Week 8

觀念補充說明

善用匯入功能: Import

- Python以模組為執行單位，若要用到別支模組以import進行匯入
- 別支模組可以是 .py檔、 .dll檔或其他可執行的程式，直譯器已經認識的不需要放副檔名
- 模組裡可能有函式、類別、整數、串列、其他模組等等。這些我們統稱為物件，所以物件有可能是模組、函式、類別、串列等。
- 模組內還包含模組的有另一個名稱叫「套件」

如何了解現況

- pip list 會顯示出你已經安裝的套件與版本
- pip show bs4 顯示已安裝的bs4版本資訊
- dir() 顯示現在已經被匯入的套件
- dir(bs4) 顯示bs4裡面的物件有哪些(比help精簡)
- 以bs4和random為例

套件 Package

- 套件中存放了多個模組，就像一個資料夾存放了很多檔案一樣。只要有 __init__.py 檔案的資料夾就會被視為 python 套件。
- 標準函式庫（standard library）／內建函式庫（built-in library）是安裝 python 時一併安裝的套件。如：math, random, time, calendar, datetime, turtle。
- 外部函式庫（external library）是需要另外安裝的模組與套件。
- import importlib as imp
  print(imp.util.find_spec('numpy'))

package_example.py # 主程式檔
package_example/ # 套件
    __init__.py
    info.py
    gui/ # 子套件
        __init__.py
        menu.py
        canvas.py
        foo.py
        bar.py
    formats/ # 子套件
        __init__.py
        jpg.py
        png py
        bmp.py
        foo.py
        bar.py
    tools/ # 子套件
        __init__.py
        rotate.py

```
import package_example.gui
print(type(package_example))
print(package_example.__name__)
print(package_example.gui.__name__)
print(gui.__name__)
```

```
import package_example.gui as gui
print(gui.__name__)
```

Or

```
from package_example import gui
print(gui.__name__)
```

如何import自己寫的 .py 檔

```
>>> import sys
>>> sys.path
['', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\Lib\\idlelib', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\python38.zip', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\DLLs', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\lib', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\lib\\site-packages', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\lib\\site-packages\\beautifulsoup4-4.9.0-py3.8.egg', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\lib\\site-packages\\soupsieve-2.0-py3.8.egg']
>>> sys.path.append('C:\Users\AU\AppData\Local\MineTest')   此為實際路徑
SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 2-3: truncated \UXXXXXXXX escape
>>> sys.path.append('/Users/AU/AppData/Local/MineTest')
>>> import gametest
請問你出甚麼拳(1是剪刀、2是石頭、3是布):
```

```
>>> sys.path.append('C:\Users\AU\AppData\Local\MineTest')
SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in p
osition 2-3: truncated \UXXXXXXXX escape
>>> sys.path.append('/Users/AU/AppData/Local/MineTest')
>>> import gametest
請問你出甚麼拳(1是剪刀、2是石頭、3是布):1
隨機數是  1
你出 剪刀; 電腦出 剪刀    結果...平手
play again?  n
game is over
>>> sys.path.append(r'C:\Users\AU\AppData\Local\MineTest')
>>> sys.path
['', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\Lib\\i
dlelib', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\py
thon38.zip', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32
\\DLLs', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\li
b', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32', 'C:\\U
sers\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\lib\\site-package
s', 'C:\\Users\\AU\\AppData\\Local\\Programs\\Python\\Python38-32\\lib\\si
te-packages\\beautifulsoup4-4.9.0-py3.8.egg', 'C:\\Users\\AU\\AppData\\Loc
al\\Programs\\Python\\Python38-32\\lib\\site-packages\\soupsieve-2.0-py3.8
.egg', '/Users/AU/AppData/Local/MineTest', 'C:\\Users\\AU\\AppData\\Local\
\MineTest']
>>>
```

```
>>> dir(gametest)
['__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__',
 '__package__', '__spec__', 'again', 'computer', 'game_dic', 'outresult', 'player
', 'random', 'result_dic', 'win']
>>>
```

```
game_dic={1:"剪刀",2:"石頭",3:"布"}
result_dic={-1:"你輸",0:"平手",1:"你贏"}
def outresult(you, computer, result):
    print("你出 "+game_dic[you]+"; 電腦出 "+game_dic[computer] + "   結果..."+result_dic[result])
import random
again='y'
while(again == 'y' or again == 'Y'):
    player = eval(input('請問你出甚麼拳(1是剪刀、2是石頭、3是布):'))
    computer=random.randint(1,3)
    print('隨機數是 ',computer)
    if(player == computer):
        win = 0
    elif(player == (computer+1) or player == (computer-2)):
        win = 1
    else:
        win = -1
    outresult(player, computer, win)
```

