

LEARN CODE WITH MIKE

搜尋



[Python爬蟲教學]快速搞懂AJAX 動態載入網頁的爬取秘訣

10月 04, 2020

分享

標籤

Python爬蟲教學



Photo by LinkedIn Sales Navigator on Unsplash

在使用Python網頁爬蟲爬取網頁時，有些網頁為了提升使用體驗及維持執行效能，會使用AJAX的技術，非同步向伺服器傳送參數，取得資料來進行顯示，使用者不會感受到畫面有任何的閃爍或停頓，依然能夠正常的使用網頁。

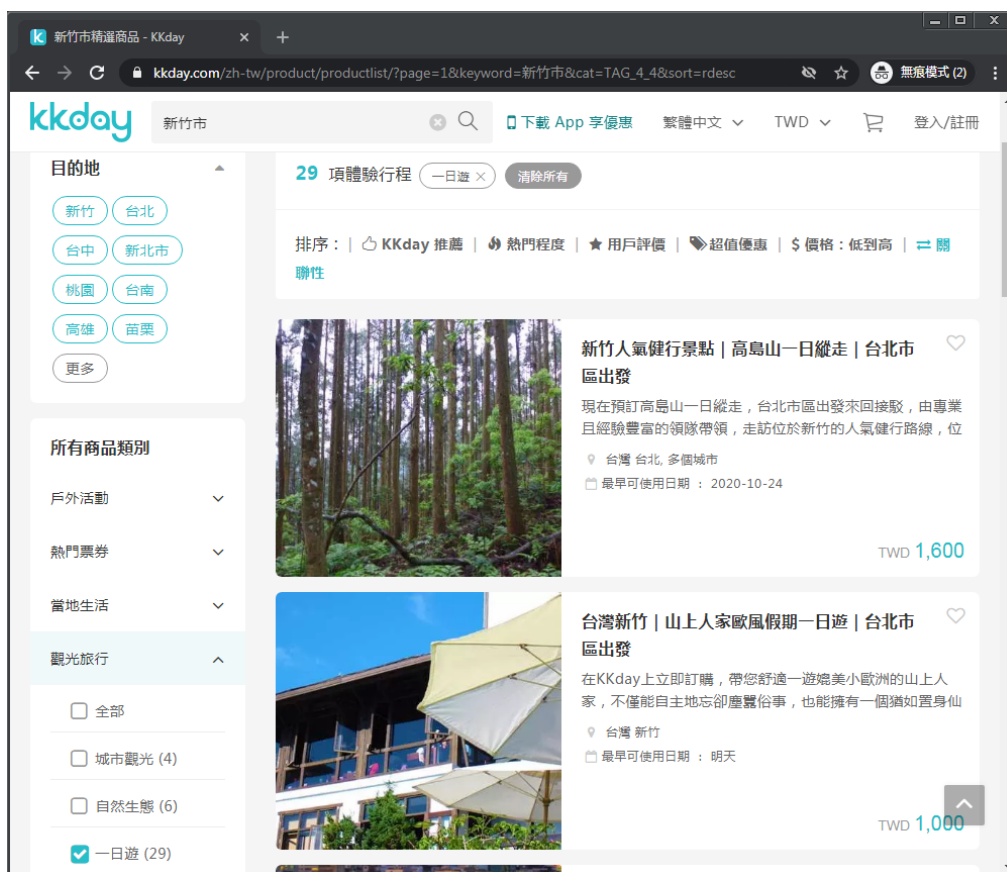
不過想要爬取這種類型的網頁，點擊滑鼠右鍵檢視原始碼時，會看到滿滿的Unicode編碼資料，沒有HTML標籤，這時候要來爬取網頁時，會完全抓不到資料。

由於筆者在爬取**KKday網站**的一日遊票券時，就是遇到這樣的情況，所以想藉此來和大家分享Python網頁爬蟲該如何爬取AJAX類型的網頁，其中的開發流程如下：

