

管理與安裝第三方函數庫

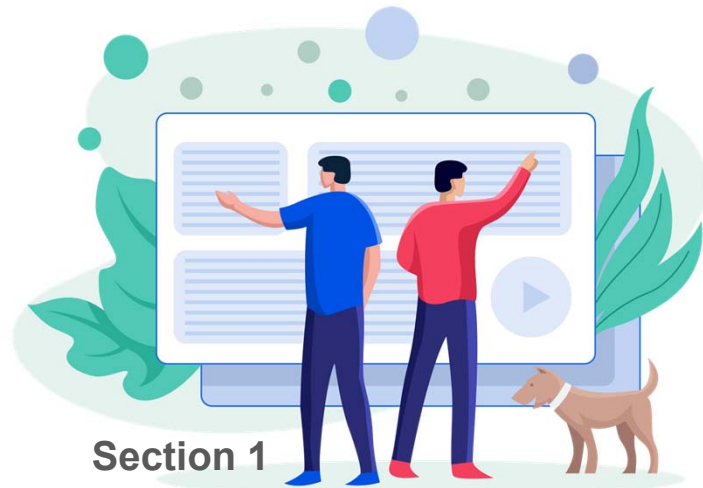
學習目標

- pip list 的用法
- pip install 的用法
- 使用第三方函數庫(Library) 相關模組



第三方程式庫(Third-party Libraries)

- 查詢與安裝第三方程式庫
- 使用使用第三方程式庫



Python程式庫

- 每一個 Python 版本都會安裝一些模組，稱為標準程式庫，提供了各式各樣領域內有用的工具與介面
- 除了標準程式庫以外，另有一個 Python Package Index (PyPI)，是官方的第三方倉庫，由其他 Python 程式設計師貢獻，從簡單到複雜的套件都有
 - ▣ 例如 numpy, scipy, pandas, matplotlib, scikit-learn
- PyPI 官網已經有幾十萬個套件了(packages)，並且持續的增加中
 - ▣ <https://pypi.org>

使用pip指令

- pip 指令是用來安裝 Python 相關套件(Packages) , 套件可以是自行下載的檔案 , 或是來自 PyPI 官網
- 確定可以在命令列(Command line) 執行 python 指令
 - `python -V` 或 `python --version`
- 確定可以在命令列(Command line) 執行 pip 指令
 - `pip -V` 或 `pip --version`
- 確定 pip 、 setuptools 與 wheel 套件是最新的
 - `python -m pip install --upgrade pip setuptools wheel`

顯示與搜尋套件

■ pip list 可以顯示已安裝的 Python 套件

- ▣ -o, --outdated : 列出過期的套件

- ▣ -u, --uptodate : 列出最新的套件

■ pip search 可以搜尋 PyPI 官網符合條件的套件

■ 範例

- ▣ pip list List installed packages, including editables.

- ▣ pip list -o List outdated packages

- ▣ pip list -u List uptodate packages

- ▣ pip search pandas

- ▣ pip search numpy

安裝套件 -1

- pip install 可以安裝 Python 套件
- 安裝最新版
 - pip install SomeProject
- 安裝特定版本
 - pip install SomeProject==1.4
- 安裝大於等於某一版本，且小於另一版本
 - pip install SomeProject>=1,<2

安裝套件 -2

■ 安裝某一版本，但有相容於特定版本

- `pip install SomeProject~=1.4.2`

- 以上代表安裝 “1.4.*” 的版本，但是要大於等於 “1.4.2” 版

■ 套件清單(Requirements files)

- `pip freeze > requirements.txt`

- 產生已安裝套件清單

■ 透過套件清單安裝多個套件

- `pip install -r requirements.txt`

更新與移除套件

■ 更新套件

- `pip install --upgrade SomeProject`

■ 透過套件清單更新多個套件

- `pip install --upgrade -r requirements.txt`

■ 移除套件

- `pip uninstall SomeProject`

■ 透過套件清單移除多個套件

- `pip uninstall -r requirements.txt`

第三方程式庫(Demo)

- 如何查詢與安裝第三方程式庫
- 如何使用第三方程式庫



使用第三方程式庫模組

■瞭解如何使用第三方程式庫模組



Python 虛擬環境套件：virtualenv

- C:\> pip install virtualenv # 安裝虛擬環境的套件virtualenv
- C:\> mkdir flask_learning
- C:\> cd flask_learning
- C:\flask_learning> virtualenv . # .表示在目前的目錄新建虛擬環境
- C:\flask_learning> cd Scripts
- C:\flask_learning\Scripts> activate # 進入虛擬環境
- (flask_learning) C:\flask_learning\Scripts> deactivate # 離開虛擬環境
- C:\flask_learning\Scripts>

Python 套件管理工具建立新環境

- (flask_learning) C:\flask_learning\> pip install flask
- (flask_learning) C:\flask_learning\> pip list # 顯示所有套件
- (flask_learning) C:\flask_learning\> pip freeze # 顯示安裝的套件
- (flask_learning) C:\flask_learning\> pip freeze > requirements.txt
將所有安裝的套件的名稱與版本存到requirements.txt文件內
- (flask_learning) C:\flask_learning\> pip install -r requirements.txt
安裝requirements.txt文件的套件

使用numpy套件 -1

■ NumPy 是使用 Python 進行科學計算的基礎套件

- 提供強大的N維陣列物件
- 複雜的陣列處理功能(例如不同維度陣列運算)
- 提供與 C / C ++ 和 Fortran 程式整合的工具
- 提供一系列有用的數學運算功能(例如線性代數、傅立葉變換和隨機亂數)

■ 安裝 numpy 套件

- `pip install numpy`
- `pip list`

使用numpy套件 -2

■使用 numpy 範例

```
□ >>> import numpy as np
>>> ar2 = np.arange(6,0,-1)
>>> ar1 = np.array([1,2,3,4,5])
>>> ar2 = np.arange(5,0,-1)
>>> ar1
array([1, 2, 3, 4, 5])
>>> ar2
array([5, 4, 3, 2, 1])
>>> ar1+ar2
array([6, 6, 6, 6, 6])
>>>
```

使用第三方程式庫模組(Demo)

■ 如何使用第三方程式庫



本章重點精華回顧

- pip 指令管理第三方程式庫(Third-party Libraries)
- 使用第三方程式庫(Third-party Libraries) 的模組



Lab: 管理與安裝第三方函數庫

- Lab01: 使用 pip 指令
- Lab02: 使用第三方程式庫(Third-party Libraries)

Lab01: 使用pip指令

1. 開啟命令列(cmd)，做以下練習

2. 檢查pip指令版本

```
pip -V  
pip --version
```

3. 更新pip、setuptools與wheel套件

```
python -m pip install --upgrade pip setuptools wheel
```

4. 顯示已安裝的套件

```
pip list  
pip list -o  
pip list -u
```

5. 搜尋套件

```
pip search pandas  
pip search numpy
```

6. 安裝套件

```
pip install numpy  
pip list
```

Lab02: 使用第三方程式庫(Third-party Libraries)

1. 啟動Python IDLE環境，做以下練習
2. 使用 “File/Open...” 開啟 “nparray.py” 程式，了解第三方程式庫的用法
3. 關閉nparray.py程式視窗