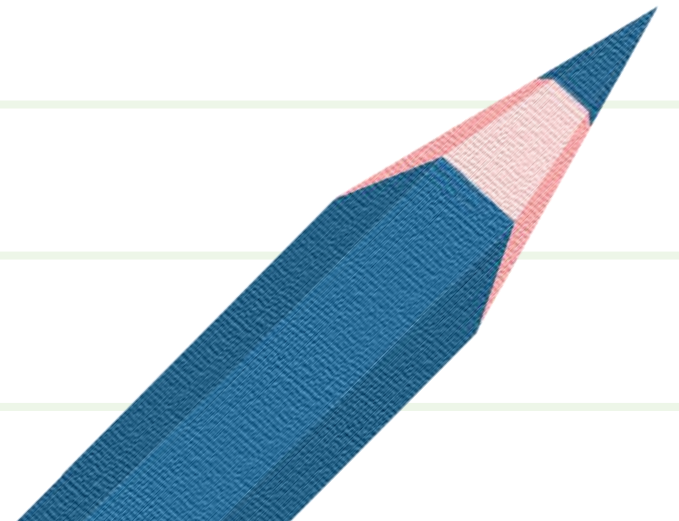


Introduction to Data Science

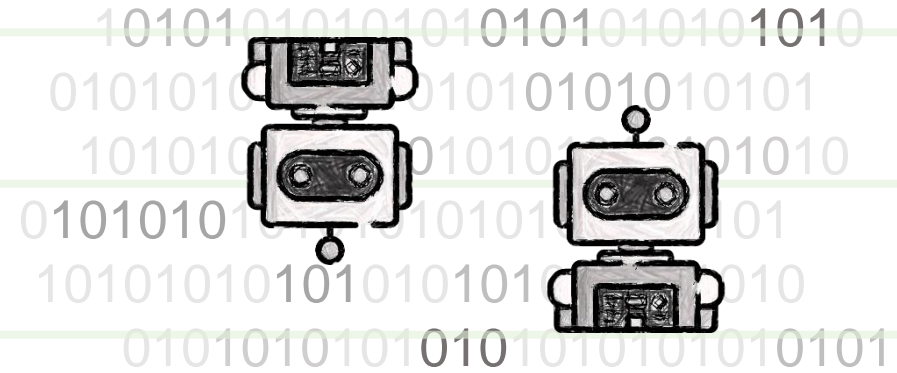
BeautifulSoup

TA: 李姍徵



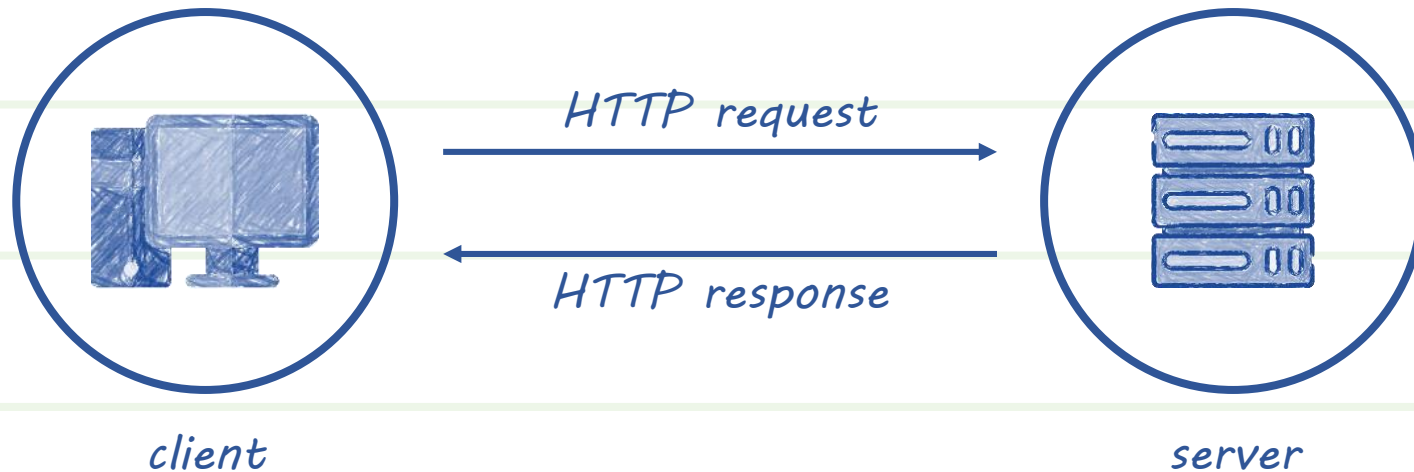
Web Crawler

- 自動瀏覽網頁的機器人
- 不必親自瀏覽網頁便能得到你想要的資訊
- 常應用於資料蒐集、自動偵測網頁變化等
- 範例：
 1. 紀錄外幣匯率，買低賣高
 2. 蒐集新聞或論壇內容，幫助預測股票價格
 3. 沒有現成資料集的情況下，取得自己想要的資料