PIP 的指令(在CMD執行)

- pip list： 可以用來列出目前已安裝的套件與版本。
- pip install 套件名稱： 可以用來安裝套件。
- pip show 套件名稱： 可以用來查詢已安裝的套件。
- pip uninstall 套件名稱： 解除安裝套件。

第三方套件 Third Party Package

- Django, Web2py, Flask：web框架，快速架設網站
- Numpy：陣列與科學計算；矩陣運算、FFT、線性代數
- SciPy：科學計算
- Pandas：數據處理與資料分析
- Matplotlib：2D視覺化工具
- PyGtk, PyQt, WxPython, tkinter：GUI程式開發
- BeautifulSoup：HTML/XML解析器
- Pillow/PIL（舊）：圖形處理
- PyGame：多媒體與遊戲開發
- Requests：存取網際網路資料
- Scikit-learn：機器學習套件
- Scrapy：網路爬蟲整合工具

爬蟲前請注意

- 每一次的爬蟲動作都會增加對方的負擔
- 請在連續的請求中加入延遲
- 如果對方往頁已有提供下載區或提供API，請勿爬
- 提供user agent的個人資訊，讓對方在需要時可以通知你

爬蟲相關套件

- 爬取
  - urllib、requests、selenium
- 剖析
  - re、ref、beautifulsoup
- 資料整理
  - pandas
- 加速爬蟲
  - Multiprocessing(多CPU)、aiohttp(減少等待回應的時間)
- 整合套件
  - scrapy: 整合前述功能

Python 存取網站方式

- 靜態網頁擷取
- 動態網頁擷取
- Logging 模組

以維基百科為練習。
https://en.wikipedia.org/

```
>>> import requests
>>> r = requests.get('https://en.wikipedia.org/')
>>> r
<Response [200]>
>>> r.ok
True
>>> type(r)
<class 'requests.models.Response'>
>>> len(r.text)
75579
>>>
```

html的狀態碼: 2xx成功、
3xx重新導向
4xx用戶端錯誤
5xx伺服器端錯誤

靜態網頁

- 靜態網頁中不包含任何.js檔
- 伺服器回傳的時候就是完整的網頁
- 此時網路爬蟲程式中最重要的部分就是如何解析網頁的HTML檔案。
- HTML定義元素
  - Tag(標籤)就是元素的名字
  - Attribute(屬性)描述元素的屬性
  - Content(內容)則是元素的內容

靜態網頁擷取

- HTML常用標籤
  - <!--註解文字-->
  - 粗體文字、斜體文字、<u>底線文字</u>
  - <head>網站的開頭</head>
  - <body>網頁檔案之主體</body>
  - 網頁檔案的一個區塊，裡面可以包含很多元素
  - <title>網頁標題名稱（顯示於視窗標題和分頁之名稱）</title>
  - <h1>HTML內文標題1(最高級)標題，通常也是標題中最重要的</h1>
  - <a href>超連結，跟著href屬性一起合用
  - <form>使用者輸入之HTML表單</from>
  - <tr> / <td> ：定義表格時最常用的兩個標籤， <tr> 是列， <td> 則是欄。

