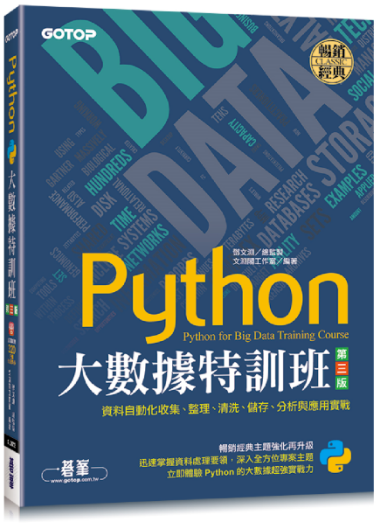


CHAPTER

02

數據資料的爬取

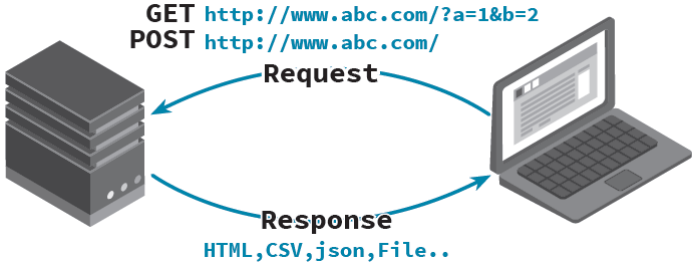
GOTOP

Python 大數據特訓班

2.1 requests 模組：讀取網站檔案

2.1.1 網路資料爬取的原理

使用者要在電腦上瀏覽網頁，基本流程是開啟瀏覽器輸入網址送出後，電腦會透過網路對網址所指定的伺服器發出要求(Request)，伺服器再根據要求透過網路回應(Response) 資料給原來的電腦，顯示在瀏覽器上。



回首頁

GOTOP

Python 大數據特訓班

2.1.2 發送GET 請求

基本語法

```
import requests
Response物件 = requests.get(網址)
```

讀取網頁原始碼

例如：讀取網頁的原始碼。

```
[1]  1  import requests
     2  url = 'http://www.ehappy.tw/demo.htm'
     3  html = requests.get(url)
     4  # 檢查HTTP回應碼是否為200(requests.code.ok)
     5  if html.status_code == requests.codes.ok:
     6      print(html.text)
```

GOTOP Python 大數據特訓班

加上 URL 參數

在 requests 模組中，URL 參數要用字典資料型態進行定義，接著用GET 請求時必須將URL 參數內容設定為 params 參數，即可完成。

例如：設定 params 參數出GET 請求。

```
[2]  1  import requests
     2  # 將查詢參數定義為字典資料加入GET請求中
     3  payload = {'key1': 'value1', 'key2': 'value2'}
     4  html = requests.get("http://httpbin.org/get",
     5                      params=payload)
     6  print(html.text)
```

GOTOP Python 大數據特訓班

2.1.3 發送POST 請求

在requests 模組中，POST 傳遞的參數要定義成字典資料型態，接著用POST 請求時必須將傳遞的參數內容設定為data 參數，即可完成。

例如：設定data 參數提出POST 請求。

```
     1  import requests
     2  # 將查詢參數加入 POST 請求中
     3  payload = {'key1': 'value1', 'key2': 'value2'}
     4  html = requests.post("http://httpbin.org/post",
     5                       data=payload)
     6  print(html.text)
```

GOTOP Python 大數據特訓班

2.1.4 自訂HTTP Headers 偽裝瀏覽器操作

在進階的網路爬蟲程式中，自訂HTTP Headers 可以將爬取的動作偽裝為瀏覽器的操作，避過網頁的檢查，這是一個常用的技術。設定的方式是在headers 中設定user-agent 的屬性，其格式如下：

```
headers = {'user-agent': 'Mozilla/5.0 (Linux; Android 8.0.0; \
            SM-G960F Build/R16NW) AppleWebKit \
            /537.36 (KHTML, like Gecko) \
            Chrome/62.0.3202.84 Mobile Safari/537.36'
}
```

例如：台灣高鐵的網路訂票頁面(https://irs.thsrc.com.tw/IMINT/)，當進行HTTP 要求時會先檢查操作者是否為瀏覽器，如果不是則無法正常讀取內容。

