網際網路大數據 & 爬蟲應用2

Dr. Cheng-Yu Ma

Postdoctoral Research Fellow

CAIM, Chang Gung Memorial Hospital

長庚醫療人工智能核心實驗室

Center for Artificial Intelligence in Medicine,
Chang Gung Memorial Hospital

馬誠佑 博士

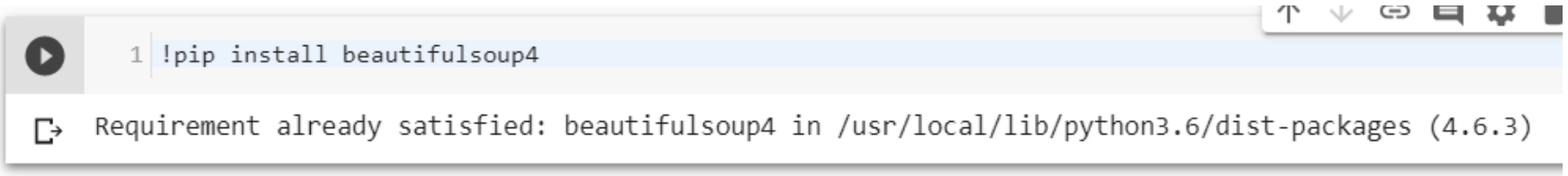
博士：國立清華大學資訊工程研究所

碩士：國立陽明大學生物醫學資訊所

Beautifulsoup

- For parsing HTML and XML documents (including having malformed markup, i.e. non-closed tags, so named after tag soup). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for web scraping.

屬性或方法	說明
title	傳回網頁標題，例如：sp.title。
text	傳回去除所有 HTML 標籤後的網頁文字內容。
find()	傳回第一個符合條件的 tag。例如：sp.find("a")。
find_all()	傳回所有符合條件的 tag。例如：sp.find_all("a")。
select()	傳回指定 CSS 選擇器如 id 或 class 的內容，例如：以 id 讀取 sp.select("#id")、以 class 讀取 sp.select(".classname")。



Beautiful Soup: example

```
# 引入 Beautiful Soup 模組
from bs4 import BeautifulSoup

# 原始 HTML 程式碼
html_doc = """ <html><head><title>Hello
World</title></head> <body><h2>Test Header</h2>
<p>This is a test.</p> <a id="link1"
href="/my_link1">Link 1</a> <a id="link2"
href="/my_link2">Link 2</a> <p>Hello, <b
class="boldtext">Bold Text</b></p> </body></html> """

# 以 Beautiful Soup 解析 HTML 程式碼
soup = BeautifulSoup(html_doc, 'html.parser')

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

排版後的 HTML

```
<html>
 <head>
  <title>
   Hello World
  </title>
 </head>
 <body>
  <h2>
   Test Header
  </h2>
  <p>
   This is a test.
  </p>
  <a href="/my_link1" id="link1">
   Link 1
  </a>
  <a href="/my_link2" id="link2">
   Link 2
  </a>
  <p>
   Hello,
   <b class="boldtext">
    Bold Text
   </b>
  </p>
 </body>
</html>
```

取得標籤文字內容

```
# 網頁標題 HTML 標籤
title_tag = soup.title
print(title_tag)

# 網頁的標題文字
print(title_tag.string)
```

```
<title>Hello World</title>
Hello World
```

find(), find_all() 搜尋標籤

```
# 所有的超連結
a_tags = soup.find_all('a')
for tag in a_tags:
# 輸出超連結的文字
    print(tag.string)
```

取出標籤屬性

```
for tag in a_tags:
# 輸出超連結網址
    print(tag.get( ‘href’ ))
```

```
/my_link1
/my_link2
```

Html tag structure

- 1.

- 2. 空元素 (無終止標籤)

：強制換行，在文本中指定下一行開始的位置

  <hr>：水平分隔線，可以在視覺上將文檔分隔成各個不同主題的部分

  Ex: <hr size="8px" align="center" width="100%">

  <img>：圖像，從網頁上鏈接圖像，有兩個必需的屬性src(引用路徑)和alt(替代文本)

  Ex: <img src="upload/20131017185230.gif" alt="替代文字二" title="範例圖片二" width="100px" height="100px">

  <input>：表單輸入元素，用於接收用戶輸入的訊息(text、radio、checkbox等等)

  <link>：規定外部資源與當前文檔的關係(放在head中)，常用於鏈接外部樣式表(引入css)

  <meta>：用來提供有關頁面的無數據訊息(放在head中)，例如網站的關鍵字和描述等等

Reference: http://blog.ywpu.me/2015/01/27-html.html

同時搜尋多種標籤

