

第 6 章

如何使用Python 外部函式庫

6-1 什麼是外部函式庫

- 外部函式庫的概念
 - 需要安裝才能使用的則是外部函式庫。
- 不斷增加的外部函式庫
 - 將特定的功能彙整成函式庫，公佈於網路上供大眾使用，這樣的函式庫就是外部函式庫。



URL

<https://pypi.python.org/pypi>

PyPI 紹介 . 豆瓣 PyPI 開源庫網站(超過 7 萬)

▼ PyPI 的首頁

The screenshot shows the PyPI homepage with a search bar at the top right. On the left, there's a sidebar with links for 'PACKAGE INDEX' (Browse packages, Package submission, List trove classifiers, List packages, RSS (latest 40 updates), RSS (newest 40 packages), Python 3 Packages, PyPI Tutorial, PyPI Security, PyPI Support, PyPI Bug Reports, PyPI Discussion, PyPI Developer Info) and 'ABOUT' (NEWS, DOCUMENTATION, DOWNLOAD, COMMUNITY, FOUNDATION, CORE DEVELOPMENT). The main content area features a 'Package Authors' section with instructions for submitting packages and a 'Get Packages' section with instructions for installing them using pip or manually. To the right, a user profile for 'kamatari' is shown with sections for 'Welcome', 'Your details (Logout)', 'Status', and 'Nothing to report'. Below these sections is an 'Infrastructure' box with information about interoperability. At the bottom, a table lists the latest packages added to the index.

Updated	Package	Description
2015-11-28	libpad 0.1	A simple lib to run code on codepad and retrieve results.
2015-11-28	peersub 1.4.0	Use peerflix with subtitles
2015-11-28	sklearn-pandas 1.0.0	Pandas integration with sklearn
2015-11-28	mnj 0.1.4.dev0	Mnj (Mongo Energy) is a helper library to simplify PyMongo interaction
2015-11-28	git-release-tagger 0.0.1	Small CLI utility to enable the CI environment to communicate the deployment progress through the use of git tags.
2015-11-28	fnvhash 0.1.0	Pure Python FNV hash implementation
2015-11-28	collective.contentexport 1.0b4	Plone add-on to export dexterity content in various formats
2015-11-28	pybuilder 0.11.3.dev20151128114423	An extensible, easy to use continuous build tool for Python

外部函式庫的使用方式

- 第 1 個是安裝外部函式庫，另外 1 個則是讓該外部函式庫可以被 Python 所取用。

pip指令：安裝或移除外部函式庫

- pip 指令的使用方式

語法

pip install [想安裝的模組名稱]

模組名稱等同於函式庫名稱或套件名稱

語法

pip uninstall [想移除的模組名稱]

- 注意：要在cmd下執行；Mac為pip3

pip show

語法

pip show [想查詢的模組名稱]



pip show Pillow ↵

Metadata-Version: 2.0

Name: Pillow

Version: 3.3.1

Summary: Python Imaging Library (Fork)

Home-page: http://python-pillow.github.io/

Author: Alex Clark (Fork Author)

Author-email: aclark@aclark.net

License: Standard PIL License

Location: /Library/Frameworks/Python.framework/Versions/3.5/lib/
python3.5/site-packages

Requires:

