

# Python程式設計#10

林奇賦 [daky1983@gmail.com](mailto:daky1983@gmail.com)

# Outline

- 內建標準函式庫
- 第三方函式庫
- PyQRCode

# 模組的使用

- 建立模組
  - 建立一個A.py檔
  - 再建立第二個B.py檔
- 載入模組
  - 在第二個檔案輸入import A
  - 亦可使用from ModuleName import FunctionName
  - Python會將載入的模組編譯成A.pyc於B.py檔旁邊
  - 使用第二個方法可以使用'\*'表示載入所有函數
- 用help()函數可以將模組定義的函數備註文字輸出

# 使用標準函數庫

- 使用os模組
  - 提供顯示系統環境參數與指令功能函數
  - `os.rename(src, dst)`
    - 對檔案或目錄更換名稱
    - `src`引數是原本的資料夾
    - `dst`引數是修改後的資料夾名稱
  - `os.remove(path)`
    - 移除檔案
    - `path`引數傳入檔案位置
    - 不會移除資料夾
  - `os.removedirs(path)`
    - 移除空的資料夾
  - `os.listdir(path)`
    - 輸出`path`引數位置的目錄和檔案名稱

# 使用標準函數庫

- 使用os模組
  - `os.chdir(path)`與`os.getcwd()`
    - `os.chdir(path)`函數是切換目錄到`path`引數位置
    - `os.getcwd()`是顯示目前所在的目錄位置
  - `os.mkdir(path[, mode])`與`os.rmdir(path)`
    - `os.mkdir()`是建立資料夾
    - `path`引數是建立/刪除目錄的位置
    - `mode`引數是Unix平台使用的
    - `os.rmdir()`函數是刪除目錄
  - `os.path.getsize(path)`
    - 取得檔案大小
  - `os.path.getctime(path)`
    - 取得檔案的建立日期

# 使用標準函數庫

- 使用os模組
  - `os.path.getmtime(path)`
    - 取得檔案的修改日期
  - `os.path.getatime(pah)`
    - 取得檔案的存取日期
  - `os.path.isfile(path)`
    - 判斷傳入的path引數是否為檔案
  - `os.path.isdir(path)`
    - 判斷傳入的path引數是否為目錄
- <http://docs.python.org/3.3/library/os.path.html>

# 使用標準函數庫

- 使用random模組
  - `random.random()`
    - 隨機產生 $0.0 \leq x < 1.0$ 之間的數字
  - `random.uniform(num1, num2)`
    - 隨機產生 $\text{num1} \leq x < \text{num2}$ 的數字
  - `random.randint(1,10)`
    - 產生一個1到10的數字
  - `random.randrange(0,101)`
    - 產生一個0到100的數字

# 使用標準函數庫

- 使用random模組
  - random.choice()
    - 將傳入的內容隨機取得
  - random.suffle()
    - 將傳入的序列型態資料內的項目順序以隨機順序產生
  - random.sample()
    - 將傳入的序列項目以指定的長度顯示一個隨機項目值的序列
- <http://docs.python.org/3.3/library/random.html?highlight=random#random>



# 使用標準函數庫

- 使用time模組
  - `time.time()`
    - 取得系統時間
  - `time.sleep(num)`
    - 設定暫停時間
  - `time.localtime()`
    - 回傳的格式如下：
    - `time.struct_time(tm_year, tm_mon, tm_mday, tm_hour, tm_min, tm_sec, tm_wday, tm_yday_, tm_isdst)`
  - `time.gmtime()`
    - 取得UTC時間

# 使用標準函數庫

- 使用sys模組
  - `sys.argv[o]`
    - 會回傳此程式檔案的位置與名稱
  - `sys.argv>1`
    - 是判斷是否有帶入參數
  - `sys.builtin_module_names`
    - 回傳Python程式語言內所有內置模組名稱
  - `sys.modules.keys()`
    - 得知目前已經載入的模組
  - `sys.platform`
    - 取得目前作業系統的版本
  - `sys.exit()`
    - 宣告`sys.exit(o)`終止程式

# 使用標準函數庫

- 使用sys模組
  - sys.version
    - 回傳目前安裝在系統上的Python版本
    - 格式：'(#build\_number, build\_date, build\_time) [compiler]'
  - sys.api\_version
    - 回傳Python直譯器的C API版本
  - sys.version\_info
    - 回傳一個tuple型態的值
    - ('主要版本' , '次要版本' , '小版本' )
  - sys.winver
    - 回傳的版本數字是註冊在Windows裡的Python版本
  - sys.path
    - 定義Python搜尋模組的路徑

# 使用標準函數庫

- 建立tar.gz壓縮檔
  - 宣告使用tarfile.open(path, mode)建立壓縮檔
  - mode引數為'a'代表要自動建立壓縮檔並將檔案添加進去
  - 使用tar.add(name)函數將檔案加入壓縮檔內
  - 用tar.close()將壓縮檔關閉
- 讀取tar.gz壓縮檔
  - tar.getmembers()
  - Ex:

```
for file in tar.getmembers():  
    print("%s\t\t\t%s" % (file.name, file.size))
```

