåőČãĂĄäÿĂäÿłåŔĆæŢřåĚČçżĎåŠŇäÿĂäÿłåŔŕéĂ
èęĄæđĎåżžåŔĆæŢřåĚČçżĎæĹŰå■ŮåĚÿïijŇä¡ååŔŕäžěä¡£çŤÍ `Py_BuildValue()`
,åęĆäÿŇïijŽ

```
double call_func(PyObject *func, double x, double y) {
  PyObject *args;
  PyObject *kwargs;

  ...
  /* Build arguments */
  args = Py_BuildValue("(dd)", x, y);
  kwargs = NULL;

  /* Call the function */
  result = PyObject_Call(func, args, kwargs);
  Py_DECREF(args);
  Py_XDECREF(kwargs);
  ...
```

åęĆædIJæšąæIJL'åĚşéŤőå■ŮåŔĆæŢřïijŇä¡ääŔŕäžěäijăéĂŠNULLãĂĆå¡Şä¡ăèęĄèřČçŤÍåĞ¡æŢřæŮűïijŇ
éIJĂèęĄçąőä£İä¡£çŤÍäžĘ `Py_DECREF()` æĹŰèĂĚ `Py_XDECREF()` æÿĚçŘĘåŔĆæŢřãĂĆ
çňňäžŇäÿłåĞ¡æŢřçŻÿåŕźåőL'åĚÍçĆźïijŇåŻääÿžåőČåĚĄèőÿäijăéĂŠNULLæŇĞéŠĹïijÍçŻť'æŐěå£¡çŢěåőČïijÍ
è£Źäź§æŸŕäÿžäžĂäźĹæĹŚäżňä¡£çŤÍåőČæİěæÿĚçŘĘåŔŕéĂL'çŽĎåĚşéŤőå■ŮåŔĆæŢřãĂĆ

èřČçŤÍäÿĞPythonåĞ¡æŢřäźŃåŘŐïijŇä¡ăå£Ěéążæč Ăæ§ěæŸŕåŘęæIJL'åijĆåÿÿåŔŚçŤ§ãĂĆ
`PyErr_Occurred()` åĞ¡æŢřåŔŕèćńçŤÍæİěåĄŽè£Źäžűäž ŃãĂĆ
åŕźåŕźäžŐåijĆåÿÿçŽĎåď ĎçŘĘåŕśæIJL'çĆźžżçČęäžEïijŇçŤśäžŐæŸŕçŤÍCèŕ■èÍĂåĘŹçŽĎïijŇä¡ăæšąæIJL'åČ
åŻăæ■ď'ïijŇä¡ăå£ĚéążèęĄèő¿ç¡őäÿĂäÿłåijĆåÿÿçŁűæĂĄçĂąïijŇæL'Şå■řåijĆåÿÿä£ąæĄŕæĹŰåĚűäžŰçŻÿåžŤ
åIJÍè£ŹéĞŇïijŇæĹŚäżňéĂL'æŃľäžĘçőĂå■ŢçŽĎ `abort()`
æİěåď ĎçŘĘãĂĆåŘęåď ŰïijŇäijăçż§CçÍŇåžŘåŚŸåŔŕèČ¡äijŽçŻť'æŐěěľ'çÍŇåžŘåěŤæžČãĂĆ

```
...
/* Check for Python exceptions (if any) */
if (PyErr_Occurred()) {
  PyErr_Print();
  goto fail;
}
...
fail:
  PyGILState_Release(state);
  abort();
```

äžŐèřČçŤÍPythonåĞ¡æŢřçŽĎè£ŤåŻđåĂijäÿ■æŔŘåŘŰä£ąæĄŕéĂŽåÿÿèęĄè£ŻèąŇçśżåđŃæčĂæ§ěåŠŇæ
èęĄè£ŹæăůåĄŽçŽĎèŕÍïijŇä¡ăå£Ěéążä¡£çŤÍPythonåŕźèśąåśĆäÿ■çŽĎåĞ¡æŢřãĂĆ
åIJÍè£ŹéĞŇæĹŚäżňä¡£çŤÍäžĘ `PyFloat_Check()` åŠŇ `PyFloat_AsDouble()`
æİěæčĂæ§ěåŠŇæŘŘåŘŰPythonæţőçĆźæŢřãĂĆ

æIJĂåŘŐäÿĂäÿłéŮőécŸæŸŕåŕźäžŐPythonåĚĺåśĂéŤĄçŽĎçőąçŘĘãĂĆ
åIJÍCèŕ■èÍĂäÿ■ëő£éŮőPythonçŽĎæŮűåĂŹïijŇä¡äéIJĂèęĄçąőä£İGILèćńæ■čçąőçŽĎèŐůåŘŰåŠŇéĞŁæŤ¿äž
äÿ■çĎűçŽĎèŕÍïijŇåŘŕèČ¡äijŽåŕijèĞť'èğčéĞŁåŹÍè£ŤåŻđéŤŹèŕŕæŢřæ■őæĹŰèĂĚçŻť'æŐěåěŤæžČãĂĆ

èřČçŤÍ PyGILState_Ensure() åŠŇ PyGILState_Release() åŔŕäžěçąőä£ÍäÿĂåĹĞéČ¡èČ¡æ■čåÿÿãĂĆ

```
double call_func(PyObject *func, double x, double y) {
  ...
  double retval;

  /* Make sure we own the GIL */
  PyGILState_STATE state = PyGILState_Ensure();
  ...
  /* Code that uses Python C API functions */
  ...
  /* Restore previous GIL state and return */
  PyGILState_Release(state);
  return retval;

fail:
  PyGILState_Release(state);
  abort();
}
```

äÿĂæŮęè£ŤåŻđïijŇPyGILState_Ensure() åŔŕäžěçąőä£ÍèřČçŤÍçž£çÍŃçŃňå■ăPythonèğčéĞŁåŹÍãĄ
åřśçőŮCäżčçăAè£ŘèąŇäžŐåŔęåď ŰäÿĂäÿłèğčéĞŁåŹÍäÿ■ç§ěéĄŞçŽĎçž£çÍŃäź§æšąäžŃãĂĆ
è£ŹæŮűåĂŹïijŇCäżčçăAąåŔŕäžěèĞłçŤśçŽĎä¡£çŤÍäżżä¡ŢåőČæČşèęĄçŽĎPython C-API
åĞ¡æŢřãĂĆ èřČçŤÍæĹŘåŁ§åŘŐïijŇPyGILState_Release()èćńçŤÍæÍěèőšèğčéĞŁåŹÍæĄćåď■åĹřåŐ§ågŃçŁűa

èęĄæşÍæĎŔçŽĎæŸŕæŕŔäÿĂäÿł PyGILState_Ensure()
èřČçŤÍå£ĚéążèůçÍĂäÿĂäÿłåŇźéĚ■çŽĎ PyGILState_Release()
èřČçŤÍâĂŤâĂŤå■şä¿£æIJL'éŤŹèŕŕåŔŚçŤ§ãĂĆ åIJÍè£ŹéĞŇïijŇæĹŚäżňä¡£çŤÍäÿĂäÿł goto
èŕ■åŔęçIJŃäÿŁåŐżæŸŕäÿłåŔŕæĂŢçŽĎèő¿èőąïijŇ ä¡ĘæŸŕåőđéŹĚäÿŁæĹŚäżňä¡£çŤÍåőČæÍěèőšæŐğåĹűæÍČè
åIJÍ fail: æăĞç■¿åŘŐéÍćçŽĎäżčçăĄåŠŇPythonçŽĎ fianl:
åİŮçŽĎçŤÍéĂŤæŸŕäÿĂæăůçŽĎãĂĆ

åęĆæđIJä¡ăä¡£çŤÍæL'ĂæIJL'è£ŹäžŻçžęåőŽæÍěçijŰåĘŹCäżčçăĄïijŇåŇĚæŃňåŕźGILçŽĎçőąçŘĘãĂĄåij
ä¡ăäijŽåŔŚçŐŕäžŐCèŕ■èÍĂäÿ■èřČçŤÍPythonèğčéĞŁåŹÍæŸŕåŔŕéÍăçŽĎâĂŤâĂŤåŕšçőŮåĘ■åď■æÍĆçŽĎçÍŃåž

17.7 15.7 äżŐCæĽľåśŢäÿ■éĞŁæŤ¿åĚÍåśĂéŤĄ

éŮőécŸ

ä¡ăæČşèőľ'CæL'ľåśŢäżčçăĄåŠŇPythonèğčéĞŁåŹÍäÿ■çŽĎåĚűäżŰè£ŻçÍŃäÿĂètůæ■čçąőçŽĎæL'ğèąŇïij
éĆčäźĹä¡ăåřśéIJĂèęĄåŐżéĞŁæŤ¿åźűéĞ■æŰřèŐůåŘŮåĚÍåśĂèğčéĞŁåŹÍéŤĄïijĹGILïijL'ãĂĆ

èğčåĘşæŰźæąĹ

åIJÍCæL'ľåśŢäżčçăĄäÿ■ïijŇGILåŔŕäžěéĂŽè£ĞåIJÍäżčçăĄäÿ■æŔŠåĚěäÿŃéÍćè£ŹæăůçŽĎåőŔæÍěéĞŁæ

```
#include "Python.h"
...

PyObject *pyfunc(PyObject *self, PyObject *args) {
   ...
   Py_BEGIN_ALLOW_THREADS
   // Threaded C code.  Must not use Python API functions
   ...
   Py_END_ALLOW_THREADS
   ...
   return result;
}
```

èőÍèőž

åŘłæIJL'å¡Şä¡ăçąőä£İæšąæIJL'Python C APIåĞ¡æŢřåIJÍCäÿ■æL'ğèąŇçŽĎæŮűåĂŹä¡ăæL'■èČ¡åőL'åĚÍçŽ
GILéIJĂèęĄèćńéĞŁæŤ¿çŽĎåÿÿèğĄçŽĎåIJžæŹŕæŸŕåIJÍèőąçőŮåŕĘéŽĘåđŇäžčçăĄäÿ■éIJĂèęĄåIJÍCæŢřçżĎ
æĹŰèĂĚæŸŕèęĄæL'ğèąŇéŸżåąđçŽĎI/OæŞ■ä¡IJæŮűïijĹæŕŤåęĆåIJÍäÿĂäÿłæŰĞäžűæŘŔè£řçňęäÿŁèŕżåŘŰa

å¡ŞGILèćńéĞŁæŤ¿åŘŐïijŇåĚűäžŰPythonçž£çÍŃæL'■èćńåĚĄèőÿåIJÍèğčéĞŁåŹÍäÿ■æL'ğèąŇãĂĆ
`Py_END_ALLOW_THREADS` åőŔäijŽéŸżåąđæL'ğèąŇçŻť'åĹřèřČçŤÍçž£çÍŃéĞ■æŰřèŐůåŘŰäžĘGILãĂĆ

17.8 15.8 CåŠŇPythonäÿ■çŽĎçž£çÍŃæůúçŤÍ

éŮőécŸ

ä¡ăæIJL'äÿĂäÿłçÍŃåžŔéIJĂèęĄæůůåŘĹä¡£çŤÍCãĂĄPythonåŠŇçž£çÍŃïijŇ
æIJL'äžŻçž£çÍŃæŸŕåIJÍCäÿ■åĹŻåżžçŽĎïijŇèűĚåĞžäžĘPythonèğčéĞŁåŹÍçŽĎæŐğåĹűèŇČåŻť'ãĂĆ
åźűäÿŤäÿĂäžŻçž£çÍŃè£Ÿä¡£çŤÍäžĘPython C APIäÿ■çŽĎåĞ¡æŢřãĂĆ

èğčåĘşæŰźæąĹ

åęĆædIJä¡ăæČşåřĘCãĂĄPythonåŠŇçž£çÍŃæůůåŘĹåIJÍäÿĂèţůïijŇä¡ăéIJĂèęĄçąőä£İæ■čçąőçŽĎåĹİåğŇ
èęĄæČşè£ŹæăůåĄŽïijŇåŘŕäžěåřĘäÿŃåĹŮäžčçăĄæŤ¿åĹřä¡ăçŽĎCäžčçăĄäÿ■åźűçąőä£İåőČåIJÍäżżä¡Ţçž£çÍŃ

```
#include <Python.h>
  ...
  if (!PyEval_ThreadsInitialized()) {
    PyEval_InitThreads();
  }
  ...
```

åŕźäžŐäżżä¡ŢèřČçŤÍPythonåŕźèśąæĹŰPython C APIçŽĎCäžčçăĄïijŇçąőä£İä¡ăéęŰåĚĹåůšçżŔæ■čçąőåI
è£ŹåŘŕäžěçŤÍ `PyGILState_Ensure()` åŠŇ `PyGILState_Release()`
æİěåĄŽåĹřïijŇåęĆäÿŃæL'Ăçď'žïijŽ

```
...
/* Make sure we own the GIL */
PyGILState_STATE state = PyGILState_Ensure();

/* Use functions in the interpreter */
...
/* Restore previous GIL state and return */
PyGILState_Release(state);
...
```

æfŔæňąèřČçŤÍ `PyGILState_Ensure()` éČ¡èęAçŻÿåžŤçŽĎèřČçŤÍ `PyGILState_Release()`.

èőÍèőž

åIJÍæűL'åŘŁåĹřCåŠŇPythonçŽĎéńŸçžğçÍŃåžŔäÿ■ïijŇå¿Ĺåď'ŽäžŃæČĚäÿĂèţůåĄŽæŸŕå¿Ĺåÿÿè ğĄçŽĎ åŘŕèČ¡æŸŕåŕźCãĂĄPythonãĂĄCçž£çÍŃãĂĄPythonçž£çÍŃçŽĎæůůåŘĹä¡£çŤÍãĂĆ åŘłèęAä¡ăçąőä£İèğčéĞŁåŹÍèćńæ■čçąőçŽĎåĹİåğŃåŇŰïijŇåźűäÿŤæűL'åŘŁåĹřèğčéĞŁåŹÍçŽĎCäžčçăĄæL'ğ

èęĄæşÍæĎŘçŽĎæŸŕèřČçŤÍ `PyGILState_Ensure()` åźűäÿ■äijŽçńŇåĹżæŁćå■ăæĹŰäÿ■æŰ■èğčéĞŁåŹÍãĂĆ åęĆæđIJæIJL'åĚűäžŰäžčçăĄæ■čåIJÍæL'ğèąŇïijŇè£Ź åIJÍåĘĚéČÍïijŇèğčéĞŁåŹÍäijŽæL'ğèąŇåŚÍæIJ§æĂğçŽĎçž£çÍŃåĹĞæ■ćïijŇåŻăæ■ď'åęĆæđIJåĚűäžŰçž£çÍŃå èřČçŤÍèĂĚæIJĂçżĹè£ŸæŸŕåŘŕäžčè£ŘèąŇçŽĎïijĹår¡çőąåŘŕèČ¡èęĄåĚĹç■L'äÿĂäijŽïijL'ãĂĆ

17.9 15.9 çŤÍWSIGåŇĚèčĚCäžčçăĄ

éŮőécŸ

ä¡ăæČşèől'ä¡ăåĘŹçŽĎCäžčçăĄä¡IJäÿžäÿĂäÿłCæL'l'åśŢæÍąåİŮæİěèő£éŮőïijŇæČşéĂŽè£Ğä¡£çŤÍ SwigåŇĚèčĚçŤ§æĹŘåŹÍ æİěåőŇæĹŘãĂĆ

èğčåĘşæŰźæąĹ

SwigéĂŽè£Ğèğčæđ ŘCåď'ťæŰĞäžűåźűèĞłåŁÍåĹŻåžžæL'l'åśŢäžčçăĄæİěæŞ■ä¡IJãĂĆ èęĄä¡£çŤÍåőČïijŇä¡ăåĚĹèęĄæIJL'äÿĂäÿłCåď'ťæŰĞäžűãĂĆä¿ŃåęĆïijŇæĹŚäžňçď'žä¿ŃçŽĎåď'ťæŰĞäžűåę

```
/* sample.h */

#include <math.h>
extern int gcd(int, int);
extern int in_mandel(double x0, double y0, int n);
extern int divide(int a, int b, int *remainder);
extern double avg(double *a, int n);

typedef struct Point {
    double x,y;
```

```
} Point;

extern double distance(Point *p1, Point *p2);
```

äÿĂæŮęä¡ăæIJL'äžĘè£Źäÿłåđ't'æŰĞäžűïijŇäÿŃäÿĂæ■ěåřśæŸŕçijŰåĘŹäÿĂäÿłSwigâĂİæŐěåŔčâĂİæŰ
æŇL'çĚğçžęåőŽïijŇè£ŹäžŻæŰĞäžűäżěâĂİ.iâĂİåŘŐçijĂåźűäÿŤçśżäijijäÿŃéİćè£Źæăůïij Ž

```
// sample.i - Swig interface
%module sample
%{
#include "sample.h"
%}

/* Customizations */
%extend Point {
    /* Constructor for Point objects */
    Point(double x, double y) {
        Point *p = (Point *) malloc(sizeof(Point));
        p->x = x;
        p->y = y;
        return p;
   };
};

/* Map int *remainder as an output argument */
%include typemaps.i
%apply int *OUTPUT { int * remainder };

/* Map the argument pattern (double *a, int n) to arrays */
%typemap(in) (double *a, int n)(Py_buffer view) {
  view.obj = NULL;
  if (PyObject_GetBuffer($input, &view, PyBUF_ANY_CONTIGUOUS |␣
↪PyBUF_FORMAT) == -1) {
    SWIG_fail;
  }
  if (strcmp(view.format,"d") != 0) {
    PyErr_SetString(PyExc_TypeError, "Expected an array of doubles
↪");
    SWIG_fail;
  }
  $1 = (double *) view.buf;
  $2 = view.len / sizeof(double);
}

%typemap(freearg) (double *a, int n) {
  if (view$argnum.obj) {
    PyBuffer_Release(&view$argnum);
  }
}
```

```
/* C declarations to be included in the extension module */

extern int gcd(int, int);
extern int in_mandel(double x0, double y0, int n);
extern int divide(int a, int b, int *remainder);
extern double avg(double *a, int n);

typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

äÿĂæŮęä¡ăåĘŹåě¡äžĘæŐěåŘčæŰĞäżűïijŇářśåŘŕäžěåIJÍåŚ¡äžd'èąŇåůěåĚůäÿ■èřČçŤÍSwigäžĘïijŽ

```
bash % swig -python -py3 sample.i
bash %
```

swigçŽĎè¿ŞåĞžářśæŸŕäÿd'äÿłæŰĞäżűïijŇsample_wrap.cåŠŇsample.pyãĂĆ åŘŐéÍćçŽĎæŰĞäżűåřśæŸŕçŤÍæĹůéIJĂèęĄåŕijåĚěçŽĎãĂĆ èĂŇsample_wrap.cæŰĞäżűæŸŕéIJĂèęĄèćńçijŰèŕŚåĽřåŘ■åŔń `_sample` çŽĎæŤŕæŇĄæÍąåİŮçŽĎCäžčçăĄãĂĆ è£ŹäÿłåŕäžěéĂŽè£ĞèůŞæŹőéĂŽæĽ'ľåśŢæÍąåİŮäÿĂæăůçŽĎæŁĂæä¿ŇåęĆïijŇä¡ăåĹŻåžžäžĘäÿĂäÿłåęĆäÿŇæĽ'Ăçd'žçŽĎ `setup.py` æŰĞäżűïijŽ