Pip list：列出已安的外部函式

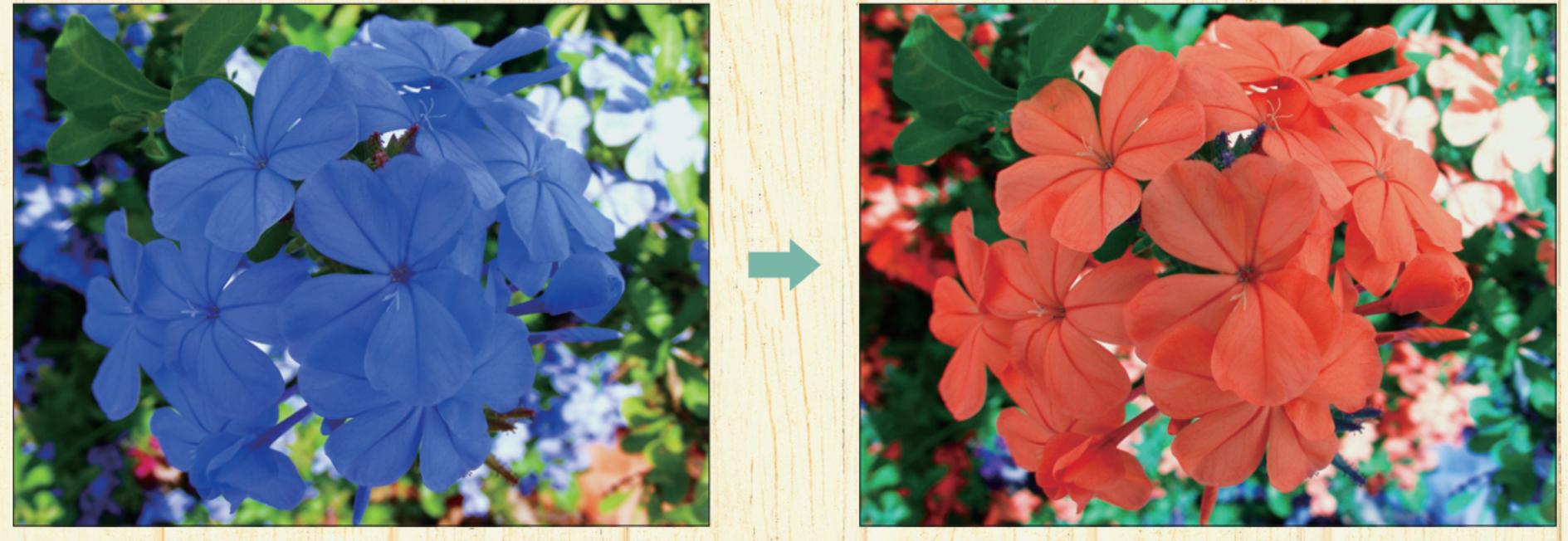
語法

```
pip list
```

6-2 使用外部函式庫的程式 (一)

- 如何以 Python 處理圖片

▼ 此小節的目標



使用Pillow函式庫

- 承繼了 PIL 的原始碼、在 2010 年開始進行開發的後繼函式庫，便是此次所要介紹的 Pillow 函式庫。
- Pillow 的官方網站
- URL <http://python-pillow.github.io/>

安裝Pillow



Windows 的操作方式

```
pip install Pillow
```



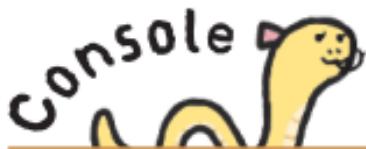
Mac 的操作方式

```
pip3 install Pillow
```

安裝後回到Python做import PIL

- 不會出現訊息表示安裝成功...

```
>>> from PIL import Image ↵
```



輸出結果

```
>>> from PIL import Image ↵  
>>>
```

沒有正確安裝Pillow

- 出現以下錯誤訊息...

Console 輸出結果

```
>>> from PIL import Image
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named PIL
```

Pillow 的功能

- 以 Python 程式顯示圖片



```
>>> from PIL import Image ↵
>>> image = Image.open('sample_image/flower.jpg') ↵
>>> image.show() ↵
```

處理三原色

- 將藍色和紅色互換後再顯示圖片

▼ 光的三原色





```
01 >>> from PIL import Image ↵
02 >>> image = Image.open('sample_image/flower.jpg') ↵
03 >>> r, g, b = image.split() ↵
04 >>> convert_image = Image.merge("RGB", (b, g, r)) ↵
05 >>> convert_image.show() ↵ ━━━━━━ 顯示互換顏色後的圖片
06 >>> convert_image.save('sample_image/rgb_to_bgr.jpg'); ↵ ━━
```

另存檔案

▼ 原本的圖片



▼ 處理後的圖片



將圖片轉換成黑白 / 灰階圖片



```
>>> from PIL import Image
>>> image = Image.open('sample_image/flower.jpg')
>>> black_and_white = image.convert('1')
>>> black_and_white.show()
>>> black_and_white.save('sample_image/b_and_w.jpg')
```