# 使用標準函數庫

- 使用shutil模組
  - 比較高階的應用層，提供數個針對檔案操作的功能
  - `shutil.copytree(src, dst)`
    - 複製整個目錄，包含目錄內的所有檔案
  - `shutil.copy(src, dst)`
    - 複製檔案
  - `shutil.rmtree(path)`
    - 移除整個目錄，包含目錄內的所有檔案
  - `shutil.move(src, dst)`
    - 移動檔案，移動時也可以進行更換檔案名稱
  - `shutil.copystat(src, dst)`
    - 複製檔案，會連同檔案屬性一同複製

# 練習

- 建立一個檔案random.txt 檔案內容為
  - 生成 100個1~1000的奇數存入檔案 (一個數字一行，且數字不重複)
- 使用os模組讀取此檔案
- 使用time模組將此檔案的印出建立日期、修改日期
- 建立一個名為test的目錄
- 使用shutil模組將random.txt移動至test目錄下

# 第三方函式庫

- python社群提供了大量的第三方模組，使用方式與標準庫類似。它們的功能無所不包，覆蓋科學計算、Web開發、資料庫介面、圖形系統多個領域，並且大多成熟而穩定。第三方模組可以使用Python或者C語言編寫。
- 您也許聽過「不要重造輪子」這句話，或是 DRY (Don't Repeat Yourself)，講得就是「別人已經寫好的東西，就拿去用吧，不用自己再重新寫一套」。

# 安裝第三方函式庫

- 方法1
  - 下載原始碼，手動執行 `python setup.py install` 安裝
- 方法2：利用第三方安裝工具（如 `pip`，`easy_install`，`distribute` 等）自動化安裝



# Python3 Packages

- 列出所有支援python3以上的函式庫
- <https://pypi.python.org/pypi?:action=browse&c=533&show=all>

# pip 1.3.1

- *A tool for installing and managing Python packages.*
- <https://pypi.python.org/pypi/pip/1.3.1>

# pip安裝步驟

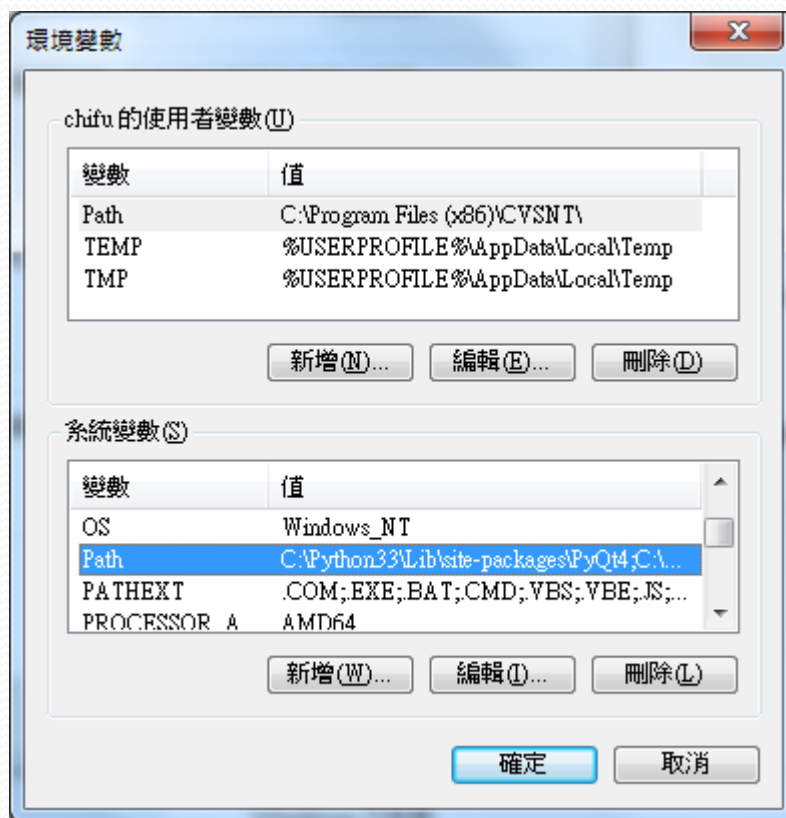
- 1.下載原始檔

<https://pypi.python.org/packages/source/p/pip/pip-1.3.1.tar.gz>

- 2.解壓縮
- 3.執行 `python setup.py install`
- 4.設定環境變數

# 設定環境變數

- 於系統變數中的Path，新增C:\Python33\Scripts



# distribute 0.6.45

- *Easily download, build, install, upgrade, and uninstall Python packages*
- <https://pypi.python.org/pypi/distribute>

# PyQRCode 0.09

- QR Code生成器，使用python3寫成的可以輸出SVG與PNG的格式
- <https://pypi.python.org/pypi/PyQRCode/0.09>

# PyQRCode 範例

- 將網址輸出成svg格式，比例設定為8

```
from pyqrcode import QRCode  
url = QRCode('http://www.ntu.edu.tw')  
#url.svg('url.svg', scale=8)
```

# PyQRCode 範例

- 將網址輸出成png格式，比例設定為10

```
from pyqrcode import QRCode  
url = QRCode('http://www.ntu.edu.tw')  
url.png('qrcode.png',scale=10)
```

**ERROR! 缺少pypng套件**



# 安裝練習

- 試著安裝 pypng 套件
- <https://pypi.python.org/pypi/pypng/0.0.15>
- 安裝成功後，試著將QR code輸出成png檔案

# 練習

- 試著讀取 `urls.txt` 檔案中的超連結
- 並將其輸出成Qrcode存成png檔
  - 1.png, 2.png ...