requests

- 客戶端發送請求 (request)
- 伺服器給予回應 (response)
- requests套件幫助我們捕捉response的內容
- 示意圖：



requests

- (cmd) pip install requests 、 (py) import requests

- `r = requests.get(<url>)` # GET請求

- `r.status_code` # 伺服器回應碼

- `r.text` # 網頁原始碼

1xx	信息請求
2xx	請求成功
3xx	重定向
4xx	客戶端錯誤
5xx	服務器錯誤

```
# GET
r = requests.get('https://www.google.com.tw/')
print('type:', type(r), '\n')
print('status code:', r.status_code, '\n')
print('text:', r.text)
```

```
type: <class 'requests.models.Response'>
```

```
status code: 200
```

```
text: <!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="zh-TW"><head><meta content="text/html; cha
rset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop
="image"><title>Google</title><script nonce="yUWPajWqA4zH0arX0meEGg==">(function(){window.google={keyI:'Zp6AYYG8La6Vr7wPi6mUkA
```

requests

- parameter # 許多GET請求會透過URL夾帶參數, 例: 搜尋
- header # 表頭包含的資訊幫助伺服器知道你是誰或正在做什麼, 例: user-agent、cookie
- cookie # 幫助伺服器儲存當前的瀏覽狀態, 例: 購物車、保持登入狀態

(F12 -> Network -> F5 -> 點擊目標網頁 -> headers -> Request Headers -> cookie)

```
# Add parameters, headers, or cookies
params = {'key1': 'value1', 'key2': 'value2'}
headers = {'user-agent': 'my-app/0.0.1'}
cookies = dict(cookie = 'some cookies')

r = requests.get('http://httpbin.org/get', params = params)
print('add parameter:', r.url)
r = requests.get('http://httpbin.org/get', headers = headers)
r = requests.get('http://httpbin.org/cookies', cookies = cookies)
```

add parameter: <http://httpbin.org/get?key1=value1&key2=value2>

```
▼ Request Headers
:authority: www.google.com.tw
:method: GET
:path: /
:scheme: https
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
accept-encoding: gzip, deflate, br
accept-language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7
cache-control: no-cache
cookie: SEARCH_SAMESITE=CgQI1ZMB; OGPC=19026531-1.; OGP=-19026531.; SID=DQhj
eFD4v_rz1v0DwzqQGhakGcb8Xwh_zK3WxG5aKEwcv8XYiDiLRKaQvHECQRpYU0-jqg.; __Secu
re-1PSID=DQhjeFD4v_rz1v0DwzqQGhakGcb8Xwh_zK3WxG5aKEwcv8XYHgckKAJQqjjIw3KLEs
```

requests

- `data = {<key1>: <value1>, <key2>: <value2>}` # 要送出的資料
- `data = ((<key1>, <value1>), (<key1>, <value2>))` # 具有重複的鍵值
- `r = requests.post(<url>, data)` # POST請求

```
# 要送出的資料
data = {'key1': 'value1', 'key2': 'value2'}
r = requests.post('http://httpbin.org/post', data = data)

# 具有重複鍵值的資料
data = (('key1', 'value1'), ('key1', 'value2'))
r = requests.post('http://httpbin.org/post', data = data)
```

GET vs. POST

- > GET: 瀏覽器會將資料內容轉為特定編碼後, 加在URL進行連線
- > POST: 瀏覽器會將資料內容封包後, 再進行傳送

BeautifulSoup

- (cmd) pip install beautifulsoup4
- (py) from bs4 import BeautifulSoup
- BeautifulSoup(response.text, 'html.parser')

以 BeautifulSoup 解析 HTML 程式碼

```
url = 'http://www.stat.ncku.edu.tw/'  
response = requests.get(url)  
soup = BeautifulSoup(response.text, 'html.parser')  
print(soup)
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta content="成功大學統計學系" name="author"/>  
<meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>  
<title>成功大學統計學系</title>  
<meta content="" name="keywords"/>  
<meta content="" name="description"/>  
<meta content="all" name="robots"/>  
<meta content="GLOBAL" name="distribution"/>  
<meta content="general" name="rating"/>
```