- 分析網頁
- 分析AJAX回傳結果
- 開發Python網頁爬蟲

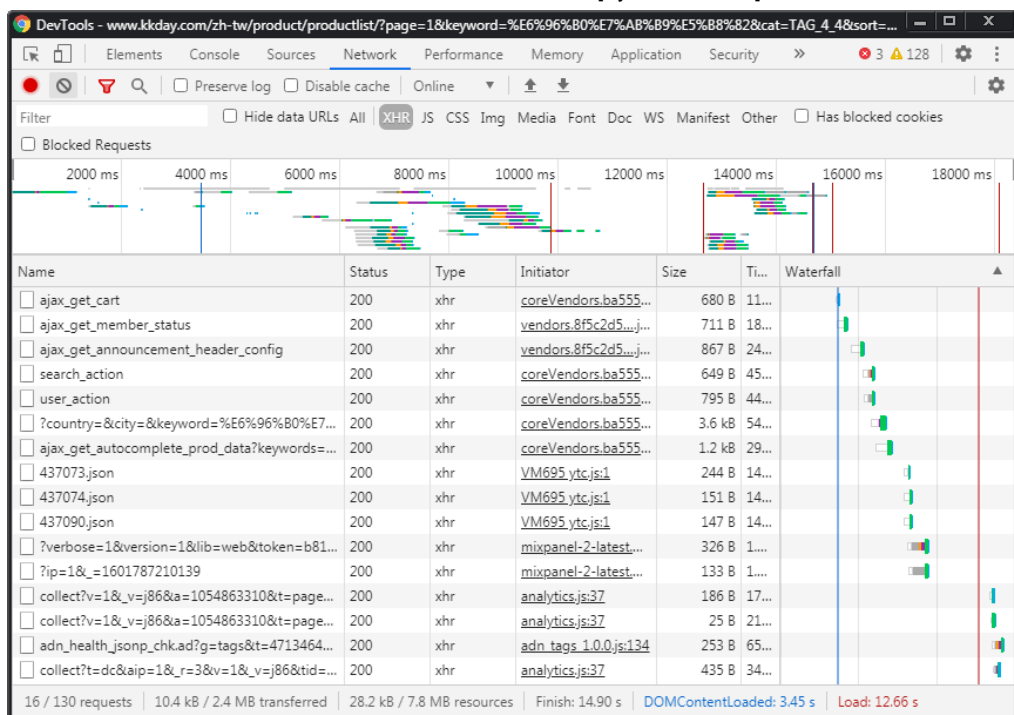
一、分析網頁

首先，前往**KKday**網站，假設在搜尋的地方輸入「新竹市」進行搜尋後，在網頁左側的「所有商品類別」中，選擇「觀光旅行」下的「一日遊」，就可以查到新竹市一日遊的相關票券，如下圖：

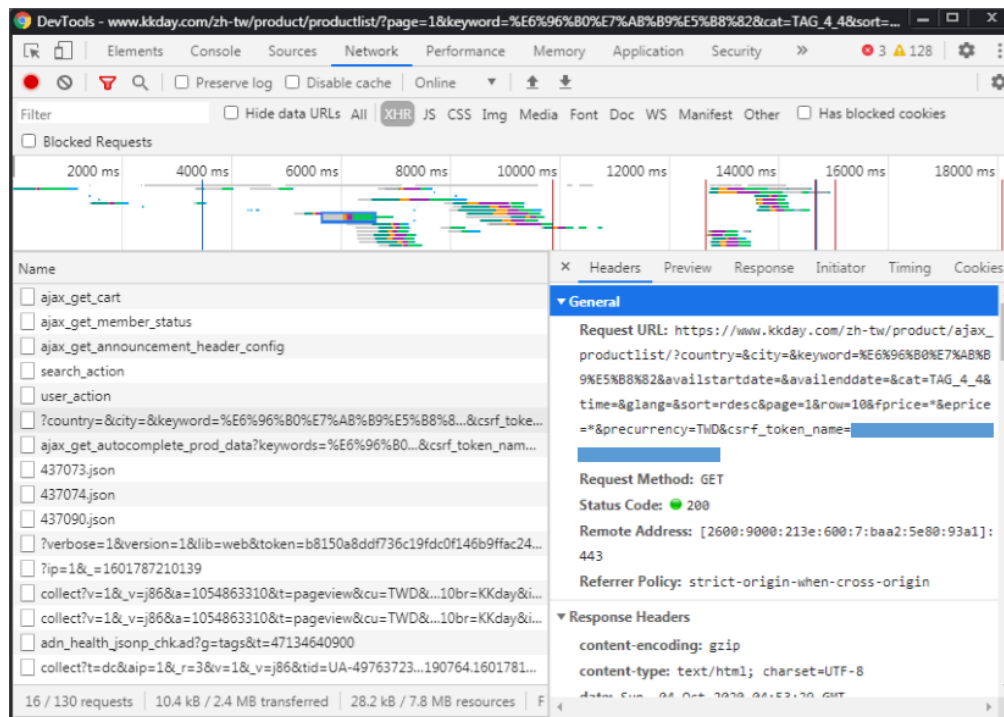


這時候，點擊滑鼠右鍵檢視網頁原始碼時，會完全找不到網頁上所顯示的票券名稱，都是Unicode編碼，這就是典型的AJAX類型網頁，也因為瀏覽器有辦法解析，所以網頁能夠正常的顯示中文字。