```
# setup.py
from distutils.core import setup, Extension

setup(name='sample',
      py_modules=['sample.py'],
      ext_modules=[
        Extension('_sample',
                  ['sample_wrap.c'],
                  include_dirs = [],
                  define_macros = [],

                  undef_macros = [],
                  library_dirs = [],
                  libraries = ['sample']
                  )
        ]
)
```

èęĄçijŰèŕŚåŠŇæţŃèŕŢïijŇåIJÍsetup.pyäÿŁæĽ'ğèąŇpython3ïijŇåęĆäÿŃïijŽ

```
bash % python3 setup.py build_ext --inplace
running build_ext
building '_sample' extension
gcc -fno-strict-aliasing -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-
 ↪prototypes
-I/usr/local/include/python3.3m -c sample_wrap.c
```

```
 -o build/temp.macosx-10.6-x86_64-3.3/sample_wrap.o
sample_wrap.c: In function âĂŸSWIG_InitializeModuleâĂŹ:
sample_wrap.c:3589: warning: statement with no effect
gcc -bundle -undefined dynamic_lookup build/temp.macosx-10.6-x86_64-
 ↪3.3/sample.o
 build/temp.macosx-10.6-x86_64-3.3/sample_wrap.o -o _sample.so -
 ↪lsample
bash %
```

åęĆæđIJäÿĂåĹĞæ■čåÿÿçŽĎèŕİïijŇä¡ăäijŽåŔŞçŐřä¡ăåřšåŔŕäžěå¿ĹæŰźä¿£çŽĎä¡£çŤĺçŢ§æĹŘçŽĎCæĽ

```
>>> import sample
>>> sample.gcd(42,8)
2
>>> sample.divide(42,8)
[5, 2]
>>> p1 = sample.Point(2,3)
>>> p2 = sample.Point(4,5)
>>> sample.distance(p1,p2)
2.8284271247461903
>>> p1.x
2.0
>>> p1.y
3.0
>>> import array
>>> a = array.array('d',[1,2,3])
>>> sample.avg(a)
2.0
>>>
```

èőĺèőž

SwigæŸŕPythonåŐĘåŔšäÿ■æđĎåžžæĽľåśŢæĺąåİŮçŽĎæIJĂåŔd'èĂĄçŽĎåůěåĚůäźŇäÿĂãĂĆ SwigèČ¡èĞłåŁĺåŇŰå¿Ĺåd'ŽåŇĚèčĚçŤ§æĹŘåŹĺçŽĎåd'ĎçŘĘãĂĆ

æĽĂæIJL'SwigæŐěåŔčéČ¡äžěçśżäijijäÿŃéİćè£Źæăůç ŽĎäÿžåijĂåd'ťïijŽ

```
%module sample
%{
#include "sample.h"
%}
```

è£ŹäÿłäżĚäżĚåŔłæŸŕåčřæŸŐäžĘæĽľåśŢæĺąåİŮçŽĎåŘ■çğřåźűæŇĞåőŽäžĘCåd'ť'æŰĞäżűïijŇ äÿžäžĘèČ¡èőľçijŰèŕŚéĂŽè£Ğå£Ěéążèę ĄåŇĚåŘńè£ŹäžŽåd'ť'æŰĞäżűïijĹä¡■äžŐ %{ åŠŇ %} çŽĎäžčçăĄïijL'ïijŇ årĘåőČäžňäźŃéŮť'åd'■åĹűçšŸèť'ť'åĹřè¿ŞåĞžäżčçăĄäÿ■ïijŇè£Źäź§æŸŕä¡äèęĄæŢ¿ç¡őæ

SwigæŐěåŔčçŽĎåžŢäÿŃéČĺåĹĘæŸŕäÿĂäÿłCåčřæŸŐåĹŮèąĺïijŇä¡äéIJĂèęĄåIJĺæĽ'ľåśŢäÿ■åŇĚåŘńåő è£ŹéĂŽåÿÿäżŐåd'ť'æŰĞäżűäÿ■èćńåd'■åĹűãĂĆåIJĺæĹŚäžňçŽĎä¿Ńå■Řäÿ■ïijŇæĹŚäžňäžĚäżĚåČŔäÿŃéİćè£

```
%module sample
%{
#include "sample.h"
%}
...
extern int gcd(int, int);
extern int in_mandel(double x0, double y0, int n);
extern int divide(int a, int b, int *remainder);
extern double avg(double *a, int n);

typedef struct Point {
    double x,y;
} Point;

extern double distance(Point *p1, Point *p2);
```

æIJL'äÿĂçĆźéIJĂèęĄåijžèřČçŽĎæŸŕè£ŹäžŻåčřæŸŐäijŽåŚŁèŕL'Swigä¡ăæČşèęĄåIJÍPythonæÍąåİŮäÿ■å
éĂŽåÿÿä¡ăéIJĂèęĄçijÚè¿Śè£ŹäÿłåčřæŸŐåĹŮèąĺæĹŰçŻÿåžŤçŽĎä£őæŤźäÿŃåőČãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJä¡ăäÿ■æČşæ§ŘäžŻåčřæŸŐèćńåŇĚåŘńè£ŻæİěïijŇä¡ăèęĄåřĘåőČäżŐåčřæŸŐåĹŮèąĺäÿ■

ä¡£çŤÍSwigæIJĂåď■æİĆçŽĎåIJřæŰźæŸŕåőČèČ¡çżŹCäžčçăĄæŔŘä¿Żåď ğéĞŔçŽĎèĞłåőŽäźL'æŞ■ä¡IJ
è£ŹäÿłäÿżécŸåď łåď ğïijŇè£ŹéĞŇæŮăæşŢåśŢåijĂïijŇä¡ĘæŸŕæĹŚäžňåIJĺæIJňèŁĆè£ŸåL'l'åśŢçď žäžĘäÿĂäž

çňňäÿĂäÿłèĞłåőŽäźL'æŸŕ `%extend` æŇĞäžď åĚĄèőÿæŰźæşŢèćńéŹĎåŁääĹřåůšå■ŸåIJÍçŽĎçżŞæđĎä
æĹŚä¿Ńå■Řäÿ■ïijŇè£ŹäÿłèćńçŤĺæİěæůżåŁääÿĂäÿłPointçżŞæđĎä¡ŞçŽĎæđĎéĂääŹĺæŰźæşŢãĂĆ
åőČåŘŕäžěèőľ'ä¡ääČŔäÿŃéİćè£Źæăůä¡£çŤĺè£ŹäÿłçżŞæđĎä¡Şïijž

```
>>> p1 = sample.Point(2,3)
>>>
```

åęĆæđIJçŢěè£ĞçŽĎèŕİïijŇPointåŕźèśąåřśå£ĚéążäžěæŻť'åŁääď'■æİĆçŽĎæŰźåijŔæİěèćńåĹŻåżžïijž

```
>>> # Usage if %extend Point is omitted
>>> p1 = sample.Point()
>>> p1.x = 2.0
>>> p1.y = 3
```

çňňäžŇäÿłèĞłåőŽäźL'æűL'åŔŁåĹřåŕź `typemaps.i` åžŞçŽĎåijŢåĚěåŠŇ
`%apply` æŇĞäžď'ïijŇ åőČäijŽæŇĞçď'žSwigåŔĆæŢřç■¿åŘ■ `int *remainder`
èęĄèćńå¡ŞåĄŽæŸŕè¿ŞåĞžåĂijãĂĆ è£ŹäÿłåőđéŹĚäÿŁæŸŕäÿĂäÿłæĺąåijŔåŇźéĚ■èğĎåĹŹãĂĆ
åIJĺæŐěäÿŃæİěçŽĎæL'ĂæIJL'åčřæŸŐäÿ■ïijŇäżżä¡ŢæŮűåĂŹåŘłèęĄçćŕäÿŁ `int
*remainder` ïijŇäżŰåřśäijŽèćńä¡IJäÿžè¿ŞåĞžãĂĆ è£ŹäÿłèĞłåőŽäźL'æŰźæşŢåŔŕäžěèőľ'
`divide()` åĞ¡æŢřè£ŤåŻđäÿd'äÿłåĂijãĂĆ

```
>>> sample.divide(42,8)
[5, 2]
>>>
```

æIJĂåŘŐäÿĂäÿłæűL'åŔŁåĹř `%typemap` æŇĞäžď'çŽĎèĞłåőŽäźL'åŔŕèČ¡æŸŕè£ŹéĞŇåśŢçď žçŽĎæIJĂ
äÿĂäÿłtypemapåřśæŸŕäÿĂäÿłåIJĺè¿ŞåĚěäÿ■çL'źåőŽåŔĆæŢřæĺąåijŔçŽĎèğĎåĹŹãĂĆ
åIJĺæIJňèŁĆäÿ■ïijŇäÿĂäÿłtypemapèćńåőŽäźL'äÿžåŇźéĚ■åŔĆæŢřæĺąåijŔ `(double *a,`

`int n)`. åIJÍtypemapåĘĚéČÍæŸŕäÿĂäÿłCäżčçăĄçL'Ğæőţïij Ňåő ČåŚŁèŕL'SwigæĂŐæăůåřĘäÿĂäÿłPythonåŕ
æIJňèŁĆäżčçăĄä¡£çŤÍäžĘPythonçŽĎçijŞå■Ÿå■Řèőőåőżå Ňźé Ě■äżżä¡ŢçIJŃäÿŁåŐżçśżäijijåŔŇçš¿åžęæŢřçż
ïijĹæŕŤåęĆNumPyæŢřçżĎãĂĄarrayæĺąåİŮåĹŻåžžçŽĎæŢřçżĎç■L'ïijL'ïijŇæŻť'åď'Žèŕůå ŔĆèĂČ15.3åřŔèŁĆ

åIJÍtypemapäżčçăĄåĘĚéČÍïijŇ$1åŠŇ$2è£Źæăůç ŽĎåŔŸéĞŔæŻ£æ■ćäijŽèŐůåŔŰtypemapæĺąåijŔçŽĎC
ïijĹæŕŤåęĆ$1æŸăářĎäÿž `double *a` ïijL'ãĂĆ$inputæŇĞåŘŠäÿĂäÿłä¡IJäÿžè¿ŞåĚěçŽĎ
`PyObject *` åŔĆæŢřïijŇ èĂŇ `$argnum` åřśäžčèąĺåŔĆæŢřçŽĎäÿłæŢřãĂĆ

çijŰåĘŹåŠŇçŘĘèğčtypemapsæŸŕä¡£çŤÍSwigæIJĂå§žæIJńçŽĎåL'■æŔŘãĂĆ
äÿ■äžĚæŸŕèŕť'äžčçăĄæŻť'çěđçğŸïijŇèĂŇäÿŤä¡ăéIJĂèęĄçŘĘèğčPython C
APIåŠŇSwigåŠŇåőČäžď'äžŠçŽĎæŰźåijŘãĂĆ SwigæŰĞæąčæIJL'æŻť'åď'Žè£ŹæŰźéİćçŽĎçżĘèŁĆïijŇåŔŕäz

äÿ■è£ĞïijŇåęĆæđIJä¡ăæIJL'åď'ğéĞŔçŽĎCäżčçăĄéIJĂèęĄèćńæŽť'éIJšäÿžæL'ľ'åśŢæĺąåİŮãĂĆ
SwigæŸŕäÿĂäÿłéİđåÿÿåijžåď'ğçŽĎåůěåĚůãĂĆåĚşéŤőçĆźåIJÍäžŐSwigæŸŕäÿĂäÿłåď'ĎçŘĘCåčřæŸŐçŽĎçij
éĂŽè£Ğåijžåď'ğçŽĎæĺąåijŔåŇźéĚ■åŠŇèĞłåőŽäźL'çžĎäżűïijŇåŔŕäžèèőľä¡ăæŻť'æŤźåčřæŸŐæŇĞåőŽåŠŇçş
æŻť'åď'Žä£ąæĄŕèŕůåŐżæ§ěéŸĚ Swigç¡ŚçńŹ ïijŇ è£ŸæIJL'
çL'źåőŽäžŐPythonçŽĎçŽÿåĚşæŰĞæąč

17.10 15.10 çŤÍCythonåŇĚèčĚCäżčçăĄ

éŮőécŸ

ä¡ăæČşä¡£çŤÍCythonæİěåĹŻåžžäÿĂäÿłPythonæL'ľ'åśŢæĺąåİŮïijŇçŤÍæİěåŇĚèčĚæ§Řäÿłåůšå■ŸåIJÍçŽĎ

èğčåĘşæŰźæąĹ

ä¡£çŤÍCythonæđĎåžžäÿĂäÿłæL'ľ'åśŢæĺąåİŮçIJŃäÿŁåŐżå¿ĹæL'ŃåĘŹæL'ľ'åśŢæIJL'äžŽçśżäijijïijŇ
åŽăäÿžä¡ăéIJĂèęĄåĹŻåžžå¿Ĺåď'ŽåŇĚèčĚåĞ¡æŢřãĂĆäÿ■è£ĞïijŇèů§åL'■éİćäÿ■åŘŇçŽĎæŸŕïijŇä¡ääÿ■éIJĂ

ä¡IJäÿžåĞĘåď'ĞïijŇåĄĞèő¿æIJńçńääżŃçż■éČĺåĹĘçŽĎçď'žä¿Ńäžčçă ĄåůšçżŕèćńçijŰèŕŚåĹřæ§Řäÿłåŕ
`libsample` çŽĎCåĞ¡æŢřåžŞäÿ■äžĘãĂĆ éęŰåĚĺåĹŻåžžäÿĂäÿłåŘ■åŕń `csample.pxd`
çŽĎæŰĞäżűïijŇåęĆäÿŃæL'Ăçď'žïijŽ

```
# csample.pxd
#
# Declarations of "external" C functions and structures

cdef extern from "sample.h":
    int gcd(int, int)
    bint in_mandel(double, double, int)
    int divide(int, int, int *)
    double avg(double *, int) nogil

    ctypedef struct Point:
         double x
         double y

    double distance(Point *, Point *)
```

è£Źäÿłæ ŰĞäżűåIJĺCythonäÿ■çŽĎä¡IJçŤĺåřśèů§CçŽĎåď'ť æŰĞäżűäÿĂæăůãĂĆ åĹİåğŃåčřæŸŐ `cdef extern from "sample.h"` æŇĞåőŽäžĘæĽ'Ăå■ęçŽĎCåď'ť æŰĞäżűãĂĆ æŐěäÿŃæİěçŽĎåčřæŸŐéČ¡æŸŕæİěèĞłäžŐéĆčäÿłåď'ť æŰĞäżűãĂĆæŰĞäżűåŘ■æŸŕ `csample.pxd` ïijŇèĂŇäÿ■æŸŕ `sample.pxd` âĂŤâĂŤè£ŹçĆźå¿ĹéĞ■èęĄãĂĆ

äÿŃäÿĂæ■ěïijŇåĹŻåžžäÿĂäÿłåŘ■äÿž `sample.pyx` çŽĎéŮőécŸãĂĆ èŕěæŰĞäżűäijŽåőŽäźĽ'åŇĚèčĚåŹĺïijŇçŤĺæİěæąěæŐěPythonèğčéĞŁåŹĺåĹř `csample.pxd` äÿ■åčřæŸŐçŽĎCäžčçăĄãĂĆ

```
# sample.pyx

# Import the low-level C declarations
cimport csample

# Import some functionality from Python and the C stdlib
from cpython.pycapsule cimport *

from libc.stdlib cimport malloc, free

# Wrappers
def gcd(unsigned int x, unsigned int y):
    return csample.gcd(x, y)

def in_mandel(x, y, unsigned int n):
    return csample.in_mandel(x, y, n)

def divide(x, y):
    cdef int rem
    quot = csample.divide(x, y, &rem)
    return quot, rem

def avg(double[:] a):
    cdef:
        int sz
        double result

    sz = a.size
    with nogil:
        result = csample.avg(<double *> &a[0], sz)
    return result

# Destructor for cleaning up Point objects
cdef del_Point(object obj):
    pt = <csample.Point *> PyCapsule_GetPointer(obj,"Point")
    free(<void *> pt)

# Create a Point object and return as a capsule
def Point(double x,double y):
    cdef csample.Point *p
    p = <csample.Point *> malloc(sizeof(csample.Point))
    if p == NULL:
```

```
        raise MemoryError("No memory to make a Point")
    p.x = x
    p.y = y
    return PyCapsule_New(<void *>p,"Point",<PyCapsule_Destructor>
 ↪del_Point)

def distance(p1, p2):
    pt1 = <csample.Point *> PyCapsule_GetPointer(p1,"Point")
    pt2 = <csample.Point *> PyCapsule_GetPointer(p2,"Point")
    return csample.distance(pt1,pt2)
```

èŕěæŰĞäżűæŻťåď ŽçŽĎçżĘèŁĆéČĺåĹĘäijŽåIJĺèőĺèőžéČĺåĹĘèŕęçżĘåśŢåijĂãĂĆ æIJĂåŘŐïijŇäÿžäžĘæđĎåžžæĽľåśŢæĺąåİŮïijŇåČŔäÿŃéİćè£ŹæăůåĹŻåžžäÿĂäÿł `setup.py` æŰĞäżűïijŽ

```
from distutils.core import setup
from distutils.extension import Extension
from Cython.Distutils import build_ext

ext_modules = [
    Extension('sample',

              ['sample.pyx'],
              libraries=['sample'],
              library_dirs=['.'])]
setup(
  name = 'Sample extension module',
  cmdclass = {'build_ext': build_ext},
  ext_modules = ext_modules
)
```

èęĄæđĎåžžæĹŚäžňæţŃèŕŢçŽĎçŻőæăĞæĺąåİŮïijŇåČŔäÿŃéİćè£ŹæăůåĄŽïijŽ

```
bash % python3 setup.py build_ext --inplace
running build_ext
cythoning sample.pyx to sample.c
building 'sample' extension
gcc -fno-strict-aliasing -DNDEBUG -g -fwrapv -O3 -Wall -Wstrict-
 ↪prototypes
 -I/usr/local/include/python3.3m -c sample.c
 -o build/temp.macosx-10.6-x86_64-3.3/sample.o
gcc -bundle -undefined dynamic_lookup build/temp.macosx-10.6-x86_64-
 ↪3.3/sample.o
  -L. -lsample -o sample.so
bash %
```

åęĆæđIJäÿĂåĹĞéąžåĹľçŽĎèŕÎïijŇä¡ååžŤèŕěæIJĽäžĘäÿĂäÿłæĽľåśŢæĺąåİŮ `sample.so` ïijŇåŔŕåIJĺäÿŃéİćä¿Ńå■Řäÿäÿ■ä¡£çŤĺïijŽ

