網頁資料擷取

擷取網頁資料的前題是不能觸犯著作權:

- https://www.tipo.gov.tw/ct.asp?xItem=219598&ctNode=7561&mp=1 (https://www.tipo.gov.tw/ct.asp?xItem=219598&ctNode=7561&mp=1)
- https://udn.com/news/story/6871/3221682 (https://udn.com/news/story/6871/3221682)

這個單元例子僅用來說明由網頁擷取資料的入門技巧。

- 使用re
- 使用BeautifulSoup
- 使用Selenium

解析網址

單純網址

In [ ]:

```
# coding=utf-8
from urllib.parse import urlparse
url = 'https://tw.stock.yahoo.com/news_list/url/d/e/'
up  = urlparse(url)
print(up)
```

In [ ]:

```
# coding=utf-8
from urllib.parse import urlparse
url = 'https://www.cwb.gov.tw/V7/forecast/index.htm'
up  = urlparse(url)
print(up)
```

有**get**參數

In [ ]:

```
# coding=utf-8
from urllib.parse import urlparse
url = 'https://tw.stock.yahoo.com/q/q?s=2330'
up  = urlparse(url)
print(up)
```

In [ ]:

```
# coding=utf-8
from urllib.parse import urlparse
url = 'https://ecshweb.pchome.com.tw/search/v3.3/?q=pc&scope=all'
up  = urlparse(url)
print(up.query.split('&'))
```

- scheme: 通訊協定
- netloc: 網域名稱
- path: 網頁所在路徑與檔名
- query；GET參數

透過**requests.get(url)**擷取網頁的內容

```
import requests
r = requests.get(url)
```

註: 萬一被阻擋，可以嘗試設定user agent

```
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/71.0.3578.80 Safari/537.36',
    'Content-Type': 'application/x-www-form-urlencoded',
    'Connection' : 'Keep-Alive',
    'Accept':'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
    'Accept-Encoding':'gzip, deflate, sdch',
    'Accept-Languate': 'en-US,en;q=0.8'}

r = requests.get(url,headers=headers)
```

你可以從chrome瀏覽器找到各式user agent字串：按Ctrl-Shift-I啟動開發者工具，在Network選單，按右上選項圖示後選More tools>Network conditions。

在**URL**裡傳遞參數

In [ ]:

```
import requests
keys = {'search_query': 'python'}
r=requests.get('https://www.youtube.com/results',params=keys)
print(r.url)
```

擷取網頁的內容

方式一**: r.content (**它是**bytes**型態**)**

可透過下面方式將bytes轉str

- 轉換方法一: r.content.decode('utf-8')
- 轉換方法二: str(r.content,encoding='utf-8')

str轉bytes

- 轉換方法一: s.encode('utf-8')
- 轉換方法二: bytes(s, encoding='utf-8')

In [ ]:

```
import requests
url = 'https://www.cwb.gov.tw/V7/forecast/index.htm'
r = requests.get(url)
for i in r.content.decode('utf-8').splitlines()[:10]:
    print(i)
```

方式二: **r.text (**它是**str**型態**)**

如下面所示，requests編碼方式為ISO-8859-1。

In [ ]:

```
import requests
url = 'https://www.cwb.gov.tw/V7/forecast/index.htm'
#html = requests.get(url).content.decode('utf-8','ignore').splitlines()
r = requests.get(url)
print(r.encoding)
```

用r.encoding='utf-8'將編碼改為'utf-8'編碼後，處理網頁內容。

In [ ]:

```
import requests
url = 'https://www.cwb.gov.tw/V7/forecast/index.htm'
r = requests.get(url)
r.encoding = 'utf-8'
for i in r.text.splitlines()[:10]:
    print(i)
```

使用**re**模組擷取資訊

HTML網頁格式

HTML標記	用途說明
<html>...</html>	標記...為HTML文件
<head>...</head>	標記...為HTML文件標頭
<title>...</title>	標記...為HTML文件標題，通常會顯示在瀏覽器標題列
<body>...</body>	標記...為HTML文件內容
<script>...</script>	標記...為描述語言
...	標記...為標題(等級為h1,...,h6)
...	標記...為文字段落
...	排版用格式標記，...通常為內文大段落或顯示分塊
...	類似 ，通常用在小段落

HTML標記	用途說明
<table>...</table>	標記...為表格呈現內容
<img src='...'>	顯示圖形檔設定
<a href='...'>	外部連結設定