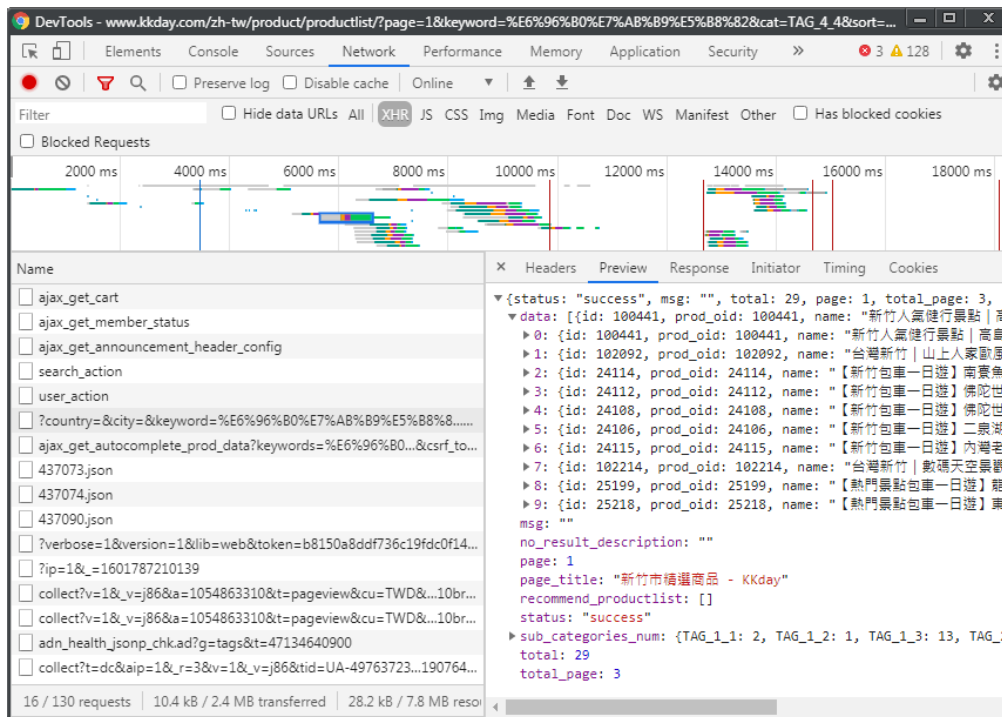
為了觀察網頁載入時，網頁向伺服器發送的請求(Request)，點擊F12鍵開啟開發者模式，並且在原網頁點擊F5鍵重新整理，就可以在開發者模式的Network頁籤下，看到每一個請求(Request)，如下圖：



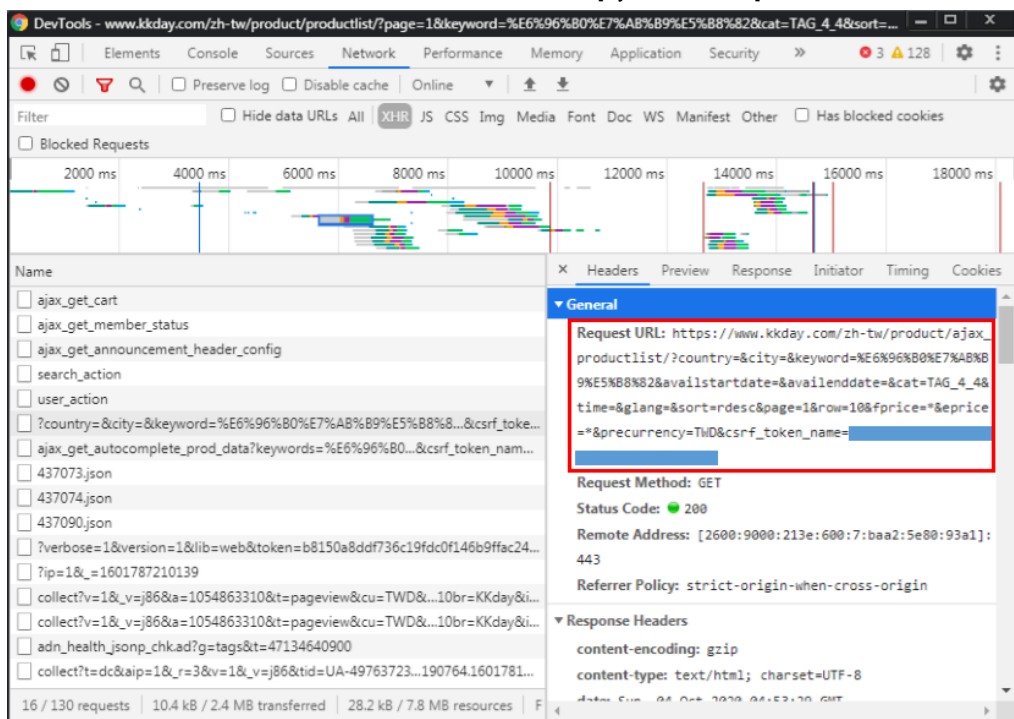
接著，就要來找哪一個請求(Request)的回應
(Response)是網頁上的資料，以上圖第6個請求
(Request)為例，點擊後，可以在「Headers」看
到它的請求(Request)內容，如下圖：



