

BEAUTIFULSOUP

Python爬蟲教學

組員

統計四 王翊芄

統計四 吳炫德

What Is Web Scraping?

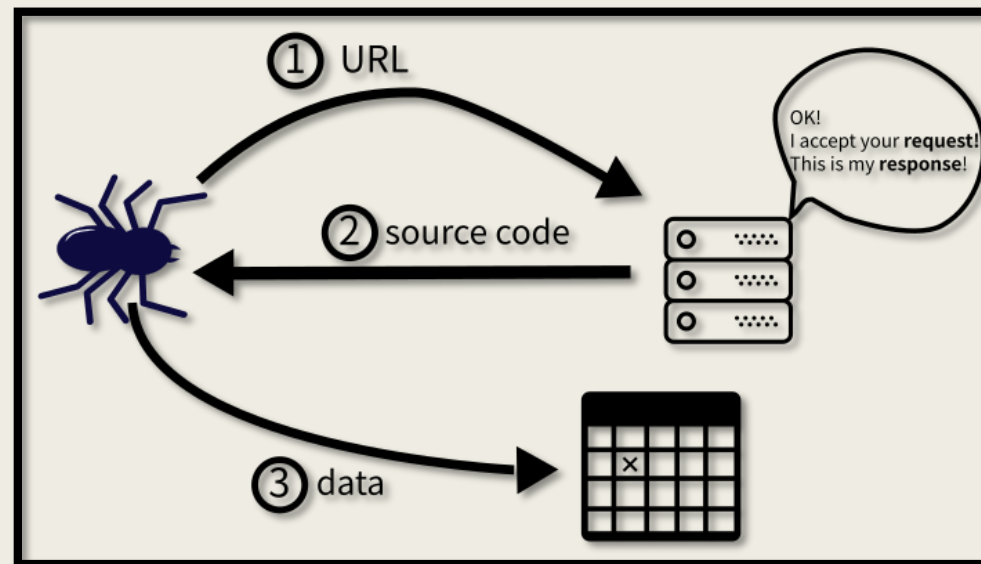
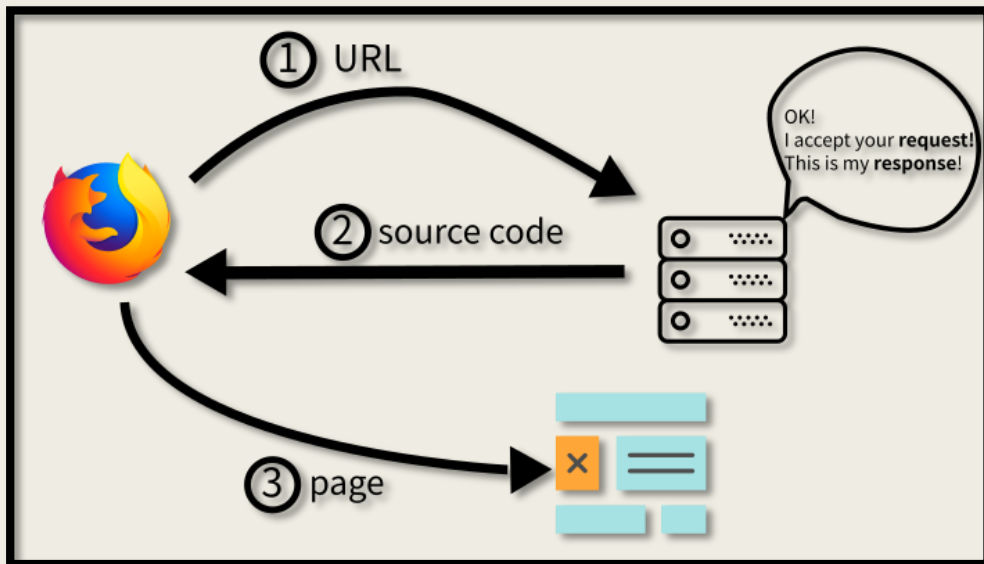
■ Definition :

- 網頁爬蟲是指利用程式去 **自動獲取網頁資訊的技術**，當你掌握了此技術之後，你就可以有源源不絕的資料進行各種運用。因此這是現今資料分析人員必定要掌握的技術。

網路爬蟲 web crawler	網頁抓取 web scraper
一般是搜尋引擎進行網路索引時(web indexing)使用的技術。	針對獲取特定網域底下的數個分頁的資料，並轉化成結構化格式的過程。
Crawler常會造成網站 伺服器的負擔 ，或 侵犯隱私資訊 。(網頁開發人員會在網站中添加robots.txt)	除了連續獲得連結連出去之外，還必須剖析各分頁的原始碼，將需要的 資料片段擷取 下來。
大部分是大規模的拜訪不特定的網站。	一般中文稱呼的網路爬蟲。

一般使用者 vs 網路爬蟲

- 一般使用者是以瀏覽器依據網址(url)向某一網站伺服器送出請求(request)，如果對方伺服器同意你的請求，會回應(response)將網頁的原始碼回傳給瀏覽器，瀏覽器再將原始碼轉換成圖文並茂的頁面
- 而網路爬蟲即是以程式碼偽裝成一般的使用者，向對方伺服器送出請求，取得回應（原始碼）。再從原始碼中抽取出需要的資訊。