```
# 搜尋所有超連結與粗體字
tags = soup.find_all(["a", "b"])
print(tags)
# 限制搜尋結果數量
tags = soup.find_all(["a", "b"], limit=2)
print(tags)
```

```
[<a href="/my_link1" id="link1">Link 1</a>, <a href="/my_link2" id="link2">Link 2</a>, <b class="boldtext">Bold Text</b>]
[<a href="/my_link1" id="link1">Link 1</a>, <a href="/my_link2" id="link2">Link 2</a>]
```

以文字內容搜尋

```
links_html = """ <a id="link1" href="/my_link1">Link
One</a> <a id="link2" href="/my_link2">Link Two</a>
<a id="link3" href="/my_link3">Link Three</a> """

soup = BeautifulSoup(links_html, 'html.parser')

# 搜尋文字為「Link One」的超連結

soup.find_all("a", string="Link One")
```

```
[<a href="/my_link1" id="link1">Link One</a>]
```

爬取頁面範例

- 中時新聞網首頁: https://www.chinatimes.com/?chdtv

```
<section class="focus-news">
 <header>
  <h4 class="section-title">
   焦點新聞
  </h4>
 </header>
 <ul class="vertical-list list-style-none row">
  <li class="col-md-6">
   <div class="row">
    <div class="col-5 col-sm-4">
     <div class="thumb-photo">
      <div class="cropper">
       <a href="https://www.chinatimes.com/realtimenews/20220808004890-260407?ctrack=pc_main_recmd_p01" target="">
        <img alt="陸無人機今夜再擾金門 守軍發射信號彈示警" class="photo" loading="lazy" src="https://images.chinatimes.com/newsphoto/2022-08-08/330/20220808004891.jp
       </a>
      </div>
     </div>
    </div>
    <div class="col">
     <h3 class="title">
      <a href="https://www.chinatimes.com/realtimenews/20220808004890-260407?ctrack=pc_main_recmd_p01" target="">
       陸無人機今夜再擾金門 守軍發射信號彈示警
      </a>
     </h3>
     <div class="meta-info">
      <time datetime="2022-08-08 21:27">
       <span class="hour">
        21:27
       </span>
      </time>
      <div class="category">
       <a href="/realtimenews/260407">
        政治
       </a>
      </div>
     </div>
    </div>
   </div>
  </li>
  <li class="col-md-6">
   <div class="row">
    <div class="col-5 col-sm-4">
     <div class="thumb-photo">
      <div class="cropper">
       <a href="https://www.chinatimes.com/realtimenews/20220808004804-260404?ctrack=pc_main_recmd_p02" target="">
        <img alt="19歲啦啦隊女神親吻告白「我愛你」 放閃照曝光2人關係" class="photo" loading="lazy" src="https://images.chinatimes.com/newsphoto/2022-08-08/330/2022080
       </a>
      </div>
```

```python
import requests
from bs4 import BeautifulSoup
from requests.models import LocationParseError
output = open('tt.html','w',encoding='utf-8')
html = requests.get('https://www.chinatimes.com/?chdtv')
soup = BeautifulSoup(html.text, 'html.parser')

output.write(soup.prettify())
output.close()

all_sections = soup.find_all('section')
target_section = None
for section in all_sections:
    #print(section.get('class'))
    if section.get('class') == ['focus-news']:
        target_section = section
        break

items = target_section.find_all('h3')
for item in items:
    print(item.string)
    print(item.find('a').get('href'))
```

```python
# Step2: 利用找出來的所有連結(links), 就可以把所有網頁的內容全部「爬」回來
# 每次捉取頁面必需休息一下
import time
count = 0
pageSoups = []
for link in links:
    r = requests.get(link)
    soup = BeautifulSoup(r.text, 'html5lib')
    # 要分析頁面，所以要先使用BeautifulSoup, https://www.crummy.com/software/BeautifulSoup/bs4/doc/
    pageSoups.append(soup)
    print('成功：%s' % link)
    time.sleep(2) # 休息2秒
    count += 1
    if count >= 10: # 全部作完會花太多時間，所以只示範捉10筆
        break
# 執行完for loop, 觀察 pages
print(pageSoups)
```

下載指定網站的圖檔 (需要python3)