而要看這個請求(Request)的回應內容，可以切換到「Preview」頁籤來檢視，如下圖：



從上圖即可看到網頁上所顯示的票券資料，也就代表如果要使用Python網頁爬蟲來發送請求(Request)時，就是要使用「Headers」頁籤下的「Request URL」網址，如下圖：



從請求網址(Request URL)可以看到其中包含了幾個查詢的條件，目前只有輸入關鍵字(keyword)及選擇商品類別(cat)，其它查詢條件都沒有輸入或保留預設值，所以，可以將請求網址(Request URL)簡化為：

「[https://www.kkday.com/zh-tw/product/ajax_productlist/?keyword=新竹市
&cat=TAG_4_4&sort=rdesc](https://www.kkday.com/zh-tw/product/ajax_productlist/?keyword=新竹市&cat=TAG_4_4&sort=rdesc)」

如果要查詢其它地區的一日遊票券，只需要替換關鍵字(keyword)參數即可。

二、分析AJAX回傳結果

在使用Python網頁爬蟲爬取一般網頁時，需要分析HTML標籤的結構，而AJAX類型的網頁，由於網頁原始碼中並沒有HTML標籤，所以，就需要分析AJAX的回傳結果。

在剛剛開發者模式下的「Preview」頁籤下，將回傳結果的「data」欄位點開來後，選擇其中一

筆資料，可以看到它所擁有的欄位，如下範例：

```
{status: "success", msg: "", total: 29, page: 1, total_page: 3,
  data: [{id: 100441, prod_oid: 100441, name: "新竹人氣傢俱",
    0: {id: 100441, prod_oid: 100441, name: "新竹人氣傢俱",
      app_cheaper_than_web: false,
      cat_key: ["TAG_3_1", "TAG_4_3", "TAG_4_4"],
      cities: [{id: "A01-001-00001", name: "台北"}],
      city_key: ["A01-001-00001", "A01-001-00009"],
      confirm_hour: 48,
      countries: [{id: "A01-001", name: "台灣", ...}],
      country_key: ["A01-001"],
      currency: "TWD",
      days: 0,
      display_price: "1,600",
      display_sale_price: "1,600",
      duration: 420,
      earliest_sale_date: "20201024",
      free_refund_before_day: null,
      free_refund_policy: false,
      hours: 7,
      id: 100441
```

```
img_url: "https://image.kkday.com/v2/image/get"
img_url_list: [...]
instant_booking: false
instant_use: false
introduction: "現在預訂高島山一日縱走・台北市區出
is_display_price: false
is_tourism_product: true
main_cat_key: "M01"
minutes: 0
name: "新竹人氣健行景點 | 高島山一日縱走 | 台北市區
order_count: 2
price: 1600
prod_oid: 100441
promo_tag_keys: []
purchase_date: null
purchase_type: null
rating_count: 0
rating_star: 0
readable_url: null
sale_price: 1600
sale_status: "1"
show_city_name: "台北, 多個城市"
```

```
show_country_name: "台灣"  
show_order_count: ""  
show_rating_star: [0, "0", "0", "0", "0"]  
show_url_order_num: ""  
star_percentage: [0, "0", "0", "0", "0"]  
theme_code: []  
url: "https://www.kkday.com/zh-tw/product/100"  
...
```

其中，筆者想要爬取票券的名稱、價格、最早可使用日期、評價及票券內容連結，分別以上是結構的name、price、earliest_sale_date、rating_star及url欄位。

三、開發Python網頁爬蟲

分析完請求連結(Request URL)及回傳結果後，筆者使用物件導向來開發一個Python網頁爬蟲，

能夠依據傳入的地區，爬取當地的一日遊票券資訊。

本文以 Visual Studio Code 為例，建立一個 scrapers.py 檔案，並且利用以下的指令安裝 requests 套件：

```
$ pip install requests
```

開啟 scrapers.py 檔案，引用 requests 模組 (Module)，如下範例：

```
1. import requests
```

建立「Kkday」類別 (Class) 及建構式 (Constructor)，讓使用者能夠在建立物件時，能夠傳入所要爬取的地區，如下範例：

```
1. import requests
```

```
2.  
3.  
4.  # KKday 網站  
5.  class Kkday:  
6.  
7.      # 建構式  
8.      def __init__(self, city_name):  
9.          self.city_name = city_name  # 城市名稱屬性
```

詳細的Python類別(Class)觀念可以參考[\[Python物件導向\]淺談Python類別\(Class\)](#)文章。接著，建立scrape()方法(Method)，用來發送請求爬取資料，如下範例：

```
1.  import requests  
2.  
3.  
4.  # KKday 網站  
5.  class Kkday:  
6.  
7.      # 建構式
```

```
8.     def __init__(self, city_name):
9.         self.city_name = city_name # 城市名稱屬性
10.
11.     def scrape(self):
```