爬蟲流程參考

1. **鎖定目標**：需要哪些資料？目標網站是哪幾個？
2. **觀察網站結構**：觀察目標網站的伺服器是如何存取資料。在網頁原始碼里嗎？又或者是動態網站，以GET或POST來產生你所看到的資訊？將資訊藏在xml、json等檔案之中嗎？是否有提供api可以直接拿到資料呢？以此來決定要用哪種方式獲取網頁資訊。
3. **嘗試爬取各網頁**：試圖獲得html原始碼。觀察是否會被對方伺服器判定成惡意程式擋掉呢？如何偽裝躲過判定？如何送get或post資料給目標網站呢？如何處理認證問題呢？
4. **剖析網頁原始碼**：是否能從html原始碼中，將所要的資訊的片段抓取下來？
5. **重複爬取**：當可以成功獲得一個分頁的特定資訊後，就可開始利用迴圈大規模的重複拜訪撈資料。這個過程中很容易會遇到例外狀況將你的爬蟲程式阻隔。
6. **資料輸出**：將獲取資料輸出，並整理成方便資料分析的格式。

禮貌原則

- 重複連續的拜訪某一網站，會造成對方伺服器的負擔
- 以下整理幾點禮貌原則：

1. 遵從*robots.txt*的規範



2. 提供*user agent*，如單位名稱或email等
3. 在連續的請求中加入延遲，減緩對方伺服器的負荷
4. 如果對方有提供資料下載區或網頁api提供資料，請不要爬

爬蟲相關套件(補充)

■ 爬取相關套件

- *urllib*
- *requests*
- *selenium*

■ 剖析html相關套件

- *re*
- *beautifulsoup*

■ 加速爬蟲相關套件

- *multiprocessing*
- *aiohttp*

■ 資料整理相關套件

- *pandas*
- *numpy*

■ 完整合一的套件

- *scrapy*

Package Installation

- requests :
 - 建立各種 *HTTP* 請求
 - 從網頁伺服器上取得想要的資料
- BeautifulSoup4 :
 - 輸入 *requests* 得來的資料
 - 指定 *HTML* 的型態來進行解析
- pandas
- numpy

套件安裝

```
pip install beautifulsoup4
```

```
pip install requests
```

```
# Web scraping
from bs4 import BeautifulSoup
import requests

# Data manipulation
import pandas as pd
import numpy as np
```

requests

```
requests.get("http://xxxx.com/")
requests.post("http://xxxx.com/post", data = {'key': 'value'})
requests.put("http://xxxx.com/put", data = {'key': 'value'})
requests.delete("http://xxxx.com/delete")
requests.head("http://xxxx.com/get")
requests.options("http://xxxx.com/get")
```

■ `page = requests.get(url)` : 用於建立適當的 HTTP 請求

- `status_code` :

確認對方伺服器回應，**200**表示正常html狀態碼

2XX成功、3XX重新導向

4XX用戶端錯誤、5XX伺服器端錯誤

- `ok` :

確認對方伺服器回應，正常則回傳**True**

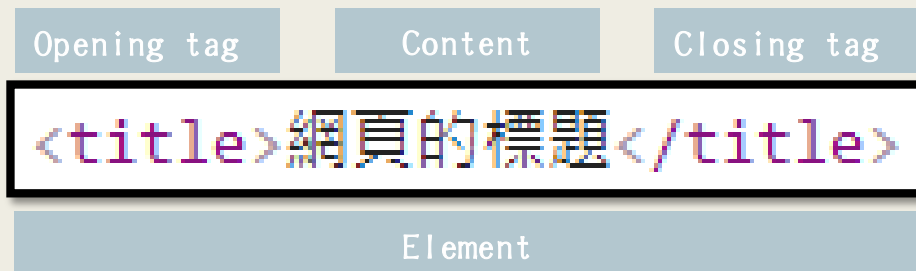
```
url = 'https://aliceinwonderland.fandom.com/wiki/Turtle_Soup'
page = requests.get(url, params=None)
print(page.status_code)
print(page.ok)
```

200

True

html 語法

- Html 由元素 (element) 構成
- 元素包含標籤 (tags) 及內容 (content)



- 標籤可以加上不同的屬性 (Attributes) 來改變瀏覽器對標籤的呈現方式

```
<html>
  <head>
    <title>有背景圖的網頁</title>
  </head>
  <body background="multimedia/CAT-WALKING-KITY.gif" class="vsc-initialized">
    加入了背景圖！
  </body>
</html> == $0
```