例子:使用**re**擷取網頁裡的新聞標題

https://udn.com/news/cate/2/7226 (https://udn.com/news/cate/2/7226) 網頁裡新聞標題寫在<h2> ... </h2>段落，如下範例

```
 <h2 style="width:100%">專利戰高通告贏蘋果 陸將禁售iPhone X以前機種 <time>22:29</time
></h2>
 <h2>先從海外版施行 日本漫畫週刊少年Jump也走向數位訂閱制<span class="i-video1"></span>
</h2>
```

In [ ]:

```
import requests
import re
url = 'https://udn.com/news/cate/2/7226'
html = requests.get(url).content.decode('utf-8')
for idx,title in enumerate(re.finditer(r'<h2[^>]*?>([^<]*?)(<time.+?/time>)?(<span.+?/s
    print('{:4d}. {}'.format(idx+1,title.group(1)))
```

使用**BeautifulSoup**模組擷取資訊

https://www.crummy.com/software/BeautifulSoup/bs4/doc/
(https://www.crummy.com/software/BeautifulSoup/bs4/doc/)

安裝

```
conda install -c anaconda beautifulsoup4
```

找到所有<h2> ... </h2>段落程式片段

```
from bs4 import BeautifulSoup
import requests
url = 'https://udn.com/news/cate/2/7226'
html = requests.get(url).content.decode('utf-8')
sp = BeautifulSoup(html,'html.parser')
for link in sp.find_all('h2'):
    print('{}'.format(link.text))
```

指令範例	說明
sp.find('a'[,key])	傳回第一個符合的內容
sp.find_all('a'[,key])	傳回所有符合的內容
sp.title/sp.title.text	傳回<title>**網頁標題**</title>

指令範例	說明
sp.text	傳回去掉HTML標籤的內容

例子:使用**BeautifulSoup**擷取網頁裡的新聞標題

In [ ]:

```
from bs4 import BeautifulSoup
import requests
url = 'https://udn.com/news/cate/2/7226'
html = requests.get(url).content.decode('utf-8')
sp = BeautifulSoup(html,'html.parser')

for idx,link in enumerate(sp.find_all('h2')):
    print('{:4d}. {}'.format(idx+1,link.text))
```

例子：擷取表格資料

中央氣象局月平均氣溫表格HTML原始碼:

https://www.cwb.gov.tw/V7/climate/monthlyMean/Taiwan_tx.htm
(https://www.cwb.gov.tw/V7/climate/monthlyMean/Taiwan_tx.htm)

```
  <table width="780" cellpadding="2" cellspacing="1" class="Form00" summary="排版用
表格">
      <tbody><tr height="44">
        <th width="150" height="44" class="tab01">地名</th>
        <th width="50" height="44" class="tab01" axis="month">一月</th>
        <th width="50" height="44" class="tab01" axis="month">二月</th>
        <th width="50" height="44" class="tab01" axis="month">三月</th>
        <th width="50" height="44" class="tab01" axis="month">四月</th>
        <th width="50" height="44" class="tab01" axis="month">五月</th>
        <th width="50" height="44" class="tab01" axis="month">六月</th>
        <th width="50" height="44" class="tab01" axis="month">七月</th>
        <th width="50" height="44" class="tab01" axis="month">八月</th>
        <th width="50" height="44" class="tab01" axis="month">九月</th>
        <th width="50" height="44" class="tab01" axis="month">十月</th>
        <th width="50" height="44" class="tab01" axis="month">十一月</th>
        <th width="50" height="44" class="tab01" axis="month">十二月</th>
        <th width="50" height="44" class="tab01">平均</th>
        <th width="100" height="44" class="tab01">統計期間</th>
      </tr>
      <tr height="44">
        <td height="44" class="active" axis="item">淡水</td>
        <td class="whitetd" width="36" height="44">15.2</td>
        <td class="whitetd" width="36" height="44">15.6</td>
        <td class="whitetd" width="36" height="44">17.4</td>
        <td class="whitetd" width="36" height="44">21.1</td>
        <td class="whitetd" width="36" height="44">24.5</td>
        <td class="whitetd" width="36" height="44">26.9</td>
        <td class="whitetd" width="36" height="44">28.8</td>
        <td class="whitetd" width="36" height="44">28.6</td>
        <td class="whitetd" width="36" height="44">26.7</td>
        <td class="whitetd" width="36" height="44">23.7</td>
        <td class="whitetd" width="36" height="44">20.6</td>
        <td class="whitetd" width="36" height="44">16.9</td>
        <td class="whitetd" width="36" height="44">22.2</td>
        <td class="whitetd" width="90" height="25">1981-2010</td>
       </tr>
       ...
       </table>
```