```
>>> import sample
>>> sample.gcd(42,10)
2
>>> sample.in_mandel(1,1,400)
False
>>> sample.in_mandel(0,0,400)
True
>>> sample.divide(42,10)
(4, 2)
>>> import array
>>> a = array.array('d',[1,2,3])
>>> sample.avg(a)
2.0
>>> p1 = sample.Point(2,3)
>>> p2 = sample.Point(4,5)
>>> p1
<capsule object "Point" at 0x1005d1e70>
>>> p2
<capsule object "Point" at 0x1005d1ea0>
>>> sample.distance(p1,p2)
2.8284271247461903
>>>
```

èőÍèőž

æIJňèŁĆåŇĚåŘńäžĘå¿Ĺåď ŽåĽ■éİćæĽĂèőšçŽĎéńŸçžğçĽźæĂğïijŇåŇĚæŃňæŢřçżĎæŞ■ä¡IJãĂĄåŇĚæŕŔäÿĂéČĺåĹĘéČ¡äijŽéĂŘäÿłèćńèőšè£řåĹřïijŇä¡ĘæŸŕæĹŚäžňæIJĂåě¡èČ¡åď■äźăäÿĂäÿŃåĽ■éİćåĞăářŔèŁåIJĺéąűåśĆïijŇä¡£çŢÍCythonæŸŕå§žäžŐCäźŃäÿŁãĂĆ.pxdæŰĞäżűäžĚäżĚåŔłåŇĚåŘńCåőŽäźĽ'ïijĹçśżäijij.hä.pyxæŰĞäżűåŇĚåŘńäžĘåőđçŐřïijĹçśżäijij.cæŰĞäżűïijĽãĂĆcimport èŕ■åŔèèćńCythonçŢÍæİěåŕijåĚě.pxdæŰĞäżűäÿ■çŽĎåőŽäźĽãĂĆ åőČèů§ä¡£çŢÍæŹőéĂŽçŽĎåŁăè¡¡PythonæĺąåİŮçŽĎåŕijåĚèèŕ■åŔěæŸŕäÿ■åŘŇçŽĎãĂĆ

åř¡çőą.*pxd* æŰĞäżűåŇĚåŘńäžĘåőŽäźĽ'ïijŇä¡ĘåőČäżňåźűäÿ■æŸŕçŢÍæİěèĞłåŁÍåĹŻåžžæĽ'ľåśŢäžčçăĄçåŻăæ■ď'ïijŇä¡ăè£ŸæŸŕèęĄåĘŹåŇĚèčĚåĞ¡æŢřãĂĆä¿ŃåęĆïijŇåřśçőŮ `csample.pxd` æŰĞäżűåčřæŸŐäžĘ `int gcd(int, int)` åĞ¡æŢřïijŇ ä¡ääż■çĎűéIJĂèęĄåIJÍ `sample.pyx` äÿ■äÿžåőČåĘŹäÿĂäÿłåŇĚèčĚåĞ¡æŢřãĂĆä¿ŃåęĆïijŽ

```
cimport csample

def gcd(unsigned int x, unsigned int y):
    return csample.gcd(x,y)
```

åŕźäžŐçőĂå■ŢçŽĎåĞ¡æŢřïijŇä¡ääźűäÿ■éIJĂèęĄåŐżåĄŽåď'łåď'ŽçŽĎæŮűãĂĆ CythonäijŽçŢ§æĹŘåŇĚèčĚäżčçăĄæİěæ■čçąőçŽĎè¡ñæ■ćåŔĆæŢřåŠŇè£ŤåŻđåĂijãĂĆ çżŚåőŽåĹřåśđæĂğäÿŁçŽĎCæŢřæ■őçśżåđŃæŸŕåŘŕéĂĽ'çŽĎãĂĆäÿ■è£ĞïijŇåęĆæđIJä¡ääŇĚåŘńäžĘåőČäżňï ä¿ŃåęĆïijŇåęĆæđIJæIJĽ'äžžä¡£çŢÍèť §æŢřæİěěřČçŢÍè£ŹäÿłåĞ¡æŢřïijŇäijŽæŁŻåĞžäÿĂäÿłåijĆåÿÿïijŽ

```
>>> sample.gcd(-10,2)
Traceback (most recent call last):
```

```
  File "<stdin>", line 1, in <module>
  File "sample.pyx", line 7, in sample.gcd (sample.c:1284)
    def gcd(unsigned int x,unsigned int y):
OverflowError: can't convert negative value to unsigned int
>>>
```

åęĆædIJä¡ăæČşåŕźåŇĚèčĚåĞ¡æŢřåĄŽåŔęåď˙ŰçŽĎæčĂæ§ëïijŇåŔłéIJĂèęĄä¡£çŤĺåŔęåď˙ŰçŽĎåŇĚèčĚ

```
def gcd(unsigned int x, unsigned int y):
    if x <= 0:
        raise ValueError("x must be > 0")
    if y <= 0:
        raise ValueError("y must be > 0")
    return csample.gcd(x,y)
```

åIJÍcsample.pxdæŰĞäżűäÿ■çŽĎ‘‘in_mandel()‘‘ åčřæŸŐæIJL’äÿłå¿ĹæIJL’èűčä¡ĘæŸŕæŕŤè¿ČéŽ¿çŘĘèğ
åIJÍè£ŹäÿłæŰĞäżűäÿ■ïijŇåĞ¡æŢřèćńåčřæŸŐäÿžçĎůåŘŐäÿĂäÿłbintèĂŇäÿ■æŸŕäÿĂäÿłintãĂĆ
åőČäijŽèőľåĞ¡æŢřåĹŻåżžäÿĂäÿłæ■čçąőçŽĎBooleanåĂijèĂŇäÿ■æŸŕçőĂå■ŢçŽĎæŢt’æŢřãĂĆ
åŻăæ■d’ïijŇè£ŤåŻđåĂij0èąĺçd’žFalseèĂŇ1èąĺçd’žTrueãĂĆ

åIJÍCythonåŇĚèčĚåŹĺäÿ■ïijŇä¡ăåŔŕäżěéĂL’æŃl’åčřæŸŐCæŢřæ■őçśżåđŃïijŇäź§åŔŕäżěä¡£çŤĺæL’ĂæIJ
åŕźäžŐ `divide()` çŽĎåŇĚèčĚåŹĺåśŢçd’žäžĘè£ŹæăůäÿĂäÿłä¿Ńå■ŘïijŇåŘŇæŮűè£ŸæIJL’åęĆä¡ŢåŐżåd’Ď

```
def divide(x,y):
    cdef int rem
    quot = csample.divide(x,y,&rem)
    return quot, rem
```

åIJÍè£ŹéĞŇïijŇrem åŘŸéĞŔèćńæŸ¿çd’žçŽĎåčřæŸŐäÿžäÿĂäÿłCæŢt’ådŃåŘŸéĞŔãĂĆ
å¡ŞåőČèćńäijăåĚě `divide()` åĞ¡æŢřçŽĎæŮűåĂŹïijŇ`&rem`
åĹŻåżžäÿĂäÿłèů§CäÿĂæăůçŽĎæŇĞåŘŚåőČçŽĎæŇĞéŠĹãĂĆ `avg()`
åĞ¡æŢřçŽĎäżčçăĄæijŤçd’žäžĘCythonæŻt’énŸçžğçŽĎçL’źæĂğãĂĆ
éęŰåĚĹ `def avg(double[:] a)` åčřæŸŐäžĘ `avg()`
æŐěåŔŮäÿĂäÿłäÿĂçżt’çŽĎåŔŇçš¿åžęåĘĚå■ŸèğĘåŻ¿ãĂĆ æIJĂæČŁåěĞçŽĎéČĺåĹĘæŸŕè£ŤåŻđçŽĎçżŞæđ

```
>>> import array
>>> a = array.array('d',[1,2,3])
>>> import numpy
>>> b = numpy.array([1., 2., 3.])
>>> import sample
>>> sample.avg(a)
2.0
>>> sample.avg(b)
2.0
>>>
```

åIJÍæ■d’åŇĚèčĚåŹĺäÿ■ïijŇ`a.size0` åŠŇ `&a[0]` åĹĘåĹńåijŢçŤĺæŢřçżĎåĚČçt’ääÿłæŢřåŠŇåžŢåśĆæŇ
èŕ■æşŢ `<double *> &a[0]` æŢŹä¡ăæĂŐæăůåřĘæŇĞéŠĹè¡ňæ■ćäÿžäÿ■åŘŇçŽĎçśżåđŃãĂĆ
åL’■æŘŘæŸŕCäÿ■çŽĎ `avg()` æŐěåŔŮäÿĂäÿłæ■čçąőçśżåđŃçŽĎæŇĞéŠĹãĂĆ
åŔĆèĂČäÿŃäÿĂèŁĆåĚşäžŐCythonåĘĚå■ŸèğĘåŻ¿çŽĎæŻt’énŸçžğèőšè£řãĂĆ

éŹď äžĘåď ĎçŘĘéĂŽåÿÿçŽĎæŢřçżĎåď ŰïijŇavg() çŽĎè£Źäÿłä¿Ńå■Řè£ŸåśŢçď žäžĘåęĆä¡Ţåď ĎçŘ
èŕ■åŔě with nogil: åčřæŸŐäžĘäÿĂäÿłäÿ■éIJĂèęĄGILåŕśèČ¡æĽğèąŇçŽĎäżčçăĄåİŮãĂĆ
åIJĺè£ŹäÿłåİŮäÿ■ïijŇäÿ■èČ¡æIJĽäżżä¡ŢçŽĎæŹőéĂŽPythonåŕźèśąâĂŤâĂŤåŔłèČ¡ä¡£çŤĺèćńåčřæŸŐäÿž
cdef çŽĎåŕźèśąåŠŇåĞ¡æŢřãĂĆ åŔęåď ŰïijŇåď ŰéČĺåĞ¡æŢřå£ĚéążçŐřåőđçŽĎåčřæŸŐåőČäżňèČ¡äÿ■ä¿İèţ
åŻăæ■ď ïijŇåIJĺcsample.pxdæŰĞäżűäÿ■ïijŇavg() èćńåčřæŸŐäÿž double avg(double
*, int) nogil.

åŕźPointçżŞæđĎä¡ŞçŽĎåď ĎçŘĘæŸŕäÿĂäÿłæŇŚæĹŸãĂĆæIJňèŁĆä¡£çŤĺèČűåŻŁåŕźèśąåŕĘPointåŕźèśąå
èęĄè£ŹæăůåĄŽçŽĎèŕİïijŇåžŢåśĆCythonäżčçăĄçĺ■å¿őæIJĽçĆźåď ■æİĆãĂĆ
éęŰåĚĹïijŇäÿŃéİćçŽĎåŕijåĚěèćńçŤĺæİěåijŢåĚěCåĞ¡æŢřåžŞåŠŇPython C
APIäÿ■åőŽäźĽçŽĎåĞ¡æŢřïijŽ

```
from cpython.pycapsule cimport *
from libc.stdlib cimport malloc, free
```

åĞ¡æŢř del_Point() åŠŇ Point() ä¡£çŤĺè£ŹäÿłåŁ§èČ¡æİěåĹŻåżžäÿĂäÿłèČűåŻŁåŕźèśąïijŇ
åőČäijŽåŇĚèčĚäÿĂäÿł Point * æŇĞéŠĹãĂĆcdef del_Point() åřĘ del_Point()
åčřæŸŐäÿžäÿĂäÿłåĞ¡æŢřïijŇ åŔłèČ¡éĂŽè£ĞCythonèő£éŮőïijŇèĂŇäÿ■èČ¡äżŐPythonäÿ■èő£éŮőãĂĆ
åŻăæ■ď ïijŇè£ŹäÿłåĞ¡æŢřåŕźåď ŰéČĺæŸŕäÿ■åŔŕèğĄçŽĎâĂŤâĂŤåőČèćńçŤĺæİěå¡ŞåĄŽäÿĂäÿłåŻđèřČåĞ¡æ
åĞ¡æŢřèřČçŤĺæŕŤåęĆ PyCapsule_New() ãĂĄPyCapsule_GetPointer()
çŻť æŐěæİěèĞłPython C APIåźűäÿŤäżěåŘŇæăůçŽĎæŰźåijŔèćńä¡£çŤĺãĂĆ

distance åĞ¡æŢřäżŐ Point() åĹŻåżžçŽĎèČűåŻŁåŕźèśąäÿ■æŔŘåŔŰæŇĞéŠĹãĂĆ
è£ŹéĞŇèęĄæşĺæĎŔçŽĎæŸŕä¡ăäÿ■éIJĂèęĄæŃĚå£ČåijĆåÿÿåď ĎçŘĘãĂĆ
åęĆæđIJäÿĂäÿłéŤŹèŕŕçŽĎåŕźèśąèćńäijăè£ŻæİěïijŇPyCapsule_GetPointer()
äijŽæŁŻåĞžäÿĂäÿłåijĆåÿÿïijŇ ä¡ĘæŸŕCythonåůšçżŔç§ěéĄŞæĂŐäźĹæ§ěæĽ¿åĹřåőČïijŇåźűåřĘåőČäżŐ
distance() äijăéĂŠåĞžåŐżãĂĆ

åď ĎçŘĘPointçżŞæđĎä¡ŞäÿĂäÿłçijžçĆźæŸŕåőČçŽĎåőđçŐřæŸŕäÿ■åŔŕèğĄçŽĎãĂĆ
ä¡ăäÿ■èČ¡èő£éŮőäżżä¡ŢåśđæĂğæİěæ§ěçIJŃåőČçŽĎåĘĚéČĺãĂĆ
è£ŹéĞŇæIJĽåŔęåď ŰäÿĂçğ■æŰźæşŢåŐżåŇĚèčĚåőČïijŇåŕśæŸŕåőŽäźĽäÿĂäÿłæĽľåśŢçśżåđŃïijŇåęĆäÿŃ

```
# sample.pyx

cimport csample
from libc.stdlib cimport malloc, free
...

cdef class Point:
    cdef csample.Point *_c_point
    def __cinit__(self, double x, double y):
        self._c_point = <csample.Point *> malloc(sizeof(csample.
↪Point))
        self._c_point.x = x
        self._c_point.y = y

    def __dealloc__(self):
        free(self._c_point)

    property x:
        def __get__(self):
```

```
            return self._c_point.x
        def __set__(self, value):
            self._c_point.x = value

    property y:
        def __get__(self):
            return self._c_point.y
        def __set__(self, value):
            self._c_point.y = value

def distance(Point p1, Point p2):
    return csample.distance(p1._c_point, p2._c_point)
```

åIJÍè£ŹéĞŇïijŇcdifçśż Point åřĘPointåčřæŸŐäÿžäÿĂäÿłæĽľåśŢçśżåđŃãĂĆ çśżåśđæĂğ cdef csample.Point *_c_point åčřæŸŐäžĘäÿĂäÿłåőđä¿ŃåŔŸéĞŔïijŇ æŃěæIJĽäÿĂäÿłæŇĞåŘŠåžŢåśĆPointçżŞæđĎä¡ŞçŽĎæŇĞéŠĹãĂĆ __cinit__() åŠŇ __dealloc__() æŰźæşŢéĂŽè£Ğ malloc() åŠŇ free() åĹŻåżžåźűéŤĂæŕĄåžŢåśĆCçżŞæđĎä¡ŞãĂĆ xåŠŇyåśđæĂğçŽĎåčřæŸŐèőľä¡ãèŐůåŔŰåŠŇèő¿ç¡őåžŢåśĆçżŞæđĎä¡ŞçŽĎåśđæĂğåĂijãĂĆ distance() çŽĎåŇĚèčĚåŹÍè£ŸåŔŕäżěèćńä£őæŤźïijŇä¡£å¿ŮåőČèČ¡æŐěåŔŮ Point æĽľåśŢçśżåđŃåőđä¿Ńä¡IJäÿžåŔĆæŢřïijŇ èĂŇäijäéĂŠåžŢåśĆæŇĞéŠĹçżŹĆåĞ¡æŢřãĂĆ

åĄŽäžĘè£ŹäÿłæŤźåŔŸåŘŐïijŇä¡ääijŽåŔŞçŐřæŞ■ä¡IJPointåŕźèśąåřśæŸ¿å¿ŮæŻť åŁäèĞłçĎűäžĘïijŽ

```
>>> import sample
>>> p1 = sample.Point(2,3)
>>> p2 = sample.Point(4,5)
>>> p1
<sample.Point object at 0x100447288>
>>> p2
<sample.Point object at 0x1004472a0>
>>> p1.x
2.0
>>> p1.y
3.0
>>> sample.distance(p1,p2)
2.8284271247461903
>>>
```

æIJňèŁĆåůšçżŔæijŤçd'žäžĘå¿Ĺåd'ŽCythonçŽĎæäÿå£ČçĽźæĂğïijŇä¡ååŔŕäžěäżěæ■d'äÿžå§žåĞĘæİěæ äÿ■è£ĞïijŇä¡ăæIJĂåě¡åĚĹåŐźéŸĚèŕżäÿŃåőŸæŰźæŰĞæąćæİěäžĘèğčæŻť åd'Žä£qæĄřãĂĆ

æŐěäÿŃæİěåĞäèŁĆè£ŸäijŽçżğçż■æijŤçd'žäÿĂäžŻCythonçŽĎåĚűäžŰçĽźæĂğãĂĆ

17.11 15.11 çŤÍCythonåĘŹéńŸæĂğèČ¡çŽĎæŢřçżĎæŞ■ä¡IJ

éŮőécŸ

ä¡ăèęĄåĘŹéńŸæĂğèČ¡çŽĎæŞ■ä¡IJæİéèĞłNumPyäźŃçśżçŽĎæŢřçżĎèőąçőŮåĞ¡æŢřãĂĆ ä¡ăåůšçżŔç§ěéĄŞäžĘCythonè£ŹæăůçŽĎåůěåĚůäijŽèőľ'åőČåŔŸå¿ŮçőĂå■ŢïijÑä¡ĘæŸŕåźűäÿ■çąőåőŽèřěæĂ

èğčåĘşæŰźæąĹ

ä¡IJäÿžäÿĂäÿłä¿Ńå■ŘïijÑäÿŃéİćçŽĎäžčçăĄæijŤçd'žäžĘäÿĂäÿłCythonåĞ¡æŢřïijÑçŤÍæİëä£őæŢt'äÿĂä

```
# sample.pyx (Cython)

cimport cython

@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip(double[:] a, double min, double max, double[:] out):
    '''
    Clip the values in a to be between min and max. Result in out
    '''
    if min > max:
        raise ValueError("min must be <= max")
    if a.shape[0] != out.shape[0]:
        raise ValueError("input and output arrays must be the same␣
↪size")
    for i in range(a.shape[0]):
        if a[i] < min:
            out[i] = min
        elif a[i] > max:
            out[i] = max
        else:
            out[i] = a[i]
```

èęĄçijŰèŕŚåŠŇædĎåžžè£ŹäÿłæL'l'åśŢïijÑä¡ăéIJĂèęĄäÿĂäÿłåČŔäÿŃéİćè£ŹæăůçŽĎ `setup.py` æŰĞäžű ïijĹä¡£çŤÍ `python3 setup.py build_ext --inplace` æİěædĎåžžåőČïijL'ïijŽ