靜態網頁擷取

- 網路爬蟲常用的屬性
  - id：獨一無二的代表網頁。
  - class：描述類似的元素的歸類。
  - href：超連結，有超連結我們就可以繼續深入下一個連結。
- 靜態網頁網路爬蟲步驟
  - 獲取網站
  - 分析網站
  - 儲存結果

BEAUTIFUL SOUP

- 解析網站的模式
- 以html.parser模式解析
- 以lxml與xml模式解析

```
>>> dir(bs4.BeautifulSoup)
['ASCII_SPACES', 'DEFAULT_BUILDER_FEATURES', 'NO_PARSER_SPECIFIED_WARNING', 'ROOT_TAG_NAME', '__bool__', '__call__', '__class__', '__contains__', '__copy__', '__delattr__', '__delitem__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattr__', '__getattribute__', '__getitem__', '__getstate__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__module__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', '__unicode__', '__weakref__', '_all_strings', '_check_markup_is_url', '_decode_markup', '_feed', '_find_all', '_find_one', '_is_xml', '_lastRecursiveChild', '_last_descendant', '_linkage_fixer', '_popToTag', '_should_pretty_print', 'append', 'childGenerator', 'children', 'clear', 'decode', 'decode_contents', 'decompose', 'decomposed', 'descendants', 'encode', 'encode_contents', 'endData', 'extend', 'extract', 'fetchNextSiblings', 'fetchParents', 'fetchPrevious', 'fetchPreviousSiblings', 'find', 'findAll', 'findAllNext', 'findAllPrevious', 'findChild', 'findChildren', 'findNext', 'findNextSibling', 'findNextSiblings', 'findParent', 'findParents', 'findPrevious', 'findPreviousSibling', 'findPreviousSiblings', 'find_all', 'find_all_next', 'find_all_previous', 'find_next', 'find_next_sibling', 'find_next_siblings', 'find_parent', 'find_parents', 'find_previous', 'find_previous_sibling', 'find_previous_siblings', 'format_string', 'formatter_for_name', 'get', 'getText', 'get_attribute_list', 'get_text', 'handle_data', 'handle_endtag', 'handle_starttag', 'has_attr', 'has_key', 'index', 'insert', 'insert_after', 'insert_before', 'isSelfClosing', 'is_empty_element', 'new_string', 'new_tag', 'next', 'nextGenerator', 'nextSibling', 'nextSiblingGenerator', 'next_elements', 'next_siblings', 'object_was_parsed', 'parentGenerator', 'parents', 'parserClass', 'popTag', 'prettify', 'previous', 'previousGenerator', 'previousSibling', 'previousSiblingGenerator', 'previous_elements', 'previous_siblings', 'pushTag', 'recursiveChildGenerator', 'renderContents', 'replaceWith', 'replaceWithChildren', 'replace_with', 'replace_with_children', 'reset', 'select', 'select_one', 'setup', 'smooth', 'string', 'string_container', 'strings', 'stripped_strings', 'text', 'unwrap', 'wrap']
```

BeautifulSoup

解析器	使用方法
Python's html.parser	BeautifulSoup(markup, "html.parser")
lxml's HTML parser	BeautifulSoup(markup, "lxml")
lxml's XML parser	BeautifulSoup(markup,"lxml-xml") BeautifulSoup(markup, "xml")
html5lib	BeautifulSoup(markup,"html5lib")

```
>>> from bs4 import BeautifulSoup
>>> soup = BeautifulSoup(r.text, 'html.parser')
>>> print(soup.title)
<title>Wikipedia, the free encyclopedia</title>
>>> soup.find_all('div', class_='otd-footer')
[<div class="otd-footer hlist noprint" style="text-align: right;">
<ul><li><b><a href="/wiki/Wikipedia:Selected_anniversaries/April" title="Wikipedia:Selected ann
iversaries/April">Archive</a></b></li>
<li><b><a class="extiw" href="https://lists.wikimedia.org/mailman/listinfo/daily-article-l" tit
le="mail:daily-article-l">By email</a></b></li>
<li><b><a href="/wiki/List_of_historical_anniversaries" title="List of historical anniversaries
">List of historical anniversaries</a></b></li></ul>
</div>]
```