convert 的
參數 1，代表
1-bit 單位
元的黑白圖片

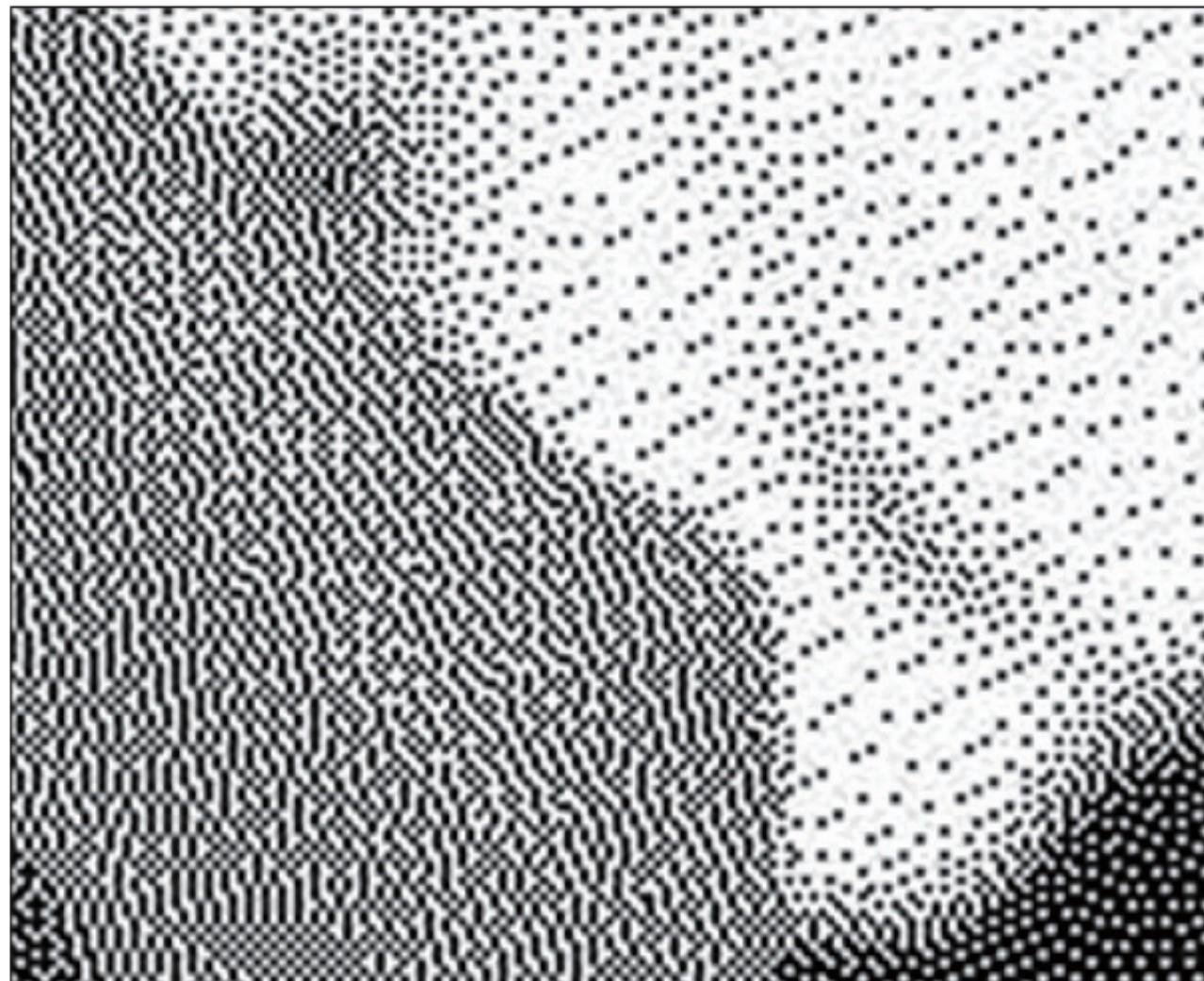


▼ 轉換後的黑白圖片



每個方塊都是1個畫素(pixwl)

▼ 圖片局部放大



把圖片轉成灰階



```
>>> from PIL import Image  
>>> image = Image.open('sample_image/flower.jpg')  
>>> gray_image = image.convert("L") •———— 轉換成灰階的圖片  
>>> gray_image.show()  
>>> gray_image.save('sample_image/gray_image.jpg') •———— 另存檔案
```

▼ 轉換後的灰階圖片





計算公式

$$L(\text{亮度}) = R(\text{紅}) * 0.299 + G(\text{綠}) * 0.587 + B(\text{藍}) * 0.114$$

將圖片旋轉90度

● 旋轉圖片方向



```
>>> from PIL import Image  
>>> image = Image.open('sample_image/flower.jpg')    顯示 90 度  
>>> image.transpose(Image.ROTATE_90).show() •———— 旋轉的圖片  
>>> image.transpose(Image.ROTATE_90).save('rotate_90.jpg') •—
```

儲存 90 度旋轉的圖片

▼ 旋轉 90 度後的圖片



旋轉圖片的參數

▼ 其它旋轉方式

參數	效果
Image.FLIP_LEFT_RIGHT	將圖片左右翻轉
Image.FLIP_TOP_BOTTOM	將圖片上下翻轉
Image.ROTATE_90	將圖片旋轉 90 度
Image.ROTATE_180	將圖片旋轉 180 度
Image.ROTATE_270	將圖片旋轉 270 度

※ 旋轉角度為逆時針方向

6-3 使用外部函式庫的程式（二）

- 用 Python 存取網路資料

- 這裡將會使用「**requests**」這個外部函式庫取得網路上的資訊。
。操作時請確認電腦有連上網路。

安裝requests函式庫

- 認識 requests 函式庫
- ▶ requests 介紹文件
 - 英文版 [URL](http://docs.python-requests.org/en/master/) <http://docs.python-requests.org/en/master/>
 - 簡中版 [URL](http://cn.python-requests.org/zh_CN/latest/) http://cn.python-requests.org/zh_CN/latest/



```
pip install requests
```

以python讀取某網頁內容



```
01 >>> import requests ↵
02 >>> r = requests.get('http://tw.yahoo.com') ↵
03 >>> print(r.text) ↵
```

使用pprint較易閱讀

● 解說



```
01 >>> import requests ↵
02 >>> import pprint ↵
03 >>> r = requests.get('http://tw.yahoo.com') ↵
04 >>> pprint.pprint(r.text) ↵
<!DOCTYPE html>
...省略
<meta http-equiv="X-UA-Compatible" content="chrome=1">
<meta name="description" content="最新 Yahoo 奇摩首頁，提供最方便的網站搜尋、即時新聞、生活資訊和 Yahoo 奇摩服務入口
">
...省略
```

打錯了...