```
from distutils.core import setup
from distutils.extension import Extension
from Cython.Distutils import build_ext

ext_modules = [
    Extension('sample',
              ['sample.pyx'])
]

setup(
  name = 'Sample app',
```

```
  cmdclass = {'build_ext': build_ext},
  ext_modules = ext_modules
)
```

ä¡ääijŽåŔŠçŐřçżŞæđIJåĞ¡æŢřçąőåőđåŕźæŢřçżĎè£ŻèąŇçŽĎä£őæ■čïijŇåźűäÿŤåŔŕäżěéĂĆçŤĺäžŐåď'Žç

```
>>> # array module example
>>> import sample
>>> import array
>>> a = array.array('d',[1,-3,4,7,2,0])
>>> a

array('d', [1.0, -3.0, 4.0, 7.0, 2.0, 0.0])
>>> sample.clip(a,1,4,a)
>>> a
array('d', [1.0, 1.0, 4.0, 4.0, 2.0, 1.0])

>>> # numpy example
>>> import numpy
>>> b = numpy.random.uniform(-10,10,size=1000000)
>>> b
array([-9.55546017,  7.45599334,  0.69248932, ...,  0.69583148,
       -3.86290931,  2.37266888])
>>> c = numpy.zeros_like(b)
>>> c
array([ 0.,  0.,  0., ...,  0.,  0.,  0.])
>>> sample.clip(b,-5,5,c)
>>> c
array([-5.        ,  5.        ,  0.69248932, ...,  0.69583148,
       -3.86290931,  2.37266888])
>>> min(c)
-5.0
>>> max(c)
5.0
>>>
```

ä¡àè£ŸäijŽåŔŠçŐřè£ŘèąŇçŤ§æĹŘçżŞæđIJéİđåÿÿçŽĎå£ñãĂĆ
äÿŃéİćæĹŚäżňåřĘæIJňä¿ŃåŠŇnumpyäÿ■çŽĎåůšå■ŸåIJĺçŽĎ clip()
åĞ¡æŢřåĄŽäÿĂäÿłæĂğèČ¡åŕźærŤïijŽ

```
>>> timeit('numpy.clip(b,-5,5,c)','from __main__ import b,c,numpy',
 ↪number=1000)
8.093049556000551
>>> timeit('sample.clip(b,-5,5,c)','from __main__ import b,c,sample
 ↪',
...           number=1000)
3.760528204000366
>>>
```

æ■čåęĆä¡äçIJŃåĹřçŽĎïijŇåőČèęĄå£ńå¿Ĺåď'ŽâĂŤâĂŤè£ŹæŸŕäÿĂäÿłå¿ĹæIJL'èűčçŽĎçżŞæđIJïijŇåŽă

èőÍèőž

æIJňèŁĆåĹľçŤĺäžĘCythonçśžåđŃçŽĎåĘĚå■ŸèğĘåŻ¿ïijŇæđĄåď˘ğçŽĎçőĂåŇŰäžĘæŢřçżĎçŽĎæŞ■ä¡I
`cpdef clip()` åčřæŸŐäžĘ `clip()` åŘŇæŮűäÿžCçžğåĹńåĞ¡æŢřäžěåŔŁPythonçžğåĹńåĞ¡æŢřãĂĆ
åIJÍCythonäÿ■ïijŇè£ŹäÿłæŸŕå¿ĹéĞ■èęĄçŽĎïijŇåŻääÿžåőČèąĺçď˘žæ■ď˘åĞ¡æŢřěřČçŤĺèęĄæŕŤåĚűäżŰCytho
ïijĹæŕŤåęĆä¡ăæČşåIJĺåŔęåď˘ŰäÿĂäÿłäÿ■åŘŇçŽĎCythonåĞ¡æŢřäÿ■èřČçŤĺclip()ïijL˘ãĂĆ

çśžåđŃåŔĆæŢř `double[:] a` åŠŇ `double[:] out` åčřæŸŐè£ŹäžŻåŔĆæŢřäÿžäÿĂçżt˘çŽĎåŔŇçš¿ážęæŢřçżĎãĂĆ
ä¡IJäÿžè¿ŞåĚëïijŇåőČäžñäijŽèő£éŰőäżä¡ŢåődçŐřäžĘåĘĚå■ŸèğĘåŻ¿æŐěåŔčçŽĎæŢřçżĎåŕźèśąïijŇè£ŹäÿłŰ
3118æIJL˘èŕęçżĘåőŽäźL˘ãĂĆ åŇĚæŃñäžĘNumPyäÿ■çŽĎæŢřçżĎåŠŇåĘĚç¡őçŽĎarrayåžŞãĂĆ

å¡Şä¡ăçijŰåĘŹçŤ§æĹŘçżŞædIJäÿžæŢřçżĎçŽĎäžčçăĄæŮűïijŇä¡ăåžŤèŕëéĄţå¿łäÿŁéİćçđ˘žä¿ŃéĆčæăůè
åőČäijŽåřĘåĹŻåžžè¿ŞåĞžæŢřçżĎçŽĎèť˘čäžżçżŹèřČçŤĺèĂĚïijŇäÿ■éIJĂèęĄç§ěéĄŞä¡ăæŞ■ä¡IJçŽĎæŢřçżĎç
ïijĹåőČäżĚäżĚåĄĞëő¿æŢřçżĎåůšçżŔåĞĘåď˘Gåě¡äžĘïijŇåŔłéIJĂèęĄåĄžäÿĂäžŽåřŘçŽĎæčÄæ§ěæŕŤåęĆçą
åIJÍåČŔNumPyäźŃçśżçŽĎåžŞäÿ■ïijŇä¡£çŤÍ `numpy.zeros()` æĹŰ `numpy.zeros_like()` åĹŻåžžè¿ŞåĞžæŢřçżĎçŻÿåŕźèĂŃèÍĂæŕŤè¿Čåőźæ ŸŞãĂĆåŔęåď˘ŰïijŇèęĄåĹŻåžžæIJłåĹİ
ä¡ääŔŕäžěä¡£çŤÍ `numpy.empty()` æĹŰ `numpy.empty_like()` .
åęĆædIJä¡ăæČşèęĘçŻŰæŢřçżĎåĘĚåőźä¡IJäÿžçżŞædIJçŽĎèŕÍéĂL˘æŃľè£Źäÿd˘äÿłäijŽæŕŤè¿Čå£ńçĆźãĂĆ

ä¡ää¡ăçŽĎåĞ¡æŢřåődçŐřäÿ■ïijŇä¡ääŔłéIJĂèęĄçőĂå■ŢçŽĎéĂŽè£ĞäÿŃæăĞè£ŘçőŮåŠŇæŢřçżĎæ§ěæ
CythonäijŽèt˘§èt˘čäÿžä¡ăçŤ§æĹŘéńŸæŢĹçŽĎäžčçăĄãĂĆ

`clip()` åőŽäźL˘äźŃåL˘■çŽĎäÿd˘äÿłèčĚéěřåŹĺåŔŕäžěäijŸåŇŰäÿŃæĂğèČ¡ãĂĆ
`@cython.boundscheck(False)` çIJĄåŐżäžĘęæL˘ĂæIJL˘çŽĎæŢřçżĎèűŁçŢŇæčÄæ§ěijŇ
å¡Şä¡ăç§ěéĄŞäÿŃæăĞèő£éŰőäÿ■äijŽèűŁçŢŇçŽĎæŮűåĂŹåŔŕäžěä¡£çŤĺåőČãĂĆ
`@cython.wraparound(False)` æűĹéŹd˘äžĘçŻÿåŕźæŢřçżĎår¿éČĺçŽĎèt˘§æŢřäÿŃæăĞçŽĎåd˘ĎçŘĘïijĺ
åijŢåĚèè£Źäÿd˘äÿłèčĚéěřåŹĺåŔŕäžěædĄåď˘ğçŽĎæŔŘå■ĞæĂğèČ¡ïijĹæţŃèŕŢè£Źäÿłä¿Ńå■ŘçŽĎæŮűåĂŹåd˘

äżżä¡ŢæŮűåĂŹåd˘ĎçŘĘæŢřçżĎæŮűïijŇçăŤçl˘űåźűæŤźåŰĎåžŢåśĆçőŮæşŢåŘŇæăůåŔŕäžěædĄåď˘ğçŽ
ä¿ŃåęĆïijŇèĂČèŹŠåŕź `clip()` åĞ¡æŢřçŽĎåęĆäÿŃä£őæ■čïijŇä¡£çŤĺæİąäżűèąĺè¿¿åijŔïijŽ

```
@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip(double[:] a, double min, double max, double[:] out):
    if min > max:
        raise ValueError("min must be <= max")
    if a.shape[0] != out.shape[0]:
        raise ValueError("input and output arrays must be the same␣
↪size")
    for i in range(a.shape[0]):
        out[i] = (a[i] if a[i] < max else max) if a[i] > min else␣
↪min
```

åődéŹĚætŃèŕŢçżŞædIJæŸŕïijŇè£ŹäÿłçL˘ĹæIJňçŽĎäžčçăĄè£ŘèąŇéĂ§åžęèęĄå£ń50%äžëäÿŁïijĹ2.44çğ
`timeit()` ætŃèŕŢçŽĎ3.76çğŠïijL˘ãĂĆ

åĹřè£ŹéĞŇäÿžæ■ćïijŇä¡ääŔŕèČ¡æČşç§ěéĄŞè£Źçğ■äžčçăĄæĂŐäźĹèČ¡èů§æL˘ŃåĘŹCèŕ■èĺĂPKåŚćïij§
ä¿ŃåęĆïijŇä¡ääŔŕèČ¡åĘŹäžĘåęĆäÿŃçŽĎCåĞ¡æŢřåźűä¡£çŤĺåL˘■éİćåĞäèŁĆçŽĎæŁĂæIJŕæİèæL˘ŃåĘŹæL˘l˘

```
void clip(double *a, int n, double min, double max, double *out) {
  double x;
```

```
  for (; n >= 0; n--, a++, out++) {
    x = *a;

    *out = x > max ? max : (x < min ? min : x);
  }
}
```

æĹŚäžňæšąæIJL’åśŢçd’žè£ŹäÿłçŽĎæL’l’åśŢäžčçăĄïijŇä¡ĘæŸŕèŕŢélŇäźŃåŘŐïijŇæĹŚäžňåŔŚçŐřäÿĂäj
æIJĂåžŢäÿŃçŽĎäÿĂèąŇæŕŤä¡ăæČşèšąçŽĎè£ŘèąŇçŽĎå£ńå¿Ĺåd’ŽãĂĆ

ä¡ăåŔŕäžěåŕźåőđä¿Ńäžčçă Ąæđ Ďåžžåd’ŽäÿłæL’l’åśŢãĂĆ åŕźäžŐæ§ŘäžŻæŢřçżĎæŞ■ä¡IJïijŇæIJĂåě¡èęĄ
èęĄè£ŹæăůåĄŽçŽĎèŕÍïijŇéIJĂèęĄä£őæŤźäžčçăĄïijŇä¡£çŤÍ `with nogil:` èŕ■åŘěïijŽ

```
@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip(double[:] a, double min, double max, double[:] out):
    if min > max:
        raise ValueError("min must be <= max")
    if a.shape[0] != out.shape[0]:
        raise ValueError("input and output arrays must be the same␣
↪size")
    with nogil:
        for i in range(a.shape[0]):
            out[i] = (a[i] if a[i] < max else max) if a[i] > min␣
↪else min
```

åęĆæđIJä¡ăæČşåĘŹäÿĂäÿłæŞ■ä¡IJäžŇçżt’æŢřçżĎçŽĎçL’ĹæIJňïijŇäÿŃéİćæŸŕåŔŕäžěåŔĆèĂČäÿŃïijŽ

```
@cython.boundscheck(False)
@cython.wraparound(False)
cpdef clip2d(double[:,:] a, double min, double max, double[:,:]␣
↪out):
    if min > max:
        raise ValueError("min must be <= max")
    for n in range(a.ndim):
        if a.shape[n] != out.shape[n]:
            raise TypeError("a and out have different shapes")
    for i in range(a.shape[0]):
        for j in range(a.shape[1]):
            if a[i,j] < min:
                out[i,j] = min
            elif a[i,j] > max:
                out[i,j] = max
            else:
                out[i,j] = a[i,j]
```

åÿŇæIJŻèŕźèĂĚäÿ■èęĄå£ŸäžĘæIJňèŁĆæL’ĂæIJL’äžčçăĄéČ¡äÿ■äijŽçźŚåőŽåĹřæ§ŘäÿłçL’źåőŽæŢřçżĎ
è£ŹæăůäžčçăĄåřśæŻt’æIJL’çĄţæt’żæĂğãĂĆ äÿ■è£ĞïijŇèęĄæşĺæĎŔçŽĎæŸŕåęĆæđIJåd’ĎçŘĘæŢřçżĎèęĄæ
è£ŹäžŻåĘĚåőźåůšçżŔèűĚåĞžæIJňèŁĆèŇČåŻt’ïijŇæŻt’åd’Žä£ąæĄŕèŕůåŔĆèĂČ PEP
3118 ïijŇ åŘŇæŮű CythonæŰĞæąčäÿ■åĚşäžŐâĂIJçśżåđŃåĘĚå■ŸèğĘåŻ¿âĂİ
çŕĞäź§åĂijå¿ŮäÿĂèŕźãĂĆ

17.12 15.12 åřĘåĞ¡æŢřæŇĞéŠĹè¡ňæ■ćäÿžåŔŕèřČçŤĺåŕźèśą

éŮőécŸ

ä¡ăåůšçżŔèŐůå¿ŮäžĘäÿĂäÿłèćńçijŰèŕŚåĞ¡æŢřçŽĎåĘĚå■ŸåIJřåİĂïijŇæČşåřĘåőČè¡ňæ■ćæĹŘäÿĂäÿłP
è£Źæăůç ŽĎèŕİä¡ăåřśåŔŕäžěåřĘåőČä¡IJäÿžäÿĂäÿłæL'ľåśŢåĞ¡æŢřä¡£çŤĺäžĘãĂĆ

èğčåĘşæŰźæąĹ

ctypes æĺąåİŮåŔŕèćńçŤĺæİěåĹŻåžžåŇĚèčĚäżżæĎŔåĘĚå■ŸåIJřåİĂçŽĎPythonåŔŕèřČçŤĺåŕźèśąãĂĆ
äÿŃéİćçŽĎä¿Ńå■ŘæijŤçd'žäžĘæĂŐæăůèŐůåŔŰCåĞ¡æŢřçŽĎåŐ§ågŃãĂĄåžŢåśĆåIJřåİĂïijŇäžěåŔŁąęĆä¡

```
>>> import ctypes
>>> lib = ctypes.cdll.LoadLibrary(None)
>>> # Get the address of sin() from the C math library
>>> addr = ctypes.cast(lib.sin, ctypes.c_void_p).value
>>> addr
140735505915760

>>> # Turn the address into a callable function
>>> functype = ctypes.CFUNCTYPE(ctypes.c_double, ctypes.c_double)
>>> func = functype(addr)
>>> func
<CFunctionType object at 0x1006816d0>

>>> # Call the resulting function
>>> func(2)
0.9092974268256817
>>> func(0)
0.0
>>>
```

èőÍèőž

èęĄæđĎåžžäÿĂäÿłåŔŕèřČçŤĺåŕźèśąïijŇä¡ăéęŰåĚĹéIJĂèęĄåĹŻåžžäÿĂäÿł
CFUNCTYPE åőđä¿ŃãĂĆ CFUNCTYPE() çŽĎçňňäÿĂäÿłåŔĆæŢřæŸŕè£ŤåŻđçśžåđŃãĂĆ
æŐěäÿŃæİěçŽĎåŔĆæŢřæŸŕåŔĆæŢřçśžåđŃãĂĆäÿĂæŰęä¡ăåőŽäźL'äžĘåĞ¡æŢřçśžåđŃïijŇä¡ăåřśèČ¡åřĘåőČ
çŤ§æĹŘçŽĎåŕźèśąèćńå¡ŞåĄŽæŹőéĂŽçŽĎåŔŕéĂŽè£Ğ ctypes
èő£éŮőçŽĎåĞ¡æŢřæİěä¡£çŤĺãĂĆ

æIJňèŁĆçIJŃäÿŁåŐżåŔŕèČ¡æIJL'çĆźçěđçğŸïijŇåĄŘåžŢåśĆäÿĂçĆźãĂĆ
ä¡ĘæŸŕïijŇä¡ĘæŸŕåőČèćńáź£æşŻä¡£çŤĺäžŐåŘĎçğ■énŸçžğäžčçăĄçŤ§æĹŘæŁĂæIJŕæŕŤåęĆå■şæŮűçijŰèŕŚ

ä¿ŃåęĆïijŇäÿŃéİćæŸŕäÿĂäÿłä¡£çŤĺ llvmpy æL'ľåśŢçŽĎçőÅå■Ţä¿Ńå■ŘïijŇçŤĺæİěæđĎåžžäÿĂäÿłåřI
åźűåřĘåĚűè¡ňæ■ćäÿžäÿĂäÿłPythonåŔŕèřČçŤĺåŕźèśąãĂĆ

```
>>> from llvm.core import Module, Function, Type, Builder
>>> mod = Module.new('example')
```

```
>>> f = Function.new(mod,Type.function(Type.double(), \
                     [Type.double(), Type.double()], False), 'foo')
>>> block = f.append_basic_block('entry')
>>> builder = Builder.new(block)
>>> x2 = builder.fmul(f.args[0],f.args[0])
>>> y2 = builder.fmul(f.args[1],f.args[1])
>>> r = builder.fadd(x2,y2)
>>> builder.ret(r)
<llvm.core.Instruction object at 0x10078e990>
>>> from llvm.ee import ExecutionEngine
>>> engine = ExecutionEngine.new(mod)
>>> ptr = engine.get_pointer_to_function(f)
>>> ptr
4325863440
>>> foo = ctypes.CFUNCTYPE(ctypes.c_double, ctypes.c_double, ctypes.
↪c_double)(ptr)

>>> # Call the resulting function
>>> foo(2,3)
13.0
>>> foo(4,5)
41.0
>>> foo(1,2)
5.0
>>>
```

åźűäÿ■æŸŕèŕťåIJĺè£ŹäÿłåśĆéİćçŁŕäžĘäżżä¡ŢéŤŹèŕŕåřśäijŽåŕijèĞťPythonèğčéĞŁåŹĺæŇĆæŐĽãĂĆ èęĄèőřå¿ŮçŽĎæŸŕä¡ăæŸŕåIJĺçŻťæŐěèů§æIJžåŹĺçžğåĹńçŽĎåĘĚå■ŸåIJřåİĂåŠŇæIJňåIJřæIJžåŹĺçăĄæĽŞä

17.13 15.13 äijăéĂŠNULLçżŞåř¿çŽĎå■ŮçňęäÿšçżŹCåĞ¡æŢřåžŞ

éŮőéćŸ

ä¡ăèęĄåĘŹäÿĂäÿłæĽľåśŢæĺąåİŮïijŇéIJĂèęĄäijăéĂŠäÿĂäÿłNULLçżŞåř¿çŽĎå■ŮçňęäÿšçżŹCåĞ¡æŢřå äÿ■è£ĞïijŇä¡ăäÿ■æŸŕå¿ĹçąőåőŽæĂŐæăůä¡£çŤĺPythonçŽĎUnicodeå■ŮçňęäÿšåŐżåőđçŐřåőČãĂĆ

èğčåĘşæŰźæąĹ

èőÿåďŽCåĞ¡æŢřåžŞåŇĚåŘńäÿĂäžŻæŞ■ä¡IJNULLçżŞåř¿çŽĎå■ŮçňęäÿšïijŇèćńåčřæŸŐçśżåđŃäÿž char *. èĂČèŹŚåęĆäÿŃçŽĎCåĞ¡æŢřïijŇæĹŚäżňçŤĺæİěåĄŽæijŤçďžåŠŇæţŃèŕŢçŤĺçŽĎïijŽ