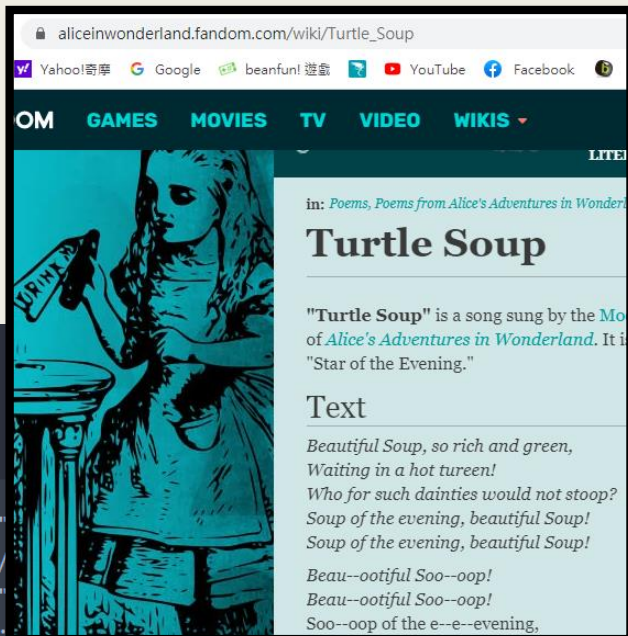
- 最基本的網頁架構如下圖

```
<html>
  <head>
    <title>網頁的標題</title>
  </head>
  <body class="vsc-initialized">
    網頁的內容
  </body>
</html> == $0
```

操作圖例

```
url = 'https://aliceinwonderland.fandom.com/wiki/Turtle_Soup'  
page = requests.get(url,params=None)  
page.text
```

```
'<!DOCTYPE html>\n<html class="client-nojs" lang="en" dir="ltr">\n<head>\n<meta charset="UTF-8" />\n<script>document.documentElement.className = document.documentElement.className.replace(/no-js/g, "js");\n</script>\n<script>(window.RLQ=window.RLQ||[]).push(function(){mw.config.set({"wgCanonicalNamespace":"","wgArticleId":3971,"wgPageName":"Turtle_Soup","wgTitle":"Turtle Soup","wgCurRevisionId":12471,"wgRevisionId":12471,"wgParentPageName":"","wgParentPageId":"","wgAction":"view","wgUserName":null,"wgUserGroups":["*"],"wgCategories":["Poems","Poems from Alice's Adventures in Wonderland"],"wgBreakFrames":false,"wgPageContentLanguage":"en","wgPageContentModel":"wikitext","wgSeparatorTransformTable":["",""],"wgDigitTransformTable":["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":["","January","February","March","April","May","June","July","August","September","October","November","December"],"wgMonthNamesShort":["","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"],"wgRelevantPageName":"Turtle_Soup","wgRelevantArticleId":3971,"wgRequestId":"73ccde2148a1dc56","wgCSPNonce":false,"wgIsProbablyEditable":false,"wgRelevantPageIsProbablyEditable":false,"wgRestrictionEdit":[],"wgRestrictionMove":[],"wgNoExternals":false,"wgArticleInterlangList":[],"wgVisualEditor":{"pageLanguageCode":"en","pageLanguageDir":"ltr","pageVariantFallbacks":"en","usePageImages":false,"usePageDescriptions":false},"wgIsTestModeEnabled":false,"wgEnableLightboxExt":true,"egMapsScriptPath":"/extensions/Maps/","egMapsDebugJS":false,"egMapsAvailableServices":["leaflet","googlemaps3"],"egMapsLeafletLayersApiKeys":{"MapBox":"","MapQuestOpen":"","Thunderforest":"","GeoportailFrance":""},"wgVisualEditorToolbarScrolloffset":0,"wgVisualEditorUnsupportedEditParams":["undo","undoafter","veswitched"],"wgEditSubmitButtonLabelPublish":false,"wgCodeMirrorEnabled":true,"egFacebookAppId":"112328095453510","comscoreKeyword":"wikiacsid_books","quantcastLabels":"","wgBlankImgUrl":"data:image/gif;base64,R0lGODlhAQABAIAAAAAAAP///yH5BAEAAAEALAAAAABAAEQAICTAEAOw%3D%3D","disclaimerMessage":"Fandom may earn an affiliate commission on sales made from links on this page.","wgCategorySelect":{"defaultNamespace":"Category","defaultNamespaces":["Category"],"wgEnableDiscussions":false,"wgEnableWikiaBarExt":true,"wgEnableWikiaBarAds":true,"wgWikiaBarMainLanguages":["de","en","es","fr"],"wgPerformanceMonitoringSamplingFactor":100,"wgPerformanceMonitoringEndpointUrl":"https://beacon.wikia-services.com/__track/special/performance_metrics?w=38772&u0026lc=en&u0026d=aliceinwonderland&u0026s=oasis&u0026u=0&u0026a=0&u0026i=sjc-prod","wgSoftwareVersion":"release-138.012@release-138.012\\n","wgRailModuleList":["Fandom\\\\\\\\Oasis\\\\\\\\Rail\\\\\\\\Modules\\\\\\\\RecentActivityModuleService","Fandom\\\\\\\\Oasis\\\\\\\\Rail\\\\\\\\Modules\\\\\\\\PopularPagesMo
```



BeautifulSoup

- 利用html.parser來剖析剛剛取得的html原始碼
 - 還有其他剖析器可以選用，如：*lxml*、*html5lib*

```
url = 'https://aliceinwonderland.fandom.com/wiki/Turtle_Soup'  
page = requests.get(url,params=None)  
page.text  
soup = BeautifulSoup(page.text , 'html.parser')
```

- 再來我們可以利用soup來快速檢視網頁內容

```
soup.title  
  
<title>Turtle Soup - Alice in Wonderland Wiki</title>
```

soup.pretty()

