Python 爬蟲

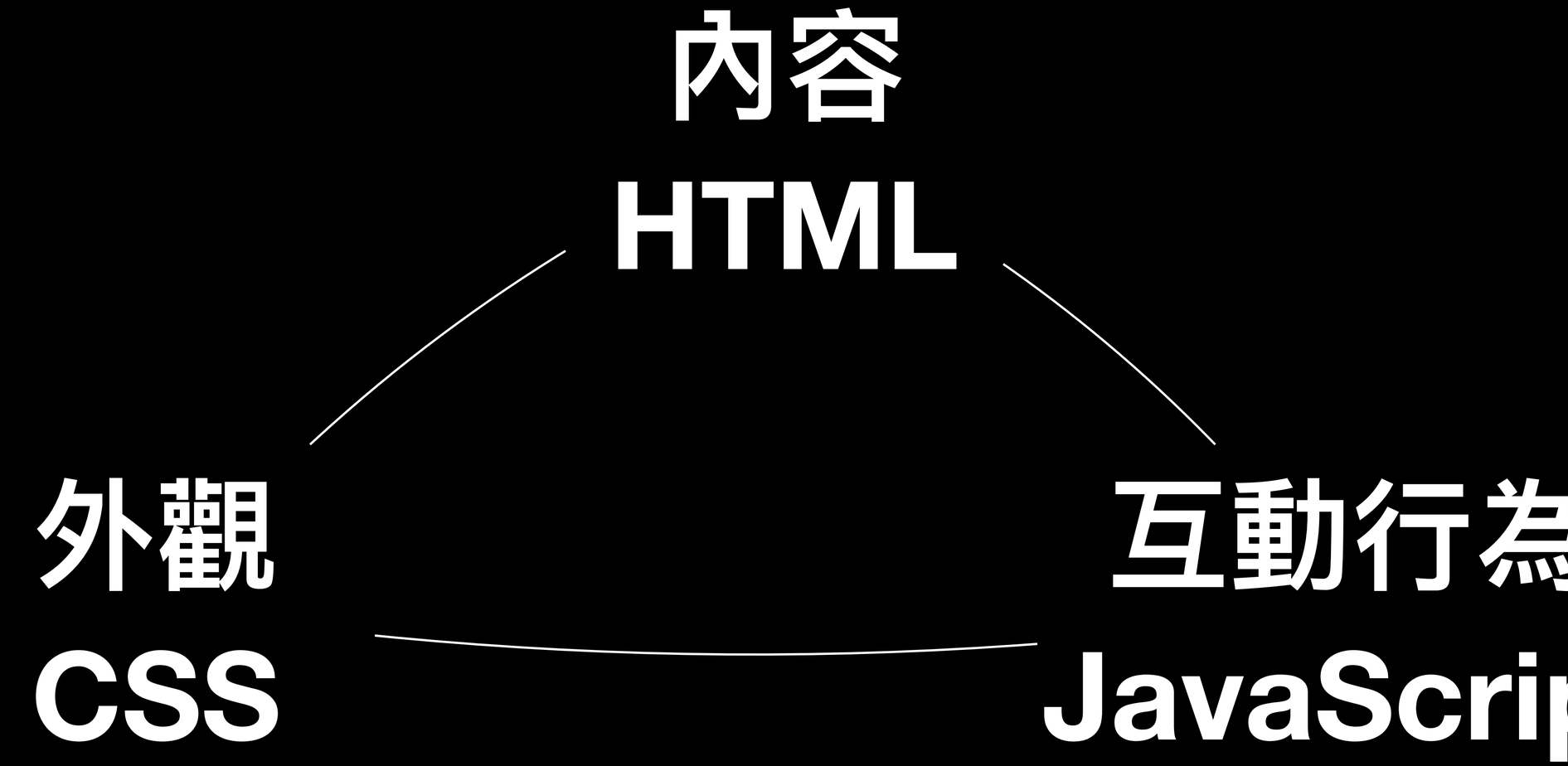
什麼是網路爬蟲

快速抓取網頁上的資料

客戶端 （瀏覽器）	伺服器端 （提供網頁的電腦）
向伺服器請求資料 →	回應客戶端的請求 ↓
將資源解析成使用者 看到的網頁 ←	根據請求回應相對應的資料

如何抓取資料？

網頁是由什麼組成


內容
HTML
外觀
CSS
互動行為
JavaScript

Quick Javascript Switcher

Selenium

如何看一個網站的原始碼？

開啟開發人員工具

按下 F12 或是滑鼠右鍵檢視元素

以 ptt joke 版為例

```
<div id="topbar-container">
    <div id="topbar" class="bbs-content">
        <a id="logo" href="/bbs/">批踢踢實業坊</a>
        <span>&rsaquo;</span>
        <a class="board" href="/bbs/joke/index.html"><span class="board-label">看板 </span>joke</a>
        <a class="right small" href="/about.html">關於我們</a>
        <a class="right small" href="/contact.html">聯絡資訊</a>
    </div>
</div>

<div id="main-container">
    <div id="action-bar-container">
        <div class="action-bar">
            <div class="btn-group btn-group-dir">
                <a class="btn selected" href="/bbs/joke/index.html">看板</a>
                <a class="btn" href="/man/joke/index.html">精華區</a>
            </div>
            <div class="btn-group btn-group-paging">
                <a class="btn wide" href="/bbs/joke/index1.html">最舊</a>
                <a class="btn wide" href="/bbs/joke/index7025.html">&lsaquo; 上頁</a>
                <a class="btn wide disabled">下頁 &rsaquo;</a>
                <a class="btn wide" href="/bbs/joke/index.html">最新</a>
            </div>
        </div>
    </div>

    <div class="r-list-container action-bar-margin bbs-screen">
        <div class="search-bar">
            <form type="get" action="search" id="search-bar">
                <input class="query" type="text" name="q" value="" placeholder="搜尋文章&#x22ef;">
            </form>
```

