資料科學大數據網路爬蟲應用

指導老師：蘇有 老師

作者：鄭羽晴 A10423014@gmail.com

查詢表格：資料很少會更動

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965/1/9	731 Fondren,Houston,TX	M	30000.0	333445555	5
Franklin	T	Wong	333445555	1955/12/8	638 Voss, Houston,TX	M	40000.0	888665555	5
Joyce	A	English	453453453	1972/7/31	5631 Rice,Houston,TX	F	25000.0	333445555	5
Ramesh	K	Narayan	666884444	1962/9/15	975 Fire Oak,Humble,TX	M	38000.0	333445555	5
James	E	Borg	888665555	1973/11/10	450 Stone,Houston,TX	M	55000.0	NULL	1
Jennifer	S	Wallace	987654321	1941/6/20	291 Berry,Bellaire,TX	F	43000.0	888665555	4
Ahmad	V	Jabbar	987987987	1969/3/29	980 Dallas,Houston,TX	M	25000.0	987654321	4
Alicia	J	Zelaya	999887777	1968/1/19	3321 Castle,Spring,TX	F	25000.0	987654321	4

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Headquarters	1	888665555	1988/6/19
Administration	4	987654321	1995/1/1
Research	5	333445555	1988/5/22

事實表格：資料會常常異動

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
333445555	10	10.0
333445555	2	10.0
333445555	20	10.0
333445555	3	10.0
453453453	1	20.0
453453453	2	20.0
666884444	3	40.0
888665555	20	Null
987654321	20	15.0
987654321	30	20.0
987987987	10	35.0
987987987	30	5.0
999887777	10	10.0
999887777	30	30.0

SQL指令

SELECT 選擇欄位

FROM 從哪一個表格

WHERE 篩選條件

pyodbc模組

conn = pyodbc.onnect('Driver=資料庫驅動程式; DBQ=資料庫檔案路徑)：可使用pyodbc.drivers()查看驅動程式，回傳pyodbc.Connection物件

```
(r'Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=D:\Python\DB and Web crawler\company.accdb')
```

cur = conn.cursor()：建立指標(才可以瀏覽資料庫)，回傳pyodbc.Cursor物件

cur.excute('SQL指令')：執行指定的SQL指令

cur.fetchone()：回傳指標位置後的第一筆SQL指令執行結果

```
('John', 'Smith', '123456789', 'Research')
```

cur.fetchall()：回傳指標位置後的所有SQL執行結果

```
[('John', 'Smith', '123456789', 'Research'), ('Franklin', 'Wong', '333445555', 'Research'), ('Ramesh', 'Narayan', '66
```

cur.description：回傳SQL指令執行後的Table結構(欄位名稱&欄位儲存的資料型態)

```
(('名', <class 'str'>, None, 10, 10, 0, True), ('姓', <class 'str'>, None, 10, 10, 0, True), ('身分證號
```

