

LEARN CODE WITH MIKE

搜尋



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

5月 31, 2020



分享

標籤

Python爬蟲教學

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，網頁爬蟲則是其中的一個方法。

網頁爬蟲就是能夠取得網頁原始碼中的元素資料技術，但是，有一些網頁較為特別，像是社群平台，需先登入後才能進行資料的爬取，或是電商網站，無需登入，但是要透過滾動捲軸，才會動態載入更多的資料，而要爬取這樣類型的網頁爬蟲，就稱為動態網頁爬蟲。

該如何實作呢？本文將使用Python Selenium及BeautifulSoup套件來示範動態網頁爬蟲的開發過程，重點包含：

- BeautifulSoup vs Selenium
- 安裝Selenium及Webdriver
- 安裝BeautifulSoup
- Selenium get()方法

- Selenium元素定位
- Selenium send_keys()方法
- Selenium execute_script方法
- BeautifulSoup find_all()方法
- BeautifulSoup getText()方法

一、BeautifulSoup vs Selenium

BeautifulSoup套件相信對於開發網頁爬蟲的人員來說，應該都有聽過，能夠解析及取得HTML原始碼各個標籤的元素資料，擁有非常容易上手的方法(Method)，但是，對於想要爬取動態網頁資料來說，則無法達成，因為BeautifulSoup套件並沒有模擬使用者操作網頁的方法(Method)，像是

輸入帳號密碼進行登入或滾動捲軸等，來讓網頁動態載入資料，進行爬取的動作。

所以，這時候，就可以使用被設計於自動化測試的Selenium套件，來模擬使用者的動作，進行登入後爬取資料或滾動卷軸，並且能夠執行JavaScript程式碼，這些就是Selenium與BeautifulSoup套件最大不同的地方。對於開發Python動態爬蟲來說，就可以結合Selenium套件以上的特點，讓網頁動態載入資料後，再利用BeautifulSoup套件簡潔的方法(Method)，將所需的資料爬取下來。

本文就是利用這樣的概念，利用Selenium套件登入Facebook後，前往粉絲專頁，執行滾動卷軸的JavaScript程式碼，讓網頁動態載入資料後，再使用BeautifulSoup套件爬取貼文標題。

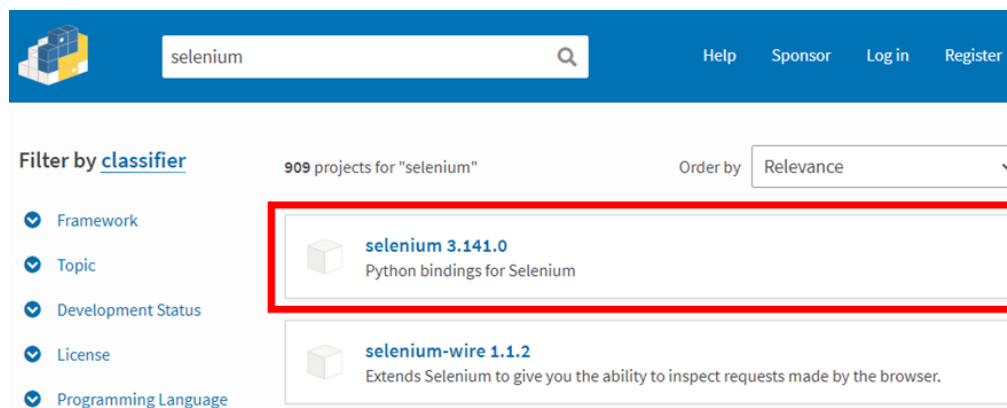
二、安裝Selenium及Webdriver

首先，利用以下指令安裝Python的Selenium套件：

```
$ pip install selenium
```

接下來，要讓Selenium套件能夠自動開啟瀏覽器，所以需要安裝對應的Webdriver(驅動程式)，本文將以Chrome瀏覽器來做示範。

前往Python 套件儲存庫 **PyPI(Python Package Index)**，輸入Selenium進行查詢，如下圖：



點擊進入後，往下可以看到Drivers的地方，如下圖：

Drivers

Selenium requires a driver to interface with the chosen browser. Firefox, for example, requires [geckodriver](https://github.com/mozilla/geckodriver/releases), which needs to be installed before the below examples can be run. Make sure it's in your *PATH*, e. g., place it in */usr/bin* or */usr/local/bin*.