- 常見的輸入錯誤

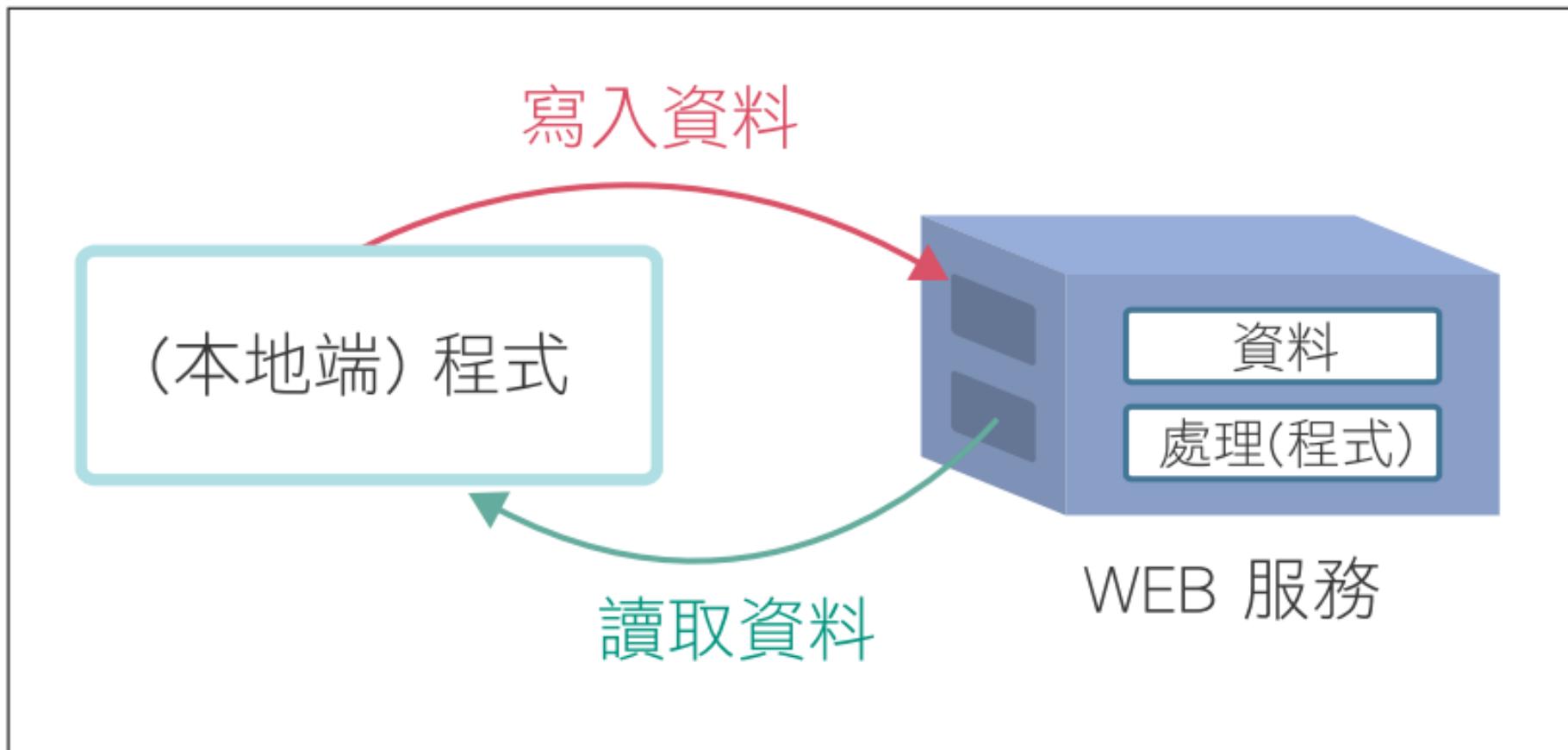
► 名稱輸入錯誤

```
NameError: name 'request' is not defined
```

利用requests存取Web API

- 什麼是 Web API

- ▼ Web API 的示意圖



某些網站會提供API，但不都會

- 利用 requests 連接 API 的方法（天氣預報篇）
- API是另外一個網址....如下....

► Weather Hacks

URL http://weather.livedoor.com/weather_hacks/

► Web API 的規格

URL http://weather.livedoor.com/weather_hacks/webservice

► 提供 RSS 的網頁（當中有日本各地區的編號）

URL http://weather.livedoor.com/forecast/rss/primary_area.xml

取得東京天氣資訊

- 130010是東京的代號，使用json()格式...



```
01 >>> import requests ↵
02 >>> import pprint ↵
03 >>> api_url = 'http://weather.livedoor.com/forecast/
           webservice/json/v1?city=130010' ↵
04 >>> weather_data = requests.get(api_url).json() ↵
05 >>> pprint.pprint(weather_data) ↵
```

善用get方法的params參數

```
>>> url = 'http://weather.livedoor.com/forecast/webservice/json/v1'  
>>> payload = {'city':'130010'}  
>>> weather_data = requests.get(url, params=payload).json()
```

取回的天氣資料



執行結果

```
{'copyright': {'image': {'height': 26,
                           'link': 'http://weather.livedoor.com/',
                           'title': 'livedoor 天氣情報',
                           'url': 'http://weather.livedoor.com/
img/cmn/livedoor.gif'},
                'width': 118},
     'link': 'http://weather.livedoor.com/',
     'provider': [{'link': 'http://tenki.jp/',
                   'name': '日本気象協会'}],
     'title': '(C) LINE Corporation'},
'description': {'publicTime': '2015-12-26T16:40:00+0900'},
```