pyodbc模組

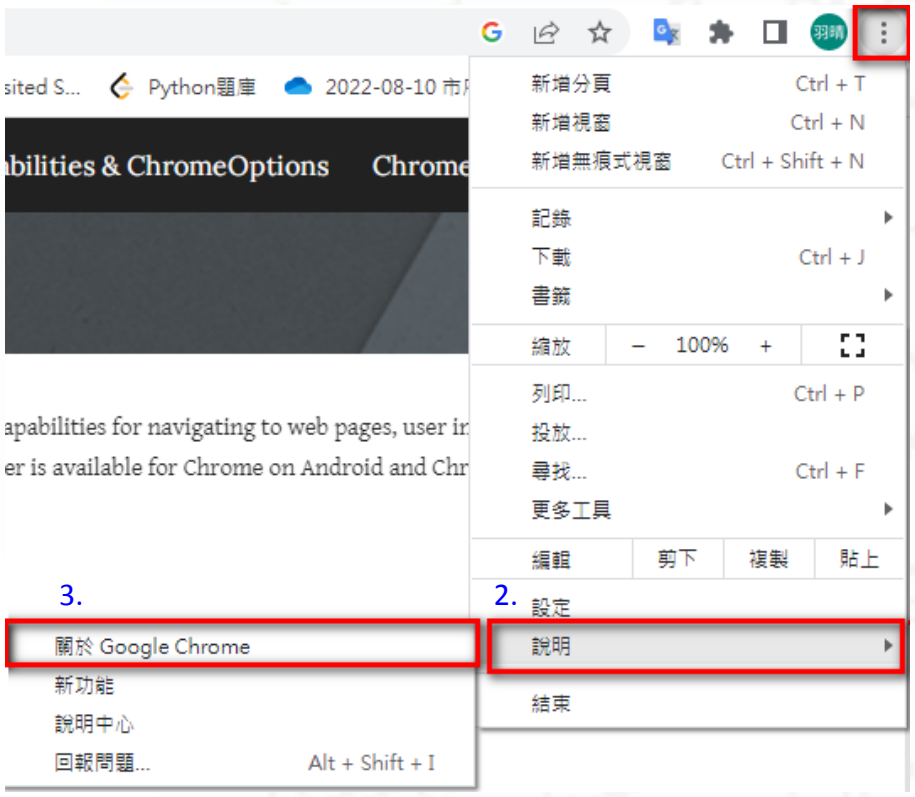
```
import pyodbc
import pandas as pd
import numpy as np


conn = pyodbc.connect(r'Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=D:\Python\DB and Web crawler\教材\資料庫與Python\CH09C2.accdb')
db_l = ['系所','教師','課程','學生','選課單']


cur = conn.cursor()#指標

writer = pd.ExcelWriter(r'D:\Python\Program\DB and Web crawler\20221102\homework.xlsx')#設置excel寫入路徑
for item in db_l:
    SQL = 'SELECT * FROM '
    SQL += item
    cur.execute(SQL)#執行SQL指令
    list1 = cur.fetchall()#SQL指令取得的資料，存為串列
    desc = cur.description#取得資料庫的結構 欄位&欄位儲存的資料型態
    df_column = pd.DataFrame(np.array(desc))#將cur.dexcription的SQL的欄位存為df，以利後續使用
    df = pd.DataFrame(np.array(list1), columns=df_column[0])#將SQL執行結果設為df的data，欄位名稱為先前儲存的欄位名稱
    df.to_excel(writer, sheet_name=item)#將完成的df寫入到Excel，並指定sheet名稱為資料庫名稱

cur.close()
conn.close()
writer.save()
```

	代碼	學院	學系
0	E01	理學院	數學系
1	E02	理學院	物理系
2	E03	理學院	地球科學系
3	E04	理學院	化學系
4	E05	理學院	光電系
5	H01	管理學院	經營管理學系
6	H02	管理學院	國際企業學系
7	H03	管理學院	統計系
8	H04	管理學院	財經管理學系
9	M01	資訊學院	資訊工程系
10	M02	資訊學院	資訊多媒體應用學系
11	M03	資訊學院	光電與通訊學系
12	M04	資訊學院	資訊傳播學系

系所 | 教師 | 課程 | 學生 | 選課單

selenium模組

Chromedriver Download #https://chromedriver.chromium.org/ 下載對應版本的chromedriver



4.確認Chrome版本

5.下載對應版本的Chromedriver

6.透過selenium開啟的網頁上面會寫Chrome目前受到自動測試軟體控制

selenium模組

WebDriver

find_element(by='id', value=None)：回傳符合條件的第一個WebElement物件

find_elements(by='id', value=None)：回傳符合條件的list of WebElement物件

Locating Elements的方法(8種)：https://selenium-python.readthedocs.io/locating-elements.html

```
find_element(By.ID, "id")
find_element(By.NAME, "name")
find_element(By.XPATH, "xpath")
find_element(By.LINK_TEXT, "link text")
find_element(By.PARTIAL_LINK_TEXT, "partial link text")
find_element(By.TAG_NAME, "tag name")
find_element(By.CLASS_NAME, "class name")
find_element(By.CSS_SELECTOR, "css selector")
```

```
<html>
 <body>
  <form id="loginForm">
   <input name="username" type="text" />
   <input name="password" type="password" />
   <input name="continue" type="submit" value="Login" />
  </form>
 </body>
</html>
```