Failure to observe this step will give you an error `selenium.common.exceptions.WebDriverException: Message: 'geckodriver' executable needs to be in PATH`.

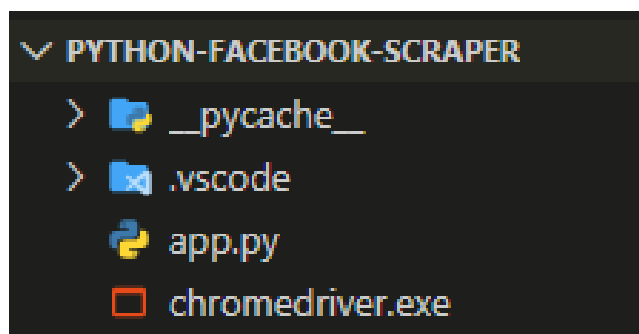
Other supported browsers will have their own drivers available. Links to some of the more popular browser drivers follow.

Chrome:	https://sites.google.com/a/chromium.org/chromedriver/downloads
Edge:	https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/
Firefox:	https://github.com/mozilla/geckodriver/releases
Safari:	https://webkit.org/blog/6900/webdriver-support-in-safari-10/

選擇要使用的瀏覽器，進行下載及解壓縮即可。

完成後，建議將這個Webdriver檔放置在Python網

頁爬蟲的專案資料夾中，如下圖：



三、安裝BeautifulSoup

接著，利用以下指令安裝BeautifulSoup套件：

```
$ pip install beautifulsoup4
```

四、Selenium get()方法

開啟開發工具，本文以Visual Studio Code為例，建立一個app.py檔案，引用time模組與Selenium套件的webdriver模組及Options類別，如下範例：

```
1. from selenium import webdriver
2. from selenium.webdriver.chrome.options import Options
3. import time
4.
5.
6. options = Options()
```

```
7. options.add_argument("--disable-notifications")
8.
9. chrome = webdriver.Chrome('./chromedriver', chrome_
10. chrome.get("https://www.facebook.com/")
```

範例中第6、7行的options物件，主要用途為取消網頁中的彈出視窗，避免妨礙網路爬蟲的執行。

而第9行就是建立webdriver物件，傳入剛剛所下載的「瀏覽器驅動程式路徑」及「瀏覽器設定(chrome_options)」，其中的「瀏覽器驅動程式路徑」一定要傳入，而「瀏覽器設定(chrome_options)」則可視情況傳入，為選擇性的。

有了Selenium webdriver的物件後，就可以透過第10行的get()方法，前往要爬取的網頁網址。

到這邊執行app.py程式碼時，就會自動開啟Chrome瀏覽器，並且前往Facebook，如下圖：



五、Selenium元素定位

到了Facebook的登入頁面後，要讓Python自動輸入「電子郵件」及「密碼」，首先需利用Selenium套件進行元素的定位，在Chrome瀏覽器按下F12後，可以看到這兩個欄位的id為「email」「pass」，如下範例：

```
▼<table cellpadding="0" role="presentation">
  ▼<tbody>
    ▶<tr>...</tr>
    ▼<tr>
      ▼<td>
        <input type="email" class="inputtext login_form_input_box"
          name="email" id="email" data-testid="royal_email">
      </td>
      ▼<td>
        <input type="password" class="inputtext login_form_input_box"
          name="pass" id="pass" data-testid="royal_pass">
      </td>
      ▶<td>...</td>
    </tr>
    ▶<tr>...</tr>
  </tbody>
</table>
```

