長榮大學
Chang Jung Christian University

網路爬蟲的基礎

互動設計系

黃詰琳 副教授

August 8, 2021

Department of Interaction Design
1 互動設計學系

長榮大學
Chang Jung Christian University

講者介紹

黃 詰 琳 長榮大學.互動設計學系 系主任

- 長榮大學.資訊工程學系 副教授
- 學歷
  - 成功大學 電機(電通所)博士
  - 成功大學 電機(計算機組)碩士
  - 台灣科技大學 電機學士(二技)
  - 台北工專 電機科(二專)
  - 高雄高工 控制科

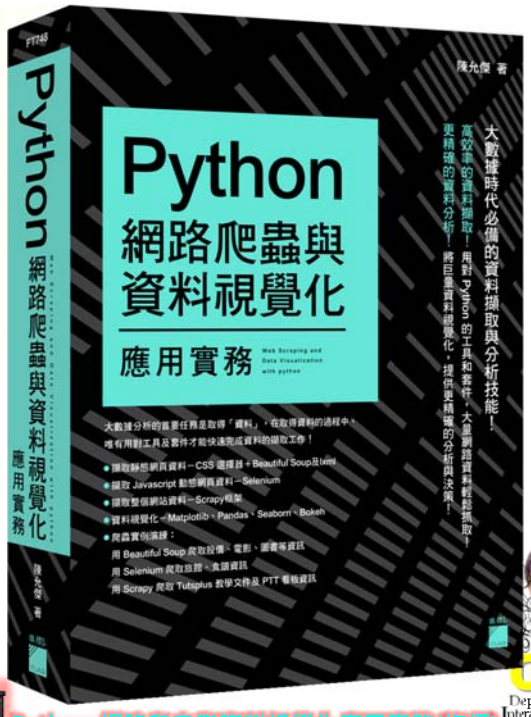
2
Department of Interaction Design
互動設計學系

講者介紹

- 專長
  - 視訊影像處理、數位影像處理、射頻識別技術、無線感測網路、3D建模、 3D列印、 VR技術
- 相關研究計畫
  - 研發基於智慧計算之無人機飛行策略以應用於人機協同捕捉作業(2020)
  - 先進智慧計算技術為基之生物辨識與生態數量控制之整合系統(2020)
  - 互動式3D醫學影像顯示、建模與列印系統之研發(2015)
  - 應用形狀基礎的等位函數法於3D列印(2014)
  - 醫學影像主體切割技術之研究 - 以乳房腫瘤為例(2014)
  - 應用紋理分析自動選取ROI於乳房MR 影像分割(2012)
  - 運用多頻譜梯度向量流場輪廓法於乳房腫瘤立體模型重建與定位(2012)

參考書籍: Python 網路爬蟲與資料視覺化應用實務

- 類別：程式設計/資料庫
- 作者：陳允傑
- 書號：FT748
- ISBN：9789863125624
- 色彩：套色



網路爬蟲的基礎

- 認識HTML標示語言
- HTML5網頁的結構
- 認識JSON
- JSON的語法
- 認識網路爬蟲
- 為什麼需要網路爬蟲
- 網路爬蟲的基本步驟
- 網路爬蟲使用的相關技術
- 使用瀏覽器瀏覽網頁的步驟
- Python網路爬蟲的相關函式庫

認識HTML標示語言

- 「HTML」（HyperText Markup Language）語法是源於SGML語言，「SGML」（Standard Generalized Markup Language）是一種功能強大的文件標示、管理和編排語言。
- HTML標示語言（Markup Lanugae）是Tim Berners-Lee在1991年建立，1993年 HTML 1.0 版由Berners-Lee和Connolly完成，經過3.2版到HTML 4.01版，目前的最新版本是HTML5，這是一種文件內容的格式編排語言，不像SGML允許定義如何標示文件的標籤。

認識HTML標示語言

- HTML使用SGML慣用語法，即標籤和屬性，如下所示：
  - 標籤（Tags）：HTML標籤是一個字串符號，可以用來標示文字內容需套用的編排格式，例如：在 開頭標籤和 結尾標籤之中的文字內容，就是使用預設格式編排成一個文字段落，如下所示：

    ```
    <p>這是一個測試網頁</p>
    ```

  - 屬性（Attributes）：HTML標籤擁有一些屬性來定義細部編排，例如：<img>標籤的src、width和height屬性，可以指定顯示的圖檔和尺寸的寬與高，如下所示：

    ```
    <img src="sample.jpg" width="20" height="30" >
    ```

長榮大學 Chang Jung Christian University

認識HTML標示語言

- 「XML」（Extensible Markup Language）可擴展標示語言也是一種標籤語言，XML 1.0版規格在1998年2月正式推出，其寫法十分類似HTML，繼承SGML自定標籤的優點，並且刪除一些SGML複雜的部分，在功能上能夠補足HTML標籤的不足，並且擁有更多的擴充性。
- 請注意！XML不是用來編排內容，而是描述資料，因此，XML沒有HTML一般的預設標籤，使用者需要自行定義描述資料所需的各種標籤。

認識HTML標示語言

- HTML5仍然遵循HTML 4.01標籤的語法，只是擴充、改進HTML標籤和API（Application Programming Interfaces）來建立複雜的Web應用程式，和處理DOM（Document Object Model）。不只如此，HTML5支援手機和平板電腦等低耗電的行動裝置，可以建立跨平台的Mobile應用程式。目前Internet Explorer（Edge）、Firefox、Safari、Chrome和Opera等瀏覽器都已經支援HTML5。.

HTML5網頁的結構

- HTML5網頁和HTML 4.x和XHTML網頁的標籤結構十分相似，其基本的標籤結構，如下所示：

```
<!DOCTYPE html>
<html lang="zh">
<head>
<meta charset="utf-8">
<title>網頁標題文字</title>
</head>
<body>
網頁內容
</body>
</html>
```

HTML5網頁的結構<!DOCTYPE>

- <!DOCTYPE>位在<html>標籤前，它並不是HTML標籤，其目的是告訴瀏覽器使用的HTML版本，以便瀏覽器使用正確引擎來產生HTML網頁內容。

HTML5網頁的結構<html>

- <html>標籤是HTML網頁的根元素，一個容器元素，其內容是其他HTML標籤，擁有<head>和<body>兩個子標籤。如果需要，<html>標籤可以使用lang屬性指定網頁使用的語言，如下所示：

```
<html lang="zh-TW">
```