GOTOP Python 大數據特訓班

程式碼

```
[3]  1  import requests
     2  url = 'https://irs.thsrc.com.tw/IMINT/'
     3  # 自訂表頭
     4  headers = {'user-agent': 'Mozilla/5.0 (Linux; Android 8.0.0; \
     5              SM-G960F Build/R16NW) AppleWebKit \
     6              /537.36 (KHTML, like Gecko) \
     7              Chrome/62.0.3202.84 Mobile Safari/537.36'
     8  }
     9  # 將自訂表頭加入 GET 請求中
    10  html = requests.get(url, headers=headers)
    11  print(html)
```

回應的HTTP 狀態碼為 200，表示正確讀取。如果不加自訂的headers 設定，執行時程式會卡住無法正確執行喔！

```
<Response [200]>
```

GOTOP

Python 大數據特訓班

2.1.5 使用Session 及Cookie 進入認證頁面

利用 Cookie 檢查篩選使用者

以熱門的批踢踢實業坊八卦討論板 (https://www.ptt.cc/bbs/Gossiping/index.html) 為例，如果想要進入討論板瀏覽內容。在第一次進入時會因為沒有認證而被重新導到「https://www.ptt.cc/ask/over18」，目的是要確定瀏覽者年滿18 歲才能進入。這是一個對於使用者資格進行確認的防護機制，不過對於網路爬蟲來說則是一個很大的考驗，因為在資料擷取的時候，必須先要經由認證的動作來取得身份才能正常的進行。

GOTOP

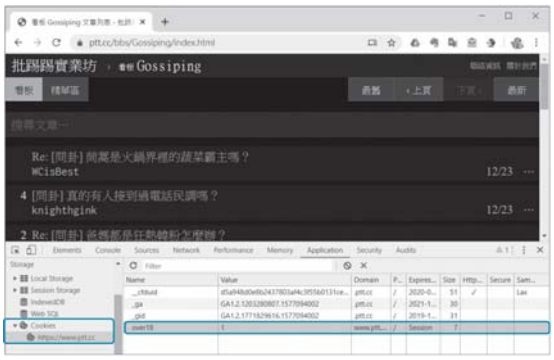
Python 大數據特訓班

檢視產生的Cookie 值

這裡將要使用Chrome 瀏覽器的開發人員工具進行Cookie 的檢視，請開啟批踢踢實業坊八卦討論板頁面，通過年滿18 歲驗證後進入討論板面。

請由瀏覽器右上角的 ⋮ / **更多工具/ 開發人員工具**，或是按 **F12** 鍵開啟 **開發人員工具**，選按**Application** 頁籤，選擇左方的**Cookies** 裡的目前網址，此時右方會顯示目前瀏覽器儲存的Cookie 值。

其中有一個Cookie 名稱為「over18」，值為1，目前的頁面就是透過這個Cookie值來判斷瀏覽者有沒有通過年滿18 歲驗證的頁面。



GOTOP

Python 大數據特訓班

在requests 請求時加入Cookie

回到剛才的範例中，如果想要順利爬取批踢踢實業坊八卦討論板的內容，就必須在請求時加入「over18=1」的cookie 值。

範例：GET 請求中設定params 參數。

```
[7]  1  import requests
     2  url = 'https://www.ptt.cc/bbs/Gossiping/index.html'
     3  # 設定cookies的值
     4  cookies = {'over18':'1'}
     5  html = requests.get(url, cookies=cookies)
     6  print(html.text)
```

GOTOP

Python 大數據特訓班

2.2 BeautifulSoup 模組：網頁解析

2.2.1 安裝 Beautifulsoup 模組

BeautifulSoup 模組可以快速的由HTML 中提取內容，只要對於網頁結構有基本的了解，即可透過一定的邏輯取出複雜頁面中指定的資料。

可以使用下列指令在Python 中安裝BeautifuleSoup：

```
! pip install -U beautifulsoup4
```

GOTOP

Python 大數據特訓班

2.2.2 認識網頁的結構

網頁的內容其實是純文字，一般都會儲存為.htm 或.html 的檔案。網頁是使用 HTML(Hypertext Markup Language) 語法利用標籤(tag) 建構內容，讓瀏覽器在讀取後能根據其敘述呈現網頁。以下的範例網頁 (http://ehappy.tw/bsdemo1.htm)，是個結構單純的頁面：

