


資料倉儲 網路爬蟲教學 1

國立臺北護理健康大學 資訊管理系

2021/04/07

A solid blue horizontal bar spanning the width of the slide, located at the bottom.

建立虛擬環境

建立一個虛擬環境，以便安裝合適的套件

Step#1 建立虛擬環境&指定python=3.5版本（相容性較高）

```
conda create -n py35 python=3.5 jupyter numpy matplotlib bs4
```

Step#2 啟動虛擬環境

```
source activate py35
```

Step#3 退出虛擬環境

```
deactivate py35
```

需要套件

(1) requests套件(抓取網頁資料用，anaconda 預設會安裝) 安裝方式

```
pip install requests
```

(2) BeautifulSoup (解析網頁用) 安裝方式

```
pip install bs4
```

建立BeautifulSoup

<http://www.pchome.com.tw>

```
import requests, bs4
```

```
htmlFile = requests.get('http://www.pchome.com.tw')
```

```
objSoup = bs4.BeautifulSoup(htmlFile.text, 'lxml')
```

```
print("列印BeautifulSoup物件資料型態 ", type(objSoup))
```

基本HTML解析

洪錦魁



一個人的極境旅行 - 南極大陸北極海

2015/2016年洪錦魁一個人到南極



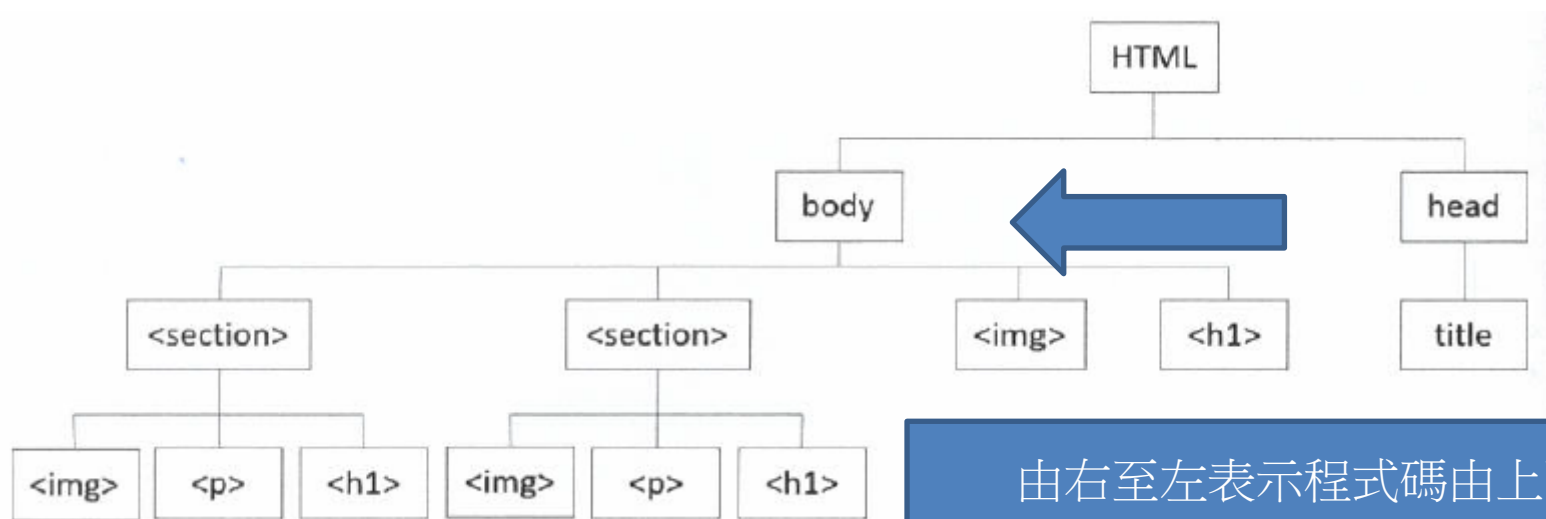
HTML5+CSS3王者歸來

本書講解網頁設計使用HTML5+CSS3



```
1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>洪錦魁著作</title>
6   <style>
7     h1#author { width:400px; height:50px; text-align:center;
8       background:linear-gradient(to right,yellow,green);
9     }
10    h1#content { width:400px; height:50px;
11      background:linear-gradient(to right,yellow,red);
12    }
13    section { background:linear-gradient(to right bottom,yellow,gray); }
14  </style>
15 </head>
16 <body>
17 <h1 id="author">洪錦魁</h1>
18 
19 <section>
20   <h1 id="content">一個人的極境旅行 - 南極大陸北極海</h1>
21   <p>2015/2016年<strong>洪錦魁</strong>一個人到南極</p>
22   
23 </section>
24 <section>
25   <h1 id="content">HTML5+CSS3王者歸來</h1>
26   <p>本書講解網頁設計使用HTML5+CSS3</p>
27   
28 </section>
29 </body>
30 </html>
```

基本HTML解析



基本HTML解析

```
import bs4
```

```
htmlFile = open('myhtml.html', encoding='utf-8')
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```

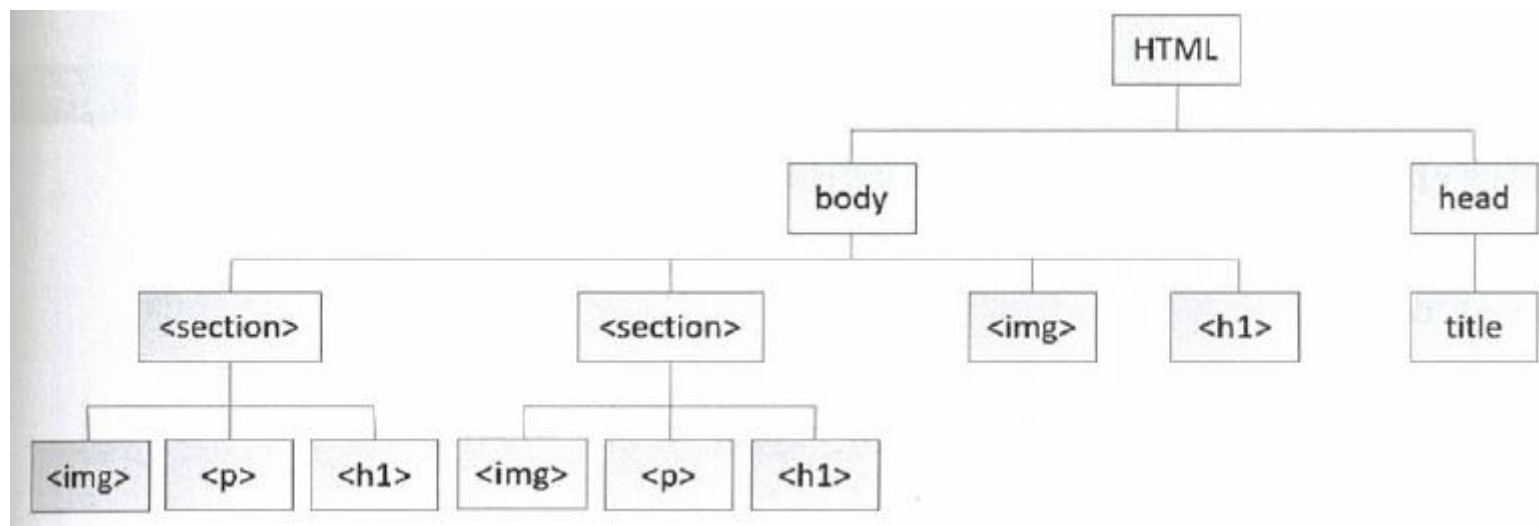
```
print("列印BeautifulSoup物件資料型態 ", type(objSoup))
```

網頁標題屬性-title