- 上述標籤的lang屬性值，常用2碼值有：zh（中文）、en（英文）、fr（法文）、de（德文）、it（義大利文）和ja（日文）等。lang屬性值也可以加上「-」分隔的2碼國家或地區，例如：en-US是美式英文、zh-TW是台灣的正體中文等。

HTML5網頁的結構<head>

- <head>標籤的內容是標題元素,包含<title>、<meta>、<script>和<style>標籤。例如: <meta>標籤可以指定網頁的編碼為utf-8,如下所示:
<meta charset="utf-8">

HTML5網頁的結構<body>

- <body>標籤才是真正編排的網頁內容，包含文字、超連結、圖片、表格、清單和表單等網頁內容，詳見書附HTML電子書。

HTML5網頁的結構

- 使用HTML5標籤建立簡單的HTML網頁內容：

```
01: <!DOCTYPE html>
02: <html lang="zh-TW ">
03: <head>
04: <meta charset="utf-8"/>
05: <title>HTML5網頁</title>
06: </head>
07: <body>
08: <h3>HTML5網頁</h3>
09: <hr/>
10: <p>第一份HTML5網頁</p>
11: </body>
12: </html>
```

認識JSON
說明

- 「JSON」的全名為（JavaScript Object Notation），這是一種類似XML的資料交換格式，事實上，JSON就是JavaScript物件的文字表示法，其內容只有文字（Text Only）。

- JSON是由Douglas Crockford創造的一種輕量化資料交換格式，因為比XML來的快速且簡單，JSON資料結構就是JavaScript物件文字表示法，不論是JavaScript語言或其他程式語言都可以輕易解讀，這是一種和語言無關的資料交換格式。

認識JSON
為什麼使用JSON

- 因為JSON格式就是文字內容，可以很容易在客戶端和伺服端之間傳送資料，現在JSON已經取代XML成為非同步瀏覽器與伺服器之間通訊使用的資料交換格式，不只如此，很多網路公司也都支援REST API，可以取得JSON格式的資料，換句話說，我們取得的網路資料，除了自行從HTML標籤取得，也可以透過AJAX下載JSON格式文件。

認識JSON
JSON文件的內容

- JSON是一種可以自我描述和容易了解的資料交換格式，使用大括號定義成對的鍵和值（Key-value Pairs），相當於物件的屬性和值，類似Python語言的字典和清單，如下所示：

```
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3",
  ...
}
```

JSON的語法
JSON的語法規則

- JSON語法並沒有關鍵字，其基本語法規則，如下所示：
  - 資料是成對的鍵和值（Key-value Pairs），使用「:」符號分隔。
  - 資料之間是使用「,」符號分隔。
  - 使用大括號定義物件。
  - 使用方括號定義物件陣列。

JSON的語法
JSON的鍵和值

- JSON資料是成對的鍵和值（Key-value Pairs），首先是欄位名稱，接著「:」符號，再加上值，如下所示：

```
"author": "陳會安“
```

- 上述"author"是欄位名稱，"陳會安"是值，JSON的值可以是整數、浮點數、字串（使用「"」括起）、布林值（true或false）、陣列（使用方括號括起）和物件（使用大括號括起）。

JSON的語法
JSON物件

- JSON物件是使用大括號包圍的多個JSON鍵和值，如下所示：

```
{
  "title": "C語言程式設計",
  "author": "陳會安",
  "category": "Programming",
  "pubdate": "06/2018",
  "id": "P101"
}
```

JSON的語法
JSON物件陣列

- JSON物件陣列可以擁有多個JSON物件，例如："Employees"欄位的值是一個物件陣列，擁有3個JSON物件，如下所示：

```
{
  "Boss": "陳會安",
  "Employees": [
    { "name" : "陳允傑", "tel" : "02-22222222" },
    { "name" : "江小魚", "tel" : "02-33333333" },
    { "name" : "陳允東", "tel" : "04-44444444" }
  ]
}
```

認識網路爬蟲
什麼是網路爬蟲

- 網路爬蟲（Web Crawler）或稱為網路資料擷取（Web Data Extraction）是一種資料擷取技術，可以讓我們直接從Web網站的HTML網頁取出所需的資料，其過程包含與Web資源進行通訊，剖析文件取出所需資料和將資料整理成資訊，即轉換成所需的資料格式。
- 網路爬蟲（Web Crawler）是一種針對目標Web網站自動擷取資訊的技術，雖然我們可以手動自行使用複製和貼上方式來收集和擷取資訊，但是透過網路爬蟲，就可以自動幫助我們收集和擷取資訊。

認識網路爬蟲
什麼不是網路爬蟲

- 請注意！並不是從網路取得資料都稱為網路爬蟲，如果取得資料已經是機器可讀取的資料，這些操作並不是網路爬蟲，例如：
  - 從網站下載資料檔：有些網站已經提供現成結構化資料的檔案可供下載，例如：Excel檔案、CSV檔案或JSON和XML檔案等。
  - 應用程式介面API：很多公司都會提供Web基礎的API介面，例如：REST API，我們可以透過REST API來下載結構化資料，例如：JSON或XML資料。

認識網路爬蟲
網路爬蟲的用途

- 線上商店可以周期使用網路爬蟲取得競爭者的商品價格，並且使用取得資訊來即時調整商品價格。
- 使用網路爬蟲從相關網路取得指定商品價格、旅館房間價格、機票價格等各種產品和服務的價格，輕鬆建立比價資訊。
- 使用網路爬蟲從網路取得各類徵才資訊和產品評論等資訊。
- 從社群網站使用網路爬蟲取得使用者評價、流行趨勢和熱門話題。

認識網路爬蟲
網路爬蟲的用途

- 用網路爬蟲從網路取得和收集電子郵件地址來進行網路行銷。
- 從房地產網站用網路爬蟲取得相關資訊來追蹤房地產的趨勢。
- 從股票資訊網站使用網路爬蟲取得相關股票資訊來追蹤股價趨勢，進而規劃投資策略。

為什麼需要網路爬蟲

- 網路爬蟲的主要工作就是從HTML網頁內容取出所需的資料，我們當然可以自行使用瀏覽器瀏覽網頁後，使用複製與貼上功能來手動取得這些資料，問題是你準備花多久的時間來收集這些資料。
- 手動取得網頁資料如果數量不大，我們並不需花費多少時間即可完成資料的取得，問題是如果有上百本圖書，我們就需要使用Python爬蟲程式。