```
void print_chars(char *s) {
    while (*s) {
        printf("%2x ", (unsigned char) *s);

        s++;
    }
```

```
    printf("\n");
}
```

æ■ď åĞ¡æŢřäijŽæĽŞå■řèćńäijăè£Żæİěå■ŮçňęäÿšçŽĎæŕŔäÿłå■ŮçňęçŽĎå■ĄåĚ■è£ŻåĹűèąĺçď žïijŇè£Ź

```
print_chars("Hello");    // Outputs: 48 65 6c 6c 6f
```

åŕźäžŐåIJĺPythonäÿ■èřČçŤĺè£ŹæăůçŽĎCåĞ¡æŢřïijŇä¡ăæIJĽåĞăçğ■éĂĽæŃľãĂĆ éęŰåĚĹïijŇä¡ăåŔŕäżěéĂŽè£ĞèřČçŤĺ PyArg_ParseTuple() åźűæŇĞåőŽâĂİyâĂIJè¡ňæ■ćçăĄæİěéŹŘåĹűåőČåŔłèČ¡æŞ■ä¡IJå■ŮèŁĆïijŇåęĆäÿŃïijŽ

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  char *s;

  if (!PyArg_ParseTuple(args, "y", &s)) {
    return NULL;
  }
  print_chars(s);
  Py_RETURN_NONE;
}
```

çżŞæđIJåĞ¡æŢřçŽĎä¡£çŤĺæŰźæşŢåęĆäÿŃãĂĆäżŤçżĘèğĆåŕ§åţŇåĚěäžĘNULLå■ŮèŁĆçŽĎå■Ůçňęäÿš

```
>>> print_chars(b'Hello World')
48 65 6c 6c 6f 20 57 6f 72 6c 64
>>> print_chars(b'Hello\x00World')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be bytes without null bytes, not bytes
>>> print_chars('Hello World')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' does not support the buffer interface
>>>
```

åęĆæđIJä¡ăæČşäijăéĂŠUnicodeå■ŮçňęäÿšïijŇåIJĺ PyArg_ParseTuple() äÿ■ä¡£çŤĺâĂİsâĂIJæăijåijŔçăĄïijŇåęĆäÿŃïijŽ

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  char *s;

  if (!PyArg_ParseTuple(args, "s", &s)) {
    return NULL;
  }
  print_chars(s);
  Py_RETURN_NONE;
}
```

å¡Şèćńä¡£çŤĺçŽĎæŮűåĂŹïijŇåőČäijŽèĞłåŁĺåřĘæĽĂæIJĽå■Ůçňęäÿšè¡ňæ■ćäÿžäżěNULLçżŞåř¿çŽĎU 8çijŰçăĄãĂĆä¿ŃåęĆïijŽ

```
>>> print_chars('Hello World')
48 65 6c 6c 6f 20 57 6f 72 6c 64
>>> print_chars('Spicy Jalape\u00f1o')  # Note: UTF-8 encoding
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f
>>> print_chars('Hello\x00World')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be str without null characters, not str
>>> print_chars(b'Hello World')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be str, not bytes
>>>
```

åęĆæđIJåŻăäÿžæ§ŘäžŻåŐ§åŻăïijŇä¡ăèęĄçŻťæŐěä¡£çŤÍ PyObject * èĂŇäÿ■èČ¡ä¡£çŤÍ PyArg_ParseTuple() ïijŇ äÿŃéİćçŽĎä¿Ńå■ŘåŘŚä¡ăåśŢçďžäžĘæĂŐæăůäżŐå■ŮèŁĆåŠŇå■Ůçňęäÿšåŕźèśąäÿ■æčĂæ§ěåŠŇæŔŘåŔŰäÿ char * åijŢçŤÍïijŽ

```
/* Some Python Object (obtained somehow) */
PyObject *obj;

/* Conversion from bytes */
{
   char *s;
   s = PyBytes_AsString(o);
   if (!s) {
      return NULL;   /* TypeError already raised */
   }
   print_chars(s);
}

/* Conversion to UTF-8 bytes from a string */
{
   PyObject *bytes;
   char *s;
   if (!PyUnicode_Check(obj)) {
       PyErr_SetString(PyExc_TypeError, "Expected string");
       return NULL;
   }
   bytes = PyUnicode_AsUTF8String(obj);
   s = PyBytes_AsString(bytes);
   print_chars(s);
   Py_DECREF(bytes);
}
```

åĽ■éİćäÿďçğ■è¡ňæ■ćéČ¡åŔŕäżěçąőä£İæŸŕNULLçżŞåř¿çŽĎæŢřæ■őïijŇ ä¡ĘæŸŕåőČäżňåźűäÿ■æčĂæ§ěå■Ůçňęäÿšäÿ■éŮťæŸŕåŘęåţŇåĚěäžĘNULLå■ŮèŁĆãĂĆ åŻăæ■ďïijŇåęĆæđIJè£Źäÿłå¿ĹéĞ■èęĄçŽĎèŕİïijŇéĆčä¡ăéIJĂèęĄèĞłåůśåŐżåĄŽæčĂæ§ěäžĘãĂĆ

èőÍèőž

åęĆæđIJåŔŕèČ¡çŽĎèŕÍïijŇä¡ăåžŤèŕěéĄ£åĚ■åŐżåĘŹäÿĂäžŻä¿ÌèţŰäžŐNULLçżŞář¿çŽĎå■Ůçňęäÿšïij
æIJĂåě¡çżŞåŘĹä¡£çŤÍäÿĂäÿłæŇĞéŠĹåŠŇéŢ£åžęåĂijæÌěåď ĎçŘĘå■Ůçňęäÿšã
äÿ■è£ĞïijŇæIJL'æŮűåĂŹä¡ăå£ĚéążåŐżåď ĎçŘĘCèŕ■èÍĂéĄŮçŢŹäżčçăĄæŮűåřśæšąå¿ŮéĂL'æŃl'äžĘãĂĆ

åř¡çőąå¿Ĺåőźæ ŸŞä¡£çŤÍïijŇä¡ĘæŸŕå¿Ĺåőźæ ŸŞå£¡ègĘçŽĎäÿĂäÿłéŮőécŸæŸŕåIJÍ
`PyArg_ParseTuple()` äÿ■ä¡£çŤÍâĂIJsâĂİæăijåijŔåŇŰçăĄäijŽæIJL'åĘĚå■Ÿæ■§èĂŮãĂĆ
ä¡Ęä¡ăéIJĂèęĄä¡£çŤÍè£Źçğ■è¡ňæ■ćçŽĎæŮűåĂŹïijŇäÿĂäÿłUTF-
8å■ŮçňęäÿšècńåĹŽåžžåźűæřÿäźĚéŹĎåŁååIJÍåŐ§ågŃå■ŮçňęäÿšåŕźèśąäÿŁéÍćãĂĆ
åęĆæđIJåŐ§ågŃå■Ůçňęäÿšå ŇĚåŘńéÍđASCIIå■Ůçňę ŽĎèŕÍïijŇåŕšäijŽåŕijèĞť'å■Ůçňęäÿšç ŽĎåřžåŕÿåćđåĹřäÿ

```
>>> import sys
>>> s = 'Spicy Jalape\u00f1o'
>>> sys.getsizeof(s)
87
>>> print_chars(s)      # Passing string
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f
>>> sys.getsizeof(s)    # Notice increased size
103
>>>
```

åęĆæđIJä¡ăåIJÍäźŐè£ŹäÿłåĘĚå■ŸçŽĎæ■§èĂŮïijŇä¡ăæIJĂåě¡éĞ■åĘŹä¡ăçŽĎCæL'l'åśŢäžčçăĄïijŇèŐl'å
`PyUnicode_AsUTF8String()` åĞ¡æŢřãĂĆåęĆäÿŃïijŽ

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  PyObject *o, *bytes;
  char *s;

  if (!PyArg_ParseTuple(args, "U", &o)) {
    return NULL;
  }
  bytes = PyUnicode_AsUTF8String(o);
  s = PyBytes_AsString(bytes);
  print_chars(s);
  Py_DECREF(bytes);
  Py_RETURN_NONE;
}
```

éĂŽè£Ğè£ŹäÿŀäŁőæŤźïijŇäÿĂäÿłUTF-8çijŰçăĄçŽĎå■Ůçňęäÿšæăźæ■őéIJĂèęĄèćńåĹŽåžžïijŇçĎűåŘĈ

```
>>> import sys
>>> s = 'Spicy Jalape\u00f1o'
>>> sys.getsizeof(s)
87
>>> print_chars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f
>>> sys.getsizeof(s)
87
>>>
```

åęĆædIJä¡ăèŕŢçİĂäijăéĂŠNULLçżŞář¿å■ŮçňęäÿšçżŹctypesåŇĚèčĚè£ĞçŽĎåĞ¡æŢřïijŇ
èęĄæşĺæĎŔçŽĎæŸŕctypesåŔłèČ¡åĚĄèőÿäijăéĂŠå■ŮèŁĆïijŇåźűäÿŤåőČäÿ■äijŽæčĂæ§ěäÿ■éŮť åţŇåĚěçŽĎ

```
>>> import ctypes
>>> lib = ctypes.cdll.LoadLibrary("./libsample.so")
>>> print_chars = lib.print_chars
>>> print_chars.argtypes = (ctypes.c_char_p,)
>>> print_chars(b'Hello World')
48 65 6c 6c 6f 20 57 6f 72 6c 64
>>> print_chars(b'Hello\x00World')
48 65 6c 6c 6f
>>> print_chars('Hello World')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ctypes.ArgumentError: argument 1: <class 'TypeError'>: wrong type
>>>
```

åęĆædIJä¡ăæČşäijăéĂŠå■ŮçňęäÿšèĂŇäÿ■æŸŕå■ŮèŁĆïijŇä¡ăéIJĂèęĄåĚĹæĽğèąŇæĽŃåŁĺçŽĎUTF-8çijŰçăĄãĂĆä¿ŃåęĆïijŽ

```
>>> print_chars('Hello World'.encode('utf-8'))
48 65 6c 6c 6f 20 57 6f 72 6c 64
>>>
```

åŕźäžŐåĚűäżŰæĽľåśŢåůěåĚůïijĹæŕŤåęĆSwigãĂĄCythonïijĽïijŇ
åIJĺä¡ää¡£çŤĺåőČäžňäijăéĂŠå■ŮçňęäÿšçżŹCäžčçăĄæŮűèęĄåĚĹåě¡åě¡å■ęäźăçŻÿåžŢçŽĎäÿIJèě£äžEãĂĆ

17.14 15.14 äijăéĂŠUnicodeå■ŮçňęäÿšçżŹCåĞ¡æŢřåžŞ

éŮőécŸ

ä¡ăèęĄåĘŹäÿĂäÿłæĽľåśŢæĹąåİŮïijŇéIJĂèęĄåřĘäÿĂäÿłPythonå■ŮçňęäÿšäijăéĂŠçżŹCçŽĎæ§ŘäÿłåžŞ

èğčåĘşæŰźæąĹ

è£ŹéĞŇæĹŚäžňéIJĂèęĄèĂČèŹŚå¿Ĺåď ŽçŽĎéŮőécŸïijŇä¡ĘæŸŕæIJĂäÿżèęĄçŽĎéŮőécŸæŸŕçŐřå■Ÿç
åŻăæ■ď ïijŇä¡ăçŽĎæŇŚæĹŸæŸŕåřĘPythonå■Ůçňęäÿšè¡ňæ■ćäÿžäÿĂäÿłèČ¡èćńCçŘĘèğčçŽĎå¡ćåijŔãĂĆ

äÿžäžĘæijŤçď žçŽĎçŻőçŽĎïijŇäÿŃéİćæIJĽäÿď äÿłCåĞ¡æŢřïijŇçŤĺæİěæŞ■ä¡IJå■ŮçňęäÿšæŢřæ■őåźűè
äÿĂäÿłä¡£çŤĺå¡ćåijŔäÿž `char *, int` å¡ćåijŔçŽĎå■ŮèŁĆïijŇ
èĂŇåŘęäÿĂäÿłä¡£çŤĺå¡ćåijŔäÿž `wchar_t *, int` çŽĎåő¡å■Ůçňęå¡ćåijŔïijŽ

```
void print_chars(char *s, int len) {
  int n = 0;

  while (n < len) {
    printf("%2x ", (unsigned char) s[n]);
    n++;
```

```
  }
  printf("\n");
}

void print_wchars(wchar_t *s, int len) {
  int n = 0;
  while (n < len) {
    printf("%x ", s[n]);
    n++;
  }
  printf("\n");
}
```

åŕźäžŐéİćåŘŚå■ÛèŁĆçŽĎåĞ¡æŢř `print_chars()` ïijŇä¡ăéIJĂèęĄåřĘPythonå■Ůçňęäÿšè¡ňæ■ćäÿžäy
8. äÿŃéİćæŸŕäÿĂäÿłè£Źæăůç ŽĎæĽ'ĺåśŢåĞ¡æŢřä¿Ńå■ŘïijŽ

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  char *s;
  Py_ssize_t  len;

  if (!PyArg_ParseTuple(args, "s#", &s, &len)) {
    return NULL;
  }
  print_chars(s, len);
  Py_RETURN_NONE;
}
```

åŕźäžŐéĆčäžŻéIJĂèęĄåd'ĎçŘĘæIJžåŹĺæIJňåIJř `wchar_t` çśżåđŃçŽĎåžŞåĞ¡æŢřïijŇä¡ăåŔŕäżěåČŔäÿŃéİćè£Źæăůçij ŰåĘŹæĽ'ĺåśŢäžčçăĄïijŽ

```
static PyObject *py_print_wchars(PyObject *self, PyObject *args) {
  wchar_t *s;
  Py_ssize_t  len;

  if (!PyArg_ParseTuple(args, "u#", &s, &len)) {
    return NULL;
  }
  print_wchars(s,len);
  Py_RETURN_NONE;
}
```

äÿŃéİćæŸŕäÿĂäÿłäžd'äžŠäijŽèŕİæİěæijŤçd'žè£ŹäÿłåĞ¡æŢřæŸŕåęĆä¡Ţåůěä¡IJçŽĎïijŽ

```
>>> s = 'Spicy Jalape\u00f1o'
>>> print_chars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f
>>> print_wchars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 f1 6f
>>>
```

äžŤçżĘèğĆåŕ§è£ŹäÿłéİćåŘŚå■ÛèŁĆçŽĎåĞ¡æŢř `print_chars()`

æŸŕæĂŐæăůæŐěåŔŮUTF-8çijŰçăĄæŢřæ■őçŽĎïijŇ äžěåŔŁ `print_wchars()` æŸŕæĂŐæăůæŐěåŔŮUnicodeçijŰçăĄåĂijçŽĎ

èőĺèőž

åIJĺçżğçż■æIJňèŁĆäźŃåL'■ïijŇä¡ăåžŤèŕěéęŰåĚĹå■ęäźăä¡ăèő£éŮőçŽĎCåĞ¡æŢřåžŞçŽĎçL'źå¿ĄãĂĆ åŕźäžŐå¿Ĺåď'ŽCåĞ¡æŢřåžŞïijŇéĂŽåÿÿäijăéĂŠå■ŮèŁĆèĂŇäÿ■æŸŕå■ŮçňęäÿšäijŽæŕŤè¿Čåě¡äžŻãĂĆèęĄè£

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  char *s;
  Py_ssize_t  len;

  /* accepts bytes, bytearray, or other byte-like object */
  if (!PyArg_ParseTuple(args, "y#", &s, &len)) {
    return NULL;
  }
  print_chars(s, len);
  Py_RETURN_NONE;
}
```

åęĆæđIJä¡ăäż■çĎűè£ŸæŸŕæČşèęĄäijăéĂŠå■ŮçňęäÿšïijŇ ä¡ăéIJĂèęĄç§ěéĄŞPython 3åŔŕä¡£çŤĺäÿĂäÿłåŘĹéĂĆçŽĎå■Ůçňęäÿšèąĺçd'žïijŇ åőČåźűäÿ■çŻť'æŐěæŸăářĎåĹřä¡£çŤĺæăĞåĞĘçśžåđŃ `char *` æĹŰ `wchar_t *` ïijĹæŻť'åď'ŽçżĘèŁĆåŔĆèĂČPEP 393ïijL'çŽĎCåĞ¡æŢřåžŞãĂĆ åŻăæ■ď'ïijŇèęĄåIJĺCäÿ■èąĺçď'žè£Źäÿłå■ŮçňęäÿšæŢřæ■őïijŇäÿĂäžŻè¡ňæ■ćè£ŸæŸŕå£Éęąžèę ĄçŽĎãĂĆ åIJĺ `PyArg_ParseTuple()` äÿ■ä¡£çŤĺâĂİs#âĂİ åŠŇâĂİu#âĂİæăijåijŔåŇŰçăĄåŔŕäžěåőL'åĚĺçŽĎæL'ğèą

äÿ■è£Ğè£Źçğ■è¡ňæ■ćæIJL'äÿłçijžçĆźåřśæŸŕåőČåŔŕèČ¡äijŽåŕijèĞť'åŐ§ăğŇå■ŮçňęäÿšåŕźèśąçŽĎåřžåŕý äÿĂæŮęè¡ňæ■ćè£ĞåŘŐïijŇäijŽæIJL'äÿĂäÿłè¡ňæ■ćæŢřæ■őçŽĎåď'■åĹűéŹĎåŁăåĹřåŐ§ăğŇå■Ůçňęäÿšåŕźèśą ä¡ăåŔŕäžěèğĆåŕ§äÿŇè£Źçğ■æŢĹæđIJïijŽ

```
>>> import sys
>>> s = 'Spicy Jalape\u00f1o'
>>> sys.getsizeof(s)
87
>>> print_chars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f
>>> sys.getsizeof(s)
103
>>> print_wchars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 f1 6f
>>> sys.getsizeof(s)
163
>>>
```