```
import bs4
```

```
htmlFile = open('myhtml.html', encoding='utf-8')  
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')  
print("物件類型 =", type(objSoup.title))  
print("列印title =", objSoup.title)
```


網頁標題屬性-title

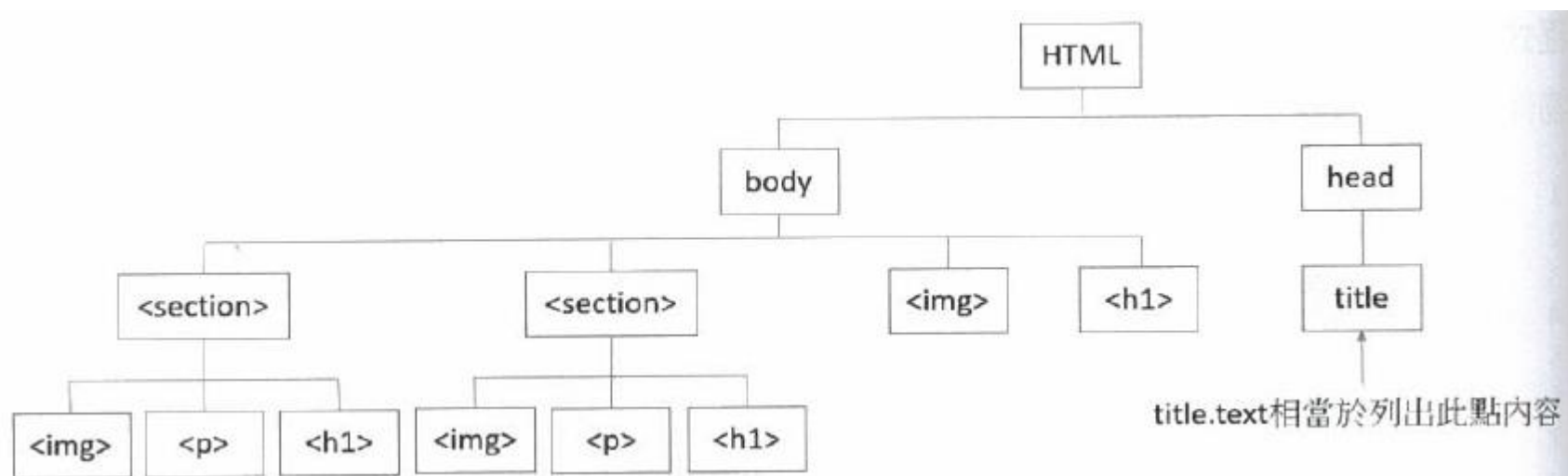


去除標籤傳回文字text屬性

```
import bs4
```

```
htmlFile = open('myhtml.html', encoding='utf-8')  
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')  
print("列印title = ", objSoup.title)  
print("title內容 = ", objSoup.title.text)
```

去除標籤傳回文字text屬性



傳回所找尋第一個符合的標籤 find ()

傳回第一個 <h1>

```
import bs4
```

```
htmlFile = open('myhtml.html', encoding='utf-8')
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```

```
objTag = objSoup.find('h1')
```

```
print("資料型態    =", type(objTag))
```

```
print("列印Tag     =", objTag)
```

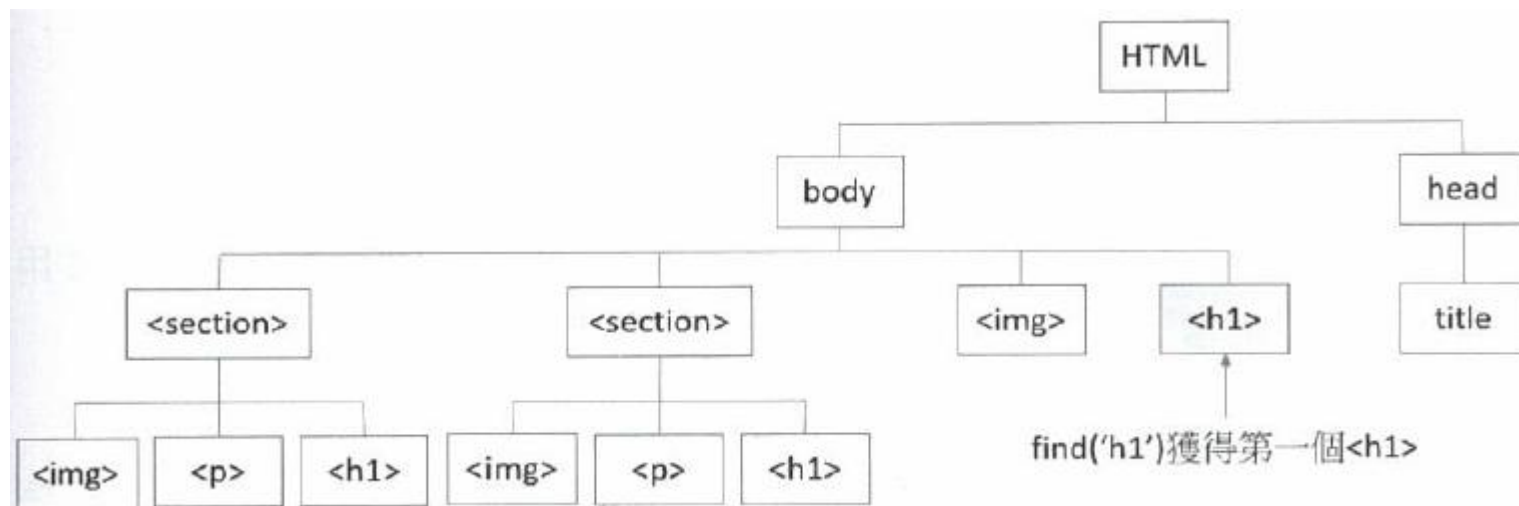
```
print("Text屬性內容 =", objTag.text)
```

```
print("String屬性內容 =", objTag.string)
```