網路爬蟲的基本步驟

- **Step 1**：識別出目標URL網址
  - 識別出目標Web資源的URL網址，可能只有一個，也可以是一組URL網址。
- **Step 2**：送出HTTP請求取得HTML網頁
  - 使用Python函式庫送出HTML請求來取回HTTP回應的HTML網頁。
- **Step 3**：分析HTML網頁
  - 使用相關視覺化工具在HTML網頁定位所需資料，並且分析如何搜尋和走訪至此標籤來取出資料。

長榮大學 Chang Jung Christian University

網路爬蟲的基本步驟

- **Step 4**：剖析HTML網頁
  - 使用Python函式庫剖析（Parse）回應文件的HTML網頁，可以建立成樹狀結構的標籤物件集合。
- **Step 5**：從剖析網頁取出所需資料
  - 我們可以透過搜尋或走訪方式來取出所需資料，在整理成指定格式後，儲存成CSV或JSON檔案。

網路爬蟲使用的相關技術

- 網路爬蟲涉及向Web網站送出HTTP請求，和在取回的HTML網頁中定位出所需的資料，在取出資料後，我們需要儲存這些資料，所以網路爬蟲需要使用的相關技術，如下所示：
  - 使用HTTP通訊協定送出HTTP請求。
  - 剖析HTML文件來定位網頁資料。
  - 將取得的資料儲存成指定的檔案格式。

使用瀏覽器瀏覽網頁的步驟

- Step 1：我們在瀏覽器輸入URL網址是用來搜尋指定的Web伺服器，也就是向Web伺服器送出HTTP請求，使用的是HTTP通訊協定。
- Step 2：Web伺服器依據HTTP請求來回應內容至瀏覽器，通常是HTML網頁，也有可能是XML或JSON檔案。
- Step 3：瀏覽器在接收到伺服器回應的HTML網頁後，就會將文件內容剖析建立成內部的樹狀結構，每一個HTML標籤是一個節點，這就是DOM（Document Object Document）。
- Step 4：瀏覽器依據DOM產生內容, 這就是我們看到的網頁內容。

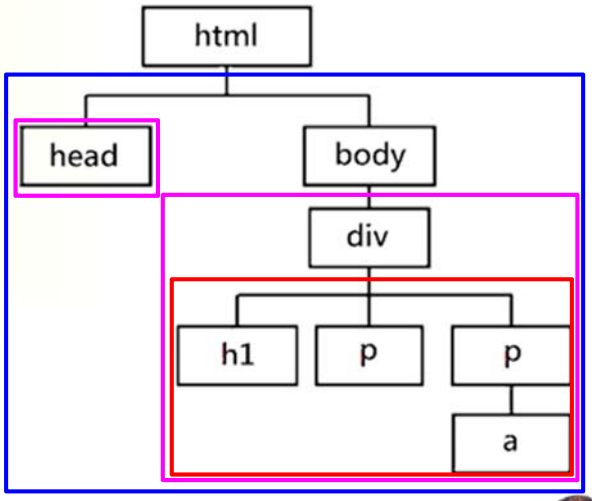
使用瀏覽器瀏覽網頁的步驟

Step 1: 輸入URL網址:
http://example.com

Step 2: 回傳HTML網頁的標籤內容

```
<html>
<head></head>
<body>
  <div>
    <h1></h1>
    <p></p>
    <p><a></p>
  </div>
</body>
</html>
```

Step 3: 剖析建立DOM節點樹



Step 4: 在瀏覽器顯示
產生的網頁內容

Example Domain

This domain is for use in illustrative examples in documents. You may use this domain in literature without prior coordination or asking for permission.

More information...

Python網路爬蟲的相關函式庫

- **Python網路爬蟲的整個過程需要使用多種工具和函式庫（Python語言就是模組和套件）來完成整個資料取得和擷取的工作，如下所示：**
  - **網路爬蟲工具**：最常使用的是瀏覽器內建的開發人員工具，可以幫助我們在HTML網頁定位出資料的所在，和找出取出此資料的特徵。例如：標籤名稱和屬性值，除此之外，一些Chrome擴充功能更是網路爬蟲不可缺的好工具，例如：Selector Gadget和XPath Helper。
  - **HTTP函式庫**：與Web伺服器進行HTTP通訊的函式庫，以便取得回應文件的HTML網頁內容，例如使用Requests。

Python網路爬蟲的相關函式庫

- **網路爬蟲函式庫**：在取得回應的HTML網頁內容後，我們需要使用函式庫來剖析文件，以便取出所需的資料，如下所示：
  - **爬取靜態網頁**：對於使用HTML標籤建立的網頁內容，例如使用Beautiful Soup和lxml來爬取網頁內容。
  - **爬取動態網頁**：如果Web網站是JavaScript產生的動態網頁內容，我們需要使用Selenium自動瀏覽器工具，也稱為WebDriver，可以幫助我們進行動態網頁的資料爬取。
  - **爬取整個網站**：如果並非單純爬取幾頁HTML網頁，而是爬取整個Web網站的內容，我們需要使用Scrapy網路爬蟲框架來幫助我們建立Python爬蟲程式。

網頁爬蟲

互動設計系

黃詰琳 副教授

August 8, 2021

網頁爬蟲

- 網路連線
- 公開資料
- Web Crawler 基本篇
- Web Crawler - Cookie
- Web Crawler – AJAX

- 參考網站

彭彭的課程 YouTube TW

- https://www.youtube.com/c/彭彭的課程
- https://training.pada-x.com/彭彭的課程教學

網頁爬蟲 - 參考網站

- 參考網站

彭彭的課程 YouTube TW

- https://www.youtube.com/c/彭彭的課程
- https://training.pada-x.com/彭彭的課程教學

彭彭的課程
8.45萬 位訂閱者

首頁 影片 播放清單 社群 頻道 簡介

長榮大學
Chang Jung Christian University

網頁爬蟲

網路連線

Department of
Interaction Design
37 互動設計學系

長榮大學
Chang Jung Christian University

網路連線、公開資料串接

網路連線

載入模組

```
import urllib.request
```

下載特定網址資料

```
import urllib.request as request
with request.urlopen(網址) as response:
    data=response.read()
print(data)
```