這時候，就可以使用 Selenium 套件的 `find_element_by_id` 方法來分別建立「電子郵件」及「密碼」的物件，如下範例：

```
1. from selenium import webdriver
2. from selenium.webdriver.chrome.options import Options
3. import time
4.
5.
6. options = Options()
7. options.add_argument("--disable-notifications")
8.
9. chrome = webdriver.Chrome('./chromedriver', chrome_options=options)
10. chrome.get("https://www.facebook.com/")
11.
12. email = chrome.find_element_by_id("email")
13. password = chrome.find_element_by_id("pass")
```

六、Selenium `send_keys()` 方法

接下來，就可以透過Selenium套件的send_keys()方法，來模擬使用者輸入資料，最後，利用submit()方法送出，進行登入的動作，如下範例：

```
1. from selenium import webdriver
2. from selenium.webdriver.chrome.options import Options
3. import time
4.
5.
6. options = Options()
7. options.add_argument("--disable-notifications")
8.
9. chrome = webdriver.Chrome('./chromedriver', chrome_options=options)
10. chrome.get("https://www.facebook.com/")
11.
12. email = chrome.find_element_by_id("email")
13. password = chrome.find_element_by_id("pass")
14.
15. email.send_keys('example@gmail.com')
```

```
16. password.send_keys('*****')
17. password.submit()
```

七、Selenium execute_script方法

由於本文要爬取Learn Code With Mike粉絲專頁的文章標題，所以登入後，需要再一次的使用Selenium套件的get()方法前往粉絲專頁，如下範例：

```
1. from selenium import webdriver
2. from selenium.webdriver.chrome.options import Options
3. import time
4.
5.
6. options = Options()
7. options.add_argument("--disable-notifications")
8.
9. chrome = webdriver.Chrome('./chromedriver', chrome_
```

```
10. chrome.get("https://www.facebook.com/")
11.
12. email = chrome.find_element_by_id("email")
13. password = chrome.find_element_by_id("pass")
14.
15. email.send_keys(config.email)
16. password.send_keys(config.password)
17. password.submit()
18.
19. time.sleep(3)
20. chrome.get('https://www.facebook.com/learncodewithmike')
```

範例中第19行的sleep()方法，主要用途是暫停執行程式碼3秒，讓網頁利用這段時間載入元素，再前往粉絲專頁，避免直接執行時，如果網頁還沒有載入完成，就會發生讀取不到所要爬取的元素或網頁卡住等例外情況，所以這行程式碼可視情況使用。

20210126補充說明

在前往粉絲專頁後，如果有彈跳出聊天室窗，而妨礙Python網頁爬蟲的執行時，則可以利用Selenium套件的find_element_by_css_selector()方法(Method)，傳入關閉(X)按鈕的樣式類別(class)來定位，並且呼叫click()點擊方法(Method)關閉聊天室窗，如下範例第22-25行：

```
1. from selenium import webdriver
2. from selenium.webdriver.chrome.options import Options
3. import time
4.
5.
6. options = Options()
7. options.add_argument("--disable-notifications")
8.
9. chrome = webdriver.Chrome('./chromedriver', chrome_
```

```
10. chrome.get("https://www.facebook.com/")
11.
12. email = chrome.find_element_by_id("email")
13. password = chrome.find_element_by_id("pass")
14.
15. email.send_keys('example@gmail.com')
16. password.send_keys('*****')
17. password.submit()
18.
19. time.sleep(3)
20. chrome.get('https://www.facebook.com/learncodewithmike')
21.
22. time.sleep(5)
23. chat = chrome.find_element_by_css_selector(
24.     '.oajrlxb2.gs1a9yip.g5ia77u1.mtkw9kbi.tlpljxtp.
25.     chat.click()
```