其中，首先定義一個result串列(List)，用來打包爬取的結果，如下範例：

```
1.  import requests
2.
3.
4.  # KKday網站
5.  class Kkday:
6.
7.      #建構式
8.      def __init__(self, city_name):
9.          self.city_name = city_name # 城市名稱屬性
10.
11.      def scrape(self):
12.
13.          result = [] # 回傳結果
```


判斷如果有傳入地區，則利用 requests 模組 (Module) 的 get() 方法 (Method) 發送請求，如下範例：

```
1. import requests
2.
3.
4. # KKday 網站
5. class Kkday:
6.
7.     # 建構式
8.     def __init__(self, city_name):
9.         self.city_name = city_name # 城市名稱屬性
10.
11.     def scrape(self):
12.
13.         result = [] # 回傳結果
14.
15.         if self.city_name: # 如果城市名稱非空值
16.
17.             # 取得傳入城市的所有一日遊票券
```

```
18.         response = requests.get(  
19.             f"https://www.kkday.com/zh-tw/produ
```

取得了回傳結果後，使用Python內建的json()方法(Method)來解析，並且存取「data」欄位，如下範例：

```
1.  import requests  
2.  
3.  
4.  # KKday網站  
5.  class Kkday:  
6.  
7.      #建構式  
8.      def __init__(self, city_name):  
9.          self.city_name = city_name # 城市名稱屬性  
10.  
11.      def scrape(self):  
12.  
13.          result = [] # 回傳結果
```

```
14.  
15.         if self.city_name: # 如果城市名稱非空值  
16.  
17.             # 取得傳入城市的所有一日遊票券  
18.             response = requests.get(  
19.                 f"https://www.kkday.com/zh-tw/produ  
20.  
21.             # 資料  
22.             activities = response.json()["data"]
```

接下來，就可以透過迴圈，讀取其中的一日遊票券資料，爬取所需的欄位，如下範例：

```
1. import requests  
2.  
3.  
4. # KKday 網站  
5. class Kkday:  
6.  
7.     # 建構式  
8.     def __init__(self, city_name):
```

```
9.         self.city_name = city_name # 城市名稱屬性
10.
11.     def scrape(self):
12.
13.         result = [] # 回傳結果
14.
15.         if self.city_name: # 如果城市名稱非空值
16.
17.             # 取得傳入城市的所有一日遊票券
18.             response = requests.get(
19.                 f"https://www.kkday.com/zh-tw/produ
20.
21.             # 資料
22.             activities = response.json()["data"]
23.
24.             for activity in activities:
25.
26.                 # 票券名稱
27.                 title = activity["name"]
28.
29.                 # 票券詳細內容連結
30.                 link = activity["url"]
```

```
31.  
32.         # 票券價格  
33.         price = activity["price"]  
34.  
35.         # 最早可使用日期  
36.         booking_date = activity["earliest_s  
37.  
38.         # 評價  
39.         star = activity["rating_star"]
```

最後，將每一筆票券資料利用Python內建的dict()方法(Method)打包成字典(Dictionary)，並且加入到result串列(List)中，如下範例：

```
1. import requests  
2.  
3.  
4. # KKday 網站  
5. class Kkday:  
6.
```

```
7.         #建構式
8.         def __init__(self, city_name):
9.             self.city_name = city_name # 城市名稱屬性
10.
11.        def scrape(self):
12.
13.            result = [] # 回傳結果
14.
15.            if self.city_name: # 如果城市名稱非空值
16.
17.                # 取得傳入城市的所有一日遊票券
18.                response = requests.get(
19.                    f"https://www.kkday.com/zh-tw/produ
20.
21.                # 資料
22.                activities = response.json()["data"]
23.
24.                for activity in activities:
25.
26.                    # 票券名稱
27.                    title = activity["name"]
28.
```

```
29.         # 票券詳細內容連結
30.         link = activity["url"]
31.
32.         # 票券價格
33.         price = activity["price"]
34.
35.         # 最早可使用日期
36.         booking_date = activity["earliest_s
37.
38.         # 評價
39.         star = activity["rating_star"]
40.
41.         result.append(
42.             dict(title=title, link=link, pr
43.
44.         return result
```