傳回所找尋第一個符合的標籤 find ()

傳回第一個 <h1>

```
IPython 7.13.0 -- An enhanced Interactive Python.  
資料型態      = <class 'bs4.element.Tag'>  
列印Tag        = <h1 id="author">洪錦魁</h1>  
Text屬性內容   = 洪錦魁  
String屬性內容 = 洪錦魁
```



傳回所找尋所有符合的標籤 find_all ()

傳回所有的<h1>

```
import bs4
```

```
htmlFile = open('myhtml.html', encoding='utf-8')
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```

```
objTag = objSoup.find_all('h1')
```

```
print("資料型態 = ", type(objTag))    # 列印資料型態
```

```
print("列印Tag串列 = ", objTag)        # 列印串列
```

```
print("以下是列印串列元素 :")
```

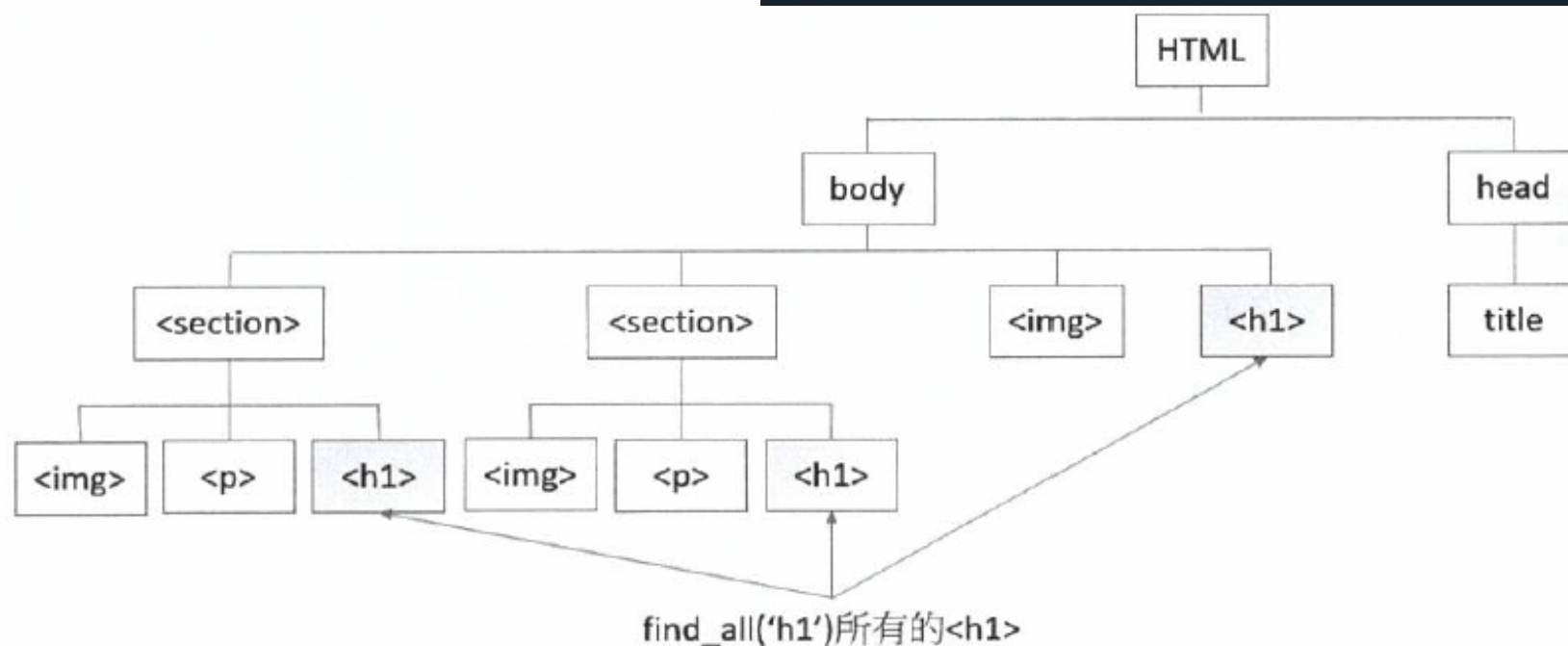
```
for data in objTag:                    # 列印串列元素內容
```

```
    print(data.text)
```

傳回所找尋所有符合的標籤 find_all()

傳回所有的<h1>

```
資料型態 = <class 'bs4.element.ResultSet'>
列印Tag串列 = [<h1 id="author">洪錦魁</h1>, <h1 id="content">一個人的極境旅行 - 南極大陸北極海</h1>, <h1 id="content">HTML5+CSS3王者歸來</h1>]
以下是列印串列元素：
洪錦魁
一個人的極境旅行 - 南極大陸北極海
HTML5+CSS3王者歸來
```



HTML元素

getText()

textContent：內容，不含任何標籤碼。

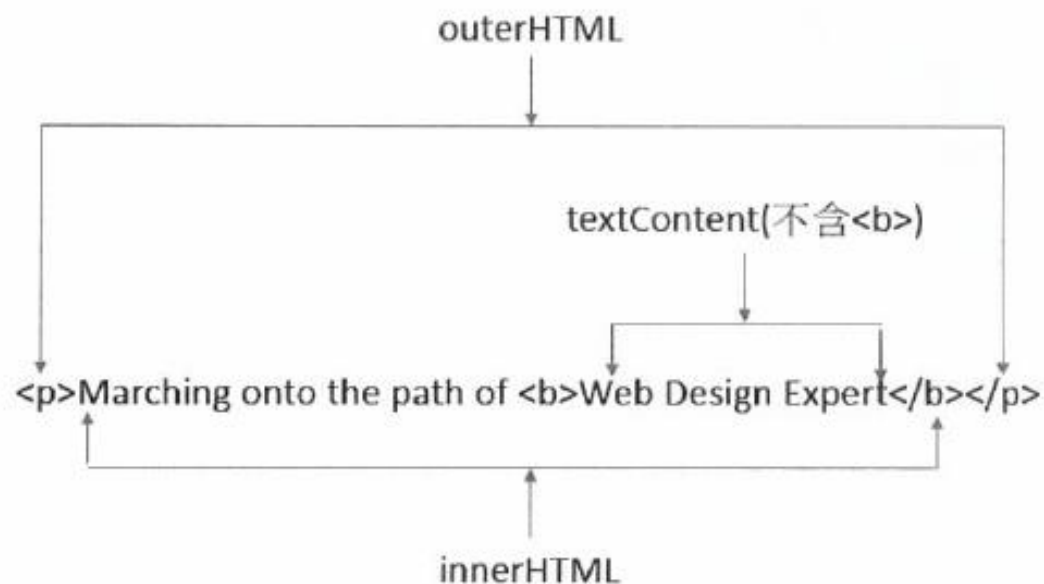
innerHTML：元素內容，含子標籤碼，但是不含本身標籤碼。

outerHTML：元素內容，含子標籤碼，也含本身標籤碼。

如果有一個元素內容如下：

`<p>Marching onto the path of Web Design Expert</p>`

3 個屬性的觀念與內容分別如下：



HTML元素

getText()