PS.如果在前往粉絲專頁後，沒彈跳出聊天室窗，則無需增加第22-25行，否則會發生例外錯誤。

接著，按下F12，可以看到HTML的原始碼中，只有顯示一些文章，而不會全部載入，必須透過滾動捲軸來動態載入更多的文章，所以這時候就可以利用Selenium套件的execute_script()方法，來執行滾動捲軸的JavaScript程式碼，如下範例：

```
1. from selenium import webdriver
2. from selenium.webdriver.chrome.options import Options
3. import time
4.
5.
6. options = Options()
7. options.add_argument("--disable-notifications")
8.
9. chrome = webdriver.Chrome('./chromedriver', chrome_
10. chrome.get("https://www.facebook.com/")
11.
```

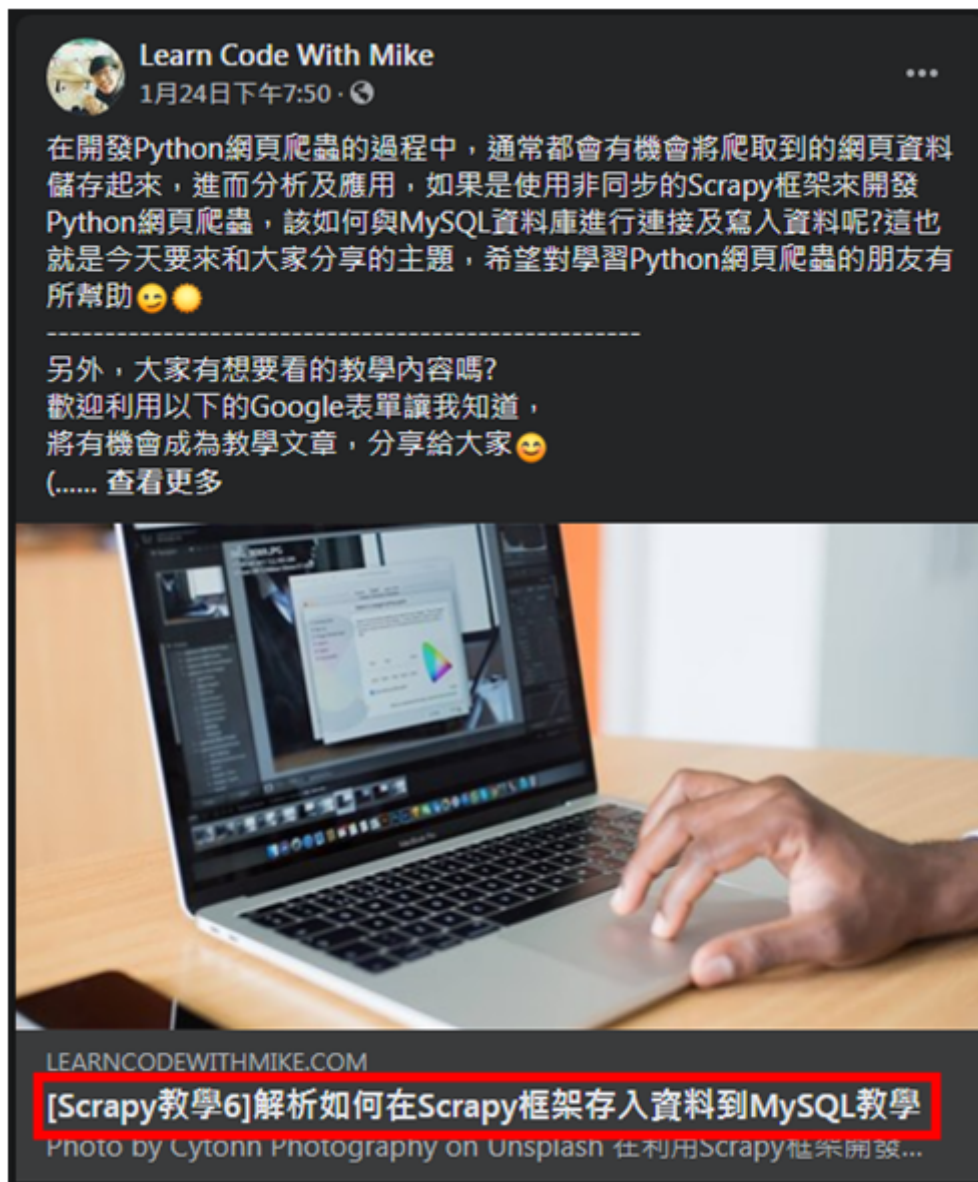
```
12. email = chrome.find_element_by_id("email")
13. password = chrome.find_element_by_id("pass")
14.
15. email.send_keys('example@gmail.com')
16. password.send_keys('*****')
17. password.submit()
18.
19. time.sleep(3)
20. chrome.get('https://www.facebook.com/learncodewithmike')
21.
22. for x in range(1, 4):
23.     chrome.execute_script("window.scrollTo(0,document.body.scrollHeight)")
24.     time.sleep(5)
```