åŕźäžŐåřŠéĞŔçŽĎå■ŮçňęäÿšåŕźèśąïijŇåŔŕèČ¡æšąäżĂäźĹå¡śåŞ■ïijŇ ä¡ĘæŸŕåęĆæđIJä¡ăéIJĂèęĄåIJĺæL'l'åśŢäÿ■åď'ĎçŘĘåď'ğéĞŔçŽĎæŰĞæIJňïijŇä¡ăåŔŕèČ¡æČşéĄ£åĚ■è£ŹäÿłäÿŃéĺćæŸŕäÿĂäÿłä£őèőćçL'ĹæIJňåŔŕäžěéĄ£åĚ■è£Źçğ■åĘĚå■Ÿæ■§èĂŮïijŽ

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  PyObject *obj, *bytes;
  char *s;
  Py_ssize_t   len;

  if (!PyArg_ParseTuple(args, "U", &obj)) {
    return NULL;
  }
  bytes = PyUnicode_AsUTF8String(obj);
  PyBytes_AsStringAndSize(bytes, &s, &len);
  print_chars(s, len);
  Py_DECREF(bytes);
  Py_RETURN_NONE;
}
```

èĂŇåŕź wchar_t çŽĎåď ĎçŘĘæŮűæČşèęĄéĄ£åĚ■åĘĚå■Ÿæ■§èĂŮåřśæŻť åŁăéŽ¿åŁđäžĘãĂĆ
åIJĺåĘĚéČĺïijŇPythonä¡£çŤĺæIJĂéńŸæŢĹçŽĎèąĺçď žæİěå■ŸåĆĺå■Ůçňęäÿšã ĂĆ
ä¿ŃåęĆïijŇåŔłåŇĚåŘńASCIIçŽĎå■Ůçňęäÿšèćńå■ŸåĆĺäÿžå■ŮèŁĆæŢřçżĎïijŇ
èĂŇåŇĚåŘńèŇČåŻť äżŐU+0000åĹřU+FFFFçŽĎå■ŮçňęçŽĎå■Ůçňęäÿšä¡£çŤĺåŔŇå■ŮèŁĆèąĺçď žãĂĆ
çŤśäžŐåŕźäžŐæŢřæ■őçŽĎèąĺçď žå¡ćåijŔäÿ■æŸŕå■ŢäÿĂçŽĎïijŇä¡ăäÿ■èČ¡åřĘåĘĚéČĺæŢřçżĎè¡ňæ■ćäÿž
wchar_t * çĎűåŘŐæIJ§æIJŻåőČèČ¡æ■čçąőçŽĎåůěä¡IJãĂĆ ä¡ăåžŤèŕěåĹŻåżžäÿĂäÿł
wchar_t æŢřçżĎåźűåŘŚåĚűäÿ■åď ■åĹűæŰĞæIJñãĂĆ PyArg_ParseTuple()
çŽĎâĂİu#âĂİæăijåijŘçăĄåŔŕäžěåÿőåŁľä¡ăénŸæŢĹçŽĎåőŇæĹŘåőČïijĹåőČåřĘåď ■åĹűçżŞæđIJéŹĎåŁăåĹi

åęĆæđIJä¡ăæČşéĄ£åĚ■éŢ£æŮűéŮťåĘĚå■Ÿæ■§èĂŮïijŇä¡ăåŤŕäÿĂçŽĎéĂĽ'æŃľ åřśæŸŕåď ■åĹűUnicoc
åřĘåőČäijăéĂŠçżŹCåĞ¡æŢřïijŇçĎűåŘŐåŻđæŤűè£ŹäÿłæŢřçżĎçŽĎåĘĚå■ŸãĂĆäÿŃéİćæŸŕäÿĂäÿłåŔŕèČ¡çŽ

```
static PyObject *py_print_wchars(PyObject *self, PyObject *args) {
  PyObject *obj;
  wchar_t *s;
  Py_ssize_t len;

  if (!PyArg_ParseTuple(args, "U", &obj)) {
    return NULL;
  }
  if ((s = PyUnicode_AsWideCharString(obj, &len)) == NULL) {
    return NULL;
  }
  print_wchars(s, len);
  PyMem_Free(s);
  Py_RETURN_NONE;
}
```

åIJĺè£ŹäÿłåőđçŐřäÿ■ïijŇPyUnicode_AsWideCharString()
åĹŻåżžäÿĂäÿłäÿťæŮűçŽĎwchar_tçijŞåĘšåźűåď ■åĹűæŢřæ■őè£ŻåŐżãĂĆ
è£Źäÿłçij ŞåĕšèćńäijăéĂŠçżŹCçĎűåŘŐèćńéĞŁæŤ¿æŐĽ'ãĂĆ ä¡ĘæŸŕæĹŚåĘŹè£ŹæIJñäźęçŽĎæŮűåĂŹïijŇè

åęĆæđIJä¡ăç§ěéĄŞCåĞ¡æŢřåžŞéIJĂèęĄçŽĎå■ŮèŁĆçijŰçăĄåźűäÿ■æŸŕUTF-8ïijŇ
ä¡ăåŔŕäžěåijžåĹűPythonä¡£çŤĺæĽ'ĺåśŢçăĄæİěæĽ'ğèąŇæ■čçąőçŽĎè¡ňæ■ćïijŇåŕśåČŔäÿŃéİćè£Źæăůïijž

```
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  char *s = 0;
  int   len;
  if (!PyArg_ParseTuple(args, "es#", "encoding-name", &s, &len)) {
    return NULL;
  }
  print_chars(s, len);
  PyMem_Free(s);
  Py_RETURN_NONE;
}
```

æIJĂåŘŐïijŇåęĆæđIJä¡ăæČşçŻťæŐěåďĎçŘĘUnicodeå■ŮçňęäÿšïijŇäÿŃéİćçŽĎæŸŕä¿Ńå■ŘïijŇæijŤç

```
static PyObject *py_print_wchars(PyObject *self, PyObject *args) {
  PyObject *obj;
  int n, len;
  int kind;
  void *data;

  if (!PyArg_ParseTuple(args, "U", &obj)) {
    return NULL;
  }
  if (PyUnicode_READY(obj) < 0) {
    return NULL;
  }

  len = PyUnicode_GET_LENGTH(obj);
  kind = PyUnicode_KIND(obj);
  data = PyUnicode_DATA(obj);

  for (n = 0; n < len; n++) {
    Py_UCS4 ch = PyUnicode_READ(kind, data, n);
    printf("%x ", ch);
  }
  printf("\n");
  Py_RETURN_NONE;
}
```

åIJÍè£ŹäÿłäžčçăĄäÿ■ïijŇPyUnicode_KIND() åŠŇ PyUnicode_DATA() è£ŹäÿďäÿłåőŘåŠŇUnicodeçŽĎåŕŕåŔŸåő¡åžęå■ŸåĆÍæIJL'åĚşïijŇè£ŹäÿłåIJÍPEP 393äÿ■æIJL'æŔŔè£řãĂĆ kind åŔŸéĞŔçijŰçăĄåžŢåśĆå■ŸåĆÍïijĹ8ä¡■ãĂĄ16ä¡■æĹŰ32ä¡■ïijL'äżěåŘŁæŇ åIJÍåőđéŹĚæČĚåĘţäÿ■ïijŇä¡ăåźűäÿ■éIJĂèęĄç§ěéĄŞäżżä¡Ţèů§è£ŹäžŻåĂijæIJL'åĚşçŽĎäÿIJèě£ïijŇ åŔłéIJĂèęĄåIJÍæŔŘåŔŰå■ŮçňęçŽĎæŮűåĂŹårĘåőČäžňäijăçżŹ PyUnicode_READ() åőŔãĂĆ

è£ŸæIJL'æIJĂåŘŐåĞăåŔěïijŽå¡ŞäżŐPythonäijăéĂŠUnicodeå■ŮçňęäÿšçżŹCçŽĎæŮűåĂŹïijŇä¡ăåžŤèŕě åęĆæđIJæIJL'UTF-8åŠŇåő¡å■Ůçňęäÿď'çğ■éĂL'æŃl'ïijŇèŕůéĂL'æŃl'UTF-8. åŕźUTF-8çŽĎæŤŕæŇĄæŻť'åŁăæŹőéĄ■äÿĂäžŻïijŇäź§äÿ■åőźæŸŞçŁŕéŤŹïijŇèğčéĞŁåŹĺäź§èČ¡æŤŕæŇĄçŽĎæŻť'åě æIJĂåŘŐïijŇçąőä£Ía¡ăäžŤçżĘéŸĚèŕżäžĘ åĚşäžŐåď'ĎçŘĘUnicodeçŽĎçŻÿåĚşæŰĞæąč

17.15 15.15 Cå■Ůçňęäÿšè¡ňæ■ćäÿžPythonå■Ůçňęäÿš

éŮőécŸ

æĂŐæăůåřĘCäÿ■çŽĎå■Ůçňęäÿšè¡ňæ■ćäÿžPythonå■ŮèŁĆæĹŰäÿĂäÿłå■Ůçňęäÿšåŕźèśąïij§

èğčåĘşæŰźæąĹ

Cå■Ůçňęäÿšä¡£çŤĺäÿĂåŕź `char *` åŠŇ `int` æİěèąĺçd'žïijŇ ä¡ăéIJĂèęĄåEşåőŽå■Ůçňęäÿšå£řåžŢæŸŕçŤĺäÿĂäÿłåŐ§ågŃå■ŮèŁĆå■Ůçňęäÿšè£ŸæŸŕäÿĂäÿłUnicodeå■Ůç å■ŮèŁĆåŕźèśąåŔŕäžěåČŔäÿŃéİćè£ŹæăůäĄ£çŤĺ `Py_BuildValue()` æİěædĎåżžïijŽ

```
char *s;      /* Pointer to C string data */
int   len;    /* Length of data */

/* Make a bytes object */
PyObject *obj = Py_BuildValue("y#", s, len);
```

åęĆædIJä¡ăèęĄåĹŻåżžäÿĂäÿłUnicodeå■ŮçňęäÿšïijŇåźűäÿŤä¡ăç§ěéĄŞ `s` æŇĞåŘŚäžĘUTF-8çijŰçăĄçŽĎæŢřæ■őïijŇåŔŕäžěä¡£çŤĺäÿŃéİćçŽĎæŰźåijŔïijŽ

```
PyObject *obj = Py_BuildValue("s#", s, len);
```

åęĆædIJ `s` ä¡£çŤĺåĚűäžŰçijŰçăĄæŰźåijŔïijŇéĆčäźĹåŔŕäžěåČŔäÿŃéİćä¡£çŤĺ `PyUnicode_Decode()` æİěædĎåżžäÿĂäÿłå■ŮçňęäÿšïijŽ

```
PyObject *obj = PyUnicode_Decode(s, len, "encoding", "errors");

/* Examples /*
obj = PyUnicode_Decode(s, len, "latin-1", "strict");
obj = PyUnicode_Decode(s, len, "ascii", "ignore");
```

åęĆædIJä¡ăæĄřåě¡æIJL'äÿĂäÿłçŤĺ `wchar_t *, len` åŕźèąĺçd'žçŽĎåő¡å■ŮçňęäÿšïijŇ æIJL'åĞăçğ■éĂL'æŃl'æĂğãĂĆéęŰåĚĹä¡ăåŔŕäžěä¡£çŤĺ `Py_BuildValue()` ïijŽ

```
wchar_t *w;     /* Wide character string */
int len;        /* Length */

PyObject *obj = Py_BuildValue("u#", w, len);
```

åŔęåd'ŰïijŇä¡ăè£ŸåŔŕäžěä¡£çŤĺ `PyUnicode_FromWideChar()`:

```
PyObject *obj = PyUnicode_FromWideChar(w, len);
```

åŕźäžŐåő¡å■ŮçňęäÿšïijŇåźűæšąæIJL'åŕźå■ŮçňęæŢřæ■őè£ŻèąŇèğčædŘâĂŤâĂŤåőČèćńåĄĞåőŽæŸŕåŐ§

èőÍèőž

åřĘCäÿ■çŽĎå■Ůçňęäÿšè¡ňæ■ćäÿžPythonå■ŮçňęäÿšéĄţå¿łåŠŇI/OåŘŇæăůçŽĎåŐ§åĹŹãĂĆ äźš§åřśæŸŕèřťˇïijŇæÍèèĞłCäÿ■çŽĎæŢřæ■őå£Ěéążæăźæ■őäÿĂäžŻèğčçăĄåŹÍèćńæŸ¿åijŔçŽĎèğčçăĄäÿžäÿĂä éĂŽåÿÿçijŰçăĄæăijåijŔåŇĚæŃňASCIIãĂĄLatin-1åŠŇUTF-8. åęĆæđIJä¡ääźűäÿ■çąőåőŽçijŰçăĄæŰźåijŔæĹŰèĂĚæŢřæ■őæŸŕäžŇè£ŻåĹűçŽĎïijŇä¡ăæIJĂåě¡åřĘå■Ůçňęäÿ å¡ŞæđĎéĂăäÿĂäÿłåŕźèšąçŽĎæŮűåĂŹïijŇPythonéĂŽåÿÿäijŽåďˇ■åĹűä¡ăæŔŘä¿ŻçŽĎå■ŮçňęäÿšæŢřæ■őãĂĆ åęĆæđIJæIJĽå£ĚèęĄçŽĎèŕÍïijŇä¡ăéIJĂèęĄåIJĺåŘŐéÍćåŐžéĞŁæŤ¿Cå■Ůçňęäÿšã ĂĆ åŘŇæŮűïijŇäÿžäžĘèőľçĺŃåžŔæŻťˇåŁäåĄěåčőïijŇä¡ääžŤèŕěåŘŇæŮűä¡£çŤĺäÿĂäÿłæŇĞéŠĹåŠŇäÿĂäÿłåďˇğ èĂŇäÿ■æŸŕä¿ÍètŰNULLçżŞår¿æŢřæ■őæÍěåĹŻäžžå■Ůçňęäÿšã ĂĆ

17.16 15.16 äÿ■çąőåőŽçijŰçăĄæăijåijŔçŽĎCå■Ůçňęäÿš

éŮőécŸ

ä¡ăèęĄåIJĺCåŠŇPythonçŻťˇæŐěæÍěåŻđè¡ňæ■ćå■ŮçňęäÿšïijŇä¡ĘæŸŕCäÿ■çŽĎçijŰçăĄæăijåijŔåźűäÿ■çă ä¿ŇåęĆïijŇåŘŕèČ¡Cäÿ■çŽĎæŢřæ■őæIJ§æIJŻæŸŕUTF-8ïijŇä¡ĘæŸŕåźűæšąæIJĽåijžåĹűåőCå£Ěéążæ ŸŕãĂĆ ä¡ăæČşçijŰåĘŹäžčçăĄæÍěäžěäÿĂçğ■äijŸéŻĚçŽĎæŰźåijŔåďˇĎçŘĘè£ŹäžŻäÿ■åŘĹæăijæŢřæ■őïijŇè£ŹæăůåĄ

èğčåĘşæŰźæąĹ

äÿŃéÍćæŸŕäÿĂäžŻCçŽĎæŢřæ■őåŠŇäÿĂäÿłåĞ¡æŢřæÍěæijŤçďˇžè£ŹäÿłéŮőécŸïijŽ

```
/* Some dubious string data (malformed UTF-8) */
const char *sdata = "Spicy Jalape\xc3\xb1o\xae";
int slen = 16;

/* Output character data */
void print_chars(char *s, int len) {
  int n = 0;
  while (n < len) {
    printf("%2x ", (unsigned char) s[n]);
    n++;
  }
  printf("\n");
}
```

åIJÍè£ŹäÿłäžčçăĄäÿ■ïijŇå■Ůçňęäÿš `sdata` åŇĚåŘńäžĘUTF-8åŠŇäÿ■åŘĹæăijæŢřæ■őãĂĆ äÿ■è£ĞïijŇåęĆæđIJçŤÍæĹůåIJÍCäÿ■èřĈçŤÍ `print_chars(sdata, slen)` ïijŇåőČçijžèČ¡æ■čåÿÿåůëä¡IJãĂĆ çŐřåIJĺåĄĞèő¿ä¡ăæČşåřĘ `sdata` çŽĎåĘĚåőźè¡ňæ■ćäÿžäÿĂäÿłPythonå■Ůçňęäÿšã ĂĆ è£ŻäÿĂæ■ěåĄĞèő¿ä¡ääIJĺåŘŐéÍćè£ŸæČşéĂŽè£ĞäÿĂäÿłæĽľåśŢåřĘéĆčäÿłå■Ůçňęäÿšäijääÿł `print_chars()` åĞ¡æŢřãĂĆ äÿŃéÍćæŸŕäÿĂçğ■çŤÍæÍěä£ÍæŁďˇåŐ§ågŃæŢřæ■őçŽĎæŰźæşŢïijŇåřśçőŮåč

```
/* Return the C string back to Python */
static PyObject *py_retstr(PyObject *self, PyObject *args) {
  if (!PyArg_ParseTuple(args, "")) {
```

```
    return NULL;
  }
  return PyUnicode_Decode(sdata, slen, "utf-8", "surrogateescape");
}

/* Wrapper for the print_chars() function */
static PyObject *py_print_chars(PyObject *self, PyObject *args) {
  PyObject *obj, *bytes;
  char *s = 0;
  Py_ssize_t   len;

  if (!PyArg_ParseTuple(args, "U", &obj)) {
    return NULL;
  }

  if ((bytes = PyUnicode_AsEncodedString(obj,"utf-8",
↪"surrogateescape"))
        == NULL) {
    return NULL;
  }
  PyBytes_AsStringAndSize(bytes, &s, &len);
  print_chars(s, len);
  Py_DECREF(bytes);
  Py_RETURN_NONE;
}
```

åęĆædIJä¡ăåIJÍPythonäÿ■åřÌèřŢè£ŹäžŻåĞ¡æŢřijŇäÿŃéİćæŸŕè£ŘèąŇæŢĹædIJïijŽ