#輸出排版後的HTML 程式碼

```
print(soup.pretty())
```

```
<!DOCTYPE html>
<html class="client-nojs" dir="ltr" lang="en">
  <head>
    <meta charset="utf-8"/>
    <title>
      Turtle Soup - Alice in Wonderland Wiki
    </title>
    <script>
      document.documentElement.className = document.do
    </script>
    <script>
```

soup.find() 用法

- 只搜尋第一個符合條件的HTML節點
- 在某些確定只需要取第一個節點的情況可以使用

```
soup.find("b")
```

```
<b>Author</b>
```

- 相比於find_all()

```
soup.find_all("b")
```

```
[<b>Author</b>,  
  <b>Book</b>,  
  <b>Written in</b>,  
  <b>Video</b>,  
  <b>"Turtle Soup"</b>]
```

soup.find_all() 用法

1. 搜尋標籤 `soup.find_all('tag')`
2. 根據id搜尋 `soup.find_all(id='tag id')`
3. 搜尋文本 `soup.find_all(text='text')`
4. 利用正規表達法
 - `soup.find_all(text=re.compile('your re'))`,
`soup.find_all(id=re.compile('your re'))`
5. 指定屬性搜尋標籤 `soup.find_all('tag', {'id': 'kkk', 'class': 'kkk'})`

1. 搜尋標籤

```
for i in soup.find_all("i"):
    print(i)
```

```
<i> </i>
<i><a href="/wiki/Alice%27s_Adventures_in_Wonderland" title="Alice's Adventur
<i>Beautiful Soup, so rich and green,</i>
<i>Waiting in a hot tureen!</i>
<i>Who for such dainties would not stoop?</i>
<i>Soup of the evening, beautiful Soup!</i>
<i>Soup of the evening, beautiful Soup!</i>
<i>Beau--ootiful Soo--oop!</i>
<i>Beau--ootiful Soo--oop!<br/></i>
```

2. 根據 id 搜尋

```
for i in soup.find_all(id="WikiaBar"):
    print(i)
```

```
<div id="WikiaBar">
<div class="WikiaBarWrapper hidden" id="WikiaBarWrapper">
<div class="wikia-bar wikia-bar-anon">
<a class="wikiabar-button" data-index="0" href="//www.wikia.com/Video_Games">
<span>Gaming: New Releases</span>
</a>
<a class="wikiabar-button" data-index="1" href="//www.wikia.com/Entertainment
<span>New in Entertainment</span>
</a>
```


3. 搜尋文本

- 利用 `find_all(text= "...")` 來搜尋文本，
需內容完全相符
- 常搭配正規表達法來使用

```
for i in soup.find_all(text="Beau--ootiful Soo--oop!"):
    print(i)
```

```
Beau--ootiful Soo--oop!
Beau--ootiful Soo--oop!
Beau--ootiful Soo--oop!
Beau--ootiful Soo--oop!
```

```
import re
```

```
for i in soup.find_all(text=re.compile("Beautiful")):
    print(i)
```

```
Beautiful Soup, so rich and green,
Beautiful, beautiful Soup
Beautiful Soup! Who cares for fish,
Pennyworth only of Beautiful Soup?
Beautiful, beauti--FUL SOUP!
```

5. 指定屬性搜尋標籤

- 欲將原始碼裡面 ... 其中的0000XXX全部印出

```
for link in soup.find_all('a'):
    print(link.get('href'))
```

```
https://www.fandom.com/
https://www.fandom.com/topics/games
```

- 欲搜尋div標籤下class含有WikiaSiteWrapper的段落

```
soup.find_all("div", class_="WikiaSiteWrapper")
```

```
[<div class="WikiaSiteWrapper">
  <div class="banner-notifications-placeholder">
    <div class="wds-banner-notification__container">
      </div>
```

class為python之保留用字，
以class_替代

Get 請求 (requests)

```
google_url = 'https://www.google.com.tw/'  
google_html = requests.get(google_url)
```