#By.ID
login_form = driver.find_element(By.ID, 'loginForm')

#By.NAME
username = driver.find_element(By.NAME, 'username')
password = driver.find_element(By.NAME, 'password')

#By.XPATH最常用，XPATH路徑[1]就是第一個，跟index從0開始不一樣
username = driver.find_element(By.XPATH, "/html/body/form[1]/input[1]")
password = driver.find_element(By.XPATH, "/html/body/form[1]/input[2]")

```
<html>
 <body>
  <p>Are you sure you want to do this?</p>
  <a href="continue.html">Continue</a>
  <a href="cancel.html">Cancel</a>
</body>
</html>
```

#By.LINK_TEXT
continue_link = driver.find_element(By.LINK_TEXT, 'Continue')

#By.PARTIAL_LINK
continue_link = driver.find_element(By.PARTIAL_LINK_TEXT, 'Conti')

```
<html>
 <body>
  <h1>Welcome</h1>
  <p>Site content goes here.</p>
</body>
</html>
```

#By.TAG_NAME
heading1 = driver.find_element(By.TAG_NAME, 'h1')

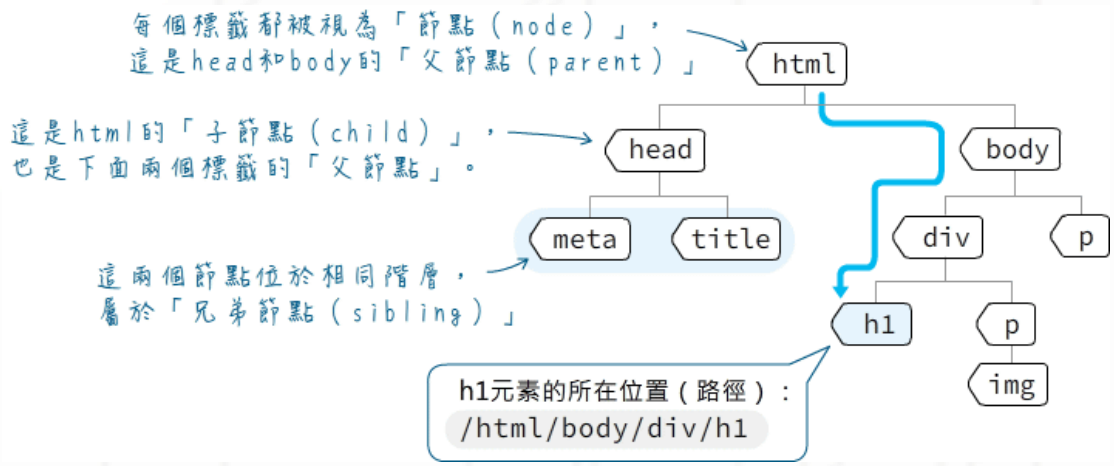
```
<html>
 <body>
  <p class="content">Site content goes here.</p>
</body>
</html>
```

#By.CLASS_NAME
content = driver.find_element(By.CLASS_NAME, 'content')

#ByCSS_SELECTOR
content = driver.find_element(By.CSS_SELECTOR, 'p.content')

selenium模組

XPATH絕對路徑與相對路徑




```
<html>
 <head>
  <meta charset="utf-8">
  <title>手作XDIY</title>
 </head>
 <body>
  <div id="main" style="width:650px">
    <h1>自製麥克風防風罩</h1>
    <p>
     <img src="img/mic_boom_12.jpg">
     Full HD高畫質攝影機可以捕捉到細膩的影像...
    </p>
  </div>
  <p>版權聲明文字</p>
 </body>
</html>
```