範例中，使用Python的迴圈，執行滾動捲軸3次，每滾動一次就暫停執行程式碼5秒，讓Facebook載入更多的文章。

八、BeautifulSoup find_all()方法

20210126補充說明

前面透過Selenium套件進行登入、前往粉絲專頁
及滾動捲軸來動態載入更多的文章，這時候就可
以來查看所要爬取的文章標題原始碼，如下圖：



在紅框的文章標題，點擊右鍵，選擇「檢查」，
可以看到它的HTML原始碼為：

```
▼<span class="a8c37x1j ni8dbmo4 stjgntxs 19j0dhe7 ojkyduve" style="-webkit-box-orient: vertical;
-webkit-line-clamp: 2; display: -webkit-box;">
  <span dir="auto">[Scrapy教學6]解析如何在Scrapy框架存入資料到MySQL教學</span> == $0
</span>
```

將目前瀏覽器的 `page_source`(原始碼)，傳入 `BeautifulSoup` 類別，來進行解析，如下範例：

```
1. from selenium import webdriver
2. from selenium.webdriver.chrome.options import Options
3. from bs4 import BeautifulSoup
4. import time
5.
6.
7. options = Options()
8. options.add_argument("--disable-notifications")
9.
10. chrome = webdriver.Chrome('./chromedriver', chrome_options=options)
11. chrome.get("https://www.facebook.com/")
12.
13. email = chrome.find_element_by_id("email")
```

```
14. password = chrome.find_element_by_id("pass")
15.
16. email.send_keys('example@gmail.com')
17. password.send_keys('*****')
18. password.submit()
19.
20. time.sleep(3)
21. chrome.get('https://www.facebook.com/learncodewithmike')
22.
23. for x in range(1, 4):
24.     chrome.execute_script("window.scrollTo(0,document.body.scrollHeight)")
25.     time.sleep(5)
26.
27. soup = BeautifulSoup(chrome.page_source, 'html.parser')
```

接著，就可以使用BeautifulSoup套件的find_all()方法，取得文章標題的元素，也就是樣式類別(class)為a8c37x1j...的所有標籤，如下範例：


```
1. from selenium import webdriver
2. from selenium.webdriver.chrome.options import Options
3. from bs4 import BeautifulSoup
4. import time
5.
6.
7. options = Options()
8. options.add_argument("--disable-notifications")
9.
10. chrome = webdriver.Chrome('./chromedriver', chrome_options=options)
11. chrome.get("https://www.facebook.com/")
12.
13. email = chrome.find_element_by_id("email")
14. password = chrome.find_element_by_id("pass")
15.
16. email.send_keys('example@gmail.com')
17. password.send_keys('*****')
18. password.submit()
19.
20. time.sleep(3)
21. chrome.get('https://www.facebook.com/learncodewithmike')
22.
```

```
23.     for x in range(1, 4):
24.         chrome.execute_script("window.scrollTo(0,document)
25.         time.sleep(5)
26.
27.     soup = BeautifulSoup(chrome.page_source, 'html.pars
28.
29.     titles = soup.find_all('span', {
30.         'class': 'a8c37x1j ni8dbmo4 stjgntxs l9j0dhe7 a
```