```python
import requests,os
from bs4 import BeautifulSoup
from urllib.request import urlopen
url = 'http://www.tooopen.com/img/87.aspx'
html = requests.get(url)
html.encoding="utf-8"
sp = BeautifulSoup(html.text, 'html.parser')
# 建立 images 目錄儲存圖片
images_dir="images/"
if not os.path.exists(images_dir):
    os.mkdir(images_dir)
# 取得所有 <a> 和 <img> 標籤
all_links=sp.find_all(['a','img'])
for link in all_links:
    # 讀取 src 和  href 屬性內容
    src=link.get('src')
    href = link.get('href')
    attrs=[src,href]

    for attr in attrs:
        # 讀取  .jpg 和  .png 檔
        if attr != None and ('.jpg' in attr or '.png' in attr):
            # 設定圖檔完整路徑
            full_path = attr
            filename = full_path.split('/')[-1]  # 取得圖檔名
            ext = filename.split('.')[-1]  #取得副檔名
            filename = filename.split('.')[-2] #取得主檔名
            if 'jpg' in ext:
                filename = filename + '.jpg'
            else:
                filename = filename + '.png'
            print(attr)
            # 儲存圖片
            try:
                image = urlopen(full_path)
                f = open(os.path.join(images_dir,filename),'wb')
                f.write(image.read())
                f.close()
            except:
                print("{} 無法讀取!".format(filename))
```

http://jwlin.github.io/py-scraping-analysis-book/ch2/table/table.html

Pycone 松果城市課程列表

Python是非常強的的程式語言, 簡潔友好的語法特別容易上手, 又有許多第三方函式庫的支援。 Python是完全物件導向的語言, 有益於減少程式碼的重複性。Python的設計哲學是優雅, 明確, 簡單。 Python的設計風格, 使其成為易讀, 易維護且具有廣泛用途的程式語言。 Python的應用範圍相當廣泛, 例如web後端開發, 機器學習, 資料分析, 自然語言處理, 網頁爬蟲與遊戲等等。如果自己常常翻閱書籍卻無法掌握重點, 上網收集資料卻覺得太過片段, 想要自己動手寫寫看卻不知道如何開始。 這們課會從最基本的環境架設開始教起, 讓所有同學都可以深入淺出一窺Python的奧妙,更透過實務專題練習的方式,使學生可以應用課堂所學來完成一個Python軟體。

課程名稱	適合對象	售價	課程連結
初心者 - Python入門	初學者	1490	python
Python 網頁爬蟲入門實戰	有程式基礎的初學者	1890	python
Python 機器學習入門實戰 (預計)	有程式基礎的初學者	1890	python
Python 資料科學入門實戰 (預計)	有程式基礎的初學者	1890	python
Python 資料視覺化入門實戰 (預計)	有程式基礎的初學者	1890	python
Python 網站架設入門實戰 (預計)	有程式基礎的初學者	1890	python

```
<table class="table">
   <thead>
    <tr><th>課程名稱</th><th>適合對象</th><th>售價</th><th>課程連結</th></tr>
   </thead>
   <tbody>
    <tr><td>初心者 - Python入門</td><td>初學者</td><td>1490</td><td><a href="http://www.pycone.com"><img class="python-logo" src="img/python-logo.png" alt="python™"></a></td></tr>
    <tr><td>Python 網頁爬蟲入門實戰</td><td>有程式基礎的初學者</td><td>1890</td><td><a href="http://www.pycone.com"><img class="python-logo" src="img/python-logo.png" alt="python™"></a></td></tr>
    <tr><td>Python 機器學習入門實戰 (預計)</td><td>有程式基礎的初學者</td><td>1890</td><td><a href="http://www.pycone.com"><img class="python-logo" src="img/python-logo.png" alt="python™"></a></td></tr>
    <tr><td>Python 資料科學入門實戰 (預計)</td><td>有程式基礎的初學者</td><td>1890</td><td><a href="http://www.pycone.com"><img class="python-logo" src="img/python-logo.png" alt="python™"></a></td></tr>
    <tr><td>Python 資料視覺化入門實戰 (預計)</td><td>有程式基礎的初學者</td><td>1890</td><td><a href="http://www.pycone.com"><img class="python-logo" src="img/python-logo.png" alt="python™"></a></td></tr>
    <tr><td>Python 網站架設入門實戰 (預計)</td><td>有程式基礎的初學者</td><td>1890</td><td><a><img class="python-logo" src="img/python-logo.png" alt="python™"></a></td></tr>
   </tbody>
  </table>
</div>
```