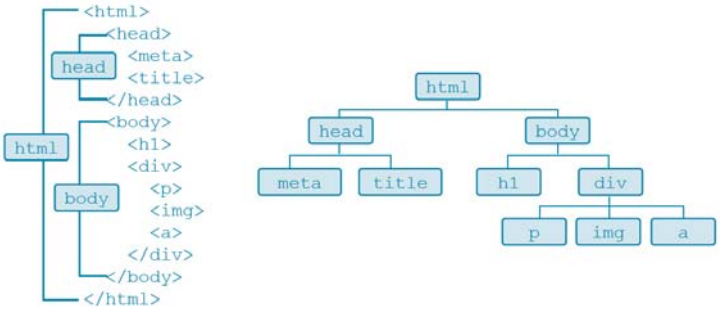
程式碼：bsdemo1.htm

```
<!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>
</html>
```

GOTOP

Python 大數據特訓班

最上層的節點是<html>，在以下分成二個部份：<head> 及<body>，<head> 之中有<meta> 及<title>，而<body> 中又有<h1> 與 ，最後在 之下又有<p>、<img> 及<a>。



GOTOP

Python 大數據特訓班

2.2.3 BeautifulSoup 的使用

語法範例如下：

```
from bs4 import BeautifulSoup
BeautifulSoup物件 = BeautifulSoup(原始碼, 解析器)
```

BeautifulSoup 常用的解析器如下，建議使用lxml 模組進行解析：

語法	說明
BeautifulSoup(原始碼,'html.parser')	python 內建，執行速度適中，文件容錯能力強。
BeautifulSoup(原始碼,'lxml')	執行速度快，文件容錯能力強。

GOTOP

Python 大數據特訓班

2.2.4 BeautifulSoup 常用的屬性

屬性	說明
標籤名稱	傳回指定標籤內容，例如：sp.title 傳回 <title> 的標籤內容。
text	傳回去除所有 HTML 標籤後的網頁文字內容。

例如：建立BeautifulSoup 型別物件sp，解析「http://ehappy.tw/bsdemo1.htm」網頁原始碼。接著用標籤名稱與text 二個屬性，取出指定的內容。

```
[10] 1  import requests
     2  from bs4 import BeautifulSoup
     3  url = 'http://ehappy.tw/bsdemo1.htm'
     4  html = requests.get(url)
     5  html.encoding = 'UTF-8'
     6  sp = BeautifulSoup(html.text, 'lxml')
     7  print(sp.title)
     8  print(sp.title.text)
     9  print(sp.h1)
    10  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>
```

GOTOP

Python 大數據特訓班

2.2.5 BeautifulSoup 常用的方法

BeautifulSoup 常用的方法如下：

方法	說明
find()	尋找第一個符合條件的標籤，以 **字串** 回傳。例如：sp.find("a")。
find_all()	尋找所有符合條件的標籤，以 **串列** 回傳。例如：sp.find_all("a")。
select()	尋找指定 CSS 選擇器如 id 或 class 的內容，以 **串列** 回傳。例如： 以 id 讀取：sp.select("#id") 以 class 讀取：sp.select(".classname")

GOTOP

Python 大數據特訓班

2.2.6 找尋指定標籤的內容：find()、find_all()

find

語法：

```
BeautifulSoup 物件 .find( 標籤名稱 )
```

find_all

語法：

```
BeautifulSoup 物件 .find_all( 標籤名稱 )
```

加入標籤屬性為搜尋條件

1. 將屬性值做為find() 或find_all() 方法的參數，語法：

```
BeautifulSoup 物件 .find 或 find_all( 標籤名稱 , 屬性名稱 = 屬性內容 )
```

2. 將屬性值化為字典資料，做為find() 或find_all() 方法的參數，語法：

```
BeautifulSoup 物件 .find 或 find_all( 標籤名稱 ,{ 屬性名稱 : 屬性內容 })
```

GOTOP Python 大數據特訓班

```
[11]  1  html = '''
      2  <html>
      3    <head><meta charset="UTF-8"><title>我是網頁標題</title></he
      4    <body>
      5        <p id="p1">我是段落一</p>
      6        <p id="p2" class='red'>我是段落二</p>
      7    </body>
      8  </html>
      9  '''
```