總是從html元素起頭　　用單一斜線分隔階層

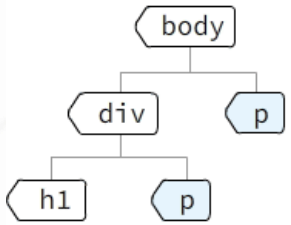
```
/html/body/div/h1
```

絕對路徑

總是用雙斜線起頭

```
//h1
```

相對路徑



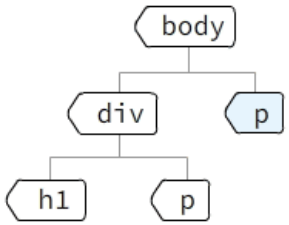

```
//p
```

選取任意路徑的p元素（共兩個）

或

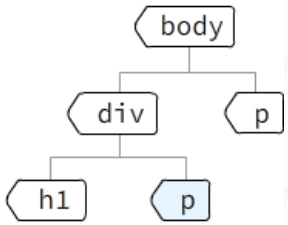
```
//body//p
```

選到body之下的所有p元素，
相對路徑可以再包含相對路徑。




```
//body/p
```

選到緊鄰在body之下的p元素




```
//div/p
```

或

```
//body/div/p
```

或

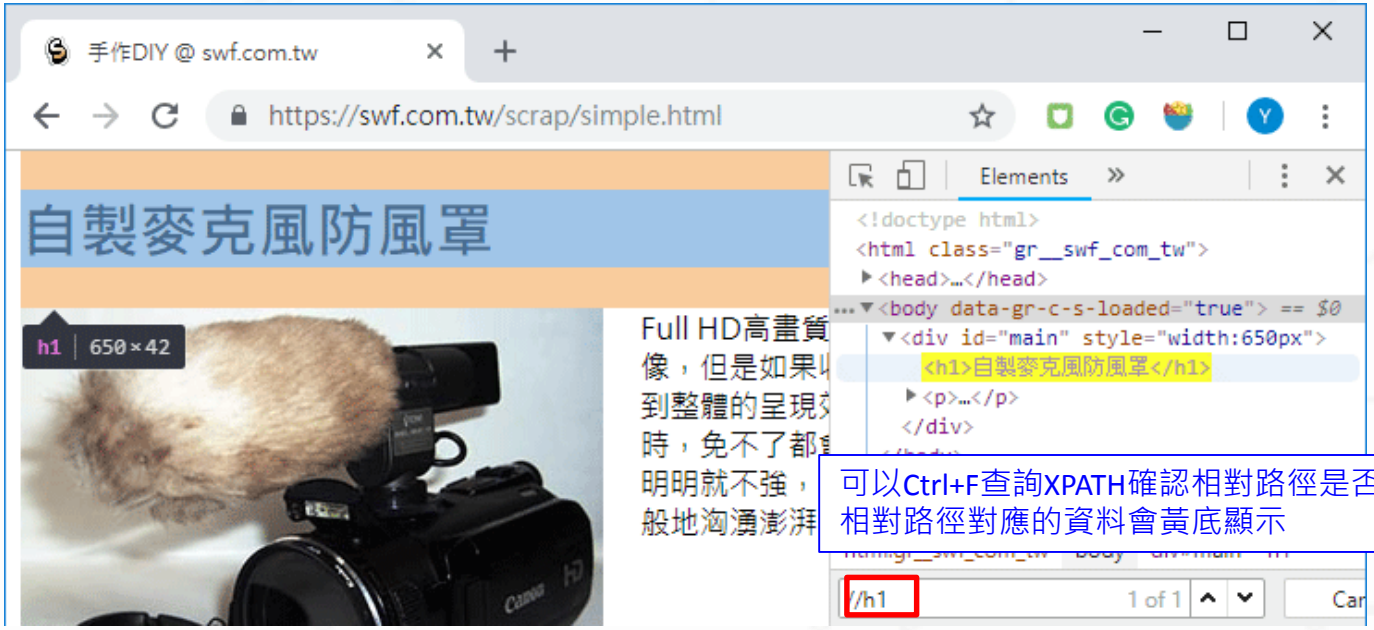
```
//body/*/p
```

星號代表任意元素

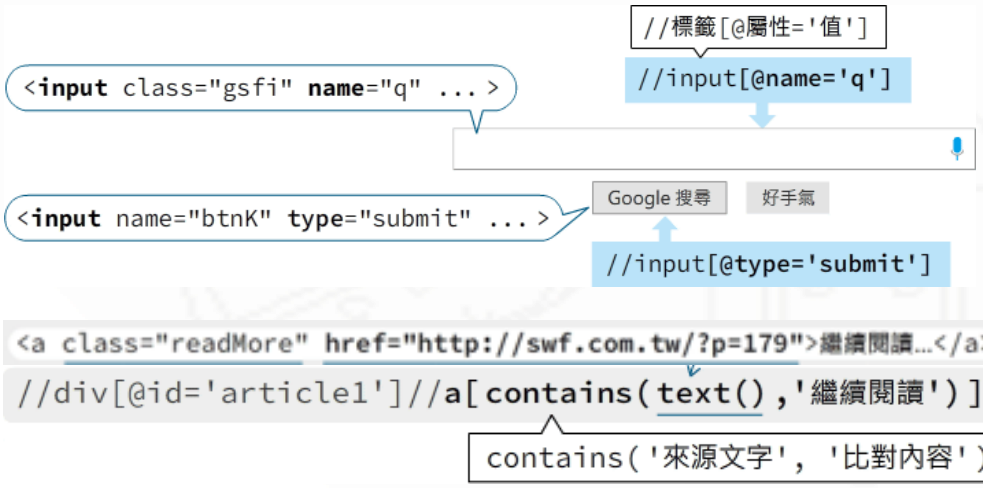
'/'絕對路徑
'//'相對路徑

selenium模組

XPATH



可以Ctrl+F查詢XPATH確認相對路徑是否寫的正確
相對路徑對應的資料會黃底顯示



透過標籤的屬性來選取

透過內文來定位XPATH

selenium模組

```
#開啟天瓏網路書店並搜尋'selenium'相關書籍