長榮大學
Chang Jung Christian University

網頁爬蟲

公開資料

Department of
Interaction Design
41 互動設計學系

長榮大學
Chang Jung Christian University

網路連線、公開資料串接

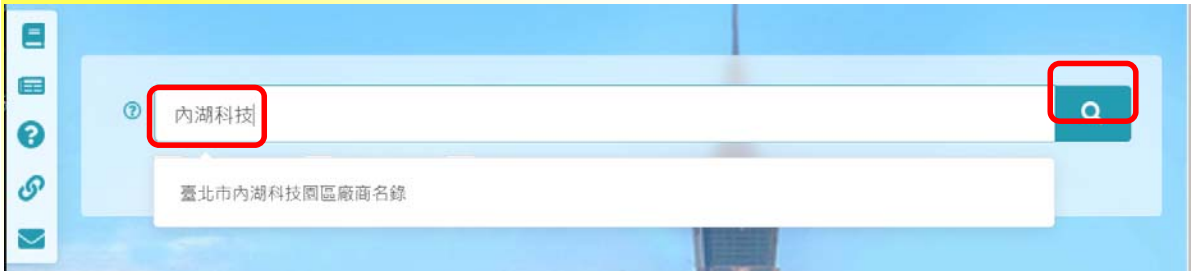
公開資料

適合的資料來源

台北市政府公開資料http://data.taipei/

公開資料下載

http://data.taipei/



資料名稱	資料集描述	更新時間	使用次數/瀏覽次數
臺北市科技工業園區	臺北市科技工業園區時間數列統計資料	2020-10-15	151次
臺北市內湖科技園區廠商名錄	臺北市內湖科技園區廠商資料。	2020-10-14	24803次

公開資料下載

臺北市內湖科技園區廠商名錄 OD ☆☆☆☆☆ 平均0 (0人/次投票) 分享：

加入收藏

資料項目

臺北市內湖科技園區廠商名錄 智慧繪圖 下載 (12668次) API 預覽

詮釋資料

主題分類	經濟

API位址與使用方式

API位址	https://data.taipei/api/v1/dataset/296acfa2-5d93-4706-ad58-e83cc951863c?scope=resourceAquire
異動時間	2020-10-14 16:35:14
API使用方式	GET

發布時間 2012-01-31

長榮大學 Chang Jung Christian University

公開資料下載

{"result":{"limit":1000,"offset":0,"count":5595,"sort":"","results":[{"公司地址":"114臺北市內湖區內湖路1段250號12
樓","ADDR_X":"307169","ADDR_Y":"2774976","統編":"53777727","_id":1,"公司名稱":"奧嫣顧問有限公司"},{"公司地址":"114臺北市內湖區洲子街
樓","ADDR_X":"307674","ADDR_Y":"2774787","統編":"53778036","_id":2,"公司名稱":"有氧生活家股份有限公司"},{"公司地址":"114臺北市內湖區
樓","ADDR_X":"307528","ADDR_Y":"2774767","統編":"53778275","_id":3,"公司名稱":"雲普科技股份有限公司"},{"公司地址":"114臺北市內湖區
樓","ADDR_X":"308249","ADDR_Y":"2773229","統編":"53778789","_id":4,"公司名稱":"詣漢國際有限公司"},{"公司地址":"114臺北市內湖區麗宗
樓","ADDR_X":"308104","ADDR_Y":"2773570","統編":"53778937","_id":5,"公司名稱":"鑽鉅建設股份有限公司"},{"公司地址":"114臺北市內湖區
2","ADDR_X":"307474","ADDR_Y":"2774728","統編":"53789651","_id":6,"公司名稱":"苗林實業有限公司"},{"公司地址":"114臺北市內湖區堤頂大
1","ADDR_X":"307699","ADDR_Y":"2774272","統編":"53792733","_id":7,"公司名稱":"大鴻生物科技股份有限公司"},{"公司地址":"114臺北市內湖
樓","ADDR_X":"307444","ADDR_Y":"2774860","統編":"53802681","_id":8,"公司名稱":"瑞普光電有限公司"},{"公司地址":"114臺北市內湖區堤頂
樓","ADDR_X":"307021","ADDR_Y":"2774612","統編":"53823139","_id":9,"公司名稱":"永梁股份有限公司"},{"公司地址":"114臺北市內湖區瑞光
樓","ADDR_X":"307737","ADDR_Y":"2774460","統編":"53823428","_id":10,"公司名稱":"永鷹有限公司"},{"公司地址":"114臺北市內湖區洲子街18
樓","ADDR_X":"307245","ADDR_Y":"2774923","統編":"53827992","_id":11,"公司名稱":"傳奇亞太科技有限公司"},{"公司地址":"114臺北市內湖區
號","ADDR_X":"308354","ADDR_Y":"2773884","統編":"53866615","_id":12,"公司名稱":"域相墨湧設計事業有限公司"},{"公司地址":"114臺北市內
樓","ADDR_X":"307996","ADDR_Y":"2773836","統編":"53901198","_id":13,"公司名稱":"鼐宇通運有限公司"},{"公司地址":"114臺北市內湖區陽光
3","ADDR_X":"308372","ADDR_Y":"2773918","統編":"53909966","_id":14,"公司名稱":"晉生企業有限公司"},{"公司地址":"114臺北市內湖區瑞光
樓","ADDR_X":"307110","ADDR_Y":"2774781","統編":"53913053","_id":15,"公司名稱":"捷聯室內裝修設計有限公司"},{"公司地址":"114臺北市內
樓","ADDR_X":"307221","ADDR_Y":"2774926","統編":"53913146","_id":16,"公司名稱":"商邦餐飲有限公司"},{"公司地址":"114臺北市內湖區陽光
5","ADDR_X":"308252","ADDR_Y":"2773991","統編":"53913899","_id":17,"公司名稱":"宜卡遠國際有限公司"},{"公司地址":"114臺北市內湖區內
2","ADDR_X":"307746","ADDR_Y":"2774850","統編":"53913954","_id":18,"公司名稱":"興又昌股份有限公司"},{"公司地址":"114臺北市內湖區洲
1","ADDR_X":"307678","ADDR_Y":"2774744","統編":"53914335","_id":19,"公司名稱":"樂快股份有限公司"},{"公司地址":"114臺北市內湖區瑞光
樓","ADDR_X":"307180","ADDR_Y":"2774779","統編":"53914721","_id":20,"公司名稱":"遠雛科技有限公司"},{"公司地址":"114臺北市內湖區麗宗
樓","ADDR_X":"307998","ADDR_Y":"2773824","統編":"53914877","_id":21,"公司名稱":"旭東國際通運有限公司"},{"公司地址":"114臺北市內湖區
樓","ADDR_X":"307413","ADDR_Y":"2774628","統編":"53916093","_id":22,"公司名稱":"台灣威亞數位科技股份有限公司"},{"公司地址":"114臺北
37號3樓","ADDR_X":"308289","ADDR_Y":"2773542","統編":"53916262","_id":23,"公司名稱":"聚達系統股份有限公司"},{"公司地址":"114臺北市
樓","ADDR_X":"307474","ADDR_Y":"2774946","統編":"53916586","_id":24,"公司名稱":"永衡開發投資股份有限公司"},{"公司地址":"114臺北市內
樓","ADDR_X":"308077","ADDR_Y":"2773750","統編":"53917543","_id":25,"公司名稱":"中意整合行銷有限公司"},{"公司地址":"114臺北市內湖區
樓","ADDR_X":"307151","ADDR_Y":"2774760","統編":"53917733","_id":26,"公司名稱":"台灣長愛企業有限公司"},{"公司地址":"114臺北市內湖區
樓","ADDR_X":"308114","ADDR_Y":"2773568","統編":"53920456","_id":27,"公司名稱":"捷力邦科技股份有限公司"},{"公司地址":"114臺北市內湖
1","ADDR_X":"308013","ADDR_Y":"2774021","統編":"53921758","_id":28,"公司名稱":"瘋狂娛樂有限公司"},{"公司地址":"114臺北市內湖區內湖
樓","ADDR_X":"307574","ADDR_Y":"2774863","統編":"53922996","_id":29,"公司名稱":"禾麟數碼科技股份有限公司"},{"公司地址":"114臺北市內
2樓","ADDR_X":"308357","ADDR_Y":"2773493","統編":"53924376","_id":30,"公司名稱":"新悅開發建設股份有限公司"},{"公司地址":"114臺北市內