```
import bs4

htmlFile = open('myhtml.html', encoding='utf-8')
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
objTag = objSoup.find_all('h1')
print("資料型態 = ", type(objTag))    # 列印資料型態
print("列印Tag串列 = ", objTag)        # 列印串列
print("\n使用Text屬性列印串列元素:")
for data in objTag:                    # 列印串列元素內容
    print(data.text)
print("\n使用getText()方法列印串列元素:")
for data in objTag:
    print(data.getText())
```

HTML元素

getText()

```
資料型態 = <class 'bs4.element.ResultSet'>  
列印Tag串列 = [<h1 id="author">洪錦魁</h1>, <h1 id="content">一個人的極境旅行 - 南極大陸北極海</h1>, <h1 id="content">HTML5+CSS3王者歸來</h1>]
```

使用Text屬性列印串列元素：

洪錦魁

一個人的極境旅行 - 南極大陸北極海

HTML5+CSS3王者歸來

使用getText()方法列印串列元素：

洪錦魁

一個人的極境旅行 - 南極大陸北極海

HTML5+CSS3王者歸來

HTML屬性的搜尋

找第一個含id='author'的節點

```
import bs4
```

```
htmlFile = open('myhtml.html', encoding='utf-8')
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```


```
objTag = objSoup.find(id='author')
```

```
print(objTag)
```


```
print(objTag.text)
```

HTML屬性的搜尋

找第一個含id='author'的節點



```
<h1 id="author">洪錦魁</h1>  
洪錦魁
```



HTML屬性的搜尋

找第一個含id=content'的節點

```
import bs4
```

```
htmlFile = open('myhtml.html', encoding='utf-8')
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```

```
objTag = objSoup.find_all(id='content')
```

```
for tag in objTag:
```

```
    print(tag)
```

```
    print(tag.text)
```

HTML屬性的搜尋

找第一個含id=content'的節點

```
<h1 id="content">一個人的極境旅行 - 南極大陸北極海</h1>  
一個人的極境旅行 - 南極大陸北極海  
<h1 id="content">HTML5+CSS3王者歸來</h1>  
HTML5+CSS3王者歸來
```

HTML屬性的搜尋

找含attrs屬性方式含"-"-之類的屬性名稱

```
import bs4
```

```
htmlFile = "<div book-info='deepmind'>深智數位</div>"  
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')  
tag = objSoup.find(attrs={'book-info': 'deepmind'})  
print(tag)  
print(tag.text)
```

HTML屬性的搜尋

找含attrs屬性方式含"- "之類的屬性名稱



```
<div book-info="deepmind">深智數位</div>  
深智數位
```


使用find ()/ find_all ()執行CSS的搜尋

使用class_和省略方式

```
import bs4
```



```
htmlFile = "<h1 class='boldtext'>深智數位</h1>"
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```

```
tag = objSoup.find('h1', class_='boldtext')
```

```
print(tag)
```

```
print(tag.text)
```

```
print('-'*70)
```

```
tag = objSoup.find('h1', 'boldtext')
```

```
print(tag)
```

```
print(tag.text)
```

使用find ()/ find_all ()執行CSS的搜尋

使用class_和省略方式



```
<h1 class="boldtext">深智數位</h1>
```

深智數位

```
-----  
<h1 class="boldtext">深智數位</h1>
```

深智數位

使用find ()/ find_all ()執行CSS的搜尋

搜尋部分字串符合的節點

```
import bs4  
import re
```



```
htmlFile = "<h1 class='boldtext'>深智數位</h1>"  
objSoup = bs4.BeautifulSoup(htmlFile, 'xml')  
tag = objSoup.find('h1', class_=re.compile('text'))  
print(tag)  
print(tag.text)
```

使用find ()/ find_all ()執行CSS的搜尋

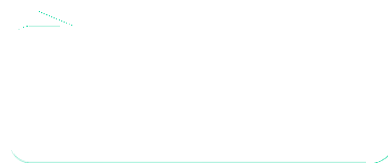
搜尋部分字串符合的節點

```
<h1 class="boldtext">深智數位</h1>  
深智數位
```

使用find ()/ find_all ()執行CSS的搜尋

搜尋任一屬性符合

```
import bs4
import re
```



```
htmlFile = "<h1 class='bold italic'>深智數位</h1>"
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
tag = objSoup.find('h1', class_='bold')
print(tag)
print(tag.text)
print('-'*70)
tag = objSoup.find('h1', class_='italic')
print(tag)
print(tag.text)
```

使用find ()/ find_all ()執行CSS的搜尋

搜尋任一屬性符合



```
<h1 class="bold italic">深智數位</h1>
```

深智數位

```
<h1 class="bold italic">深智數位</h1>
```

深智數位

select ()

搜尋id屬性是author的內容

```
import bs4
```

```
htmlFile = open('myhtml.html', encoding='utf-8')
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```

```
objTag = objSoup.select('#author')
```

```
print("資料型態    =", type(objTag))          # 列印資料型態
```

```
print("串列長度    =", len(objTag))           # 列印串列長度
```

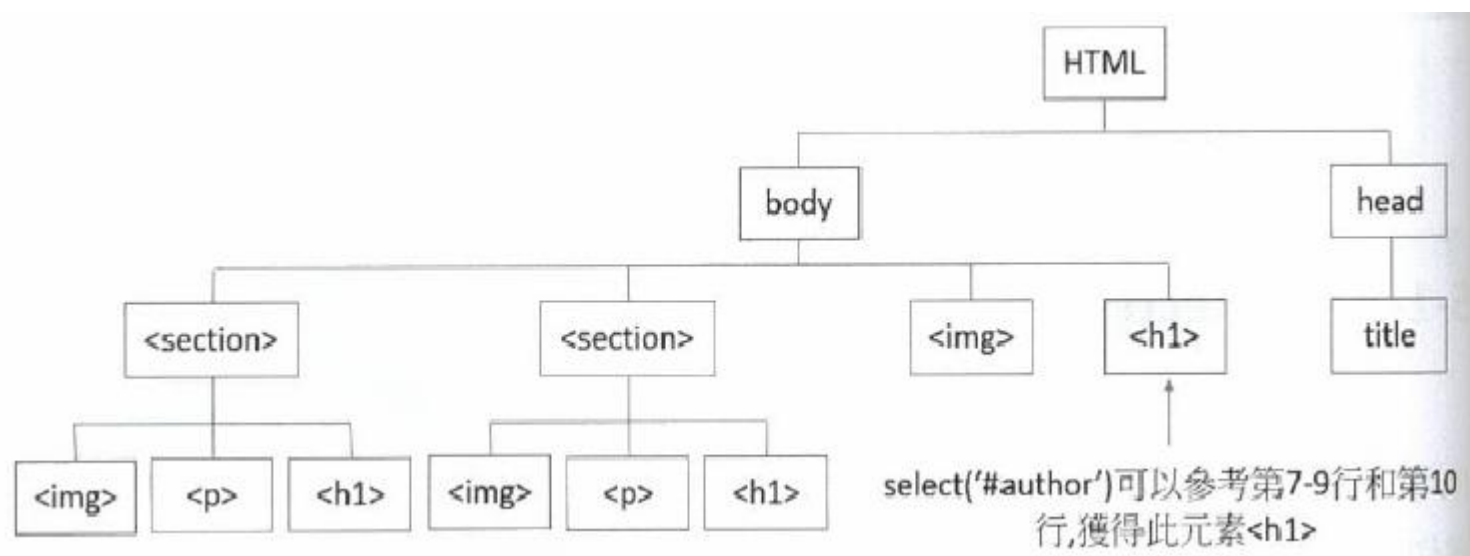
```
print("元素資料型態 =", type(objTag[0]))      # 列印元素資料型態
```