Cont...

```
'text': '西日本や東日本は、西から高気圧に覆われています。一方、東北地方を低気圧が東に進んでいます。¥n'
        '【関東甲信地方】関東甲信地方は、晴れや曇りで、長野県の山沿いでは雪や雨の降っている所があります。¥n'
...
},
'forecasts': [ {'date': '2015-12-26',
    'dateLabel': '今日',
    'image': {'height': 31,
        'title': '晴れ',
        'url': 'http://weather.livedoor.com/
img/icon/1.gif',
        'width': 50} ,
```

Cont...

```
'telop': '晴れ',
'temperature': {'max': None, 'min': None}],
...省略
    'temperature': {'max': None, 'min': None}]],
'link': 'http://weather.livedoor.com/area/forecast/130010',
'location': {'area': '関東', 'city': '東京', 'prefecture': '東京都'},
'pinpointLocations': [{['link': 'http://weather.livedoor.com/area/
forecast/1310100',
    'name': '千代田区'},
...省略
'publicTime': '2015-12-26T17:00:00+0900',
'title': '東京都 東京 の天気'}
```

檢查資料有沒有設索引鍵 (key)

- 有8組可供查詢



```
>>> weather_data.keys() ↴  
dict_keys(['copyright', 'title', 'description', 'location',  
'publicTime', 'link', 'pinpointLocations', 'forecasts'])
```

► Weather Hacks > 網站服務的規格

URL http://weather.livedoor.com/weather_hacks/webservice

用forecast查詢第一筆資料



```
>>> pprint.pprint(weather_data['forecasts'][0]) ↵
{'date': '2016-10-02',
 'dateLabel': '今日',
 'image': {'height': 31,
            'title': '晴れ',
            'url': 'http://weather.livedoor.com/img/icon/1.gif',
            'width': 50},
 'telop': '晴れ',
 'temperature': {'max': None, 'min': None}}
```

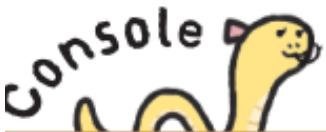
寫一支py程式在cmd視窗執行

● 輕鬆取得天氣預報資訊（基本篇）



get_weather1.py

```
import requests
api_url = 'http://weather.livedoor.com/forecast/webservice/json/v1'
payload={'city':'130010'}
weather_data = requests.get(api_url, params=payload).json()
print(weather_data['forecasts'][0]['dateLabel'] + '的天氣是：' +
weather_data['forecasts'][0]['telop'])
```



python get_weather1.py ↵

今天的天氣是：晴曇（晴時多雲）

使用requests.get取得資料

● 解說

```
weather_data = requests.get(api_url, params=payload).json()
```

● 無法順利執行？



輸出結果

```
python: can't open file 'get_weather1.py': [Errno 2] No such  
file or directory
```

取得三天的天氣資訊

● 輕鬆取得天氣預報資訊（應用篇）



get_weather2.py / py

```
import requests
api_url = 'http://weather.livedoor.com/forecast/webservice/json/v1'
payload = {'city':'130010'}
weather_data = requests.get(api_url, params = payload).json()
print(weather_data['forecasts'][0]['dateLabel'] + '的天氣是：'
+ weather_data['forecasts'][0]['telop'])
print(weather_data['forecasts'][1]['dateLabel'] + '的天氣是：'
+ weather_data['forecasts'][1]['telop'])
print(weather_data['forecasts'][2]['dateLabel'] + '的天氣是：'
+ weather_data['forecasts'][2]['telop'])
```

用for迴圈來改寫...



get_weather3.py

```
import requests
api_url = 'http://weather.livedoor.com/forecast/webservice/json/v1'
payload={'city' : '130010'}
weather_data = requests.get(api_url, params = payload).json()
for weather in weather_data['forecasts']:
    print(weather['dateLabel'] + '的天氣是：' + weather['telop'])
```

在cmd下執行...



```
python3 get_weather3.py ↵
```

今天的天氣是：晴時々曇

明天的天氣是：晴れ

明後天的天氣是：晴時々曇

● 成為達人的重點



get_weather4.py

```
import requests
api_url = 'http://weather.livedoor.com/forecast/webservice/json/v1'
payload = {'city' : '130010'}
weather_data = requests.get(api_url, params = payload).json()
for weather in weather_data['forecasts']:
    print(weather)
```



輸出結果

```
{'temperature': {'min': None, 'max': {'celsius': '26', 'fahrenheit': '78.8'}}, 'image': {'width': 50, 'url': 'http://weather.livedoor.com/img/icon/15.gif', 'height': 31, 'title': '雨'}, 'date': '2016-10-03', 'dateLabel': '今天', 'telop': '雨'}
```

利用 requests 連接 Wikipedia API

- 利用 requests 連接 API 的方法（Wikipedia 篇）

- ▶ Wikipedia

- URL <https://zh.wikipedia.org/>

- ▶ MediaWiki 的主頁（繁體中文）

- URL <https://www.mediawiki.org/wiki/MediaWiki/zh-tw>

- ▶ MediaWiki API 的說明

- URL https://www.mediawiki.org/wiki/API:Main_page/zh



● 取得 Wikipedia 的資訊



```
01 >>> import requests, pprint ↵
02 >>> api_url = 'https://zh.wikipedia.org/w/api.php' ↵
03 >>> api_params = {'format':'json', 'action':'query', 'titles': ↵
    '椎名林檎', 'prop':'revisions', 'rvprop':'content'} ↵
04 >>> wiki_data = requests.get(api_url, params=api_params).json() ↵
05 >>> pprint.pprint(wiki_data) ↵
```