長榮大學 Chang Jung Christian University

確認資料格式

JSON、CSV、或其他格式

長榮大學 Chang Jung Christian University

解讀 JSON 格式

使用內建的 json 模組

範例程式

```
#網路連線
import urllib .request as request
src="https://www.cjcu.edu.tw/"
with request.urlopen(src)as response:
    data=response.read() #取得長榮大學網站的原始碼(HTML、CSS、JASON)
print(data)

with request.urlopen(src)as response:
    data=response.read().decode("utf-8") #utf-8解碼
print(data)
```

範例程式

```
#串接、擷以公開資料
import urllib .request as request
import json#使用JSON格式讀取
src="https://data.taipei/api/v1/dataset/296acfa2-5d93-4706-ad58-e83cc951863c?scope=resourceAquire"
with request.urlopen(src) as response:
    data=json.load(response)#使用JSON格式讀取
print(data)

#將公司名稱列表出來
clist = data["result"]["results"]
with open("data.txt","w",encoding="utf-8") as file:
    for company in clist:
        file.write(company["公司名稱"]+"\n")
```

長榮大學
Chang Jung Christian University

網頁爬蟲

Web Crawler
基本篇

Department of
Interaction Design
51 互動設計學系

長榮大學
Chang Jung Christian University

基本流程

1. 連線到特定網址, 抓取資料
2. 解析資料, 取得實際想要的部份

Web Crawler 基本篇

抓取資料

關鍵心法

盡可能讓程式模仿一個普通使用者的樣子

Web Crawler 基本篇

解析資料

長榮大學 Chang Jung Christian University

JSON 格式資料

使用內建的 json 模組即可

HTML 格式資料

```
<html>
    <head>
        <title>HTML 格式</title>
    </head>
    <body>
        <div class="list">
            <span>階層結構</span>
            <span>樹狀結構</span>
        </div>
    </body>
</html>
```


長榮大學 Chang Jung Christian University

HTML 格式資料

使用第三方套件BeautifulSoup來做解析

HTML 格式資料

未安裝 BeautifulSoup 套件而 import bs4

```
import bs4 #解譯HTML格式的套件 - BeautifulSoup
root=bs4.BeautifulSoup(data, "html.parser") #讓BeautifulSoup協助解析HTML格式文件
print(root)
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-14-67d06d40cf5f> in <module>
----> 1 import bs4 #解譯HTML格式的套件
      2 root=bs4.BeautifulSoup(data, "html.parser") #讓BeautifulSoup協助解析HTML
格式文件
      3 print(root)

ModuleNotFoundError: No module named 'bs4'
```

```
ModuleNotFoundError: No module named 'bs4'
```