以find()、find_all() 方法尋找指定標籤：

```
[12]  1  from bs4 import BeautifulSoup
      2  sp = BeautifulSoup(html, 'lxml')
      3  print(sp.find('p'))
      4  print(sp.find_all('p'))
      5  print(sp.find('p', {'id':'p2', 'class':'red'}))
      6  print(sp.find('p', id='p2', class_= 'red'))
```

```
<p id="p1">我是段落一</p>
[<p id="p1">我是段落一</p>, <p class="red" id="p2">我是段落二</p>]
<p class="red" id="p2">我是段落二</p>
<p class="red" id="p2">我是段落二</p>
```

GOTOP Python 大數據特訓班

2.2.7 利用CSS 選擇器找尋內容：select()

選取標籤、id 及class 類別

1. **選取標籤**：直接設定標籤是最常用的方式，例如：讀取<title> 標籤：

```
datas = sp.select("title")
```

2. **選取id 編號**：因為標籤中的id 屬性不能重複，會是唯一的值，讀取時最明確。

   例如：讀取id 為firstdiv 的標籤內容，請記得id 前必須加上「#」符號。

```
內容範例：<div id="firstdiv"> 文件內容 </div>
選取方式：datas = sp.select("#firstdiv")
```

3. **選取css 類別名稱**：類別名稱前必須加上「.」符號。例如：

```
內容範例：<p class="title"><b> 文件標題 </b></p>
選取方式：data1 = sp.select(".title")
```

GOTOP Python 大數據特訓班

4. **複合選取**：當有多層標籤、id 或類別嵌套時，也可以使用select 方法逐層尋找。

```
datas = sp.select("html head title") #html 下的 head 下的 title 內容
```

特別要再提醒，**select() 的回傳即使只有一個值，它還是會以串列表示。**

```
[13]  1  from bs4 import BeautifulSoup
      2  sp = BeautifulSoup(html, 'lxml')
      3  print(sp.select('title'))
      4  print(sp.select('p'))
      5  print(sp.select('#p1'))
      6  print(sp.select('.red'))
```

```
[<title>我是網頁標題</title>]
[<p id="p1">我是段落一</p>, <p class="red" id="p2">我是段落二</p>]
[<p id="p1">我是段落一</p>]
[<p class="red" id="p2">我是段落二</p>]
```

GOTOP Python 大數據特訓班

2.2.8 取得標籤的屬性內容

如果要取得回傳值中屬性的內容，可以使用get() 方法或是以字典取值的方式：

```
回傳值 .get(" 屬性名稱 ")
回傳值 [" 屬性名稱 "]
```

定義變數：html，其內容為一個網頁原始碼：

```
[14]  1  html = '''
      2  <html>
      3    <head><meta charset="UTF-8"><title>我是網頁標題</title></hea
      4    <body>
      5        <img src="http://www.ehappy.tw/python.png">
      6        <a href="http://www.e-happy.com.tw">超連結</a>
      7    </body>
      8  </html>
      9  '''
```

以get() 方法和串列元素取得標籤的屬性內容：

```
[15]  1  from bs4 import BeautifulSoup
      2  sp = BeautifulSoup(html, 'lxml')
      3  print(sp.select('img')[0].get('src'))
      4  print(sp.select('a')[0].get('href'))
      5  print(sp.select('img')[0]['src'])
      6  print(sp.select('a')[0]['href'])
```

GOTOP

Python 大數據特訓班

2.2.9 專題：威力彩開獎號碼

以下是台灣彩券(https://www.taiwanlottery.com.tw) 的官方網站，在首頁中會將各種獎項最新一期的得獎號碼全部整理在頁面上，乍看之下內容非常豐富，不過有時要找到想要的資訊就不是那麼容易了！這裡我們將要挑戰用程式把頁面上威力彩的開獎號碼擷取下來，整理後顯示在螢幕上。