```
>>> s = retstr()
>>> s
'Spicy JalapeÃśo\udcae'
>>> print_chars(s)
53 70 69 63 79 20 4a 61 6c 61 70 65 c3 b1 6f ae
>>>
```

äžŤçżĘèğĆåŕ§çżŞædIJä¡ăäijŽåŔŞçŐřïijŇäÿ■åŘĹæăijå■ŮçňęäÿšèćńçijŰçăĄåĹřäÿĂäÿłPythonå■Ůçňęäÿ
åźűäÿŤå¡ŞåőčèćńåžđäijäçżŹCçŽĎæŮűåĂŹïijŇèćńè¡ňæ■ćäÿžåŠŇäźŃåĽ■åŐ§ågŃCå■Ůçňęäÿšäÿ Äæăůç ŽĎå

èőÍèőž

æIJňèŁĆåśŢçd'žäžĘåIJÍæĽl'åśŢæÍąåİŮäÿ■åd'ĎçŘĘå■Ůçňęäÿšæ Ůűäij ŽéĚ■åĹřçŽĎäÿĂäÿłæčŸæĽŃåŔÍ
äź§åřśæŸŕèŕť'ïijŇåIJÍæĽl'åśŢäÿ■çŽĎCå■ŮçňęäÿšåŕŕèČ¡äÿ■äijŽäÿěæäijéĄţå¿łPythonæĽĂæIJ§æIJŻçŽĎUn
åŻăæ■d'ïijŇå¿ĹåŔŕèČ¡äÿĂäžŽäÿ■åŘĹæäijCæŢřæ■őäijăéĂŠåĹřPythonäÿ■åŐżãĂĆ
äÿĂäÿłå¿Ĺåě¡çŽĎä¿Ńå■Řåřśæ ŸŕæűĽåŘŁåĹřåžŢåśĆçşżçż§èřČçŤĺæŕŤåęĆæŰĞäžűåŘ■è£Źæăůç ŽĎå■Ůçňęä
ä¿ŃåęĆïijŇåęĆædIJäÿĂäÿłçşżçż§èřČçŤĺè£ŤåŻđçżŹèğčéĞŁåŹĺäÿĂäÿłæ■§åİŔçŽĎå■ŮçňęäÿšïijŇäÿ■èČ¡èćńa

äÿĂèĹňæİěèőšïijŇåŔŕäżěéĂŽè£ĞåĹűåőŽäÿĂäžŽéŤŹèŕŕç■ŰçŢěæŕŤåęĆäÿěæäijãĂĄå£¡çŢěãĂĄæŽ£äżč
äÿ■è£ĞïijŇè£ŹäžŽç■ŰçŢěçŽĎäÿĂäÿłçijžçĆźæŸŕåőČäżňæřÿäźĚæĂğçăť'åİŔäžĘåŐ§ågŃå■Ůçňęäÿšç ŽĎåĘĚ
ä¿ŃåęĆïijŇåęĆædIJä¿Ńå■Řäÿ■çŽĎäÿ■åŘĹæäijæŢřæ■őä¡£çŤĺè£ŹäžŽç■ŰçŢěäźŃäÿĂèğčçĂąïijŇä¡ăäijŽå¿Ů

```
>>> raw = b'Spicy Jalape\xc3\xb1o\xae'
>>> raw.decode('utf-8','ignore')
'Spicy JalapeÃńo'
>>> raw.decode('utf-8','replace')
'Spicy JalapeÃńo?'
>>>
```

surrogateescape éŤŹèŕŕåď'ĎçŘĘç■ŰçŢěäijŽářĘæL'ĂæIJL'äÿ■åŔŕèğčçăĄå■ŮèŁĆè¡ňåŇŰäÿžäÿĂä¿ŃåęĆïijŽ

```
>>> raw.decode('utf-8','surrogateescape')
'Spicy JalapeÃńo\udcae'
>>>
```

å■ŢçŃňçŽĎä¡Őä¡■äžčçŘĘå■Ůçňęæŕ'ŤåęĆ \udcae åIJÍUnicodeäÿ■æŸŕéİđæşŢçŽĎãĂĆ åŻăæ■ď'ïijŇè£Źäÿłå■ŮçňęäÿšářśæŸŕäÿĂäÿłéİđæşŢèąĺçď'žãĂĆ åőđéŹĚäÿŁïijŇåęĆæđIJä¡ăářĘåőČäijăäÿłäÿĂäÿłæL'ğèąŃè¿ŞåĞžçŽĎåĞ¡æŢřïijŇä¡ăäijŽå¿ŮåĹřäÿĂäÿłéŤŹèŕŕ

```
>>> s = raw.decode('utf-8', 'surrogateescape')
>>> print(s)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
UnicodeEncodeError: 'utf-8' codec can't encode character '\udcae'
in position 14: surrogates not allowed
>>>
```

çĎűèĂŇïijŇåĚĄèőÿäžčçŘĘè¡ňæ■ćçŽĎåĚşéŤőçĆźåIJÍäžŐäžŐCäijăçżŹPythonåŔĹåŻđäijăçżŹCçŽĎäÿ■å¡Şè£ŹäÿłåŮçňęäÿšåE■æňąä¡£çŤÍ surrogateescape çijŰçăĄæŮűïijŇäžčçŘĘå■ŮçňęäijŽè¡ňæ■ćåŻđåŐ

```
>>> s
'Spicy JalapeÃńo\udcae'
>>> s.encode('utf-8','surrogateescape')
b'Spicy Jalape\xc3\xb1o\xae'
>>>
```

ä¡IJäÿžäÿĂèĹňåĞĘåĹŹïijŇæIJĂåě¡éĄ£åĚ■äžčçŘĘçijŰçăĄâĂŤâĂŤåęĆæđIJä¡ăæ■čçąőçŽĎä¡£çŤÍäžĘçij äÿ■è£ĞïijŇæIJL'æŮűåĂŹçąőåőđäijŽåĞžçŐřä¡ăåźűäÿ■èČ¡æŐğåĹűæŢřæ■őçijŰçăĄåźűäÿŤä¡ăåŔĹäÿ■èČ¡å£¡ç éĆčäźĹářśåŔŕäžěä¡£çŤÍæIJňèŁĆçŽĎæŁĂæIJŕäžĘãĂĆ

æIJĂåŘŐäÿĂçĆźèęĄæşÍæĎŘçŽĎæŸŕïijŇPythonäÿ■èőÿåď'ŽéİćåŘŚçşżçż§çŽĎåĞ¡æŢřïijŇçL'źåĹńæŸŕå éČ¡äijŽä¡£çŤÍäžčçŘĘçijŰçăĄãĂĆä¿ŃåęĆæđIJä¡ăä¡£çŤÍåČŘ os.listdir() è£ŹæăůçŽĎåĞ¡æŢřïijŇ äijååĚěäÿĂäÿłåŇĚåŔńäžĘäÿ■åŔŕèğčçăĄæŰĞäžűåŘ■çŽĎçŻőå¡ŢçŽĎèŕİïijŇåőČäijŽ åŔĆèĂČ5.15çŽĎçŻÿåĚşçńäèŁĆãĂĆ

PEP 383 äÿ■æIJL'æŻť'åď'ŽåĚşäžŐæIJňæIJžæŔŘåĹřçŽĎäžěåŔŁåŠŇsurrogateescapeéŤŹèŕŕåď'ĎçŘĘçŻÿåĚşçŽĎä£ąæĄŕãĂĆ

17.17 15.17 äijăéĂŠæŰĞäžűåŘ■çżŹCæĽľåśŢ

éŮőécŸ

ä¡ăéIJĂèęĄåŘŠCåžŞåĞ¡æŢřäijăéĂŠæŰĞäžűåŘ■ïijŇä¡ĘæŸŕéIJĂèęAçąőä£ÍæŰĞäžűåŘ■æăźæ■őçşżçż§a

èğčåĘşæŰźæąĹ

åĘŹäÿĂäÿłæŐěåŔŮäÿĂäÿłæŰĞäžűåŘ■äÿžåŔĆæŢřçŽĎæĽľåśŢåĞ¡æŢřïijŇåęĆäÿŃè£ŹæăůïijŽ

```
static PyObject *py_get_filename(PyObject *self, PyObject *args) {
  PyObject *bytes;
  char *filename;
  Py_ssize_t len;
  if (!PyArg_ParseTuple(args,"O&", PyUnicode_FSConverter, &bytes)) {
    return NULL;
  }
  PyBytes_AsStringAndSize(bytes, &filename, &len);
  /* Use filename */
  ...

  /* Cleanup and return */
  Py_DECREF(bytes)
  Py_RETURN_NONE;
}
```

åęĆædIJä¡ăåůšçżŔæIJL'äžĘäÿĂäÿł`PyObject *` ïijŇåÿŇæIJŻåřĘåĚűè¡ñæ■ćæĹŘäÿĂäÿłæŰĞäžűåŘ■ï

```
PyObject *obj;    /* Object with the filename */
PyObject *bytes;
char *filename;
Py_ssize_t len;

bytes = PyUnicode_EncodeFSDefault(obj);
PyBytes_AsStringAndSize(bytes, &filename, &len);
/* Use filename */
...

/* Cleanup */
Py_DECREF(bytes);

If you need to return a filename back to Python, use the following␣
↪code:

/* Turn a filename into a Python object */

char *filename;       /* Already set */
int   filename_len;   /* Already set */
```

```
PyObject *obj = PyUnicode_DecodeFSDefaultAndSize(filename, filename_
↪len);
```

èőÍèőž

äžěåŘŕçğžæď■æŰźåijŔæÍěåď ĎçŘĘæŰĞäžűåŘ■æŸŕäÿĂäÿłå¿ĹæčŸæĽ ŃçŽĎéŮőécŸïijŇæIJĂåŘŐäžď åęĆæđIJä¡ăåIJÍæĽľ åśŢäžčçăĄäÿ■ä¡£çŤÍæIJňèŁĆçŽĎæŁĂæIJŕïijŇæŰĞäžűåŘ■çŽĎåď ĎçŘĘæŰźåijŔåŠŇåŠ åŇĚæŃňçijŰçăA/çŢŇéÍcå■ŮèŁĆïijŇåď ĎçŘĘåİŔå■Ůçňęïij Näžčç ŘĘè¡ňæ■ćåŠŇåĚűäžŰåď ■æİĆæČĚåĘţãĂĆ

17.18 15.18 äijäéĂŠåůšæĽŞåijĂçŽĎæŰĞäžűçżŹCæĽľåśŢ

éŮőécŸ

ä¡ăåIJÍPythonäÿ■æIJL'äÿĂäÿłæĽŞåijĂçŽĎæŰĞäžűåŕźèśąïijŇä¡ĘæŸŕéIJĂèęĄåřĘåőČäijăçżŹèęĄä¡£çŤÍ

èğčåĘşæŰźæąĹ

èęĄåřĘäÿĂäÿłæŰĞäžűè¡ňæ■ćäÿžäÿĂäÿłæŢť åđŃçŽĎæŰĞäžűæŔŔè£řçňęïijŇä¡£çŤÍ `PyFile_FromFd()` ïijŇåęĆäÿŃïijŽ

```
PyObject *fobj;     /* File object (already obtained somehow) */
int fd = PyObject_AsFileDescriptor(fobj);
if (fd < 0) {
   return NULL;
}
```

çżŞæđIJæŰĞäžűæŔŔè£řçňęæŸŕéĂŽè£ĞèřČçŤÍ `fobj` äÿ■çŽĎ `fileno()` æŰźæşŢèŐůå¿ŮçŽĎãĂĆ åŽăæ■ď ïijŇäžžä¡Ţäžèè£Źçğ■æŰźåijŔæŽť éIJšçżŹäÿĂäÿłæŔŔè£řåŹĺçŽĎåŕźèśąéČ äÿĂæŮęä¡ăæIJL'äžĘè£ŹäÿłæŔŔè£řåŹĺïijŇåőČåřśèČ¡èćńäijăéĂŠçżŹåď Žäÿłä¡ŐçžğçŽĎåŘŕåď ĎçŘĘæŰĞäžű

åęĆæđIJä¡ăéIJĂèęĄè¡ňæ■ćäÿĂäÿłæŢť åđŃæŰĞäžűæŔŔè£řçňęäÿžäÿĂäÿłPythonåŕźèśąïijŇéĂĆçŤÍäÿŃé `PyFile_FromFd()` :

```
int fd;      /* Existing file descriptor (already open) */
PyObject *fobj = PyFile_FromFd(fd, "filename","r",-1,NULL,NULL,NULL,
↪1);
```

`PyFile_FromFd()` çŽĎåŔĆæŢřåŕźåžŤåĘĚç¡őçŽĎ `open()` åĞ¡æŢřãĂĆ NULL-LèąÍçď žçijŰçăĄãĂĄéŤŹèŕŕåŠŇæ■ćèąŇåŔĆæŢřä¡£çŤÍéżŸèőď åĂijãĂĆ

èőÍèőž

åęĆæđIJåřĘPythonäÿ■çŽĎæŰĞäžűåŕźèśąäijăçżŹCïijŇæIJL'äÿĂäžŽæşÍæĎŔäžŃéąźãĂĆ éęŰåĚĹïijŇPythonéĂŽè£Ğ `io` æÍąåİŮæL'ğèąŇèĞłåůśçŽĎI/OçijŞåĘšãĂĆ

åIJÍäijăéĂŠäżżä¡Ţçśżåđ́ÑçŽĎæŰĞäżűæŔŕè£řçňęçżŹCäźŃåĽ■ïijÑä¡ăéČ¡èęĄéęŰåĚĹåIJÍçŻÿåžŤæŰĞäżűåŕź
äÿ■çĎűçŽĎèŕİïijÑä¡ăäijŽæĽŞäźśæŰĞäżűçşżçż§äÿŁéİćçŽĎæŢřæ■őãĂĆ

åĚűæňąïijÑä¡ăéIJĂèęĄçĽźåĹńæşĺæĎŔæŰĞäżűçŽĎå¡Šåśđè ĂĚäżěåŔŁåĚşéŮ■æŰĞäżűçŽĎèĄÑèť čãĂĆ
åęĆæđIJäÿĂäÿłæŰĞäżűæŔŕè£řçňęèćńäijäçżŹCïijÑä¡ĘæŸŕåIJÍPythonäÿ■è£ŸåIJÍèćńä¡£çŤÍçİĂïijÑä¡ăéIJĂèę
çśżäijijçŽĎïijÑåęĆæđIJäÿĂäÿłæŰĞäżűæŔŕè£řçňęèćńè¡ňæ■ćäÿžäÿĂäÿłPythonæŰĞäżűåŕźèśąïijÑä¡ăéIJĂèęA
`PyFile_FromFd()` çŽĎæIJĂåŘŐäÿĂäÿłåŔĆæŢřèćńèő¿ç¡őæĹŘ1ïijÑçŤÍæİěæŇĞåĞżPythonåžŤèŕěåĚşéŮ

åęĆæđIJä¡ăéIJĂèęĄäżŐCæăĞåĞĘI/OåžŞäÿ■ä¡£çŤÍåęĆãĂĂ`fdopen()`
åĞ¡æŢřæİěåĹŻåžžäÿ■åŘÑçśżåđ́ÑçŽĎæŰĞäżűåŕźèśąæŕŤåęĆ `FILE *` åŕźèśąïijÑ
ä¡ăéIJĂèęĄçĽźåĹńåřŔå£ĆäžĘãĂĆè£ŹæăůåĄŽäijŽåIJÍI/OåäĘæăĹäÿ■äžğçŤ§äÿd'äÿłåőÑåĚÍäÿ■åŘÑçŽĎI/Oç
ïijÍäÿĂäÿłæŸŕæİěèĞłPythonçŽĎ `io` æĺąåİŮïijÑåŔęäÿĂäÿłæİěèĞłCçŽĎ `stdio`
ïijĽãĂĆ åČŔCäÿ■çŽĎ `fclose()` äijŽåĚşéŮ■PythonèęĄ¡£çŤÍçŽĎæŰĞäżűãĂĆ
åęĆæđIJèőľä¡ăéĂĽçŽĎèŕİïijÑä¡ăåžŤèŕěäijŽéĂĽæŃľåŐżæđĎåžžäÿĂäÿłæĽ'ľåšŢäžčçĄąæİěåd'ĎçŘĘåžŢåśĆ
èĂÑäÿ■æŸŕä¡£çŤÍæİěèĞł<stdio.h>çŽĎéńŸåśĆæŁ¡èśąåŁ§èČ¡ãĂĆ

17.19 15.19 äžŐCèŕ■èÍĂäÿ■èŕżåŘŰçśżæŰĞäżűåŕźèśą

éŮőécŸ

ä¡ăèęĄåĘŹCæĽ'ľåšŢæİěèŕżåŘŰæİěèĞłäżżä¡ŢPythonçśżæŰĞäżűåŕźèśąäÿ■çŽĎæŢřæ■őïijĹæŕŤåęĆæŹőé

èğčåĘşæŰźæąĹ

èęĄèŕżåŘŰäÿĂäÿłçśżæŰĞäżűåŕźèśąçŽĎæŢřæ■őïijÑä¡ăéIJĂèęĄéĞ■åd'■èřČçŤÍ
`read()` æŰźæşŢïijÑçĎűåŘŐæ■čçąőçŽĎèğčçăĄèŐůå¿ŮçŽĎæŢřæ■őãĂĆ

äÿŃéİćæŸŕäÿĂäÿłCæĽ'ľåšŢåĞ¡æŢřä¿Ńå■ŘïijÑäžĚäžĚåŘłæŸŕèŕżåŘŰäÿĂäÿłçśżæŰĞäżűåŕźèśąäÿ■çŽĎ

```
#define CHUNK_SIZE 8192

/* Consume a "file-like" object and write bytes to stdout */
static PyObject *py_consume_file(PyObject *self, PyObject *args) {
  PyObject *obj;
  PyObject *read_meth;
  PyObject *result = NULL;
  PyObject *read_args;

  if (!PyArg_ParseTuple(args,"O", &obj)) {
    return NULL;
  }

  /* Get the read method of the passed object */
  if ((read_meth = PyObject_GetAttrString(obj, "read")) == NULL) {
    return NULL;
  }

  /* Build the argument list to read() */
  read_args = Py_BuildValue("(i)", CHUNK_SIZE);
```

```
  while (1) {
    PyObject *data;
    PyObject *enc_data;
    char *buf;
    Py_ssize_t len;

    /* Call read() */
    if ((data = PyObject_Call(read_meth, read_args, NULL)) == NULL)
↪{
      goto final;
    }

    /* Check for EOF */
    if (PySequence_Length(data) == 0) {
      Py_DECREF(data);
      break;
    }

    /* Encode Unicode as Bytes for C */
    if ((enc_data=PyUnicode_AsEncodedString(data,"utf-8","strict
↪"))==NULL) {
      Py_DECREF(data);
      goto final;
    }

    /* Extract underlying buffer data */
    PyBytes_AsStringAndSize(enc_data, &buf, &len);

    /* Write to stdout (replace with something more useful) */
    write(1, buf, len);

    /* Cleanup */
    Py_DECREF(enc_data);
    Py_DECREF(data);
  }
  result = Py_BuildValue("");

 final:
  /* Cleanup */
  Py_DECREF(read_meth);
  Py_DECREF(read_args);
  return result;
}
```

èęĄæţŃèŕŢè£ŹäÿłäżčçăĄïijŇåĚĹæđĎéĂăäÿĂäÿłçśżæŰĞäżűåŕźèśąæŕŤåęĆäÿĂäÿłStringIOåőđä¿ŃïijŇç

```
>>> import io
>>> f = io.StringIO('Hello\nWorld\n')
>>> import sample
>>> sample.consume_file(f)
```