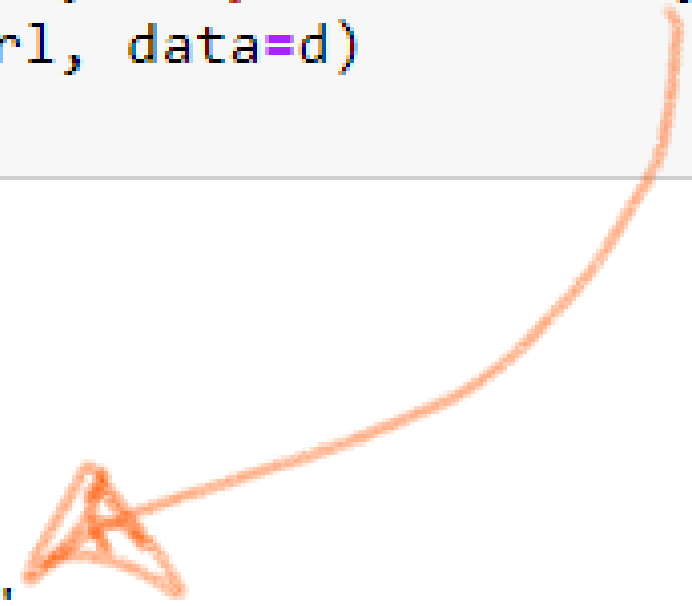
- google_html 即是得到的 response

Post 請求

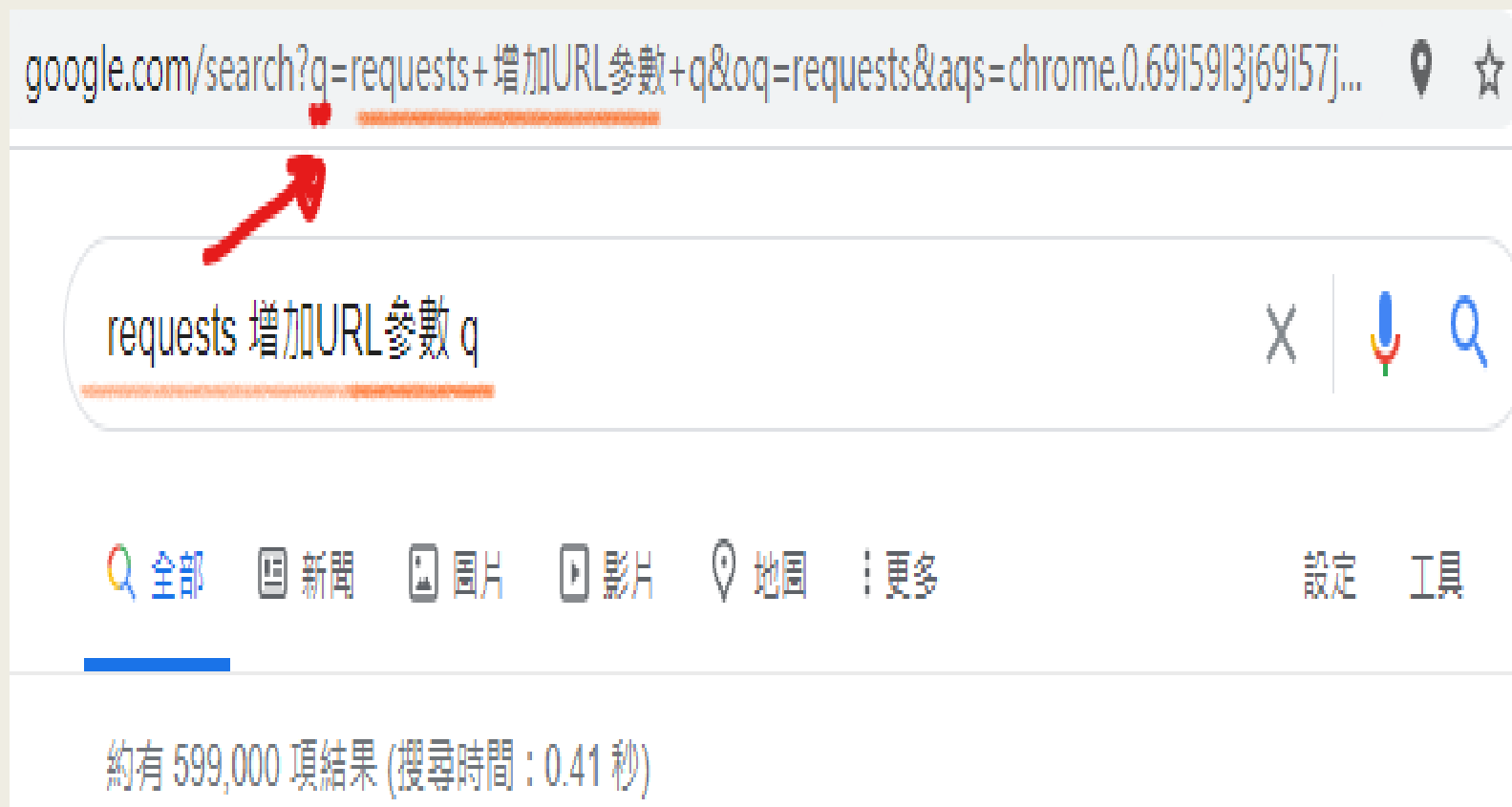
- Requests 提交 form 表單，提交使用者名稱與密碼

```
: url = 'http://httpbin.org/post'
d = {'key1': 'value1', 'key2': 'value2'}
r = requests.post(url, data=d)
print(r.text)
```

```
{
  "args": {},
  "data": "",
  "files": {},
  "form": {
    "key1": "value1",
    "key2": "value2"
```



加入參數



```
In [89]: google_url = 'https://www.google.com.tw/'  
        ## 要加入的參數  
        my_params = {'key1' : 'value1', 'key2': "value2"}  
  
        ## 將參數加入 get()請求  
        requests.get(google_url, params = my_params).url  
  
Out[89]: 'https://www.google.com.tw/?key1=value1&key2=value2'
```

- `.url` 可以回傳的response的網址

→ ↻ ⓘ 檔案 | C:/Users/hello/Documents/school/巨資/巨資py/ptt_searchContent.html ☆ 炫德

Google

ptt × 🔍

全部 新聞 圖片 影片 地圖 購物 書籍 搜尋工具

批踢踢實業坊
term.ptt.cc

使用網頁版終端機連線至批踢踢實業坊-- 以學術性質為目的，提供各專業學生實習的平台，而以電子佈告欄系統為主的一系列服務。期許在網際網路上建立起一個 ...