HTML 基礎

- <標籤名 屬性名1 = “屬性值” 屬性名2 = “屬性值” … 屬性名N = “屬性值”>內容</標籤名>

寫爬蟲的思路

- 取得網頁內容
- 選擇想要取得的資訊
- 透過標籤鎖定目標
- 取出目標標籤
- 處理標籤中的資訊

必要的套件

- import requests
- from bs4 import BeautifulSoup

可能會用到的語法

- Python 基礎語法
- requests 以及 BeautifulSoup 中的套件

Python 基礎語法

- print
- if
- for
- while
- def function()
- list
- .append()
- dict

requests 套件

- requests.get(url)

```
<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Re: [地獄] Kobe和女兒在直升機上的談話 - 看板 joke - 批踢踢實業坊</title>
<meta name="robots" content="all">
<meta name="keywords" content="Ptt BBS 批踢踢">
<meta name="description" content="上帝:Kobe你終於來了，快 快教我打球，你肯定很想教我投籃
Kobe:好，我教你傳球，你只管傳給我就好，我投給你看
上帝:http://i.imgur.com/XcoOKGP.jpg
引述《aa0529 (頁乘)》之銘言：
: 2020年一月的某一天
">
<meta property="og:site_name" content="Ptt 批踢踢實業坊">
<meta property="og:title" content="Re: [地獄] Kobe和女兒在直升機上的談話">
<meta property="og:description" content="上帝:Kobe你終於來了，快 快教我打球，你肯定很想教我投籃
Kobe:好，我教你傳球，你只管傳給我就好，我投給你看
上帝:http://i.imgur.com/XcoOKGP.jpg
引述《aa0529 (頁乘)》之銘言：
: 2020年一月的某一天
">
<link rel="canonical" href="https://www.ptt.cc/bbs/joke/M.1583147270.A.395.html">

<link rel="stylesheet" type="text/css" href="//images.ptt.cc/bbs/v2.27/bbs-common.css">
<link rel="stylesheet" type="text/css" href="//images.ptt.cc/bbs/v2.27/bbs-base.css" media="screen">
<link rel="stylesheet" type="text/css" href="//images.ptt.cc/bbs/v2.27/bbs-custom.css">
<link rel="stylesheet" type="text/css" href="//images.ptt.cc/bbs/v2.27/pushstream.css" media="screen">
<link rel="stylesheet" type="text/css" href="//images.ptt.cc/bbs/v2.27/bbs-print.css" media="print">




        </head>
    <body>

<div id="topbar-container">
        <div id="topbar" class="bbs-content">
                <a id="logo" href="/bbs/">批踢踢實業坊</a>
                <span>&rsaquo;</span>
```

requests 範例

```
response = requests.get(url)
# 以 GET 傳請求給目標伺服器，伺服器回傳 response 物件
# response 接收回傳值
print(response.text)
# 輸出網頁原始碼
```

BeautifulSoup 套件

- soup = BeautifulSoup(response.text , ‘html.parser’)
- soup.find_all(‘標籤’ , ‘屬性’＝’屬性值’) (屬性跟屬性值可加可不加)
- soup.find(‘標籤’ , ‘屬性’=‘屬性值’) (屬性跟屬性值可加可不加)
- .get(‘屬性’)
- .get_text()

BeautifulSoup(response.text , ‘html.parser’)

- 以 html.parser 為解析器解析 response.text 的內容 存入 soup 中
- response.text 為網站原始碼

soup.find_all(‘標籤’ , ‘屬性’=’屬性值’)

- content = soup.find_all(‘div’ , class_=‘title’)
- 找尋所有標籤為 div 且 class 為 title 的所有目標 存入 content
- class_ 是因為避免跟保留字 class 衝突

soup.find(‘標籤’，‘屬性’=‘屬性值’)

- content = soup.find(‘a’)
- 找尋第一個標籤為 a 的目標 存入 content

.get(‘屬性’)

- content = soup.find(‘a’)
- content.get(‘href’)
- 在 content 中抓取屬性為 href 中的資料

.get_text()

- content = soup.find(‘a’)
- content.get_text()
- 抓取標籤 a 中的所有內容

HTML 解析器

解析器	使用方法	優點	缺點
Python’s html.parser	BeautifulSoup(markup , ‘html.parser’)	• Python 本身就有 • 速度中等	• 不能很好的兼容 • before Python 2.7.3 or 3.2.2
lxml’s HTML parser	BeautifulSoup(markup , ‘lxml’)	• 速度快 • 兼容性好	• 須安裝 C 語言的函式庫
lxml’s XML parser	BeautifulSoup(markup , ‘xml’)	• 速度快 • 支持 XML 解析器	• 須安裝 C 語言的函式庫
html5lib	BeautifulSoup(markup , ‘html5lib’)	• 兼容性好 • 以瀏覽器方式解析文件 • 生成 html5 格式文件	• 速度慢 • 須安裝 html5lib

開始實作吧～～～


感謝聆聽!