● 關於 MediaWiki API 的查詢參數

- 1. format = json
- 2. action = query
- 3. titles = 椎名林檎
- 4. prop = revisions
- 5. rvprop = content

► action 可指定的項目

URL <https://zh.wikipedia.org/w/api.php>

► action 指定為「query」時, prop 可指定的項目

URL <https://zh.wikipedia.org/w/api.php?action=help&modules=query>

► prop 指定為「revisions」時, rvprop 可指定的項目

URL <https://zh.wikipedia.org/w/api.php?action=help&modules=query+revisions>

► 可用來測試 API 的網頁

URL <https://zh.wikipedia.org/wiki/Special:ApiSandbox>

▼ Wikipedia API 沙盒的頁面



● 取得 Wikipedia 的資訊（應用篇）



wiki1.py / py

```
01 import requests  
02 import codecs •———— Windows 需要匯入此模組  
03 api_base_url = 'https://zh.wikipedia.org/w/api.php'  
04 api_params = {'format':'xmlfm', 'action':'query', 'titles':  
    '椎名林檎', 'prop':'revisions', 'rvprop':'content'}  
05 wiki_data = requests.get(api_base_url, params=api_params)  
06 fo = codecs.open('C:\\\\Users\\\\使用者帳號\\\\Desktop\\\\wiki.html',  
    'w', 'utf-8')  
07 fo = open('/Users/(使用者帳號)/Desktop/wiki.html', 'w') •———— Mac 的寫法  
08 fo.write(wiki_data.text)  
09 fo.close() •———— Windows 的寫法
```

2 行擇一使用即可

以瀏覽器打開取回的內容...

● MediaWiki API 的結果

```
<?xml version="1.0"?>
<api batchcomplete="">
<query>
  <pages>
    <page _idx="263745" pageid="263745" ns="0" title="椎名林檎">
      <revisions>
        <rev contentformat="text/x-wiki" contentmodel="wikitext"
          xml:space="preserve">{{藝人
| 姓名      = 椎名林檎
| 類型      = 女歌手
...省略

```

● 傳遞參數給檔案的方法



try_sys.py / py

```
import sys  
print(sys.argv)
```



```
python try_sys.py one two three four! ↵  
['try_sys.py', 'one', 'two', 'three', 'four!']
```

取得 Wikipedia 的資訊（續）

語法

```
python 程式檔案名稱 想搜尋的標題
```

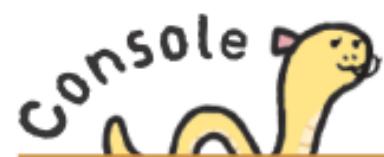
寫一支可以指定名稱的程式..wiki2.py



↓ wiki2.py py

```
01 import requests, sys
02 import codecs •———— Windows 需要匯入此模組
03 search_word = sys.argv[1]
04 api_url = 'https://zh.wikipedia.org/w/api.php'
05 api_params = {'format':'xmlfm', 'action':'query',
                 'prop':'revisions', 'rvprop':'content'}
06 api_params['titles'] = search_word
07 wiki_data = requests.get(api_url, params=api_params)
08 fo = codecs.open('C:\\\\Users\\\\使用者帳號\\\\Desktop\\\\'+ search_word
                  + '.html', 'w', 'utf-8') •———— Windows 的寫法
09 #fo = open('/Users/使用者帳號/Desktop/'+ search_word + '.html',
           'w') •———— Mac 的寫法
10 fo.write(wiki_data.text)
11 fo.close()
```

第 8、9 行擇一使用即可



python wiki2.py 東京事變 ↵

Web API使用上需要注意的地方

- Web API 會不斷變動

- 免費使用的 Web API 有可能在毫無預警的狀況下改變其規格，因此，原本正常運作的 Web API 相關程式，可能會突然無法使用，這樣的情形相當常見。

- 注意 Web API 的連線頻率

- 對 Web API 以程式送出大量的連線要求，即使我們這些 Web API 的正常使用者沒有惡意，也可能造成類似 DoS 攻擊的效果。因此，不同的 Web API 會有不同的預防措施，也許是限制單位時間內的使用次數，而需付費的 Web API 可能會按照費用等級限制使用的次數。

6-4 使用外部函式庫的程式（三）

- 以 Python 收集資訊

- 利用 Python 從網路上抓取資訊的方法。而這樣獲取資訊的動作有個專門用語，稱為「Crawling」和「Scraping」

Crawling 和 Scraping

- 從網站直接抓取資料的動作稱為 **Crawling**, 因為從各頁面獲取資料的過程有如趴在地上爬行, 所以使用了爬行 **Crawl** 這個英文單字。
- **Scraping** 是根據 **Crawling** 所獲取的原始資料, 從中篩選出需要的部分、甚至對資料進行分析重組, 成為更有價值的資訊。**Scrape** 這個單字具有「刮削、挖出」的意思。