GOTOP

Python 大數據特訓班

範例：查詢威力彩開獎號碼

```
[16]  1  import requests
      2  from bs4 import BeautifulSoup
      3  url = 'https://www.taiwanlottery.com.tw/'
      4  r = requests.get(url)
      5  sp = BeautifulSoup(r.text, 'lxml')
      6  # 找到威力彩的區塊
      7  datas = sp.find('div', class_='contents_box02')
      8  # 開獎期數
      9  title = datas.find('span', 'font_black15').text
     10  print('威力彩期數：', title)
     11  # 開獎號碼
     12  nums = datas.find_all('div', class_='ball_tx ball_green')
     13  # 開出順序
     14  print('開出順序：', end=' ')
     15  for i in range(0,6):
     16      print(nums[i].text, end=' ')
     17  # 大小順序
     18  print('\n大小順序：', end=' ')
     19  for i in range(6,12):
     20      print(nums[i].text, end=' ')
     21  # 第二區
     22  num = datas.find('div', class_='ball_red').text
     23  print('\n第二區：', num)
```

GOTOP Python 大數據特訓班

2.3 使用正規表達式

正規表達式(regular expression，簡稱regex)，簡單來說就是用一定的規則處理字串的方法。它能透過一些特殊符號的輔助，讓使用者輕易對於資料內容進行檢查格式或搜尋取代的處理。

2.3.1 正規表達式的使用

檢查行動電話號碼格式

台灣行動電話「xxxx-xxxxxx」的格式，可用如下正規表達式。

```
'\d\d\d\d-\d\d\d\d\d\d'
```

也可以再簡化如下：

```
r'\d{4}-\d{6}'
```

GOTOP Python 大數據特訓班

正規表達式特殊字元表

正規表達式	功能說明	範例
.	代表一個除了換列字元 (\n) 以外的所有字元。	a.c 匹配 **a1c**23 => a1c
^	代表輸入列的開始。	^ab 匹配 **ab**c23 => ab ^ab 匹配 a1c23 => None
$	代表輸入列的結束。	23$ 匹配 a1c**23** => 23 34$ 匹配 a1c23 => None
*	代表前一個項目可以出現 0 次或無限多次。	ac* 匹配 **acc**123 => acc ac* 匹配 **ac**123 => ac
+	代表前一個項目可以出現 1 次或無限多次。	ac+ 匹配 **accc**123 => accc ac+ 匹配 **ac**123 => ac
?	代表前一個項目可以出現 0 次或 1 次。	ac? 匹配 **acc**123 => ac ac? 匹配 **a**123 => a
[abc]	代表符合 a 或 b 或 c 的任何字元。	[abc] 匹配 d12**bc**3 => bc [abc]+ 匹配 d**ab**12**bc**3 => abbc
[a-z]	代表符合 a、b、c ~z 的任何字元。	[a-z]+ 匹配 **cd**12**bc**3 => cdbc
\	代表後面的字元以一般字元處理。	a\+ 匹配 **a+**aaaa => a+

GOTOP Python 大數據特訓班

正規表達式	功能說明	範例
{m}	代表前一個項目必須正好出現 m 數。	a{2} 匹配 **aa**abbb=> aa
{m,}	代表前一個項目出現次數最少 m 次，最多無限次。	a{2,} 匹配 **aaa**bbb => aaa
{m,n}	代表前一個項目出現次數最少 m 次，最多 n 次。	a{2,4} 匹配 **aaaa**abb => aaaa
\d	數字字元，相當於 [0123456789] 或 [0-9]。	\d+ 匹配 a**12**bc => 12
^	反運算，例如：[^a-d] 代表除了 a、b、c、d 以外的所有字元。	[^a-d]+ 匹配 a**12s**bc => 12s
\D	非數字字元，相當於 [^0-9]。	'[\D]+ 匹配 12**cd**34 => cd
\n	換列字元。	
\r	回列首字元 (carriage return)。	
\t	tab 定位字元。	
\s	空白、定位、Tab 鍵、跳列、換頁字元，相當於 [\r\t\n\f]。	[a\sb]+ 匹配 **a b**c => a b
\S	非空白、定位、Tab 鍵、跳列、換頁字元，相當於 [^ \r\t\n\f]。	[a\S]+ 匹配 **a bc** => abc
\w	數字、字母或底線字元，相當於 [0-9a-zA-Z_]。	[\w]+ 匹配 **12bc_AB***% => 12bc_AB
\W	非數字、字母或底線字元，相當於 [^\w]，即 [^0-9a-zA-Z_]。	[\W]+ 匹配 12bc_AB***%** =>*%