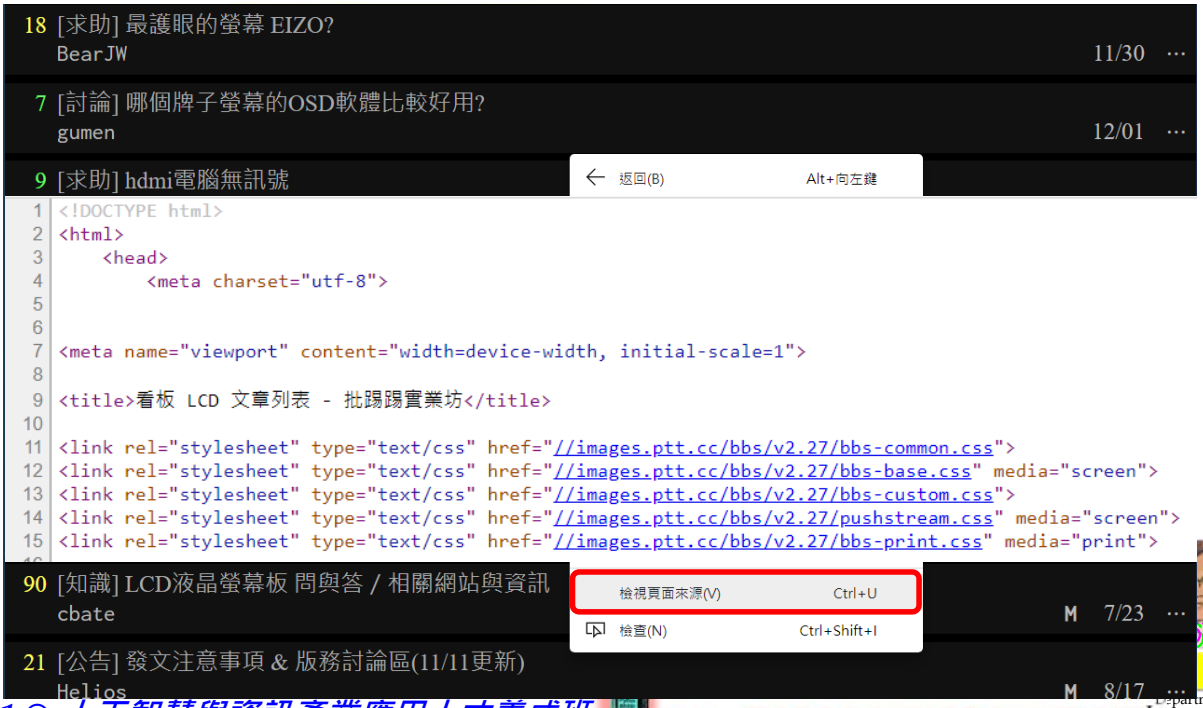
Web Crawler 基本篇

安裝套件

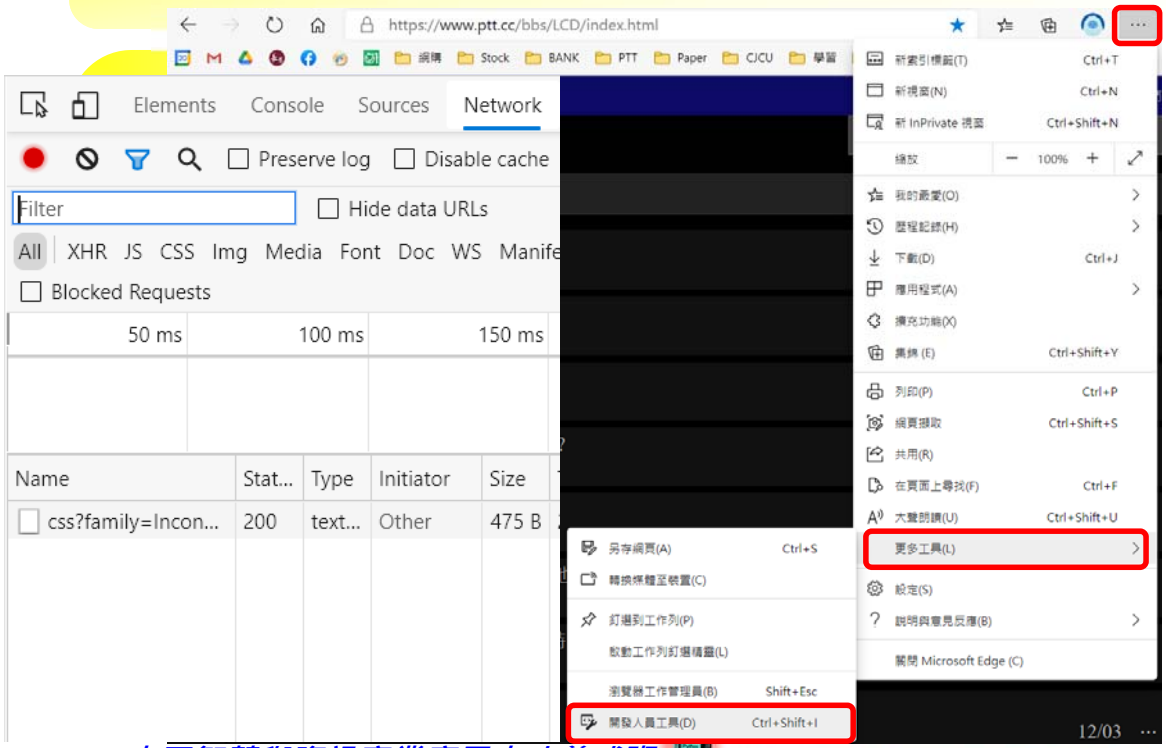
BeautifulSoup

pip install beautifulsoup4

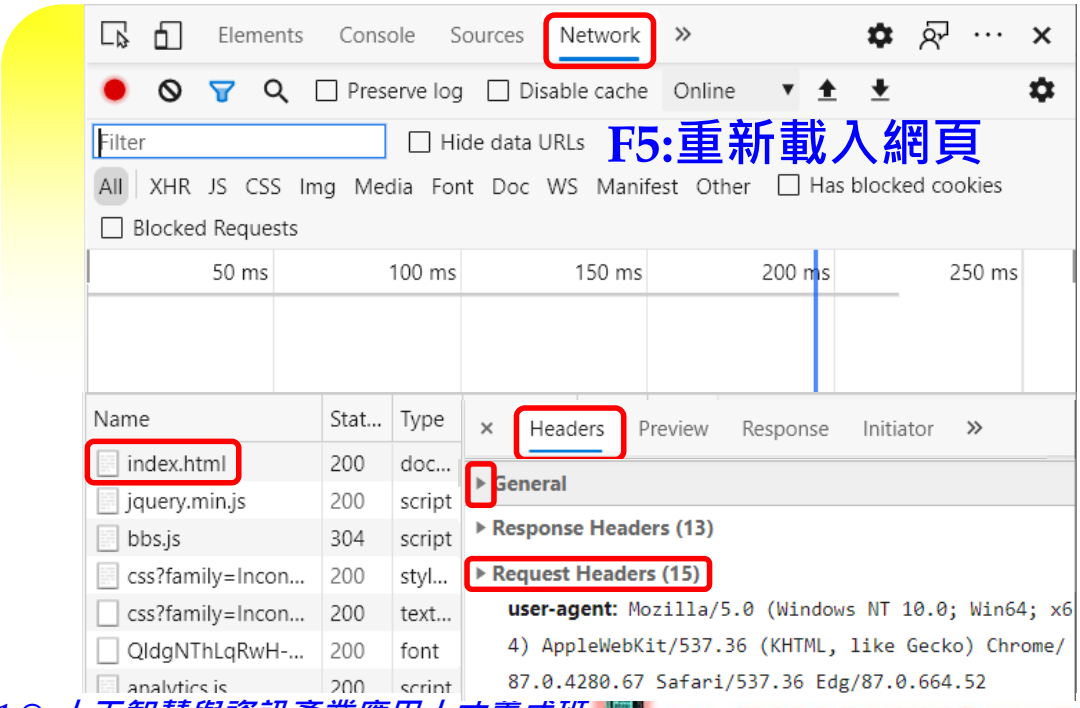
顯示網頁原始碼



進入開發人員工具(F12)



取得 user-agent: 資料



範例程式

```
#抓取PTT的原始碼
import urllib.request as request
src="https://www.ptt.cc/bbs/Lifeismoney/index.html"
with request.urlopen(src) as response:
    data=response.read().decode("utf-8")
print(data)  #HTTPError: HTTP Error 403: Forbidden
```

PTT會驗證"User-Agent",未通證驗證無法直接爬到資料

➔爬不到資料會產生

HTTPError: HTTP Error 403: Forbidden

範例程式

```
#抓取PTT省錢版網頁的原始碼
import urllib.request as request
src="https://www.ptt.cc/bbs/Lifeismoney/index.html"
requestUA=request.Request(src, headers={
   "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.193 Safari/537.36 Edg/86.0.622.68"
})
with request.urlopen(requestUA) as response:
   data=response.read().decode("utf-8")
print(data)
```