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
s = Service(r"C:\chromedriver.exe")
driver = webdriver.Chrome( service = s )

url = 'https://www.tenlong.com.tw/'
driver.get(url)

str = '/html/body/div[3]/nav[1]/nav/div/form/input[2]' #Copy full Xpath取得
search_field = driver.find_element( By.XPATH, str ) #使用By.XPATH方式
search_field.send_keys('selenium') #輸入selenium到對應位置
search_field.submit() #提交
```

selenium模組

#iframe網頁爬蟲 switch to
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
s = Service(r"C:\chromedriver.exe")
driver = webdriver.Chrome(service = s)

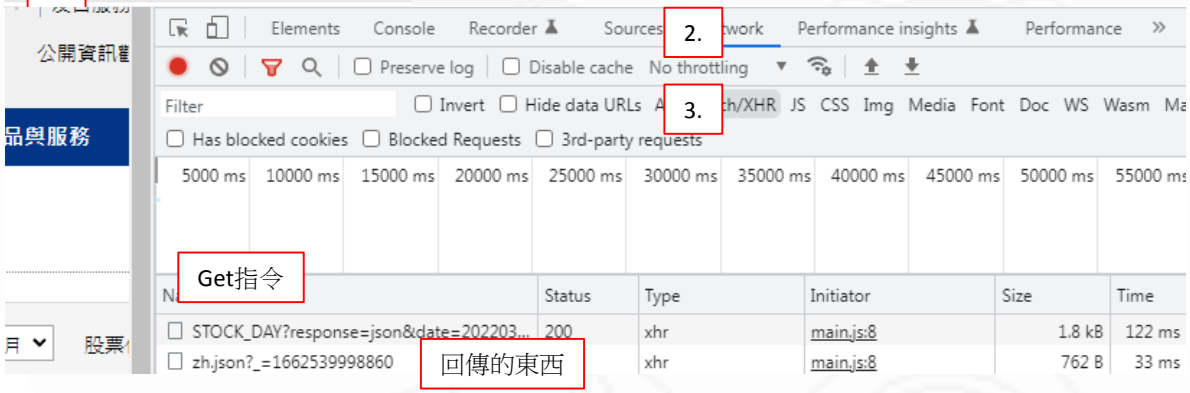
url = 'https://www.w3schools.com/html/tryit.asp?filename=tryhtml_id_css'
driver.get(url)