九、BeautifulSoup getText()方法

最後，利用Python迴圈讀取每一個文章標題的元素，透過BeautifulSoup套件的getText()方法，取得所有標籤中的文字，如下範例：

```
1.     from selenium import webdriver
2.     from selenium.webdriver.chrome.options import Options
3.     from bs4 import BeautifulSoup
4.     import time
```

```
5.
6.
7.  options = Options()
8.  options.add_argument("--disable-notifications")
9.
10. chrome = webdriver.Chrome('./chromedriver', chrome_
11. chrome.get("https://www.facebook.com/")
12.
13. email = chrome.find_element_by_id("email")
14. password = chrome.find_element_by_id("pass")
15.
16. email.send_keys('example@gmail.com')
17. password.send_keys('*****')
18. password.submit()
19.
20. time.sleep(3)
21. chrome.get('https://www.facebook.com/learncodewithm
22.
23. for x in range(1, 4):
24.     chrome.execute_script("window.scrollTo(0,docume
25.     time.sleep(5)
26.
```

```
27. soup = BeautifulSoup(chrome.page_source, 'html.parser')
28.
29. titles = soup.find_all('span', {
30.     'class': 'a8c37x1j ni8dbmo4 stjgntxs l9j0dhe7 o
31.
32. for title in titles:
33.
34.     post = title.find('span', {'dir': 'auto'})
35.
36.     if post:
37.         print(post.getText())
38.
39. chrome.quit()
```

20210126補充說明

為了要更精準的爬取到粉絲專頁的文章標題，在迴圈的讀取過程中，第34行再透過 find() 方法 (Method)，取得所有樣式類別(class)為a8c37x1j...

的標籤子元素，第36行判斷如果不為None，則印出爬取結果，如下圖：

```
[Django教學18]善用Django Filter來快速建構多條件的查詢功能
[Django教學17]Django專案整合Amazon S3雲端服務5步驟就上手
[Django教學16]建構Django網站該知道的Email寄送方式
[Django教學15]Django連接PostgreSQL資料庫手把手教學
[Django教學14]解析5個常用的Django Class-based Views使用方式
[Django教學13]Django Allauth套件整合Google登入驗證實作教學
[Django教學12]利用輕量的Django tastypie來打造屬於自己的API
[Django教學11]Django部署到Heroku雲端平台教學指南
[Django教學10]客製化Django ModelForm表單欄位驗證的技巧
[Django教學9]6個步驟快速搞懂Django上傳圖片的功能
Pluralsight - Unlimited Online Developer, IT, and Cyber Security Training
[Django教學8]Django UserCreationForm實作網站登入驗證及註冊功能分享
[Django教學7]透過Django ModelForm快速開發CRUD專案教學
[Django教學6]Django Template(樣板)整合Bootstrap實戰教學
[Django教學5]Django Template(樣板)開發快速上手
[Django教學]實用的Django Administration操作及客製化技巧
[Django教學]Django Migration(資料遷移)的重要觀念
[Django教學]建立Django應用程式(APP)
[Django教學]3步驟快速安裝Django網站框架
Python寄送電子郵件實作教學-以Gmail為例
7個Python使用BeautifulSoup開發網頁爬蟲的實用技巧
```

十、小結

以上就是結合Python的Selenium及BeautifulSoup
套件實現動態登入的網頁爬蟲，完整的程式碼可

以參考下方的GitHub網址，如果在實作的過程中有碰到問題，歡迎在底下留言分享。

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。



GitHub網址：

<https://github.com/mikeku1116/python-facebook-scraper>

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]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爬蟲教學]快速搞懂AJAX動態載入網頁的爬取秘訣