```
print("元素內容    =", objTag[0].getText())   # 列印元素內容
```

select ()

搜尋id屬性是author的內容

```
資料型態 = <class 'bs4.element.ResultSet'>  
串列長度 = 1  
元素資料型態 = <class 'bs4.element.Tag'>  
元素內容 = 洪錦魁
```



select ()

將解析的串列元素傳給str()

```
import bs4

htmlFile = open('myhtml.html', encoding='utf-8')
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
objTag = objSoup.select('#author')
print("列出串列元素的資料型態 = ", type(objTag[0]))
print(objTag[0])
print("列出str()轉換過的資料型態 = ", type(str(objTag[0])))
print(str(objTag[0]))
```

select ()

將解析的串列元素傳給str()

```
列出串列元素的資料型態    = <class 'bs4.element.Tag'>  
<h1 id="author">洪錦魁</h1>  
列出str()轉換過的資料型態 = <class 'str'>  
<h1 id="author">洪錦魁</h1>
```

select ()

將**attrs**屬性應用在串列元素，列出字典結果

```
import bs4
```

```
htmlFile = open('myhtml.html', encoding='utf-8')  
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')  
objTag = objSoup.select('#author')  
print(str(objTag[0].attrs))
```

select ()

將attrs屬性應用在串列元素，列出字典結果

```
{'id': 'author'}
```

select ()

搜尋<p>標籤，最後列出串列內容與不含子標籤的元素內容

```
import bs4
```

```
htmlFile = open('myhtml.html', encoding='utf-8')
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```

```
pObjTag = objSoup.select('p')
```

```
print("含<p>標籤的串列長度 = ", len(pObjTag))
```

```
for pObj in pObjTag:
```

```
    print(str(pObj))          # 內部有子標籤<strong>字串
```

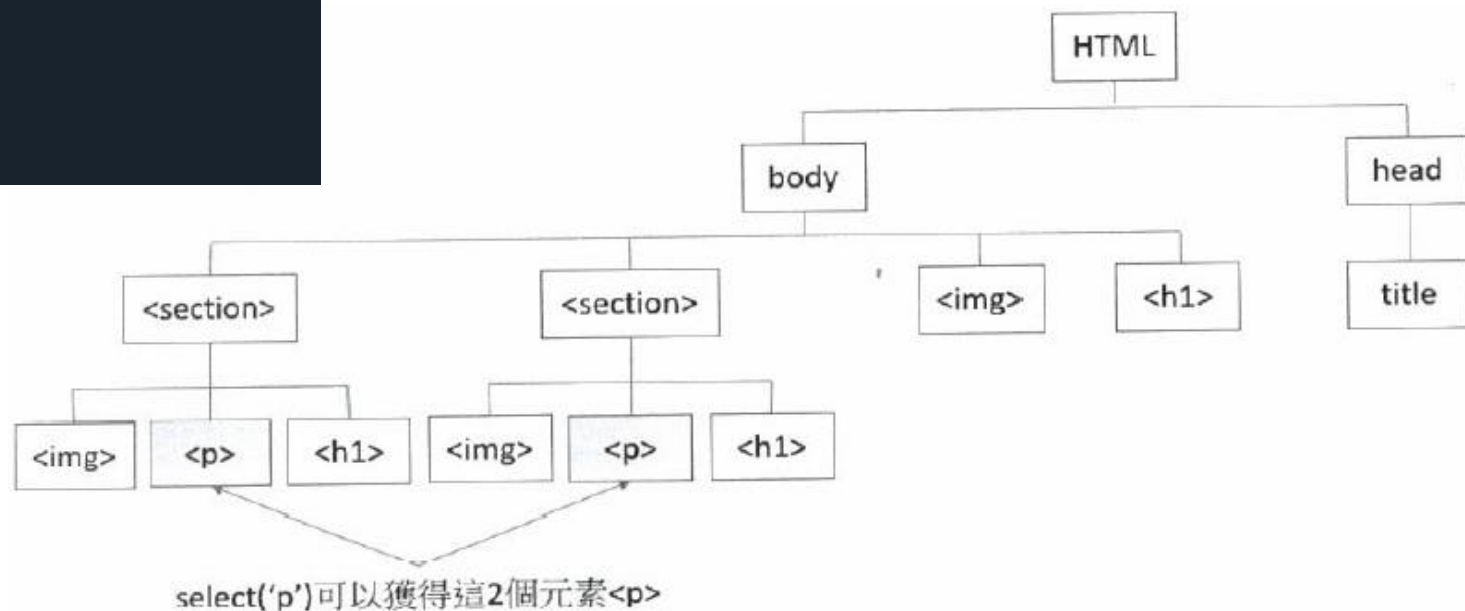
```
    print(pObj.getText())     # 沒有子標籤
```

```
    print(pObj.text)         # 沒有子標籤
```

select ()

搜尋<p>標籤，最後列出串列內容與不含子標籤的元素內容

```
含<p>標籤的串列長度 = 2  
<p>2015/2016年<strong>洪錦魁</strong>一個人到南極</p>  
2015/2016年洪錦魁一個人到南極  
2015/2016年洪錦魁一個人到南極  
<p>本書講解網頁設計使用HTML5+CSS3</p>  
本書講解網頁設計使用HTML5+CSS3  
本書講解網頁設計使用HTML5+CSS3
```



標籤字串的get ()

搜尋標籤

```
import bs4

htmlFile = open('myhtml.html', encoding='utf-8')
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
imgTag = objSoup.select('img')
print("含<img>標籤的串列長度 = ", len(imgTag))
for img in imgTag:
    print(img)
```

標籤字串的get ()

搜尋標籤

```
含<img>標籤的串列長度 = 3  
  
  

```


標籤字串的get ()