分類看板- 批踢踢實業坊
www.ptt.cc > cls

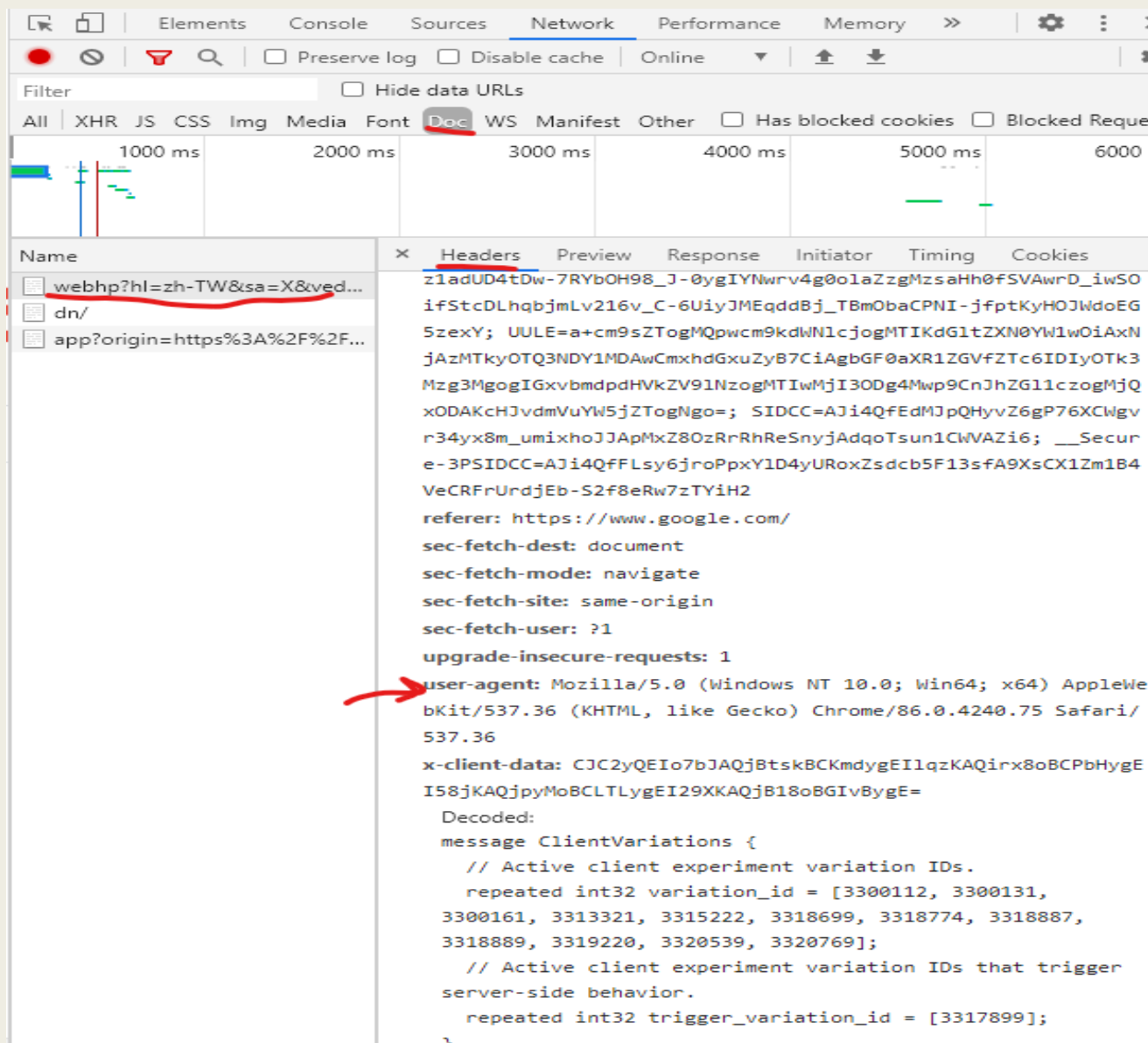
Σ市民廣場報告站長PTT咬我 · B_Group. 一一. Σ臺灣大學臺大, 臺大, 臺大 · C_Group. 一一. Σ政治大學政大, 政大, 政大 · D_Group. 一一. Σ青蘋果樹校園, 班板, 社團.
熱門看板 · J_Group 一一Σ生活娛樂館生活... · L_Group 一一Σ國家體育場...