```
req = requests.get('http://jwlin.github.io/py-scraping-analysis-book/ch2/table/table.html')
soup = BeautifulSoup(req.text,'html.parser')
prices = []
rows = soup.find('table','table').tbody.find_all('tr')
for row in rows:
        price = row.find_all('td')[2].text
        prices.append(int(price))
print(sum(prices)/len(prices))
```

parent, children, siblings

```
rows = soup.find('table','table').tbody.find_all('tr')
for row in rows:
        all_tds = row.find_all('td')
        #all_tds = [td for td in row.children]

        if 'href' in all_tds[3].a.attrs:
                href = all_tds[3].a['href']
        else:
                href = None
        print(all_tds[0].text, all_tds[1].text, all_tds[2].text, href, all_tds[3].a.img['src'])
```

Exercise:

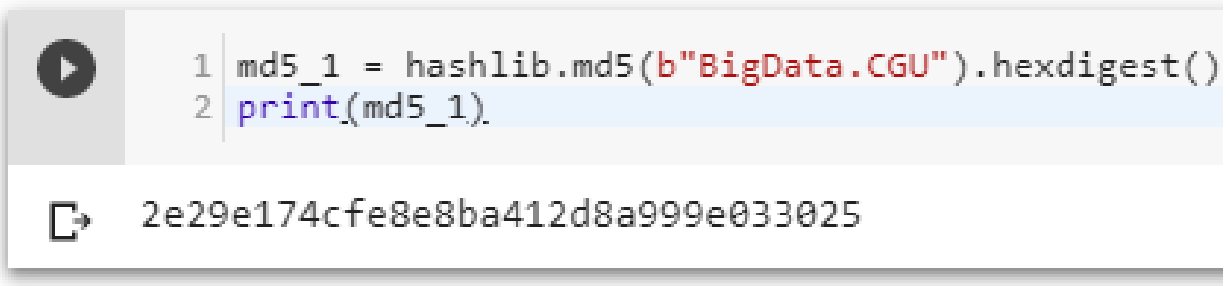
- 請以BeautifulSoup取得台灣彩券https://www.taiwanlottery.com.tw/

  當期威力彩開獎結果並將開出順序，大小順序，第二區分別擷取出來

網頁測試自動化

Hashlib

```
import hashlib
# 建立 MD5 物件
md5 = hashlib.md5()
# 要計算 MD5 雜湊值的資料
data = “BigData.CGU"
# 更新 MD5 雜湊值
md5.update(data.encode("big5"))
# 取得 MD5 雜湊值
h = md5.hexdigest()
print(h)
```



```
import hashlib,os,requests
url = "http://opendata.epa.gov.tw/ws/Data/REWXQA/?\
$orderby=SiteName&$skip=0&$top=1000&format=json"
# 讀取網頁原始碼
html=requests.get(url).text.encode('utf-8-sig')
# 判斷網頁是否更新
md5 = hashlib.md5(html).hexdigest()
if os.path.exists('old_md5.txt'):
    with open('old_md5.txt', 'r') as f:
        old_md5 = f.read()
    with open('old_md5.txt', 'w') as f:
        f.write(md5)
else:
    with open('old_md5.txt', 'w') as f:
        f.write(md5)
if md5 != old_md5:
    print('資料已更新...')
else:
    print('資料未更新，從資料庫讀取...')
```

網頁爬蟲

網路爬蟲 = 自動化 + 捉取網頁 + 分析內容 + 存取

For loop
if… else…

Requests
BeautifulSoup

- 土法煉鋼
- API ex: https://pm25.lass-net.org/

https://opendata.epa.gov.tw/

Homework

1. 請試著利用beautifulsoup完成作業1

2. https://www.chinatimes.com/?chdtv 抓取

    a. 熱搜關鍵字

    b. 政治焦點

    c. 生活焦點

    d. 娛樂焦點

    加分: 圖片