** 這裡的語法是用" class_"，因為class是python內建的關鍵字之一，所以beatifulsoup使用class_替代 **

```
>>> soup.find("h1")
<h1 class="firstHeading" id="firstHeading" lang="en">Main Page</h1>
>>> soup.find("h1").contents
['Main Page']
>>> soup.find('h1').contents
['Main Page']
>>> soup.find_all('h1').contents
Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    soup.find_all('h1').contents
  File "C:\Users\Li-Ling\AppData\Local\Programs\Python\Python38-32\lib\site-packages\beautifuls
oup4-4.9.0-py3.8.egg\bs4\element.py", line 2127, in __getattr__
    raise AttributeError(
AttributeError: ResultSet object has no attribute 'contents'. You're probably treating a list o
f elements like a single element. Did you call find_all() when you meant to call find()?
```

```
>>> for title in soup.find_all('h1'):
        print(title.contents)


['Main Page']
>>>
```

```
>>> soup.find("h2")
<h2 id="mp-tfa-h2" style="margin:0.5em; background:#cef2e0; font-family:inherit; font-size:120%
; font-weight:bold; border:1px solid #a3bfb1; color:#000; padding:0.2em 0.4em;"><span id="From_
today.27s_featured_article"></span><span class="mw-headline" id="From_today's_featured_article"
>From today's featured article</span></h2>
```

```
>>> test=soup.find_all("h2")
>>> for t in test:
        print(t.span)


<span id="From_today.27s_featured_article"></span>
<span class="mw-headline" id="Did_you_know_...">Did you know ...</span>
<span class="mw-headline" id="In_the_news">In the news</span>
<span class="mw-headline" id="On_this_day">On this day</span>
<span id="Today.27s_featured_picture"></span>
<span class="mw-headline" id="Other_areas_of_Wikipedia">Other areas of Wikipedia</span>
<span id="Wikipedia.27s_sister_projects"></span>
<span class="mw-headline" id="Wikipedia_languages">Wikipedia languages</span>
None
>>> for t in test:
        print(t.span.contents)


[]
['Did you know\xa0...']
```

要找的資訊藏在哪段原始碼?