認識 BeautifulSoup4

- BeautifulSoup4 是設計用來執行 Scraping 的模組
- ▶ BeautifulSoup4 的官方網站
 - URL <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

BeautifulSoup4 的安裝方式



```
pip install beautifulsoup4 ↵
```



```
>>> from bs4 import BeautifulSoup ↵
>>> soup = BeautifulSoup("<html> Lollipop </html>", "html.parser") ↵
```

以 BeautifulSoup4 進行 Scraping



```
01 >>> import requests ↵
02 >>> from bs4 import BeautifulSoup ↵
03 >>> html_data = requests.get('      http://tw.yahoo.com') ↵
04 >>> soup = BeautifulSoup(html_data.text,"html.parser") ↵
05 >>> soup.title ↵
<title>Yahoo 奇摩</title>
```

● 對「Yahoo 奇摩科技新聞」的 RSS 進行Scraping

► Yahoo 奇摩可訂閱的 RSS

URL <https://tw.info.yahoo.com/rss/>

▼ Yahoo 奇摩訂閱 RSS 的畫面



▼ 按下 **RSS** 按鈕後 RSS 格式顯示的新聞

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<rss xmlns:media="https://search.yahoo.com/mrss/" version="2.0">
  <channel>
    <title>科技新聞 - 頭條新聞 - Yahoo奇摩新聞</title>
    <link>https://tw.news.yahoo.com/technology/</link>
    <description>瀏覽 Yahoo奇摩新聞上的最新科技頭條新聞。尋找最新新聞報導，包括科技頭條新聞的相關分析與意見。</description>
    <language>zh-Hant-TW</language>
    <copyright>Copyright (c) 2016 Yahoo! Inc. All rights reserved</copyright>
    <pubDate>Thu, 06 Oct 2016 12:55:37 +0800</pubDate>
    <ttl>5</ttl>
    <image>
      <title>科技新聞 - 頭條新聞 - Yahoo奇摩新聞</title>
      <link>https://tw.news.yahoo.com/technology/</link>
      <url>
        https://s.yimg.com/rz/d/yahoo_news_zh-Hant-TW_s_f_p_bestfit_news.png
      </url>
    </image>
    <item>
      <title>廣告藏病毒？Spotify又傳醜聞</title>
      <description>
        <p><a href="https://tw.news.yahoo.com/spotify-%E5%8D%8B%BB%E7%89%88%E5%86%8D%E8%A2%AB%E6%8F%AD%E9%80%8F%E9%81%8E%E5%BB%A3%E5%91%8A%E5%90%91%E7%94%A8%E6%88%B6%E6%95%A3%E6%92%045537498.html"></a>Spotify 的 Windows 版本曾發生廣告散播病毒程式的醜聞，以致大量用戶的電腦受害，如今情況再次發生，同樣是出現在免費版本 ...</p><br clear="all"/>
      </description>
      <link>
        https://tw.news.yahoo.com/spotify-%E5%8D%8B%BB%E7%89%88%E5%86%8D%E8%A2%AB%E6%8F%AD%E9%80%8F%E9%81%8E%E5%BB%A3%E5%91%8A%E5%90%91%E7%94%A8%E6%88%B6%E6%95%A3%E6%92%045537498.html
      </link>
```

Yahoo 奇摩科技新聞」的 RSS 網頁是 XML

- XML 標籤具有下列這些特點

- 開始標籤**<title>**和結束標籤**</title>**, b名稱前面有加上b/斜線符號的標籤為結束標籤。
- 開始標籤和結束標籤會成對出現。
- 標籤之間會形成巢狀結構。舉例來說, b在b**<channel>**b的開始和結束標籤之間可以看到**<title>**b標籤,b也就是「titleb標籤被包含於bchannelb標籤中」的狀態。



```
01 >>> import requests ↵
02 >>> from bs4 import BeautifulSoup ↵
03 >>> yahoo_tech_news_xml = requests.get('https://tw.news.yahoo.
com/rss/technology') ↵
04 >>> soup = BeautifulSoup(yahoo_tech_news_xml.text, "html.parser") ↵
05 >>> type(soup) ↵
<class 'bs4.BeautifulSoup'>
06 >>> soup.findAll('item') ↵
[<item>
  <title>廣告藏病毒 ? Spotify 又傳醜聞</title>
  <description>...省略</description>
  <link>...省略</link>
  <pubDate>Thu, 06 Oct 2016 12:55:37 +0800</pubDate>
  <source url="http://qooah.com/">Qooah</source>
  <guid isPermaLink="false">...省略</guid>
  <media:content url="...省略"/>
  <media:text type="html">...省略</media:text>
  <media:credit role="publishing company"/>
  </item>,
  ...省略
  </item>
]
```



```
>>> for news in soup.findAll('item'):
...     tab print(news.title)
...
<title>廣告藏病毒？Spotify 又傳醜聞</title>
...省略
<title>白太陽現象 小冰河期恐來臨</title>
```

