

Python 程式設計

林奇賦 daky1983@gmail.com

Outline

- ▶ 內建模組
- ▶ 第三方模組

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 Function/Class Name
 - ▶ 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

`__name__`

- ▶ `__name__` 在 Python 是一個特別的變數 (variable) ，若這個變數的值等於 `'__main__'` ，就表示目前執行的是相同的 Python 檔案
- ▶ EX08_02_1, EX08_02_2

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os模組

- ▶ `os.chdir(path)` 函數
 - ▶ 切換目錄到 `path` 引數位置
- ▶ `os.getcwd()`
 - ▶ 顯示目前所在的目錄位置
- ▶ `os.mkdir(path[, mode])`
 - ▶ 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
 - ▶ 刪除目錄
- ▶ `os.path.getsize(path)`
 - ▶ 取得檔案大小

使用os模組

- ▶ `os.path.getctime(path)`
 - ▶ 取得檔案的建立日期
- ▶ `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

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 $num1 \leq x < num2$ 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random模組

- ▶ `random.choice(seq)`
 - ▶ 將傳入的內容隨機取得
- ▶ `random.shuffle(x [, random])`
 - ▶ 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(population, k)`
 - ▶ 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

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
- ▶ `time.strftime()`
 - ▶ 設定時間格式以字串形式輸出
 - ▶ `time.strftime('%Y-%m-%d %H:%M:%S', time.localtime())`

使用標準函數庫

- 使用sys模組
 - `sys.argv[0]`
 - 會回傳此程式檔案的位置與名稱
 - `sys.argv`
 - 帶入參數以list形式儲存
 - `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

- 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

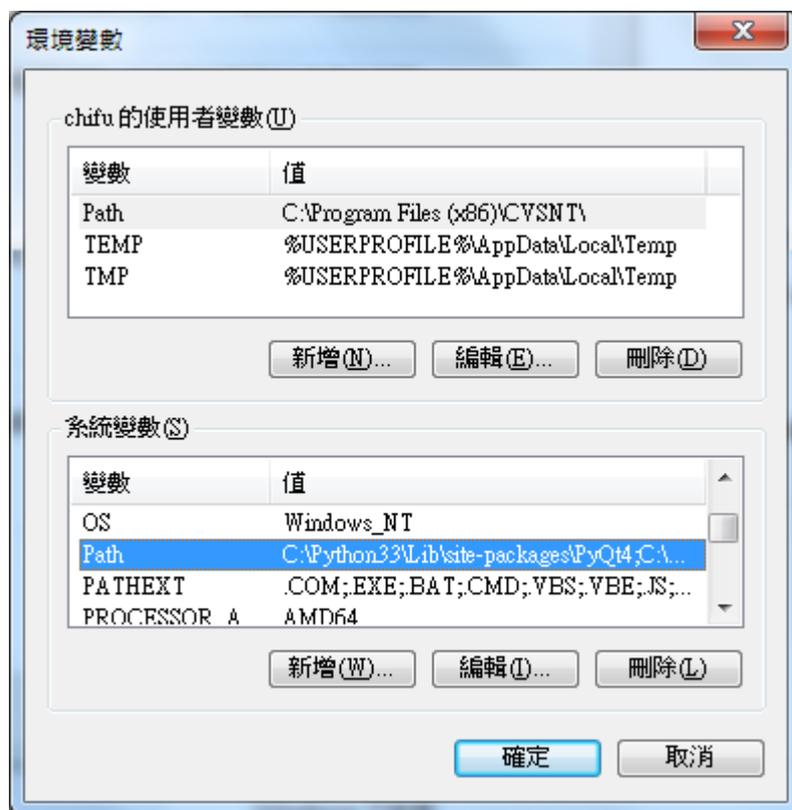
- ▶ 利用第三方安裝工具（如 `pip`，`easy_install`，`distribute` 等）
自動化工具安裝

安裝pip

- ▶ pip可正常工作在Windows、Mac OS、Unix/Linux等作業系統上
- ▶ 需要至少2.6+和3.2+的CPython版本
- ▶ python 2.7.9 和3.4以后的版本已經內建pip程式，所以不需要另外安裝。其他的版本可以使用以下命令進行安裝
 - ▶ 下載 <https://bootstrap.pypa.io/get-pip.py>
 - ▶ 安裝（Linux下需要root權限）
 - ▶ 於command line模式下執行，`python get-pip.py`
 - ▶ 加入環境變數（Windows）
 - ▶ 將 **C:\Python34\Scripts** 加入PATH
- ▶ 更多的安裝方法可以參考：
 - ▶ <https://pip.pypa.io/en/latest/installing.html>