```
Hello
World
>>>
```

èőĺèőž

åŠŇæŹőéĂŽçşżçż§æŰĞäżűäÿ■åŘŇçŽĎæŸŕïijŇäÿĂäÿłçśżæŰĞäżűåŕźèśąåźűäÿ■éIJĂèęĄä¡£çŤĺä¡Őçžğ
åŻăæ■ď'ïijŇä¡ăäÿ■èČ¡ä¡£çŤĺæŹőéĂŽçŽĎCåžŞåĞ¡æŢřæİěèő£éŮőåőČãĂĆ
ä¡ăéIJĂèęĄä¡£çŤĺPythonçŽĎC APIæİěåČŔæŹőéĂŽæŰĞäżűçśżäijijçŽĎéĆčæăůæŞ■ä¡IJçśżæŰĞäżűåŕźèśąãĂ

åIJĺæĹŚäżňçŽĎèğčåĘşæŰźæąĹäÿ■ïijŇ`read()` æŰźæşŢäżŐèćńäijăéĂŠçŽĎåŕźèśąäÿ■æŔŘåŔŰåĞžæİě
äÿĂäÿłåŔĆæŢřåĹŮèąĺèćńæđĎåżžçĎűåŘŐäÿ■æŰ■çŽĎèćńäijăçżŹ `PyObject_Call()`
æİěèřČçŤĺè£ŹäÿłæŰźæşŢãĂĆ èęĄæčĂæ§ěæŰĞäżűæIJńåř¿ïijĹEOFïijĽïijŇä¡£çŤĺäžĘ
`PySequence_Length()` æİěæ§ěçIJŃæŸŕåŘęè£ŤåŻđåŕźèśąéŢ£åžęäÿž0.

åŕźäžŐæĽĂæIJĽçŽĎI/OæŞ■ä¡IJïijŇä¡ăéIJĂèęĄåĚşæşĺåžŢåśĆçŽĎçijŰçăĄæăijåijŔïijŇè£ŸæIJĽå■ŮèŁ
æIJňèŁĆæijŤçď'žäžĘåęĆä¡ŢäżěæŰĞæIJňæĺąåijŔèŕżåŔŰäÿĂäÿłæŰĞäżűåźűåřĘçżŞæđIJæŰĞæIJňèğčçăĄäÿž
åęĆæđIJä¡ăæČşäżěäžŇè£ŻåĹűæĺąåijŔèŕżåŔŰæŰĞäżűïijŇåŔłéIJĂèęĄä£őæŤźäÿĂçĆźçĆźå■şåŔŕïijŇä¿ŃåęĆ

```
...
/* Call read() */
if ((data = PyObject_Call(read_meth, read_args, NULL)) == NULL) {
  goto final;
}

/* Check for EOF */
if (PySequence_Length(data) == 0) {
  Py_DECREF(data);
  break;
}
if (!PyBytes_Check(data)) {
  Py_DECREF(data);
  PyErr_SetString(PyExc_IOError, "File must be in binary mode");
  goto final;
}

/* Extract underlying buffer data */
PyBytes_AsStringAndSize(data, &buf, &len);
...
```

æIJňèŁĆæIJĂéŽ¿çŽĎåIJřæŰźåIJĺäžŐåęĆä¡Ţè£ŻèąŇæ■čçąőçŽĎåĘĚå■ŸçőąçŘĘãĂĆ
å¡Şåď'ĎçŘĘ `PyObject * `` åŔŸéĞŔçŽĎæŮűåĂŹïijŇéIJĂèęĄæşĺæĎŔçőąçŘĘåijŢçŤĺèőąæŢ
åŕź ``Py_DECREF()` çŽĎèřČçŤĺåŕśæŸŕæİěåĄŽè£ŹäÿłçŽĎãĂĆ

æIJňèŁĆäżčçăĄäżěäÿĂçğ■éĂŽçŤĺæŰźåijŔçijŰåĘŹïijŇåŻăæ■ď'äżŰäź§èČ¡éĂĆçŤĺäžŐåĚűäżŰçŽĎæŰĞ
ä¿ŃåęĆïijŇèęĄåĘŹæŢřæ■őïijŇåŔłéIJĂèęĄèŐůåŔŰçśżæŰĞäżűåŕźèśąçŽĎ `write()`
æŰźæşŢïijŇåřĘæŢřæ■őè¡ňæ■ćäÿžåŘĹéĂĆçŽĎPythonåŕźèśą ïijĹå■ŮèŁĆæĹŰUni-
codeïijĽïijŇçĎűåŘŐèřČçŤĺèŕěæŰźæşŢåřĘè¿ŞåĚěåĘŹåĚěåĹřæŰĞäżűãĂĆ

æIJĂåŘŐïijŇåř¡çőąçśżæŰĞäżűåŕźèśąéĂŽåÿÿè£ŸæŔŘä¿ŻåĚűäżŰæŰźæşŢïijĹæŕŤåęĆreadline(),

read_info()ïijL'ïijÑ æĹŚäžňæIJĂåě¡åŔłä¡£çŤĺå§žæIJňçŽĎ `read()` åŠŇ `write()` æŰźæşŢãĂĆ åIJĺåĘŹCæL'ľåśŢçŽĎæŮűåĂŹïijŇèČ¡çőĂå■Ţåřśår¡éĞŔçőĂå■ŢãĂĆ

17.20 15.20 åďĎçŘĘCèŕ■èĺĂäÿ■çŽĎåŔŕè£■äžčåŕźèśą

éŮőécŸ

ä¡ăæČşåĘŹCæL'ľåśŢäžčçăĄåď'ĎçŘĘæİěèĞłäżżä¡ŢåŔŕè£■äžčåŕźèśąåęĆåĹŮèąĺãĂĄåĚČçżĎãĂĄæŰĞä

èğčåĘşæŰźæąĹ

äÿŃéİćæŸŕäÿĂäÿłCæL'ľåśŢåĞ¡æŢřä¿Ńå■ŘïijŇæijŤçď'žäžĘæĂŐæăůåď'ĎçŘĘåŔŕè£■äžčåŕźèśąäÿ■çŽĎå

```
static PyObject *py_consume_iterable(PyObject *self, PyObject␣
↪*args) {
  PyObject *obj;
  PyObject *iter;
  PyObject *item;

  if (!PyArg_ParseTuple(args, "O", &obj)) {
    return NULL;
  }
  if ((iter = PyObject_GetIter(obj)) == NULL) {
    return NULL;
  }
  while ((item = PyIter_Next(iter)) != NULL) {
    /* Use item */
    ...
    Py_DECREF(item);
  }

  Py_DECREF(iter);
  return Py_BuildValue("");
}
```

èőÍèőž

æIJňèŁĆäÿ■çŽĎäžčçăĄåŠŇPythonäÿ■åŕźåžŤäžčçăĄçśżäijijãĂĆ `PyObject_GetIter()` çŽĎèřČçŤĺåŠŇèřČçŤÍ `iter()` äÿĂæăůåŔŕèŐůå¿ŮäÿĂäÿłè£■äžčåŹĺãĂĆ `PyIter_Next()` åĞ¡æŢřèřČçŤÍ `next` æŰźæşŢè£ŤåŻđäÿŃäÿĂäÿłåĚČçť'ăæĹŰNULL(åęĆæđIJæšąæIJL'åĚČçť'ääžĘ)ãĂĆ èęĄæşÍæĎŔæ■čçąőçŽĎåĘĚå■ŸçőąçŘĘâĂŤâĂŤ `Py_DECREF()` éIJĂèęĄåŘŇæŮűåIJÍäžğçŤ§çŽĎåĚČçť'ääŠŇè£■äžčåŹĺåŕźèśąæIJňèžńäÿŁåŘŇæŮűèćńèřČçŤÍïijŇ äžěéĄ£åĚ■åĞžçŐřåĘĚå■ŸæşĎéIJšãĂĆ

17.21 15.21 èŕŁæŰ■åĹĘæőţéŤŹèŕŕ

éŮőécŸ

èğčéĞŁåŹĺåŽääÿžæ§Řäÿłå Ĺ Ęæőţé ŤŹèŕŕãĂĄæĂżçž£éŤŹèŕŕãĂĄèő£éŮőèűŁçŢŇæĹŰåĚűäžŰèĞťåŚ¡éŤ
ä¡ăæČşèŐůå¿ŮPythonåăĘæăĹä£ąæĄŕïijŇäžŐèĂŇæĽ¿åĞžåIJĺåŔŚçŤ§éŤŹèŕŕçŽĎæŮűåĂŹä¡äçŽĎçÍŃåžŔè£

èğčåĘşæŰźæąĹ

faulthandler æĺąåİŮèČ¡èćńçŤĺæİěåÿőä¡äèğčåĘşè£ŹäÿłéŮőécŸãĂĆ
åIJĺä¡äçŽĎçÍŃåžŔäÿ■åijŢåĚěäÿŃåĹŮäžčçăĄïijŽ

```
import faulthandler
faulthandler.enable()
```

åŔęåď Űè£ŸåŔŕäžěåČŔäÿŃéİćè£ŹæăůäĄ£çŤÍ -Xfaulthandler
æİěè£ŘèąŇPythonïijŽ

```
bash % python3 -Xfaulthandler program.py
```

æIJĂåŘŐïijŇä¡ääŔŕäžěèő¿ç¡ő PYTHONFAULTHANDLER çŐŕåćČåŔŸéĞŔãĂĆ åijĂåŘŕ-
faulthandleråŘŐïijŇåIJÍCæĽľåśŢäÿ■çŽĎèĞťåŚ¡éŤŹèŕŕäijŽåŕijèĞťäÿĂäÿłPythonéŤŹèŕŕääĘæăĹèćńæĽŞå■ř

```
Fatal Python error: Segmentation fault

Current thread 0x00007fff71106cc0:
  File "example.py", line 6 in foo
  File "example.py", line 10 in bar
  File "example.py", line 14 in spam
  File "example.py", line 19 in <module>
Segmentation fault
```

ař¡çőąè£Źäÿłåźűäÿ■èČ¡åŚŁèŕĽä¡ăCäžčçăĄäÿ■åŞłéĞŇåĞžéŤŹäžĘïijŇä¡ĘæŸŕèĞşařŚèČ¡åŚŁèŕĽä¡ăPyth

èőÍèőž

faulthandleräijŽåIJÍPythonäžčçăĄæĽğèąŇåĞžéŤŹçŽĎæŮűåĂŹåŘŚä¡ääśŢçd'žèů§èÿłä£ąæĄŕãĂĆ
èĞşařŚïijŇåőČäijŽåŚŁèŕĽä¡ääĞžéŤŹæŮűèćńèřČçŤÍçŽĎæIJĂéąűçžğæĽľåśŢåĞ¡æŢřæŸŕåŞłäÿłãĂĆ
åIJÍpdbåŠŇåĚűäžŰPythonèřČèŕŢåŹÍçŽĎåÿőåŁľäÿŃïijŇä¡ääřśèČ¡è£¡æăźæžŕæžŘæĽ¿åĹŕéŤŹèŕŕæĽĂåIJÍçŽĎ

faulthandleräÿ■äijŽåŚŁèŕĽä¡ääżžä¡ŢCèŕ■èÍĂäÿ■çŽĎéŤŹèŕŕä£ąæĄŕãĂĆ
åŻăæ■ď ïijŇä¡ăéIJĂèęĄä¡£çŤÍäijăçż§çŽĎCèřČèŕŢåŹÍïijŇæŕŤåęĆgdbãĂĆ
äÿ■è£ĞïijŇåIJÍfaulthandlerè£¡èÿłä£ąæĄŕåŘŕäžěèőľä¡ääŐżåĹď æŰ■äžŐåŞłéĞŇçİĂæĽŃãĂĆ
è£ŸèęĄæşÍæĎŘçŽĎæŸŕåIJÍCäÿ■æ§ŘäžŽçśżåđŃçŽĎéŤŹèŕŕåŘŕèČ¡äÿ■åď łåőźæŸŞæĄćåď ■ãĂĆ
ä¿ŃåęĆïijŇåęĆæđIJäÿĂäÿłCæĽľåśŢäÿćåijČäžĘçÍŃåžŘåăĘæăĹä£ąæĄŕïijŇåőČäijŽèőľfaulthandleräÿ■åŘŕçŤ
éĆčäźĹä¡ääź§å¿Ůäÿ■åĹřäžżä¡Ţè¿ŞåĞžïijĹéŹď äžĘçÍŃåžŘåěŤæžČåď ŰïijL'ãĂĆ

18 éŹĎå¡ŢA

18.1 åIJÍçž£èţĎæžŘ

http://docs.python.org

åęĆæđIJä¡ăéIJĂèęĄæůśåĚěäžĘèğčæŐćçľ˝űèŕ■èÍĂåŠŇæÍąåİŮçŽĎçżĘèŁĆïijŇéĆčäźĹäÿ■å£Ěèŕť˝ïijŇPyth
3 çŽĎæŰĞæąčèĂŇäÿ■æŸŕäžěåL’■çŽĎèĂĄçL’ĹæIJň

http://www.python.org/dev/peps

åęĆæđIJä¡ăåŘŠçŘĘèğčäÿžpythonèŕ■èÍĂæůżåŁăæŰřçL’źæĂğçŽĎåŁÍæIJžäžěåŔŁåőđçŐřçŽĎçżĘèŁĆïijŇ
Enhancement ProposalsâĂŤ-PythonåijĂåŔŠçijŰçăĄèğĎèŇČïijL’çżİåŕźæŸŕéİđåÿÿåőİèť˝tçŽĎèţĎæžŘãĂĆåřď˝

http://pyvideo.org

è£ŹéĞŇæIJL’æİěèĞłæIJĂè£ŠçŽĎPyConåď˝ğäijŽãĂĄçŤÍæĹůçżĎèğĄéİćäijŽç■L’çŽĎåď˝ğéĞŔèğĘéćŚæi
3äÿ■æůżåŁăçŽĎçŽĎæŰřçL’źæĂğãĂĆ

http://code.activestate.com/recipes/langs/python

éŢ£æIJ§äžěæİěïijŇActiveStateçŽĎPythonçL’ĹåİŮåůšçżŔæĹŘäÿžäÿĂäÿłæL’¿åĹřæŢřäžěå■Čè őąçŽĎéŠĹ

http://stackoverflow.com/questions/tagged/python

Stack Overflow çŻőåL’■æIJL’èűĚè£Ğ175,000äÿłéŮőécŸècńæăĞèőřäÿžPythonçŻÿåĚşïijÍèĂŇåĚűäÿ■åď˝
3çŽĎïijL’ãĂĆår¡çőąéŮőécŸåŠŇåŻđç■ŤçŽĎèť˝íéĞŔäÿ■åŘŇïijŇä¡ĘæŸŕäż■çĎűèČ¡åŔŠçŐřå¿Ĺåď˝Žåě¡äijŸçğ

18.2 Pythonå■ęäźžääźęçś■

äÿŇéİćè£ŹäžŻäźęçś■æŔŘä¿ŻäžĘåŕźPythonçijŰçÍŇçŽĎåĚěéŮÍäżŇçż■ïijŇäÿŤéĞ■çĆźæŤ¿åIJÍäžĘPytho
3äÿŁãĂĆ

Beginning Python: From Novice to Professional, 2nd Edition, by Magnus Lie HetâĂŘ land, Apress (2008). Programming in Python 3, 2nd Edition, by Mark Summerfield, Addison-Wesley (2010).

- *Learning Python*ïijŇçňňåŻŻçL’Ĺ ïijŇä¡IJèĂĚ Mark LutzïijŇ OâĂŹReilly & Associates åĞžçL’Ĺ (2009)ãĂĆ
- *The Quick Python Book*ïijŇä¡IJèĂĚ Vernon CederïijŇ Manning åĞžçL’Ĺ(2010)ãĂĆ
- *Python Programming for the Absolute Beginner*ïijŇçňňäÿL’çL’ĹïijŇä¡IJèĂĚ Michael DawsonïijŇCourse Technology PTR åĞžçL’Ĺ(2010).
- *Beginning Python: From Novice to Professional*ïijŇçňňäžŇçL’ĹïijŇ ä¡IJèĂĚ Magnus Lie HetâĂŘ landïijŇ Apress åĞžçL’Ĺ(2008).
- *Programming in Python 3*ïijŇçňňäžŇçL’ĹïijŇä¡IJèĂĚ Mark SummerfieldïijŇAddison-Wesley åĞžçL’Ĺ (2010).

18.3 éńŸçžğäźęçś■

äÿŃéİćçŽĎè£ŹäžŻäźęçś■æŔŘä¿ŻäžĘæŻťåď'ŽéńŸçžğçŽĎèŇČåŻť'ïijŇäź§åŇĚåŘńPython 3æŰźéİćçŽĎåĘĚåőźãĂĆ

- *Programming Python*ïijŇçňňåŻŻçL'Ĺ, by Mark Lutz, OâĂŹReilly & Associates åĞžçL'Ĺ(2010).
- *Python Essential Reference*ïijŇçňňåŻŻçL'ĹïijŇä¡IJèĂĚ David Beazley, Addison-Wesley åĞžçL'Ĺ(2009).
- *Core Python Applications Programming*ïijŇçňňäÿL'çL'ĹïijŇä¡IJèĂĚ Wesley Chun, Prentice Hall åĞžçL'Ĺ(2012).
- *The Python Standard Library by Example* ïijŇ ä¡IJèĂĚ Doug HellmannïijŇAddison-Wesley åĞžçL'Ĺ(2011).
- *Python 3 Object Oriented Programming*ïijŇä¡IJèĂĚ Dusty Phillips, Packt Publishing åĞžçL'Ĺ(2010).
- *Porting to Python 3*ïijŇ ä¡IJèĂĚ Lennart RegebroïijŇCreateSpace åĞžçL'Ĺ(2011), http://python3porting.com.

19 åĚşäžŐèŕŚèĂĚ

åĚşäžŐèŕŚèĂĚ

- åğŞåŘ■ïijŽ çĘŁèČ¡
- å¿őä£ąïijŽ yidao620
- EmailïijŽ yidao620@gmail.com
- å■Žåőćïij Ž http://yidao620c.github.io/
- GitHubïijŽ https://github.com/yidao620c

---

20 Roadmap

2014/08/10 - 2014/08/31:

```
|    githubéąźçŽőæŘ■åżžïijŇreadthedocsæŰĞæąčçŤ§æĹŘãĂĆ
|    æŢt'äÿłéąźçŻőçŽĎæąĘæđűåőŇæĹŘ
```

2014/09/01 - 2014/10/31:

```
|    åL'■4çńăç£żèŕŚåőŇæĹŘ
```

2014/11/01 - 2015/01/31:

```
|    åL’■8çńăç£żèŕŚåőŇæĹŘ
```

2015/02/01 - 2015/03/31:

```
|    åL’■9çńăç£żèŕŚåőŇæĹŘ
```

2015/04/01 - 2015/05/31:

```
|    10çńăç£żèŕŚåőŇæĹŘ
```

2015/06/01 - 2015/06/30:

```
|    11çńăç£żèŕŚåőŇæĹŘ
```

2015/07/01 - 2015/07/31:

```
|    12çńăç£żèŕŚåőŇæĹŘ
```

2015/08/01 - 2015/08/31:

```
|    13çńăç£żèŕŚåőŇæĹŘ
```

2015/09/01 - 2015/11/30:

```
|    14çńăç£żèŕŚåőŇæĹŘ
```

2015/12/01 - 2015/12/20:

```
|    15çńăç£żèŕŚåőŇæĹŘ
```

2015/12/21 - 2015/12/31:

```
|    åŕźåĚĺéČĺç£żèŕŚè£ŻèąŇæăąåŕźäÿĂæňą
```

2016/01/01 - 2016/01/10:

```
|    åŕźåd’ŰåĚňåijĂåŔŚåÿČåőŇæŢt’çL’Ĺ1.
↪0ïijŇåŇĚæŃňè¡ňæ■ćåŘŐçŽĎPDFæŰĞäžű
```