GOTOP Python 大數據特訓班

2.3.2 正規表達式的範例

用途途	正規表達式	範例
整數	[0-9]+	33025
浮點數	[0-9]+\.[0-9]+	75.93
英文單字	[A-Za-z]+	Python
變數名稱	[A-Za-z_][A-Za-z0-9_]*	_pointer
Email	[a-zA-Z0-9._-]+@[a-zA-Z0-9\._-]+	guest@mail.com
URL	http://[a-zA-Z0-9\./_-]+	http://e-happy.com.tw/

2.3.3 建立正規表達式物件

```
import re
回傳結果物件 = re.方法(正規表示式, 搜尋字串)
```

GOTOP Python 大數據特訓班

2.3.4 正規表達式物件的方法

方法	說明
match(string)	由字串起頭開始傳回指定字串中符合正規表達式的字串，直到不符合字元為止，並把結果存入 MatchObject 物件中；若無符合字元，傳回 None。
search(string)	傳回指定字串中第一組符合正規表達式的字串，並把結果存入 MatchObject 物件中；若無符合字元會傳回 None。
findall(string)	傳回指定字串中所有符合正規表達式的字串，並傳回一個串列；若無符合字元，傳回空的串列。

match() 方法

```
[21]  1  import re
      2  m = re.match(r'[a-z]+','abc123xyz')
      3  print(m)
```

```
<re.Match object; span=(0, 3), match='abc'>
```

GOTOP Python 大數據特訓班

利用以下方法取得結果，如下：

方法	說明
group()	傳回符合正規表達式的字串，若無符合則傳回 None。
start()	傳回 match 的開始位置。
end()	傳回 match 結束位置。
span()	傳回 (開始位置 , 結束位置) 的元組物件。

在上例中match 物件得到的結果如下：

```
[22]  1  if m != None:
      2      print(m.group())    #abc
      3      print(m.start())    #0
      4      print(m.end())      #3
      5      print(m.span())     #(0, 3)
```

GOTOP Python 大數據特訓班

search() 方法

```
1  import re
2  m = re.search(r'[a-z]+', 'abc123xyz')
3  print(m)    # <re.Match object; span=(0, 3), match='abc'>
4  if m != None:
5      print(m.group())  # abc
6      print(m.start())  # 0
7      print(m.end())    # 3
8      print(m.span())   # (0,3)
```

findall() 方法

```
[24]
1  import re
2  m = re.findall(r'[a-z]+', 'abc123xyz')
3  print(m)    # ['abc', 'xyz']
```

GOTOP Python 大數據特訓班

2.3.5 使用正規表達式取代內容

語法如下：

```
回傳結果 = re.sub( 正規表達式 , 取代字串 , 搜尋字串 , count=0)
```

```
[25]
1  import re
2  result = re.sub(r"\d+", "*", "Password:1234,ID:5678")
3  print(result)   # Password:*,ID:*
```

2.3.6 範例：正規表示式練習

定義變數：html，其內容為一個網頁原始碼。

```
[26]
1  html = """
2  <div class="content">
3      E-Mail：<a href="mailto:mail@test.com.tw">
4        mail</a><br>
5      E-Mail2：<a href="mailto:mail2@test.com.tw">
6        mail2</a><br>
7      <ul class="price">定價：360元 </ul>
8      <img src="http://test.com.tw/p1.jpg">
9      <img src="http://test.com.tw/p2.png">
10     電話：(04)-76543210、0937-123456
11 </div>
12 """
```

GOTOP

Python 大數據特訓班 第三版

讀取正規表達式指定的內容。

```
[20]  1  import re
      2  pattern=r'[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+'
      3  emails = re.findall(pattern,html)
      4  for email in emails: #顯示 email
      5      print(email)
      6
      7  price=re.findall(r'[\d]+元',html)[0].split('元')[0] #價格
      8  print(price) #顯示定價金額
      9
     10  imglist = re.findall(r'[http://]+[a-zA-Z0-9-/.]+\.[jpgpng]+',html)
     11  for img in imglist: #
     12      print(img) #顯示圖片網址
     13
     14  phonelist = re.findall(r'\(?\d{2,4}\)?\-\d{6,8}',html)
     15  for phone in phonelist:
     16      print(phone) #顯示電話號碼
```

```
mail@test.com.tw
mail2@test.com.tw
360
http://test.com.tw/p1.jpg
http://test.com.tw/p2.png
(04)-76543210
0937-123456
```