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

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



tsung · 2020年6月1日 凌晨2:55

您好：

我有在您的Github上發PR

主要是針對Selenium自動下載對應Chrome Webdriver 版本的
套件

再請您查看～



Mike Ku · 2020年6月5日 晚上10:00

哈囉TSUNG YU CHEN您好：

WOW原來有這麼好的實作方式，非常感謝您的分享，

讓我學到了一課～：)

已經Merge PR囉～

回覆



MINI · 2020年6月1日 上午11:18

感謝分享好文章～



Mike Ku · 2020年6月6日 晚上7:52

MINI感謝您的支持唷：)

[回覆](#)**Unknown** · 2020年7月19日 下午4:20

你好 為什麼完整的程式碼中多了

```
import webdriver_manager  
from webdriver_manager.chrome import  
ChromeDriverManager
```

ChromeDriverManager.install() 跟你原本文章中用
executable_path有什麼不同呢？

我下載你的代碼之後，把ChromeDriverManager的部分拿掉，
改回executable_path的方式，運行之後，每一步都有執行，但是，就是最後一步，沒有把標題print out不知道是哪裡出了問題呢？以下是我的代碼。謝謝

```
from selenium import webdriver  
from selenium.webdriver.chrome.options import Options  
from bs4 import BeautifulSoup  
import time
```

```
options = Options()  
options.add_argument("--disable-notifications")
```

```
chrome =  
webdriver.Chrome(executable_path='./chromedriver',  
chrome_options=options)  
chrome.get("https://www.facebook.com/")
```

```
email = chrome.find_element_by_id("email")
```

```
password = chrome.find_element_by_id("pass")

email.send_keys(' * * * * * ')
password.send_keys(' * * * * * * * ')
password.submit()

time.sleep(3)
chrome.get('https://www.facebook.com/learncodewithmike')

for x in range(1, 4):
    chrome.execute_script("window.scrollTo(0,document.body.scrollHeight)")
    time.sleep(5)

soup = BeautifulSoup(chrome.page_source, 'html.parser')

titles = soup.find_all('span', {
    'class': 'oi732d6d ik7dh3pa d2edcug0 qv66sw1b c1et5uql
a8c37x1j s89635nw ew0dbk1b a5q79mjw g1cxx5fr lrzzd5p
oo9gr5id hzawbc8m'})

for title in titles:
    print(title.getText())

chrome.quit()
```



Mike Ku · 2020年7月19日 晚上9:17

您好：

1.本文的寫法與使用webdriver-manager不一樣的地方是，本文的寫法需要手動到PyPI下載對應版本的

driver，並且未來如果您的本機Chrome版本升級時，舊專案就需要手動到PyPI下載較新版本的driver。而使用webdriver-manager，在「每一次」執行時，它就會自動使用與Chrome版本相同的driver，無需手動到PyPI下載，當您的Chrome版本更新時，舊專案也都不會影響。

2.您所貼的程式碼如果要使用本文的寫法，應該是：

```
chrome = webdriver.Chrome('./chromedriver',  
chrome_options=options)  
而不是  
chrome =  
webdriver.Chrome(executable_path='./chromedriver', chrome_options=options)
```

希望以上有回答到您的疑問 :)

回覆

匿名 · 2020年10月15日 下午2:23

感謝大大的教學~~~
受益良多 :D



Mike Ku · 2020年10月17日 晚上8:20

不客氣 :) 感謝您的支持~~

回覆



? · 2020年12月9日 晚上8:05

感謝寫的這麼詳細。
在大大的blog受益良多



Mike Ku · 2020年12月13日 下午3:41

謝謝您的支持和鼓勵，會持續分享Python教學文章給大家，如果這樣的教學內容對您有所幫助，別忘了追蹤Learn Code With Mike粉絲專頁與訂閱部落格文章唷 :)