假設要爬取花蓮一日遊的票券資料，就可以在建立Kkday物件時，傳入「花蓮」，並且呼叫scrape()方法(Method)來進行爬取，如下範例：

```
1. import requests
2.
3.
4. # KKday網站
5. class Kkday:
6.
7.     #建構式
8.     def __init__(self, city_name):
9.         self.city_name = city_name # 城市名稱屬性
10.
11.     def scrape(self):
12.
13.         result = [] # 回傳結果
14.
15.         if self.city_name: # 如果城市名稱非空值
16.
17.             # 取得傳入城市的所有一日遊票券
18.             response = requests.get(
19.                 f"https://www.kkday.com/zh-tw/produ
20.
21.             # 資料
22.             activities = response.json()["data"]
```



```
23.  
24.         for activity in activities:  
25.  
26.             # 票券名稱  
27.             title = activity["name"]  
28.  
29.             # 票券詳細內容連結  
30.             link = activity["url"]  
31.  
32.             # 票券價格  
33.             price = activity["price"]  
34.  
35.             # 最早可使用日期  
36.             booking_date = activity["earliest_s  
37.  
38.             # 評價  
39.             star = activity["rating_star"]  
40.  
41.             result.append(  
42.                 dict(title=title, link=link, pr  
43.  
44.         return result
```

◀ ▶

- { 'title': '【季節限定】花蓮賞鯨半日遊 | 多羅滿賞鯨團 | 市區接駁' }
- { 'title': '花蓮包車一日遊 | 太魯閣&清水斷崖&七星潭 | 花蓮包車' }
- { 'title': '花蓮一日遊 | 太魯閣&清水斷崖 | 花蓮市區出發接送' , 'url': 'http://www.taiwan.com.tw/News/1013036' }
- { 'title': '花蓮六十石山金針花海一日遊 | 2020 金針花季 | 花蓮市' }
- { 'title': '【86折優惠】花蓮包車一日遊 | 太魯閣&七星潭&佳興' }
- { 'title': '台灣花蓮 | 2020 金針花季&富興小火車一日遊 | 花蓮市' }
- { 'title': '【花蓮太平洋左岸一日遊】石門洞、石梯坪、親不知子' }
- { 'title': '【2020 金針花季】花蓮六十石山 / 赤柯山專屬包車一日' }
- { 'title': '花蓮一日遊 | 賞鯨 & 太魯閣 & 清水斷崖 & 七星潭 | 花' }
- { 'title': '【親子遊首選】花蓮遠雄海洋公園門票&來回接駁' , 'url': 'http://www.taiwan.com.tw/News/1013036' }

四、小結

當想要使用Python網頁爬蟲爬取AJAX類型網站時，首先利用開發者模式觀察網頁向伺服器取得資料的網址 (Request URL)，接著，從「Preview」頁籤進行回傳結果的確認，並且決定所要爬取的欄位名稱，最後，即可開發Python網頁爬蟲來取得所需的資料。

讀者可以試著調整請求網址(Request URL)的查詢參數，來更精準的爬取所需的票券資訊，希望本文的分享對您有所幫助。

如果您喜歡我的文章，**請幫我按五下Like**(使用Google或Facebook帳號免費註冊)，支持我創作

教學文章，回饋由LikeCoin基金會出資，完全不會花到錢，感謝大家。



**有想要看的教學內容嗎?歡迎利用以下的Google
表單讓我知道，將有機會成為教學文章，分享給
大家😊**

<https://forms.gle/UW8u9XddoY17HjaSA>

Python學習資源

- [Python學習資源整理](#)

Python網頁爬蟲推薦課程

- **Python基礎課程和網路爬蟲入門實戰**
- **Python 網頁爬蟲入門實戰**

Python網頁爬蟲 - BeautifulSoup教學

- **[Python爬蟲教學]7個Python使用
BeautifulSoup開發網頁爬蟲的實用技巧**

Python網頁爬蟲 - Selenium教學

- **[Python 爬蟲教學] 整合 Python Selenium 及
BeautifulSoup實現動態網頁爬蟲**
- **[Python爬蟲教學]3個建構Python動態網頁爬
蟲重要的等待機制**
- **[Python 爬蟲教學] 學會使用 Selenium 及
BeautifulSoup套件爬取查詢式網頁**

Python非同步網頁爬蟲

- [Python爬蟲教學]整合asyncio與aiohttp打造Python非同步網頁爬蟲
- [Python爬蟲教學]非同步網頁爬蟲使用GRequests套件提升爬取效率的實作技巧

Python網頁爬蟲應用

- [Python爬蟲教學]Python網頁爬蟲結合LINE Notify打造自動化訊息通知服務
- [Python爬蟲教學]有效利用Python網頁爬蟲幫你自動化下載圖片
- [Python+LINE Bot教學]建構具網頁爬蟲功能的LINE Bot機器人
- [Django+Python網頁爬蟲教學]打造具有網頁爬蟲功能的關鍵字查詢網頁