設定環境變數

- ▶ 於系統變數中的Path，新增C:\PythonXY\Scripts (XY為版本號)



pip 使用方式

- ▶ 安裝套件
 - ▶ `pip install SomePackage` (自動安裝最新版本)
 - ▶ `pip install SomePackage==1.0.4` (指定版本)
 - ▶ `pip install 'SomePackage>=1.0.4'` (指定最小版本)
- ▶ 更新套件
 - ▶ `pip install -U SomePackage`
- ▶ 刪除套件
 - ▶ `pip uninstall SomePackage`

更多的使用方法可以參考：

https://pip.pypa.io/en/stable/user_guide.html

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>

PyQRCode

- ▶ 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>
- ▶ EX08_01.py

第三方函式庫

- ▶ **PyPI (Python Package Index)**，Python的第三方套件集散地，擁有數萬個套件，包羅萬象的各種需求幾乎都可以找到合適的套件。而使用 **pip** 套件管理程式，能夠方便我們管理、安裝這些套件。

常用的套件簡介

- ▶ 列出一些有名、常用到的或是有趣的第三方套件，這些套件可以單獨使用或是配合Django來完成強大的功能。
- ▶ 網站框架：
 - ▶ Django：完整強大的Web框架
 - ▶ Pyramid：另一個完整強大的Web框架
 - ▶ web2py：Google app engine 預設使用的框架
 - ▶ flask：相較而言是輕量的網站框架

常用的套件簡介

- ▶ 圖片處理：
 - ▶ PIL：Python Imaging Library，PIL 是 Python 下最有名的影像處理套件，由許多不同的模組所組成，並且提供了許多的處理功能，允許我們在簡單的 Python 程式裡進行影像的處理。
 - ▶ Pillow：PIL的fork版本，另一套圖形處理的套件。
- ▶ 科學計算：
 - ▶ Numpy：支援非常多的科學計算，包含矩陣運算、線性代數、傅立葉轉換等。大多數的科學計算領域上都派得上用場。
 - ▶ Matplotlib：可以畫出各種圖形如長條圖、分布圖、立體圖等...
 - ▶ pandas：提供處理特殊資料結構，具有數據處理與資料分析的功能
 - ▶ scikit-learn：機器學習的套件，包含內建的分群分類計算、回歸、統計等功能

常用的套件簡介

- ▶ 命令列操作及遠端登入：
 - ▶ **fabric**：可以直接撰寫shell命令，透過**fabric**執行，也支援遠端登入和自定義shell命令。
 - ▶ **paramiko**：提供遠端登入和部分指令呼叫的功能。
- ▶ 網路爬蟲：
 - ▶ **Scrapy**：Python爬蟲框架，可以輕易的與**Django**合作。
- ▶ 文件解析器：
 - ▶ **beautifulsoup**：可以處理HTML、XML等標記格式。
 - ▶ **lxml**：可以處理HTML、XML等標記格式，使用**xpath**選取內容。

常用的套件簡介

- ▶ 自然語言處理：
 - ▶ **nltk**：理論基礎及功能很強大的語言處理套件，但相對低階，需要對相關領域熟悉的才能流利使用。
 - ▶ **textblob**：較高階的分詞、分句、語言分析工具。
 - ▶ **jieba**：針對中文的分詞、分句、語言分析工具。
- ▶ 網路請求client端：
 - ▶ **requests**：可以模擬各種網路請求，如：**get**、**post**、**put**、**delete**等。
 - ▶ **pycurl**：在Linux、Unix系統上的命令：**curl**的python版本。

常用的套件簡介

- ▶ 背景程序，定時任務：
 - ▶ **celery**：可以輕易編寫、呼叫非同步及背景程序，或是執行定時任務 **cronjob**。
- ▶ 資料庫介接：
 - ▶ **mysql-python**：MySQL資料庫介接套件，Django連接MySQL時預設使用的套件。
 - ▶ **psycopg2**：PostgreSQL資料庫介接套件，Django連接PostgreSQL預設使用的套件。
 - ▶ **pymongo**：MongoDB資料庫介接套件。