回覆



Unknown · 2020年12月31日 下午4:27

感謝Mike前輩的教學，教學step by step，使人容易瞭解，讓我能爬社群網站Dcard時有所借鑑。
因為同樣需要捲動下滑頁面(≧▽≦)(≧▽≦)
爬動態網頁真得有些難度呢(-`д´-。)(-`д´-。)
再次感謝大佬的教學(,,ᄇᄇ,,)
立馬繼續瀏覽大大的其他好文章~~



Mike Ku · 2021年1月2日 晚上8:11

感謝您的支持阿，爬動態網頁真的有些難度，希望有幫助到您囉~ :)

回覆

匿名 · 2021年2月2日 晚上11:33

Hello Mike

謝謝你寫這麼詳盡的文章可以讓大家參考學習
但是有個問題想要請教：

我的程式碼如下：

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.options import Options
import time
```

```
options = Options()
options.add_argument("--disable-notifications")
```

```
driver = webdriver.Chrome()
driver.get('https://www.google.com/')
element = driver.find_element_by_class_name('gLfyf.gsfi')
element.send_keys('123')
```

目前可以順利開啟到<https://www.google.com/> 這個頁面，但無法輸入想要搜尋的字進去輸入框，以上面的程式碼來說，想要搜尋的字就是123，也無任何報錯，希望可以請教Mike這個問題可能是出在哪邊？

最後再次謝謝你詳盡的文章：)



Unknown · 2021年5月6日 上午11:26

你最後要加這個 `element.submit()`
才能輸入123

回覆



Unknown · 2021年6月9日 上午10:19

我用selenium的webdriver使用chrome

要爬蟲的網頁裡有一個element的html如下(發言不允許html語法，所以改用全形〈〉)：

```
〈select onchange="ChgFD(this.value);"〉  
〈option value="指標"〉指標 〈/option〉  
〈option value="統計"〉統計 〈/option〉  
〈option value="EB"〉EB 〈/option〉  
〈option value="年度"〉年度 〈/option〉  
〈option value="季度"〉季度 〈/option〉  
〈/select〉
```

我select_element_by_tag_name抓到這個tag 〈select〉
如下程式碼

```
# python的程式碼  
myelement=browser.find_element_by_tag_name('select')  
myoptions =  
myelement.find_elements_by_tag_name('option')  
myselect.select_by_value(myoptions[4].get_attribute("value"))  
# 選項設了，但沒有觸發onchange="ChgFD(this.value);" 這個  
# 事件，請問如何執行onchange事件  
time.sleep(5)  
print(browser.page_source)
```

但是select元件的onchange事件不能自動觸發
要如何trigger這個event？

[回覆](#)



GARY · 2021年8月27日 凌晨12:27

請問若是要將整理出來的資料匯出至text檔，該怎麼做呢？

[回覆](#)



Unknown · 2021年9月3日 下午2:13

感謝Mike的教學，讓人清楚瞭解程式，謝謝！

執行捲軸時出現USB: usb_device_handle_win.cc:1048 Failed to read descriptor from node connection: 連結到系統的某個裝置失去作用。

[回覆](#)



Unknown · 2021年9月3日 下午2:36

應該是在這行chrome = webdriver.Chrome('./chromedriver')
出現USB: usb_device_handle_win.cc:1048 Failed to read descriptor from node connection: 連結到系統的某個裝置失去作用。

[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)，它是一個具有物件 (Object) 概念的開發方式，能...

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

[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，所以現代程式語言如 Java、C# 及 Python 等都支援 ...

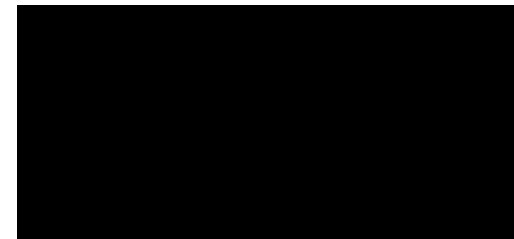
[分享](#) [7 則留言](#)

[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