熱門看板- 批踢踢實業坊
www.ptt.cc > bbs

◎PTT重機板~交通起義 · 公總償命 · BB-Love. 62. 屬性. ◎[絕愛] 行行出狀元活動開跑囉 · DC_SALE. 60. 攝影. ◎數位相機交易- 注意詐騙 · TY_Research. 57. 天氣.
分類看板 · 熱門看板 · 看板Gossiping · 看板C_Chart

加入表頭 (headers)

- 瀏覽器在送出request的時候會送出header告訴server一些資訊
- 通常不用登入、用get傳參數，還會發生問題的，就送假user-agent去試試看(假裝)



反爬蟲

- 最簡單的方式就是改表頭 (header) 的 user-agent, 偽裝成瀏覽器可能就可以。
- `my_headers = {'user-agent': 'anything'}`
- `r = requests.get(url, headers = my_headers)`

正規表達法 (regex)

找出所有 'h' 開頭的標題文字

```
titles = soup.find_all(['h1', 'h2', 'h3', 'h4', 'h5', 'h6'])  
for title in titles:  
    print(title.text.strip())
```

利用 regex 找出所有 'h' 開頭的標題文字

```
for title in soup.find_all(re.compile('h[1-6]')):  
    print(title.text.strip())
```

常用的正規表達法

- *：代表前面的字元可以不出現，也可以出現一次或多次。
 - 例如：*0*42*可以符合*42*、*042*、*0042*、*00042*等。
- +：代表前面的字元必須至少出現一次。（1次或多次）。
 - 例如：*goo+gle*可以符合*google*、*gooogle*、*goonoogle*等。
- ?：代表前面的字元最多只可以出現一次。（0次或1次）。
 - 例如：*colou?r*可以符合*color*或者*colour*

規則及範例大全

<https://atedev.wordpress.com/2007/11/23/%E6%AD%A3%E8%A6%8F%E8%A1%A8%E7%A4%BA%E5%BC%8F-regular-expression>

實際操作

抓取ETTODAY 台南美食相關文章

旅遊首頁 > 文章搜尋

請輸入關鍵字：

搜尋

搜尋結果：符合關鍵字「台南美食」的查詢結果共 181 筆。

文章搜尋

列表模式： 



雞排吐司咬下牽絲！高雄人氣早午餐插旗台南 香蕉可可厚片太邪惡

新店報報！【愛開玩笑的人台南店】來自高雄的人氣早午餐店，來台南插旗囉！最近台南新店不少耶！這家【愛開玩笑的人台南店】就是今天熱騰騰新開幕的店家，胖比一早就來朝聖！店名很有趣也很好記，之前看到鳳山店的我就想去，但一直沒去成，結果【愛開玩笑的人】都開來台南了！（2020-08-21 12:00）



姓「蔡」吃鍋打折！台南平價人氣火鍋 必點番茄鍋酸甜清爽

火鍋控看過來！發現一家不錯的火鍋店「小蔡什鍋」就在熱鬧的國華街上，主打每日現熬湯頭，都是主廚手炒，湯頭搭配選擇多樣化，就連店內使用的沙茶醬也是自製，餐點的擺盤上也很用心，喜歡吃火鍋的朋友，一起還看看吧！（2020-07-31 12:00）



大腸麵線鋪龍蝦！特搜台南巷弄美食 綿密碗粿+浮水魚羹超滿足

來到美食之都，看著大街小巷琳瑯滿目的餐廳、小吃店，你會想過要怎麼選擇？在地人就是最直接的台南美食地圖啊！想吃最local的食物，跟著我們走就對，美食地圖上每一間都是當地居民的口袋名單，讓我們一起探訪那些隱藏的巷弄美食，不吃遍台南美食地圖不罷休！（2020-07-26 16:54）










連空氣都chill！精選台南十間特色店家 必吃超綿密手拉胚淇淋

台南人嗜糖程度是眾所周知，但「在台南，不好吃的店家是存活不久的。」這句話來自於我的台南友人，第一次聽到時很震撼，不過在我陸續拜訪多次台南、累積了不少美食清單、隨意挑選一間街邊小店都不踩雷的經驗，才慢慢驗證出這句好有道理的話！這次就跟著我們的腳步，

```
In [128]: articles = tainan_soup.find("div", {"class": "part_pictxt_1"})
titles = articles.find_all("h3")
for title in titles:
    print(title.find('a').text)
    print(title.find('a').get('href'))
```

從肉燥飯到擔仔麵！台南肉燥攻略地圖收錄700家店 用手機就能查
<https://travel.ettoday.net/article/1834957.htm>
在地人超愛！台南廟前「臭豆腐」超酥脆 一周賣4天還要拿號碼牌
<https://travel.ettoday.net/article/1833080.htm>
豬油拌飯 + 半熟蛋太欠吃！台南巷弄老宅居酒屋 烤豬排嫩又多汁
<https://travel.ettoday.net/article/1825730.htm>
濃縮義式 + 冰淇淋超療癒！台南新開質感咖啡店 還吃得到脆脆司康
<https://travel.ettoday.net/article/1800574.htm>
大骨湯免費喝到飽！台南人氣無名麵店 陽春麵 + 滿滿麻醬一吃上癮
<https://travel.ettoday.net/article/1795416.htm>
9隻天使紅蝦 + 鮭魚卵鋪滿飯上！台南巷弄日料 鮮蚵沙拉超邪惡
<https://travel.ettoday.net/article/1790364.htm>
料多、湯鮮甜！台南邪惡干貝海鮮粥 蒜蓉蟹肉 + 洋蔥爆炒肥美
<https://travel.ettoday.net/article/1789471.htm>
漫遊七夕鵲橋！三重蘆洲吃玩攻略 滷肉飯鋪鮮蚵 + 蒜泥白肉超過癮
<https://travel.ettoday.net/article/1792604.htm>
快訊 / 東京米其林二星姐妹店logy摘二星 菜色台灣味搶戲
<https://travel.ettoday.net/article/1792064.htm>