```
...     tab print(news.title.string)
```

對電玩網站進行Scraping

- ▶ 巴哈姆特網站上的 Android 遊戲排行榜

URL <https://acg.gamer.com.tw/billboard.php?t=2&p=Android>

● 以開發人員工具查閱HTML

► Chrome 的操作方式（開發人員工具）：

Windows：按下鍵盤最上排的 `F12` 鍵

Mac：同時按下 `command` + `option` + `I` 鍵

► Firefox 的操作方式（開發者工具）：

Windows：同時按下 `Ctrl` + `shift` + `I` 鍵

Mac：同時按下 `command` + `option` + `K` 鍵

► Safari 的操作方式（"開發"選單）：

需要先在 Safari 偏好設定的進階中，勾選「在選單列中顯示「開發」選單」

Mac：同時按下 `command` + `option` + `U` 鍵

► Internet Explorer 的操作方式（開發人員工具）：

Windows：按下 `F12` 鍵

…省略

```
<div class="ACG-mainbox1">
…省略
<h1 class="ACG-maintitle">
<span class="ACG-mainnumber">No.1</span>
<a href="http://acg.gamer.com.tw/…省略">Pokemon GO</a>
…省略
</h1>
…省略
</div>
…省略
```

取出電玩網站的資料...

- ACG-mainbox1是目標標籤 (ref.前一頁)



```
>>> import requests ↵
>>> from bs4 import BeautifulSoup ↵
>>> game_raking_html = requests.get('https://acg.gamer.com.tw/
    billboard.php?t=2&p=Android') ↵
>>> game_raking_html.encoding = 'UTF-8' ↵ └── 將編碼改為 UTF-8
>>> soup = BeautifulSoup(game_raking_html.text, "html.parser") ↵
>>> soup.find(class_='ACG-mainbox1').find(class_='ACG-
    maintitle').find('a').string ↵
'Pokemon GO'
```

- ACG-mainbox1標籤放遊戲名次

- ACG-maintitle標籤放遊戲名稱



```
>>> for game in soup.findAll(class_='ACG-mainbox1'): ↵
...   tab print(game.find(class_='ACG-mainumber').string + ' ' +
      game.find(class_='ACG-maintitle').find('a').string) ↵
...
No.1 Pokemon GO
No.2 劍俠情緣手機版
No.3 艾爾戰紀
...省略
No.19 Shadowverse
No.20 軒轅劍之天之痕
```

6-5 自行製作外部函式庫

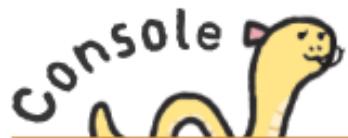
● 模組的製作方式

- 1 撰寫想製作成模組的處理程式
- 2 將完成的程式以喜歡的檔名儲存成檔案



newyear1.py py

```
print('happy new year !!')
```



```
>>> import newyear1 ↵  
happy new year !!
```

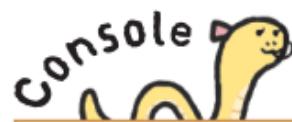


```
>>> dir() ↵
```



輸出結果

```
['__builtins__', '__cached__', '__doc__', '__loader__', '__name__',  
'__package__', '__spec__', 'newyear1']
```



```
>>> print(__name__) ↵  
__main__
```



newyear2.py

```
print(__name__)
```



```
>>> import newyear2 ↵  
newyear2
```



newyear3.py

```
if __name__ == '__main__':
    print('happy new year !!')
```



```
>>> import newyear3 ↵
>>>
```

製作更實用的模組



monthname.py / py

```
def japanese(month):
    month_name = {
        1:"睦月", 2:"如月", 3:"生", 4:"卯月", 5:"月", 6:"水無月",
        7:"文月", 8:"葉月", 9:"長月", 10:"神無月", 11:"霜月", 12:"師走"
    }
    try:
        response = month_name[month]
    except:
        response = '請輸入月份數字。'

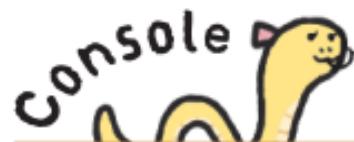
    return response
if __name__ == '__main__':
    print('此為模組檔案，請 import 匯入後再使用')
```



```
>>> import monthname ↵
>>> monthname.japanese(1) ↵
'睦月'
>>> monthname.japanese(8) ↵
'葉月'
>>> monthname.japanese(12) ↵
'師走'
```



```
>>> monthname.japanese(15)
'請輸入月份數字。'
>>> monthname.japanese('nyan')
'請輸入月份數字。'
```



python monthname.py ↵

此為模組檔案，請 import 匯入後再使用

關於函式庫、套件和模組

- 模組 (Module)

語法

```
import (檔案主檔名)
```

- 套件 (Package)

語法

```
import yyy.aaa
```

- 函式庫 (Library)

- 「lib」資料夾內的模組和套件檔案都屬於 Python 的標準函式庫。

● 函式庫的程式檔案放在哪裡？



```
>>> import calendar ↵
>>> print(calendar.__file__) ↵
/Library/Frameworks/Python.framework/Versions/3.5/lib/
    python3.5/calendar.py
```

- ▶ 上面是 Mac 的結果, Windows 的結果為 C:\Users\使用者帳號\AppData\Local\Programs\Python\Python35-32\lib\calendar.py）。

```
def isleap(year):
    """Return True for leap years, False for non-leap years."""
    return year % 4 == 0 and (year % 100 != 0 or year % 400 == 0)
```