取得所有圖檔

```
import bs4

htmlFile = open('myhtml.html', encoding='utf-8')
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
imgTag = objSoup.select('img')
print("含<img>標籤的串列長度 = ", len(imgTag))
for img in imgTag:
    print("列印標籤串列 = ", img)
    print("列印圖檔    = ", img.get('src'))
    print("列印圖檔    = ", img['src']))
```

標籤字串的get ()

取得所有圖檔

```
含<img>標籤的串列長度 = 3  
列印標籤串列 =   
列印圖檔      = hung.jpg  
列印圖檔      = hung.jpg  
列印標籤串列 =   
列印圖檔      = travel.jpg  
列印圖檔      = travel.jpg  
列印標籤串列 =   
列印圖檔      = html5.jpg  
列印圖檔      = html5.jpg
```

爬取項目清單文件

使用ch5_2_1.html

台灣旅遊景點排名

- a. 故宮博物院
- b. 日月潭
- c. 阿里山

台灣夜市排名

- A. 士林夜市
- B. 永康夜市
- C. 逢甲夜市

台灣人口排名

- i. 新北市
- ii. 台北市
- iii. 桃園市

台灣最健康大學排名

- I. 明志科大
- II. 台灣體院
- III. 台北體院

```
1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>ch5_2_1.html</title>
6 </head>
7 <body>
8 <h1>台灣旅遊景點排名</h1>
9 <ol type="a">
10   <li>故宮博物院</li><li>日月潭</li><li>阿里山</li>
11 </ol>
12 <h2>台灣夜市排名</h2>
13 <ol type="A">
14   <li>士林夜市</li><li>永康夜市</li><li>逢甲夜市</li>
15 </ol>
16 <h2>台灣人口排名</h2>
17 <ol type="i">
18   <li>新北市</li><li>台北市</li><li>桃園市</li>
19 </ol>
20 <h2>台灣最健康大學排名</h2>
21 <ol type="I">
22   <li>明志科大</li><li>台灣體院</li><li>台北體院</li>
23 </ol>
24 </body>
25 </html>
```

爬取項目清單文件

使用ch5_2_1.html

```
import requests, bs4

url = 'ch5_2_1.html'
htmlFile = open(url, encoding='utf-8')
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
titleobj = objSoup.find_all('h2')          # h2標題
print(titleobj[2].text)

itemobj = objSoup.find('ol', type='I')     # type='I'
items = itemobj.find_all('li')
for item in items:
    print(item.text)
```

爬取項目清單文件

使用ch5_2_1.html

台灣最健康大學排名
明志科大
台灣體院
台北體院

爬取項目清單文件

使用ch5_2_2.html

國家首都資料表

Washington	美國首都
Tokyo	日本首都
Paris	法國首都

```
1 <!doctype html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <title>ch5_2_2.html</title>
6 </head>
7 <body>
8 <h1>國家首都資料表</h1>
9 <dl>
10     <dt>Washington</dt>
11         <dd>美國首都</dd>
12     <dt>Tokyo</dt>
13         <dd>日本首都</dd>
14     <dt>Paris</dt>
15         <dd>法國首都</dd>
16 </dl>
17 </body>
18 </html>
```

爬取項目清單文件

使用ch5_2_2.html

```
import requests, bs4

url = 'ch5_2_2.html'
htmlFile = open(url, encoding='utf-8')
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')

mycity = []
cityobj = objSoup.find('dl')
cities = cityobj.find_all('dt')
for city in cities:
    mycity.append(city.text)          # mycity串列

mycountry = []
countryobj = objSoup.find('dl')
countries = countryobj.find_all('dd')
for country in countries:
    mycountry.append(country.text)    # mycountry串列

print("國家 = ", mycountry)
print("首都 = ", mycity)
data = dict(zip(mycountry, mycity))
print(data)
```

爬取項目清單文件

使用ch5_2_2.html

```
國家 = ['美國首都', '日本首都', '法國首都']  
首都 = ['Washington', 'Tokyo', 'Paris']  
{'美國首都': 'Washington', '日本首都': 'Tokyo', '法國首都': 'Paris'}
```


爬取表格文件

使用ch5_2_3.html

聯合國水資源中心		
河流名稱	國家	洲名
長江	中國	亞洲
尼羅河	埃及	非洲
亞馬遜河	巴西	南美洲
製表2017年5月30日		

```
1 <!doctype html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <title>ch5_2_3.html</title>
6 </head>
7 <body>
8 <table border="1">
9     <thead><!-- 建立表頭 -->
10         <tr><th colspan="3">聯合國水資源中心</th></tr>
11         <tr><th>河流名稱</th><th>國家</th><th>洲名</th></tr>
12     </thead>
13     <tbody><!-- 建立表格本體 -->
14         <tr><td>長江</td><td>中國</td><td>亞洲</td></tr>
15         <tr><td>尼羅河</td><td>埃及</td><td>非洲</td></tr>
16         <tr><td>亞馬遜河</td><td>巴西</td><td>南美洲</td></tr>
17     </tbody>
18     <tfoot><!-- 建立表尾 -->
19         <tr><td colspan="3">製表2017年5月30日</td></tr>
20     </tfoot>
21 </table>
22 </body>
23 </html>
```

爬取表格文件

使用ch5_2_3.html

```
import requests, bs4

url = 'ch5_2_3.html'
htmlFile = open(url, encoding='utf-8')
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')

myriver = []          # 河川
tableobj = objSoup.find('table').find('tbody')
tables = tableobj.find_all('tr')
for table in tables:
    river = table.find('td')
    myriver.append(river.text)

mycountry = []        # 國家
for table in tables:
    countries = table.find_all('td')
    country = countries[1]
    mycountry.append(country.text)

print("國家 = ", mycountry)
print("河川 = ", myriver)
data = dict(zip(mycountry, myriver))
print(data)           # 字典顯示結果
```

爬取表格文件

使用ch5_2_3.html

```
國家 = ['中國', '埃及', '巴西']  
河川 = ['長江', '尼羅河', '亞馬遜河']  
{ '中國': '長江', '埃及': '尼羅河', '巴西': '亞馬遜河' }
```

find_next_sibling、find_previous_sibling

使用ch5_2_3.html-另解：find_next_sibling

```
import requests, bs4

url = 'ch5_2_3.html'
htmlFile = open(url, encoding='utf-8')
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')

myriver = []                # 河川
mycountry = []              # 國家
tableobj = objSoup.find('table').find('tbody')
tables = tableobj.find_all('tr')
for table in tables:
    river = table.find('td')
    myriver.append(river.text)
    country = river.find_next_sibling('td') # 下一個節點
    mycountry.append(country.text)

print("國家 = ", mycountry)
print("河川 = ", myriver)
data = dict(zip(mycountry, myriver))
print(data)                  # 字典顯示結果
```

```
國家 = ['中國', '埃及', '巴西']
河川 = ['長江', '尼羅河', '亞馬遜河']
{'中國': '長江', '埃及': '尼羅河', '巴西': '亞馬遜河'}
```

find_next_sibling、find_previous_sibling

使用ch5_2_3.html-

比較：find_next_sibling find_previous_sibling

```
import requests, bs4
```

```
url = 'ch5_2_3.html'
```

```
htmlFile = open(url, encoding='utf-8')
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```