從台灣銀行網站 抓取匯率

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)	28.295	28.965	28.62	28.77	查詢	查詢
 港幣 (HKD)	3.547	3.751	3.668	3.738	查詢	查詢
 英鎊 (GBP)	35.89	38.01	36.785	37.415	查詢	查詢
 澳幣 (AUD)	19.76	20.54	19.975	20.32	查詢	查詢
 加拿大幣 (CAD)	21.26	22.17	21.59	21.92	查詢	查詢
 新加坡幣 (SGD)	20.56	21.47	21.03	21.25	查詢	查詢
 瑞士法郎 (CHF)	30.79	31.99	31.4	31.79	查詢	查詢
 日圓 (JPY)	0.2623	0.2751	0.2693	0.2739	查詢	查詢
 南非幣 (ZAR)	-	-	1.689	1.779	查詢	查詢
 瑞典幣 (SEK)	2.88	3.4	3.21	3.33	查詢	查詢
 紐元 (NZD)	18.33	19.18	18.66	18.96	查詢	查詢
 泰幣 (THB)	0.789	0.979	0.8984	0.9444	查詢	查詢
 菲國比索 (PHP)	0.5177	0.6507	-	-	查詢	查詢
 印尼幣 (IDR)	0.00158	0.00228	-	-	查詢	查詢
 歐元 (EUR)	33.07	34.41	33.585	34.185	查詢	查詢


```
In [129]: import pandas as pd
dict = {"currency_name": currency,
        "cash_buy": cash_buy,
        "cash_sell": cash_sell
        }
df = pd.DataFrame(dict)
print(df)
```

	currency_name	cash_buy	cash_sell
0	美金 (USD)	28.3	28.97
1	港幣 (HKD)	3.547	3.751
2	英鎊 (GBP)	35.89	38.01
3	澳幣 (AUD)	19.8	20.58
4	加拿大幣 (CAD)	21.25	22.16
5	新加坡幣 (SGD)	20.56	21.47
6	瑞士法郎 (CHF)	30.7	31.9
7	日圓 (JPY)	0.2627	0.2755
8	南非幣 (ZAR)	-	-
9	瑞典幣 (SEK)	2.86	3.38
10	紐元 (NZD)	18.37	19.22
11	泰幣 (THB)	0.7905	0.9805
12	菲國比索 (PHP)	0.5173	0.6503
13	印尼幣 (IDR)	0.00163	0.00233
14	歐元 (EUR)	32.94	34.28
15	韓元 (KRW)	0.02344	0.02734
16	越南盾 (VND)	0.00087	0.00137
17	馬來幣 (MYR)	5.81	7.435
18	人民幣 (CNY)	4.198	4.36

爬取證交所網站 上的資料

109年10月 2330 台積電 各日成交資訊

單位：元、股

日期	成交股數	成交金額	開盤價	最高價	最低價	收盤價	漲跌價差	成交筆數
109/10/05	33,238,832	14,450,407,588	438.00	438.00	431.50	432.50	-0.50	18,148
109/10/06	36,276,395	15,935,558,262	441.50	442.00	437.50	439.50	+7.00	20,189
109/10/07	44,691,239	19,693,987,093	435.00	445.00	435.00	443.00	+3.50	24,438
109/10/08	60,853,811	27,461,867,933	450.00	454.00	448.50	453.00	+10.00	37,378
109/10/12	67,645,223	31,061,721,030	458.00	462.00	457.50	460.00	+7.00	36,118
109/10/13	46,303,738	21,346,346,714	465.00	465.00	456.50	462.00	+2.00	25,938
109/10/14	52,677,241	24,157,138,176	460.50	461.00	457.00	459.00	-3.00	22,599
109/10/15	49,987,278	22,668,836,399	456.00	456.50	450.00	453.00	-6.00	23,794
109/10/16	48,274,801	21,775,749,575	451.00	456.00	448.50	449.00	-4.00	26,602
109/10/19	34,846,906	15,854,021,670	453.00	457.50	451.00	457.50	+8.50	17,112
109/10/20	26,567,638	12,065,806,730	455.50	457.00	451.00	451.00	-6.50	12,725

1	1	0	9	年	1	0	月		2
2	日期	成交股數	成交金額	開盤價	最高價	最低價	收盤價	漲跌價差	成交筆數
3	109/10/05	#####	#####	438	438	431.5	432.5	-0.5	18,148
4	109/10/06	#####	#####	441.5	442	437.5	439.5	7	20,189
5	109/10/07	#####	#####	435	445	435	443	3.5	24,438
6	109/10/08	#####	#####	450	454	448.5	453	10	37,378
7	109/10/12	#####	#####	458	462	457.5	460	7	36,118
8	109/10/13	#####	#####	465	465	456.5	462	2	25,938
9	109/10/14	#####	#####	460.5	461	457	459	-3	22,599
10	109/10/15	#####	#####	456	456.5	450	453	-6	23,794
11	109/10/16	#####	#####	451	456	448.5	449	-4	26,602
12	109/10/19	#####	#####	453	457.5	451	457.5	8.5	17,112
13	109/10/20	#####	#####	455.5	457	451	451	-6.5	12,717