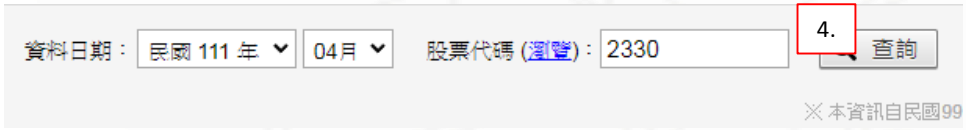
driver.switch_to.frame('iframeResult') #先使用switch_to切換到對應的iframe區，後續可以依照原本的find_emelent方法尋找資料

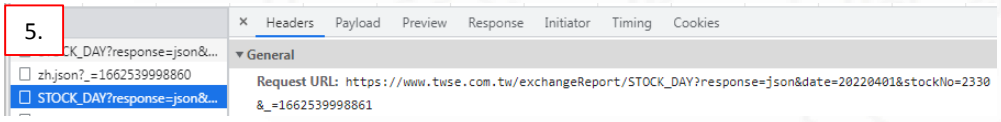
element1 = driver.find_element(By.ID, 'myHeader')
print(element1.tag_name) #h1 元素的標籤
Print(element1.text) #My Header 元素的內容

網頁中還有小網頁(有多個畫面，程式碼也還有別組<html><body>標籤

Request模組







1.右鍵檢查
2.點選Network
3.查看Fetch/XHR
4.再重新用網頁查詢一次
5.得到查詢的結果
6.找到關鍵request指令並複製

https://www.twse.com.tw/exchangeReport/STOCK_DAY?response=json&date=20220401&stockNo=2330&_=1662539998861
會變動的只有=後方➔後續只要修改藍字就可以得到別的資料

Request模組

```
import requests
url = 'https://www.twse.com.tw/exchangeReport/STOCK_DAY?response=json&date=20221001&stockNo=2603'
response = requests.get(url)
# 檢查HTTP回應碼是否為200(requests.code.ok)
if response.status_code == response.codes.ok:
   print(response.text) #這個股票網頁會取得json資料格式，後續可以用.json()方法解析
```


```
{"stat":"OK",
 "date":"20221001",
 "title":"111年10月 2603 長榮                各日成交資訊",
 "fields":["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"],
 "data":[["111/10/03","68,807,955","10,333,283,314","146.00","154.00","145.50","149.50","+3.50","31,306"],
         ["111/10/04","50,430,274","7,639,565,212","155.00","156.00","149.00","150.50","+1.00","21,898"],
         ["111/10/05","55,645,069","8,636,232,429","152.00","158.50","151.50","155.50","+5.00","29,515"],
         ["111/10/06","47,863,098","7,308,536,348","154.50","156.00","150.00","155.50"," 0.00","22,521"],
         ["111/10/07","28,713,476","4,465,127,604","153.50","157.50","152.50","155.50"," 0.00","14,516"],
         ["111/10/11","44,325,705","6,878,400,962","153.50","158.00","153.00","156.00","+0.50","20,200"],
         ["111/10/12","58,867,494","8,796,050,717","155.50","156.00","147.00","147.00","-9.00","28,819"],
         ["111/10/13","34,347,020","5,028,144,367","149.00","150.00","143.50","144.50","-2.50","18,022"],
         ["111/10/14","52,345,707","7,904,414,707","149.50","154.00","146.50","152.50","+8.00","25,152"],
         ["111/10/17","79,770,023","11,174,396,498","149.00","149.00","137.50","140.00","-12.50","42,154"],
         ["111/10/18","49,463,773","7,047,042,854","142.00","145.00","140.50","141.00","+1.00","24,279"],
         ["111/10/19","47,102,292","6,789,975,935","142.50","146.50","141.50","144.00","+3.00","24,806"],
         ["111/10/20","63,034,550","8,511,501,386","139.00","140.00","132.50","137.00","-7.00","29,730"],
         ["111/10/21","33,630,401","4,605,008,671","136.00","138.50","135.50","135.50","-1.50","15,625"],
         ["111/10/24","43,291,729","6,069,725,416","140.50","143.50","137.00","137.50","+2.00","17,749"],
         ["111/10/25","30,602,847","4,245,526,933","138.50","140.50","136.00","139.00","+1.50","14,401"],
         ["111/10/26","47,602,583","6,493,857,123","138.00","139.50","133.50","139.00"," 0.00","21,033"],
         ["111/10/27","31,866,416","4,467,994,615","140.50","142.50","137.00","142.50","+3.50","16,561"],
         ["111/10/28","30,536,385","4,236,431,352","140.00","141.50","136.00","137.50","-5.00","14,092"],
         ["111/10/31","42,218,018","5,709,837,116","135.50","137.50","132.00","137.50"," 0.00","19,307"]],
"notes":["符號說明:+/-/X表示漲/跌/不比價",
         "當日統計資訊含一般、零股、盤後定價、鉅額交易，不含拍賣、標購。",
         "ETF證券代號第六碼為K、M、S、C者，表示該ETF以外幣交易。",
         "權證證券代號可重複使用，權證顯示之名稱係目前存續權證之簡稱。"]}
```