Python網頁爬蟲部署

- [Python爬蟲教學]教你如何部署Python網頁爬蟲至Heroku雲端平台

Python網頁爬蟲資料儲存

- [Python爬蟲教學]輕鬆學會Python網頁爬蟲與MySQL資料庫的整合方式
- [Python爬蟲教學]活用openpyxl套件將爬取的資料寫入Excel檔案
- [Python爬蟲教學]解析如何串接Google Sheet試算表寫入爬取的資料

Python網頁爬蟲技巧

- [Python爬蟲教學]Python網頁爬蟲動態翻頁的實作技巧
- [Python爬蟲教學]7個降低Python網頁爬蟲被偵測封鎖的實用方法
- [Python爬蟲教學]開發Python網頁爬蟲前需要知道的五個基本觀念
- 解析Python網頁爬蟲如何有效整合Pandas套件提升資料處理效率

大數據時代的資料應用力 PYTHON網頁爬蟲 線上課程問卷

learncodewithmike.com

大數據時代的資料應用力 | Python網頁爬蟲線上課程問卷

現在網路發達的時代，如何在有限的時間及專注力下，有效掌握網路上的資料並且加以應用，將會是大數據時代下非常重要的能力。

而**網頁爬蟲**就是最常使用的技術，透過定時自動化蒐集資料及應

0%



Learn Code With Mike

歡迎訂閱以收到免費的Python教學內容及活動訊息

怎麼稱呼您

電子郵件

對什麼主題有興趣呢

- ☐ 入門教學
- ☐ 爬蟲應用
- ☐ 資料分析
- ☐ 機器學習
- ☐ 網頁開發(Django)

免費訂閱

BUILT WITH  ConvertKit

標籤：PYTHON爬蟲教學

分享

留言



記萱 · 2020年10月7日 上午10:10

請問一下，當我再進一層連結到行程介紹的時候，找不到任何有關行程介紹的媒介，請問除了selenium以外，還有別的辦法嗎？



Mike Ku · 2020年10月11日 晚上8:16

看起來只能使用selenium了 :D

回覆



kevin · 2021年4月28日 晚上9:29

感謝教學，寫得很詳細，但是[將請求網址(Request URL)簡化為：

「`https://www.kkday.com/zh-tw/product/ajax_productlist/?keyword=新竹市&cat=TAG_4_4&sort=rdesc`」]這部分有點不知道為甚麼是這樣簡化，有什麼參考的教學嗎，謝謝

回覆



輸入留言

這個網誌中的熱門文章

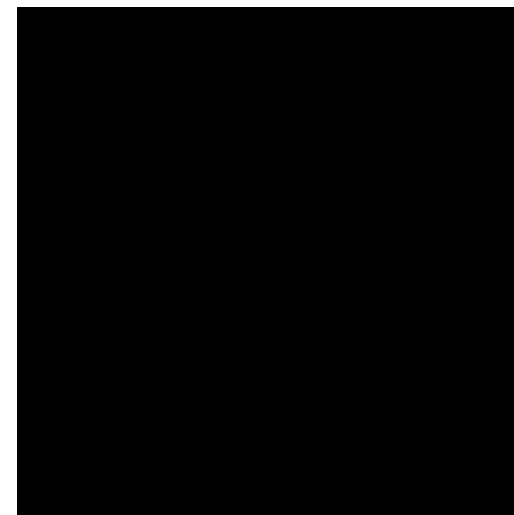
[Python爬蟲教學]7個Python使用BeautifulSoup開發網頁爬蟲的實用技巧

2月 28, 2020

Photo by Stanley Dai on Unsplash 在實務上開發專案時，很多時候會利用其他網站的資料來進行分析或運用，而取得的方式除了透過網站所提供的 API(Application Programming Interface) ...

分享 8 則留言

閱讀完整內容



[Pandas教學]資料分析必懂的Pandas DataFrame處理雙維度資料方法

11月 01, 2020

Photo by Slidebean on Unsplash 現在有許多的企業或商家，都會利用取得的使用者資料來進行分析，瞭解其中的趨勢或商機，由此可見，資料分析越來越受到重視，而這時候，能夠懂得使用...

分享 1 則留言

[閱讀完整內容](#)

[Python物件導向]淺談Python類別(Class)

1月 02, 2020

Photo by Bram Naus on Unsplash 在學習程式語言時，或多或少都有聽過物件導向程式設計 (Object-oriented programming，簡稱 OOP)，

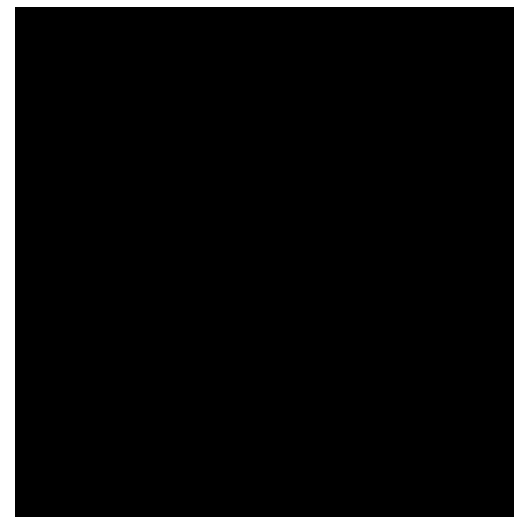
分享 27 則留言

閱讀完整內容

[Python教學]搞懂5個Python迴圈常見用法

12月 04, 2019

Photo by Scott Webb on Unsplash 在撰寫程式的過程中，都有機會要重複執行一些相同的運算，但是重複撰寫好幾次同樣的運算看起來非常的沒有效率，所以在這個情況下我們通常會使用...