```
myriver = [] # 河川
```

```
mystate = [] # 洲名
```

```
tableobj = objSoup.find('table').find('tbody')
```

```
tables = tableobj.find_all('tr')
```

```
for table in tables:
```

```
    countries = table.find_all('td')
```

```
    country = countries[1] # 國家節點
```

```
    river = country.find_previous_sibling('td') # 前一個節點
```

```
    myriver.append(river.text)
```

```
    state = country.find_next_sibling('td') # 下一個節點
```

```
    mystate.append(state.text)
```

```
print("洲名 = ", mystate)
```

```
print("河川 = ", myriver)
```

```
data = dict(zip(mystate, myriver))
```

```
print(data) # 字典顯示結果
```

```
洲名 = ['亞洲', '非洲', '南美洲']  
河川 = ['長江', '尼羅河', '亞馬遜河']  
{ '亞洲': '長江', '非洲': '尼羅河', '南美洲': '亞馬遜河' }
```

find_next_sibling 、 find_previous_sibling

使用ch5_2_1.html找同父節點的同層節點

```
import bs4
```

```
url = 'ch5_2_1.html'
```

```
htmlFile = open(url, encoding='utf-8')
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```

```
titleobj = objSoup.find('h2') # h2標題
```

```
title = titleobj.find_next_siblings('h2') # 下一系列節點
```

```
print('find_next_siblings = ', title)
```

```
titleobj = objSoup.find_all('h2')
```

```
title = titleobj[2].find_previous_siblings('h2') # 前一系列節點
```

```
print('find_previous_siblings = ', title)
```

find_next_sibling 、 find_previous_sibling

使用ch5_2_1.html找同父節點的同層節點

```
find_next_siblings    = [<h2>台灣人口排名</h2>, <h2>台灣最健康大學排名</h2>]  
find_previous_siblings = [<h2>台灣人口排名</h2>, <h2>台灣夜市排名</h2>]
```

parent()

使用ch5_2_3.html

```
import bs4
```

```
url = 'ch5_2_3.html'
```

```
htmlFile = open(url, encoding='utf-8')
```

```
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
```

```
myriver = []                # 河川
```

```
tableobj = objSoup.find('table').find('tbody')
```

```
tables = tableobj.find_all('tr')
```

```
river = tables[1].find('td')
```

```
print(river.text)
```

```
river_parent = river.parent()
```

```
print(river_parent)
```


parent()

使用ch5_2_3.html

尼羅河

```
[<td>尼羅河</td>, <td>埃及</td>, <td>非洲</td>]
```

parent()、find_next_sibling、find_previous_sibling

使用ch5_2_3.html

```
import bs4
url = 'ch5_2_3.html'
htmlFile = open(url, encoding='utf-8')
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
myriver = []          # 河川
tableobj = objSoup.find('table').find('tbody')
tables = tableobj.find_all('tr')
river = tables[1].find('td')
print(river.text)
```

上移至父節點爬取後一個節點

```
previous_row = river.parent.find_previous_sibling()
print(previous_row)
next_row = river.parent.find_next_sibling()
print(next_row)
```

上移至父節點爬取前一個節點

parent()、find_next_sibling、find_previous_sibling

使用ch5_2_3.html

尼羅河

```
<tr><td>長江</td><td>中國</td><td>亞洲</td></tr>  
<tr><td>亞馬遜河</td><td>巴西</td><td>南美洲</td></tr>
```

上移至父節點爬取後一個節點

parent()、find_next_sibling、find_previous_sibling

使用ch5_2_3.html

```
import bs4
url = 'ch5_2_3.html'
htmlFile = open(url, encoding='utf-8')
objSoup = bs4.BeautifulSoup(htmlFile, 'lxml')
myriver = []          # 河川|
tableobj = objSoup.find('table').find('tbody')
tables = tableobj.find_all('tr')
river = tables[0].find('td')
print(river.text)
previous_rows = river.parent.find_next_siblings()
print(previous_rows)
```

上移至父節點爬取前一個節點

```
river = tables[2].find('td')
print(river.text)
next_rows = river.parent.find_previous_siblings()
print(next_rows)
```

上移至父節點爬取後一個節點

parent()、find_next_sibling、find_previous_sibling

使用ch5_2_3.html

上移至父節點爬取前一個節點

長江

```
[<tr><td>尼羅河</td><td>埃及</td><td>非洲</td></tr>, <tr><td>亞馬遜河</td><td>巴西</td><td>南美洲</td></tr>]
```

亞馬遜河

```
[<tr><td>尼羅河</td><td>埃及</td><td>非洲</td></tr>, <tr><td>長江</td><td>中國</td><td>亞洲</td></tr>]
```

上移至父節點爬取後一個節點

圖片下載實作

<http://aaa.24ht.com.tw/>

```
import bs4, requests, os
```

```
headers = { 'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64)\n\nAppleWebKit/537.36 (KHTML, like Gecko) Chrome/45.0.2454.101\n\nSafari/537.36', }
```

```
url = 'http://aaa.24ht.com.tw/' # 這個伺服器會擋住網頁
```

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

```
print("網頁下載中 ...")
```

```
html.raise_for_status() # 驗證網頁是否下載成功
```

```
print("網頁下載完成")
```

```
destDir = 'out5_15' # 設定儲存資料夾
```

```
if os.path.exists(destDir) == False:
```

```
    os.mkdir(destDir) # 建立目錄供未來儲存圖片
```

```
objSoup = bs4.BeautifulSoup(html.text, 'lxml') # 建立BeautifulSoup物件
```

圖片下載實作

<http://aaa.24ht.com.tw/>

```
imgTag = objSoup.select('img') # 搜尋所有圖片檔案
print("搜尋到的圖片數量 = ", len(imgTag)) # 列出搜尋到的圖片數量
if len(imgTag) > 0: # 如果有找到圖片則執行下載與儲存
    for i in range(len(imgTag)): # 迴圈下載圖片與儲存
        imgUrl = imgTag[i].get('src') # 取得圖片的路徑
        print("%s 圖片下載中 ... " % imgUrl)
        finUrl = url + imgUrl # 取得圖片在Internet上的路徑
        print("%s 圖片下載中 ... " % finUrl)
        picture = requests.get(finUrl, headers=headers) # 下載圖片
        picture.raise_for_status() # 驗證圖片是否下載成功
        print("%s 圖片下載成功" % finUrl)

# 先開啟檔案, 再儲存圖片
pictFile = open(os.path.join(destDir, os.path.basename(imgUrl)), 'wb')
for diskStorage in picture.iter_content(10240):
    pictFile.write(diskStorage)
pictFile.close() # 關閉檔案
```