觀察爬回來的資料

```
<!DOCTYPE html>

<html>
<head>
<meta charset="utf-8"/>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<title>看板 Lifeismoney 文章列表 - 批踢踢實業坊</title>
<link href="//images.ptt.cc/bbs/v2.27/bbs-common.css" rel="stylesheet" type="text/css"/>
<link href="//images.ptt.cc/bbs/v2.27/bbs-base.css" media="screen" rel="stylesheet" type="text/css"/>
<link href="//images.ptt.cc/bbs/v2.27/bbs-custom.css" rel="stylesheet" type="text/css"/>
<link href="//images.ptt.cc/bbs/v2.27/pushstream.css" media="screen" rel="stylesheet" type="text/css"/>
<link href="//images.ptt.cc/bbs/v2.27/bbs-print.css" media="print" rel="stylesheet" type="text/css"/>
</head>
<body
<div id="topbar-container">
<div class="bbs-content" id="topbar">
<a href="/bbs/" id="logo">批踢踢實業坊</a>
<span>›</span>
<a class="board" href="/bbs/Lifeismoney/index.html"><span class="board-label">看板 </span>Lifeismoney</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/Lifeismoney/index.html">看板</a>
<a class="btn" href="/man/Lifeismoney/index.html">精華區</a>
```

HTML格式

範例程式

```
#抓取PTT省錢版網頁的原始碼
import urllib.request as request
src="https://www.ptt.cc/bbs/Lifeismoney/index.html"
requestUA=request.Request(src, headers={
   "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.193 Safari/537.36 Edg/86.0.622.68"
})
with request.urlopen(requestUA) as response:
   data=response.read().decode("utf-8")
import bs4
root=bs4.BeautifulSoup(data, "html.parser")
print(root)
print(root.title)#印出"title"標籤
print(root.title.string)#印出"title"標籤字串
```


觀察爬回來的資料

```
<div class="title">
<a href="/bbs/Lifeismoney/M.1628417722.A.8B6.html">[情報] 08/09 汽油降0.4 柴油漲0.2！</a>
</div>
<div class="meta">
<div class="author">samok</div>
<div class="article-menu">
<div class="trigger">···</div>
<div class="dropdown">
<div                                                    class="item"><a
href="/bbs/Lifeismoney/search?q=thread%3A%5B%E6%83%85%E5%A0%B1%5D+08%2F09+%E6%B1%BD%E6
%B2%B9%E9%99%8D0.4+%E6%9F%B4%E6%B2%B9%E6%BC%B20.2%EF%BC%81">搜尋同標題文章</a></div>
<div class="item"><a href="/bbs/Lifeismoney/search?q=author%3Asamok">搜尋看板內 samok 的文章</a></div>
</div>
</div>
<div class="date"> 8/08</div>
<div class="mark"></div>
</div>
</div>
<div class="r-ent">
<div class="nrec"><span class="hl f3">27</span></div>
<div class="title">
<a href="/bbs/Lifeismoney/M.1628417919.A.004.html">Re: [情報] 全聯 愛之味鮪魚片185g*3入 108元</a>
</div>
<div class="meta">
<div class="author">Guevera</div>
<div class="article-menu">
<div class="trigger">···</div>
<div class="dropdown">
<div class="item"><a
```

範例程式

```
import bs4
root=bs4.BeautifulSoup(data, "html.parser")
print(root)
print(root.title)#尋找"title"標籤
print(root.title.string)#尋找"title"標籤字串
titles=root.find("div", class_="title") 找一筆
print(titles)
print(titles.div)
print(titles.div.string)
titles=root.find_all("div", class_="title") 找全部
print(titles)
count=0
for title in titles:
    print(count, title)
    count=count+1
```

```
for title in titles:
    if title.a != None:
print(title.a.string)
```

網頁爬蟲

Web Crawler - Cookie

基本流程

1. 連線到特定網址, 抓取資料
2. 解析資料, 取得實際想要的部份

關鍵心法

盡可能讓程式模仿一個普通使用者的樣子

Web Crawler - Cookie

Cookie

長榮大學 Chang Jung Christian University

什麼是Cookie?

網站存放在瀏覽器的一小段內容

與伺服器的互動

連線時，放在Request Headers中送出

Web Crawler - Cookie

追蹤連結

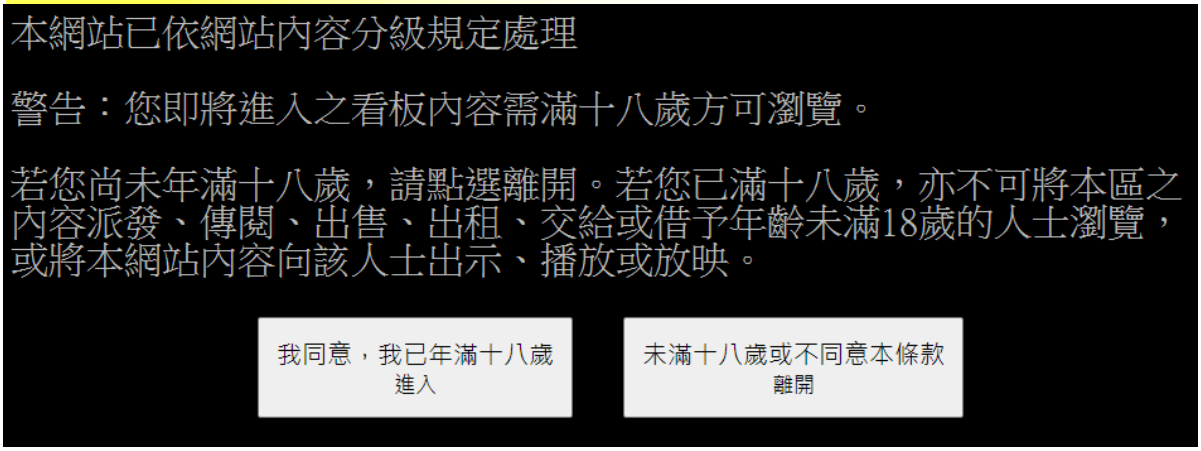
HTML 超連結

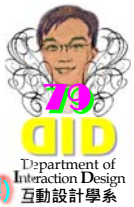
```
<html>
    <head>
        <title>HTML 格式</title>
    </head>
    <body>
        <a href="https://www.google.com/">Google</a>
    </body>
</html>
```