平均氣溫表格放在<table></table>標記內。可是那個網頁有許多<table>標記，不過其中class為Form00為平均氣溫表格。因此table = sp.find_all('table',{'class':'Form00'})可以鎖定以'class'為key, value為'Form00'的那個<table>標記。

```
url = 'https://www.cwb.gov.tw/V7/climate/monthlyMean/Taiwan_tx.htm'
html = requests.get(url).content.decode('utf-8')
sp = BeautifulSoup(html,'html.parser')
table = sp.find('table',{'class':'Form00'})
```

表格裡每一列以<tr>，</tr>標記標示。下面指令找出<table></table>標記內每一列。

```
rows = table.find_all('tr')
```

第一列，為標題列，每一欄以<th>，</th>標記標示。下面指令找出<tr>，</tr>標記內每一欄。

```
title = [c.text for c in rows[0].find_all('th')]
```

其他列每一欄以<td>，</td>標記標示。下面指令找出<tr>，</tr>標記內每一欄，並存放在各自list。

```
data  = [list() for _ in range(len(title))]

for r in rows[1:]:
    for col,cell_data in zip(data,r.find_all('td')):
        try:
            col.append(float(cell_data.text))
        except ValueError:
            col.append(cell_data.text)
```

In [ ]:

```
from bs4 import BeautifulSoup
import requests
import numpy as np
import matplotlib.pyplot as plt

url = 'https://www.cwb.gov.tw/V7/climate/monthlyMean/Taiwan_tx.htm'
html = requests.get(url).content.decode('utf-8')
sp = BeautifulSoup(html,'html.parser')
table = sp.find('table',{'class':'Form00'})
rows = table.find_all('tr')

title = [c.text for c in rows[0].find_all('th')]
data  = [list() for _ in range(len(title))]

for r in rows[1:]:
    for col,cell_data in zip(data,r.find_all('td')):
        try:
            col.append(float(cell_data.text))
        except ValueError:
            col.append(cell_data.text)

#放入 numpy.ndarray

data_table= np.core.records.fromarrays(data)
data_table.dtype.names = title

#資料標題
print(data_table.dtype.names)

#取得第0列資料
print(data_table[0])

#取得各觀測站五月均溫
print(data_table['五月'])

```

使用matplotlib繪出平均氣溫圖。

In [ ]:

```
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties
import os

font = FontProperties(fname=os.environ['WINDIR']+'\\Fonts\\kaiu.ttf', size=12)

plt.figure(figsize=(8,4))
for i in range(5):
    r = list(data_table[i])
    plt.plot(np.arange(len(data_table.dtype.names)-3),r[1:-2],label=r[0])
plt.legend(prop=font)
plt.title('平均氣溫',fontproperties=font)
plt.xticks(np.arange(len(title)-3),title[1:-2],fontproperties=font)
plt.ylabel('攝氏',fontproperties=font)
plt.show()
```

例子：擷取所有連結

HTML連結格式為:

```
<a href='https://tw.yahoo.com/?p=us'>本文</a>
```

所以

```
all_links = sp.find_all('a')
```

得到所有<a ....>....段落。假設link為以上連結為例子，

- link.get('href')得到'https://tw.yahoo.com/?p=us (https://tw.yahoo.com/?p=us)'
- link.text得到'本文'

In [ ]:

```
from bs4 import BeautifulSoup
import requests
url = 'https://udn.com/news/index'
html = requests.get(url).content.decode('utf-8')
sp = BeautifulSoup(html,'html.parser')

for idx,link in enumerate(sp.find_all('a')):
    href = link.get('href')
    if href is not None and href.startswith('http'):
        print('{:4d} text:{:<s}, link:{:>s}'.format(idx+1,link.text,href))
```

例子：擷取所有圖形檔

下面例子需要用到pillow模組

```
conda install -c anaconda pillow
```

In [ ]:

```
from bs4 import BeautifulSoup
import requests
from urllib.parse import urlparse
from urllib.request import urlopen
import matplotlib.pyplot as plt
from PIL import Image, ImageDraw, ImageFont
import numpy as np
import io
import re

url = 'https://udn.com/news/story/7934/3526132'

headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHT
           'Content-Type': 'application/x-www-form-urlencoded',
        'Connection' : 'Keep-Alive',
        'Accept':'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
        'Accept-Encoding':'gzip, deflate, sdch',
        'Accept-Languate': 'en-US,en;q=0.8'}

uc  = urlparse(url)
print(uc.scheme,uc.hostname)
domain = '{}://{}'.format(uc.scheme,uc.hostname)
print(domain)

html = requests.get(url,headers=headers).content.decode('utf-8')

sp = BeautifulSoup(html,'html.parser')
for idx,link in enumerate(sp.find_all(['a','img'])):
    href = link.get('href')
    src  = link.get('src')
    for t in [href, src]:
        if t is not None and ('.jpg' in t or '.png' in t):
            if t.startswith('http'):
                img_path = t
            elif t.startswith('//'):
                img_path = 'https:'+t
            else:
                domain+t
            print(img_path)
            print('filename:{}'.format(re.search('[^/]+((.jpg)|(.png))',img_path).grou
            image = urlopen(img_path)
            img = Image.open(image)
            plt.imshow(img)
            plt.axis('off')
            plt.show()