BeautifulSoup模組(解析HTML)

定位方式

```
#By tag name
import requests
from bs4 import BeautifulSoup
url = 'http://ehappy.tw/bsdemo1.htm'
html = requests.get(url)
html.encoding = 'UTF-8'
sp = BeautifulSoup(html.text, 'lxml')
#BeautifulSoup物件
print(sp.title)
print(sp.title.text)
print(sp.h1)
print(sp.p)
```

```
<title>我是網頁標題</title>
我是網頁標題
<h1 class="large">我是標題</h1>
<p>我是段落</p>
```

```
#By tag name + 屬性
from bs4 import BeautifulSoup
sp = BeautifulSoup(html, 'html.parser')
print(sp.find('p'))
print(sp.find_all('p'))
print(sp.find('p', {'id':'p2', 'class':'red'}))
print(sp.find('p', id='p2', class_= 'red'))
```

```
#By CSS， #代表ID，.代表class
from bs4 import BeautifulSoup
sp = BeautifulSoup(html, 'lxml')
print(sp.select('title'))
print(sp.select('p'))
print(sp.select('#p1'))
print(sp.select('.red'))
```

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>我是網頁標題</title>
  </head>
  <body>
    <h1 class="large">我是標題</h1>
    <div>
      <p>我是段落</p>
      <img src="https://www.w3.org/html/logo/
downloads/HTML5_Logo_256.png" alt="我是圖片">
      <a href="http://www.e-happy.com.tw">我是超連結</a>
    </div>
      </body>
```

BeautifulSoup模組(解析HTML)

取得標籤的屬性內容

```
from bs4 import BeautifulSoup
sp = BeautifulSoup(html, 'lxml')
print(type(sp.select('img')[0])) # <class
'bs4.element.Tag'>
print(sp.select('img')[0].get('src'))
print(sp.select('a')[0].get('href'))
print(sp.select('img')[0]['src'])
print(sp.select('a')[0]['href'])
```

```
<class 'bs4.element.Tag'>
http://www.ehappy.tw/python.png
http://www.e-happy.com.tw
http://www.ehappy.tw/python.png
http://www.e-happy.com.tw
```

```
html = '''
<html>
  <head><meta charset="UTF-8"><title>我是網頁標題</title></head>
  <body>
      <img src="http://www.ehappy.tw/python.png">
      <a href="http://www.e-happy.com.tw">超連結</a>
  </body>
</html>
'''
```