圖片下載實作

<http://aaa.24ht.com.tw/>

```
網頁下載中 ...  
網頁下載完成  
搜尋到的圖片數量 = 3  
hung.jpg 圖片下載中 ...  
http://aaa.24ht.com.tw/hung.jpg 圖片下載中 ...  
http://aaa.24ht.com.tw/hung.jpg 圖片下載成功  
travel.jpg 圖片下載中 ...  
http://aaa.24ht.com.tw/travel.jpg 圖片下載中 ...  
http://aaa.24ht.com.tw/travel.jpg 圖片下載成功  
html5.jpg 圖片下載中 ...  
http://aaa.24ht.com.tw/html5.jpg 圖片下載中 ...  
http://aaa.24ht.com.tw/html5.jpg 圖片下載成功
```


台灣彩券威力彩開獎結果

www.taiwanlottery.com.tw

```
import bs4, requests
```

```
url = 'http://www.taiwanlottery.com.tw'
```

```
html = requests.get(url)
```

```
print("網頁下載中 ...")
```

```
html.raise_for_status()
```

```
print("網頁下載完成")
```

驗證網頁是否下載成功

```
objSoup = bs4.BeautifulSoup(html.text, 'lxml') # 建立BeautifulSoup物件
```

```
dataTag = objSoup.select('.contents_box02') # 尋找class是contents_box02
```

```
print("串列長度", len(dataTag))
```

```
for i in range(len(dataTag)):
```

列出含contents_box02的串列

```
    print(dataTag[i])
```

台灣彩券威力彩開獎結果

www.taiwanlottery.com.tw

找尋開出順序與大小順序的球

```
balls = dataTag[0].find_all('div', {'class':'ball_tx ball_green'})
```

```
print("開出順序 :", end="")
```

```
for i in range(6):
```

前6球是開出順序

```
    print(balls[i].text, end=' ')
```

```
print("\n大小順序 :", end="")
```

```
for i in range(6, len(balls)):
```

第7球以後是大小順序

```
    print(balls[i].text, end=' ')
```

找出第二區的紅球

```
redball = dataTag[0].find_all('div', {'class':'ball_red'})
```

```
print("\n第二區 :", redball[0].text)
```

台灣彩券威力彩開獎結果

網頁下載中 ...

網頁下載完成

串列長度 4

```
<div class="contents_box02">
<div id="contents_logo_02"></div><div class="contents_mine_tx02"><span class="font_black15">110/4/5 第110000027期 </span><span class="font_red14"><a
href="Result_all.aspx#01">開獎結果</a></span></div><div class="contents_mine_tx04">開出順序<br/>大小順序<br/>第二區</div><div class="ball_tx ball_green">04 </
div><div class="ball_tx ball_green">05 </div><div class="ball_tx ball_green">11 </div><div class="ball_tx ball_green">02 </div><div class="ball_tx
ball_green">32 </div><div class="ball_tx ball_green">24 </div><div class="ball_tx ball_green">02 </div><div class="ball_tx ball_green">04 </div><div
class="ball_tx ball_green">05 </div><div class="ball_tx ball_green">11 </div><div class="ball_tx ball_green">24 </div><div class="ball_tx ball_green">32 </
div><div class="ball_red">01 </div>
</div>
<div class="contents_box02">
<div id="contents_logo_03"></div><div class="contents_mine_tx02"><span class="font_black15">110/4/5 第110000027期 </span><span class="font_red14"><a
href="Result_all.aspx#07">開獎結果</a></span></div><div class="contents_mine_tx04">開出順序<br/>大小順序</div><div class="ball_tx ball_green">04 </div><div
class="ball_tx ball_green">05 </div><div class="ball_tx ball_green">11 </div><div class="ball_tx ball_green">02 </div><div class="ball_tx ball_green">32 </
div><div class="ball_tx ball_green">24 </div><div class="ball_tx ball_green">02 </div><div class="ball_tx ball_green">04 </div><div class="ball_tx
ball_green">05 </div><div class="ball_tx ball_green">11 </div><div class="ball_tx ball_green">24 </div><div class="ball_tx ball_green">32 </div>
</div>
<div class="contents_box02">
<div id="contents_logo_04"></div><div class="contents_mine_tx02"><span class="font_black15">110/4/6 第110000037期 </span><span class="font_red14"><a
href="Result_all.aspx#02">開獎結果</a></span></div><div class="contents_mine_tx04">開出順序<br/>大小順序<br/>特別號</div><div class="ball_tx ball_yellow">22
</div><div class="ball_tx ball_yellow">01 </div><div class="ball_tx ball_yellow">38 </div><div class="ball_tx ball_yellow">28 </div><div class="ball_tx
ball_yellow">34 </div><div class="ball_tx ball_yellow">26 </div><div class="ball_tx ball_yellow">01 </div><div class="ball_tx ball_yellow">22 </div><div
class="ball_tx ball_yellow">26 </div><div class="ball_tx ball_yellow">28 </div><div class="ball_tx ball_yellow">34 </div><div class="ball_tx ball_yellow">38
</div><div class="ball_red">35 </div>
</div>
ball_yellow">34 </div><div class="ball_tx ball_yellow">26 </div><div class="ball_tx ball_yellow">01 </div><div class="ball_tx ball_yellow">22 </div><div
class="ball_tx ball_yellow">26 </div><div class="ball_tx ball_yellow">28 </div><div class="ball_tx ball_yellow">34 </div><div class="ball_tx ball_yellow">38
</div><div class="ball_red">35 </div>
</div>
<div class="contents_box02">
<div id="contents_logo_05"></div><div class="contents_mine_tx02"><span class="font_black15">110/4/6 第110000037期 </span><span class="font_red14"><a
href="Result_all.aspx#08">開獎結果</a></span></div><div class="contents_mine_tx04">開出順序<br/>大小順序</div><div class="ball_tx ball_yellow">22 </div><div
class="ball_tx ball_yellow">01 </div><div class="ball_tx ball_yellow">38 </div><div class="ball_tx ball_yellow">28 </div><div class="ball_tx ball_yellow">34
</div><div class="ball_tx ball_yellow">26 </div><div class="ball_tx ball_yellow">01 </div><div class="ball_tx ball_yellow">22 </div><div class="ball_tx
ball_yellow">26 </div><div class="ball_tx ball_yellow">28 </div><div class="ball_tx ball_yellow">34 </div><div class="ball_tx ball_yellow">38 </div>
</div>
開出順序 : 04    05    11    02    32    24
大小順序 : 02    04    05    11    24    32
第二區   : 01
```

.tw