```

透過Selenium擷取網頁資料

安裝

```
conda install -c conda-forge selenium
```

安裝各瀏覽器的webdriver載點https://www.seleniumhq.org/about/platforms.jsp (https://www.seleniumhq.org/about/platforms.jsp)

- 如Chrome web driver:https://sites.google.com/a/chromium.org/chromedriver/home (https://sites.google.com/a/chromium.org/chromedriver/home)

並將webdriver(如chromedriver.exe)放在Python執行的目錄內。測試下面範例：

操作瀏覽器函式

webdriver方法	說明
refresh()	重新整理頁面
back()	回上一頁
forward()	到下一頁
close()	關視窗
quit()	結束瀏覽器
get(url)	瀏覽url這網址
current_url	目前網址
title	網頁標題
page_source	網頁原始碼
save_screenshot(pngfile)	存目前網頁畫面於png檔
get_window_position()	取得視窗左上角位置
set_window_position(x,y)	設定視窗左上角位置
maximize_window()	最大化視窗
get_window_size()	取得視窗大小
set_window_size(x,y)	設定視窗大小

In [ ]:

```
from selenium import webdriver
urls = ['https://www.cwb.gov.tw/V7/','https://tw.yahoo.com/?p=us']
web = webdriver.Chrome()
for idx,url in enumerate(urls):
    web.get(url)
    web.save_screenshot('screenshot_{}.png'.format(idx))
web.quit()
```

In [ ]:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
fig = plt.figure(figsize=(15,30))
for idx in range(len(urls)):
    img = mpimg.imread('screenshot_{}.png'.format(idx))
    plt.subplot(1,2,idx+1)
    plt.imshow(img)
    plt.axis('off')
plt.show()
```

網頁元素檢索功能

基本步驟範例

```
web = webdriver.Chrome()
web.get(url)

# 檢索網頁元素，通常元素id為唯一，比較好找
element = web.find_element_by_id(id)

# 操作網頁元素
element.send_keys(value)
element.submit() # 提交
```

webdriver元素檢索方法	說明
find_element_by_X(value)	使用X檢索，取得第一個符合的元素
---	---
find_element_by_class_name(name)	使用類別名稱檢索
find_element_by_css_selector(selector)	使用CSS選擇器檢索
find_element_by_id(id)	使用id檢索
find_element_by_link_text(text)	使用連結文字檢索
find_element_by_name(name)	使用名稱檢索
find_element_by_tag_name(name)	使用HTML標籤檢索
---	---
find_elements_by_X	使用X檢索，取得所有符合的元素

webdriver元素操作方法	說明
clear()	清除內容
click()	點擊，通常用於按鈕、連結、選單
send_keys(value)	對此元素送出字串
submit()	提交
is_displayed()	此元素是否可見
is_enabled()	此元素是否可用
is_selected()	此元素是否被選定

In [ ]:

```
from selenium import webdriver

web = webdriver.Chrome()
web.maximize_window()
web.get("https://www.google.com")

#找到輸入框
element = web.find_element_by_name("q")

#輸入
element.send_keys("中央氣象局")

#提交
element.submit()

#web.close()
```

In [ ]:

```
from selenium import webdriver
import time

web = webdriver.Chrome()
web.maximize_window()
web.get("https://www.youtube.com")

#找到輸入框
element = web.find_element_by_id("search")

#輸入
element.send_keys("selenium Python")

#按搜尋
search_btn = web.find_element_by_id("search-icon-legacy")
search_btn.click()
# Get scroll height
last_height = -1
for idx in range(500):
    # Scroll down to bottom
    web.execute_script("window.scrollTo(0, window.scrollY + 800);")
    # Wait to load page
    time.sleep(.5)
    current_height = web.execute_script("return window.scrollY")
    if last_height == current_height:
        print('stop')
        break
    last_height = current_height

```

In [ ]:

```
print(web.page_source.splitlines()[:5])
```

In [ ]:

```

```