分享 4 則留言

閱讀完整內容

[Python教學]5個必知的Python Function觀念整理

12月 22, 2019

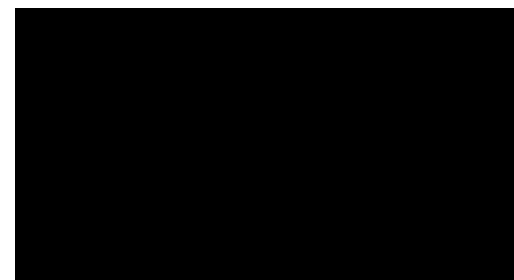


Photo by Susan Holt Simpson on Unsplash 在寫程式碼時有一個非常重要的觀念是 DRY(Don't Repeat Yourself) , 意思是避免同樣的程式碼重複出現在很多個地方 , 除了可讀性很低外 , 也不...

分享 張貼留言

閱讀完整內容

[Python+LINE Bot教學]6步驟快速上手LINE Bot機器人

6月 28, 2020

Photo by Yura Fresh on Unsplash 每當朋友或家人要聚餐時 , 是不是總要花很長的時間尋找評價不錯的餐廳?不但要確認營業時間、消費價格及地點 , 還要觀看許多的美食文章才有辦法決定 , 這...

分享 47 則留言

閱讀完整內容

[Pandas教學]5個實用的Pandas讀取Excel檔案資料技巧

12月 06, 2020

Photo by LinkedIn Sales Navigator on Unsplash

日常生活中，不免俗的都會有需要整理大量資料的需求，而最常用的文書軟體就是Excel，這時候該如何有效讀取Excel檔中的資料，進行額外的...

[分享](#) [張貼留言](#)

[閱讀完整內容](#)

[Python教學]Python Lambda Function應用技巧分享

12月 29, 2019

Photo by Fatos Bytyqi on Unsplash Lambda 函

式，也就是匿名函式，不需要定義名稱，只有一行運算式，語法非常簡潔，功能強大，所以現代

分享 7 則留言

閱讀完整內容

[Python爬蟲教學]整合Python Selenium及BeautifulSoup實現動態網頁爬蟲

5月 31, 2020

Photo by LAUREN GRAY on Unsplash 相信大家
都知道，取得資料後能夠進行許多的應用，像是
未來的趨勢預測、機器學習或資料分析等，而有
效率的取得資料則是這些應用的首要議題，網頁...

分享 18 則留言

閱讀完整內容

[Python爬蟲教學]有效利用Python網頁爬蟲爬取免費的Proxy IP清單

10月 24, 2021

Photo by Cytonn Photography on Unsplash 在開發網頁爬蟲的過程中，是不是會擔心被偵測或封鎖，而爬不到所需的資料呢？有些大型網站為了保護網頁上的資料不被大量的爬取，會特別偵測 ...

[分享](#) [張貼留言](#)

[閱讀完整內容](#)



關於創辦人



Mike Ku

是一位喜歡旅遊的網頁工程師，由於對Python程式語言非常有興趣，所以創辦了「Learn Code With Mike」網站，提供線上

的Python「入門教學、爬蟲應用、資料分析與網頁開發」等主題的教學，透過小專案實作的方式來幫助初學者們學習Python程式語言，並且有能力開發屬於自己的應用程式。

瀏覽簡介

取得最新發佈的免費Python教學

請輸入電子郵件

免費訂閱

文章分類



Django套件應用 (3)

Django教學系列 (16)

LINE Bot應用教學 (6)

Pandas教學系列 (22)

Pandas資料清理 (8)

Python初學入門 (13)

Python爬蟲教學 (29)

[Python物件導向](#) (6)

[Python資料結構](#) (3)

[Python實戰應用](#) (5)

[顯示更多](#)



MikeKu1116



Learn Code With Mike



learncodewithmike@gmail.com

Python推薦課程

[The Complete Python Course](#)



Python - 輕鬆學會寫程式



線上課程平台推薦



資料分析學習平台推薦

 技術提供 : Blogger

COPYRIGHT © 2019-2021 BY MIKE 古耕全 @ Learn Code With Mike版權所有