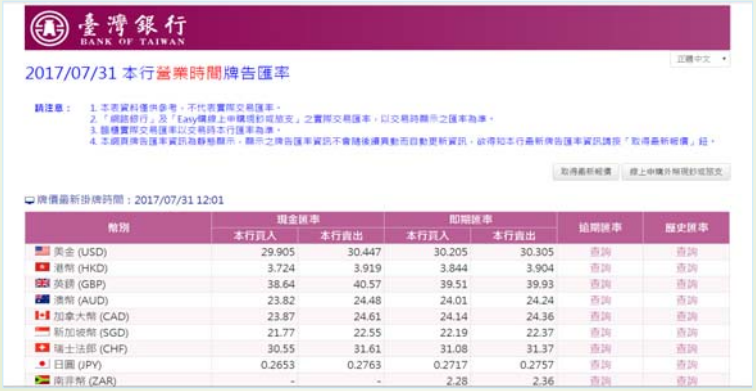
- 可善用檢查功能

作業

- 找新冠肺炎或武漢肺炎的相關網頁
- 找出當中整理好的累計數字抓出存入檔案
  - 可以是病例、措施或其他相關資料都可以

靜態網頁網路爬蟲概念

- 靜態網頁網路爬蟲實作
  - 獲取網頁： 以 http://rate.bot.com.tw/xrt?Lang=zh-TW (台銀牌告匯率)舉例


臺灣銀行
BANK OF TAIWAN

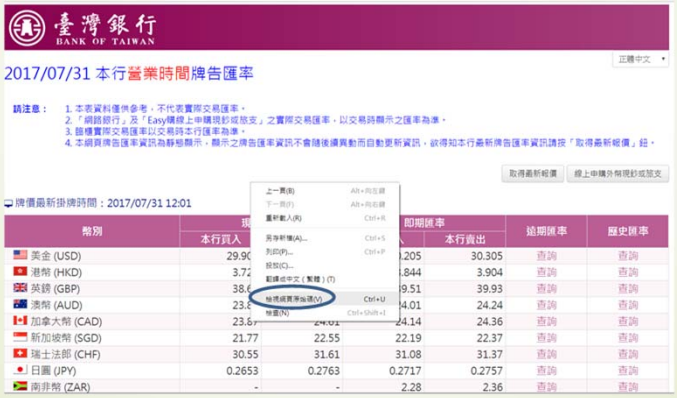
2017/07/31 本行營業時間牌告匯率

牌價最新掛牌時間：2017/07/31 12:01

幣別	現金匯率		即期匯率		遠期匯率	歷史匯率
	本行買入	本行賣出	本行買入	本行賣出		
美金 (USD)	29.905	30.447	30.205	30.305	查詢	查詢
港幣 (HKD)	3.724	3.919	3.844	3.904	查詢	查詢
英鎊 (GBP)	38.64	40.57	39.51	39.93	查詢	查詢
澳幣 (AUD)	23.82	24.48	24.01	24.24	查詢	查詢
加拿大幣 (CAD)	23.87	24.61	24.14	24.36	查詢	查詢
新加坡幣 (SGD)	21.77	22.55	22.19	22.37	查詢	查詢
瑞士法郎 (CHF)	30.55	31.61	31.08	31.37	查詢	查詢
日圓 (JPY)	0.2653	0.2763	0.2717	0.2757	查詢	查詢
南非幣 (ZAR)	-	-	2.28	2.36	查詢	查詢

靜態網頁網路爬蟲概念

- **靜態網頁網路爬蟲實作**
  - 分析網頁：從前面網頁中按滑鼠右鍵，出現快顯功能表後，點選檢視網頁原始碼按鈕，接著會出現我們想分析的內容



靜態網頁網路爬蟲概念

- **分析網頁：**從原始碼可以看出，這個網頁的表格包括由tr(表格的列標籤)分割的各個幣別，各幣別內還有td(表格的行標籤) 搭配div class和visible-phone描述的幣別資訊(如美金(USD))與由td標籤描述的各種匯率資料(如上圖中本行現金買入匯率為29.915)所組成

```
<html lang="zh-TW" class="no-js">
<head>
    <meta charset="utf-8" />
    <title>臺灣銀行牌告匯率</title>

<tr>
    <td data-table="幣別" class="currency phone-small-font">
        <div>
            <div class="sp-div sp-america-div">
                <img title="幣別國旗" alt="幣別國旗" src="/Content/images/sprite_lateral.png" class="sp-img sp-america-img" />
            </div>
            <br class="visible-phone print_hide" />
            <div class="visible-phone print_hide">
                美金 (USD)
            </div>
            <div class="hidden-phone print_show" style="text-indent:30px;">
                美金 (USD)
            </div>
        </div>
    </td>
    <td data-table="本行現金買入" class="rate-content-cash text-right print_hide">29.915</td>
    <td data-table="本行現金賣出" class="rate-content-cash text-right print_hide">30.457</td>
    <td data-table="本行即期買入" class="rate-content-sight text-right print_hide" data-hide="phone">30.215</td>
    <td data-table="本行即期賣出" class="rate-content-sight text-right print_hide" data-hide="phone">30.315</td>
```