BeautifulSoup

- `print(soup.prettify())` # 輸出排版後的 HTML 程式碼

```
print(soup.prettify())
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta content="成功大學統計學系" name="author"/>
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <title>
    成功大學統計學系
  </title>
  <meta content="" name="keywords"/>
  <meta content="" name="description"/>
  <meta content="all" name="robots"/>
  <meta content="GLOBAL" name="distribution"/>
  <meta content="general" name="rating"/>
  <meta content="7 days" name="revisit-after"/>
  <meta content="all" name="webcrawlers">
  <meta content="all" name="spiders">
  <meta content="http://www.stat.ncku.edu.tw/" property="og:url">
  <meta content="website" property="og:type">
  <meta content="成功大學統計學系" property="og:title">
  <meta content="" property="og:description">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
```


BeautifulSoup

- `a_tag = soup.find('a')` # 只取出感興趣節點標籤的第一個
- `print(a_tag)` # 印出該標籤包含的整段語法
- `print(a_tag.string)` # 印出該標籤包含的內文
- `print(a_tag.get('href'))` # 印出該節點我們感興趣的屬性的值


```
a_tag = soup.find('a')
print('a_tag:\n', a_tag)
print('')
print('a_tag.string:\n', a_tag.string)
print('')
print('a_tag.get:\n', a_tag.get('href'))
```

```
a_tag:
<a class="micon" href="#menu" id="nav-toggle"><span></span></a>

a_tag.string:
None

a_tag.get:
#menu
```

BeautifulSoup

- `a_tags = soup.find_all('a')` # 找出所有特定的 HTML 標籤節點
 - 使用迴圈依序輸出
-  # 可以放 list, 同時搜尋多個標籤

```
a_tags = soup.find_all('a')
for tag in a_tags:
    print(tag.string)
```

```
None
成功大學統計學系
None
系所公告
課程公告
```

可搭配上一頁提到的方式,
輸出自己想要的東西

BeautifulSoup

- `soup.find(id= 'iqlink')` # 利用 id 來搜尋
- `soup.find(class_= 'quicklink5')` # 利用 css 的 class 來搜尋

```
iqlink_tag = soup.find(id='iqlink')  
print(iqlink_tag.text)
```

數據科學研究所
調查統計研究中心
統計諮詢中心
工業統計與品質實驗室
資料採礦研究室
榕園統計文教基金會

```
stat_ds = soup.find(class_='quicklink5')  
print(stat_ds.text)
```

數據科學研究所

BeautifulSoup

1. `<td>some text</td>`

沒有子標籤，但有文本

2. `<td></td>`

沒有子標籤，且沒有文本

3. `<td><p>more text</p></td>`

有子標籤，且僅子標籤有文本

4. `<td>even <p>more text</p></td>`

有子標籤，且各包含一段文本

`.string`

1. `some text`
2. `None`
3. `more text`
4. `None`

`.text`

1. `some text`
2.
3. `more text`
4. `even more text`

re

- 正規表示式 (Regular Expression)
- 字串比對技巧

BeautifulSoup 中，可用來搜尋

也可針對爬取的文本進行處理

```
text = 'Introduction to Data Science'

# find all
find_all_capital = re.findall('[A-Z]', text)
print('find all:', find_all_capital)

# split
split_space = re.split(' ', text)
print('split space:', split_space)

# sub
sub_space = re.sub(' ', '-', text)
print('sub space:', sub_space)
```

```
find all: ['I', 'D', 'S']
split space: ['Introduction', 'to', 'Data', 'Science']
sub space: Introduction-to-Data-Science
```

查表與範例：[就是愛程式](#)

線上練習：[Regex101](#)

re

- *：代表前面的字元可以不出現，也可以出現一次或多次

例：0*42 可以符合 42、042、00042 等

- +：代表前面的字元必須至少出現一次。

例：go+od 可以符合 good、good、gooooooooooooooooood 等

- ?：代表前面的字元最多只可出現一次。

例：colou?r 可以符合 color 或 colour

re

- .：符合除「\r」及「\n」之外的任何單個字元
- \d：捕捉字串中數字的部分，不限位數
- \w：捕捉任何的字元，包含英文及數字
- ()：找出字串中符合括號內字串的子字串
- a[bc]：表示a後面接b或c # 例：ab、ac
- [a-z]：表示字元範圍 # a-z 為所有小寫字母，0-9 為所有數字



End

爬蟲是自動化蒐集資料的方式之一，但在使用爬蟲的時候也要多加注意！

盡量不要不間斷的重複拜訪相同頁面，可能會造成伺服器延遲，甚至癱瘓QQ