連續抓取頁面實務

解析頁面的超連結，並結合程式邏輯完成

PTT 性版





顯示網頁原始碼

18 [求助] 最護眼的螢幕 EIZO?
BearJW 11/30 ···

7 [討論] 哪個牌子螢幕的OSD軟體比較好用?
gumen 12/01 ···

9 [求助] hdmi電腦無訊號

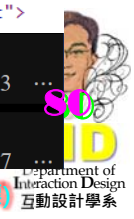
返回(B) Alt+向左鍵

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">


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

<title>看板 LCD 文章列表 - 批踢踢實業坊</title>

<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">
```

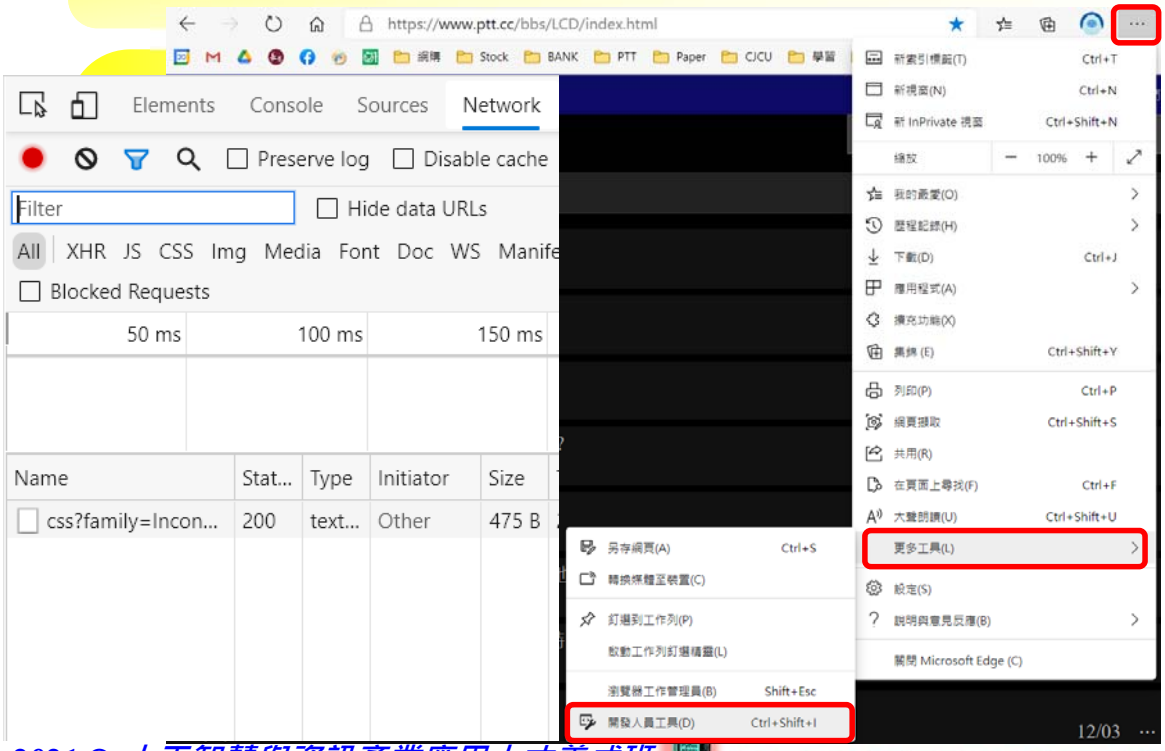
90 [知識] LCD液晶螢幕板 問與答 / 相關網站與資訊
cbate M 7/23 ···

檢視頁面來源(V) Ctrl+U
檢查(N) Ctrl+Shift+I

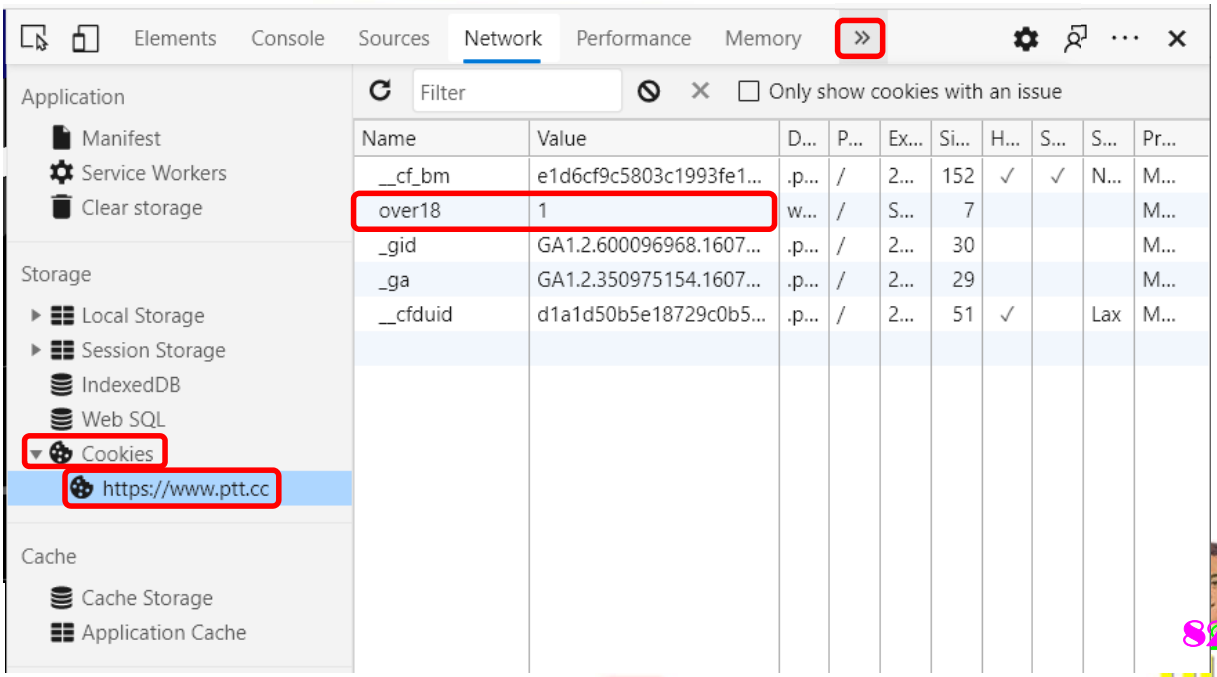
21 [公告] 發文注意事項 & 版務討論區(11/11更新)
Helios M 8/17 ···



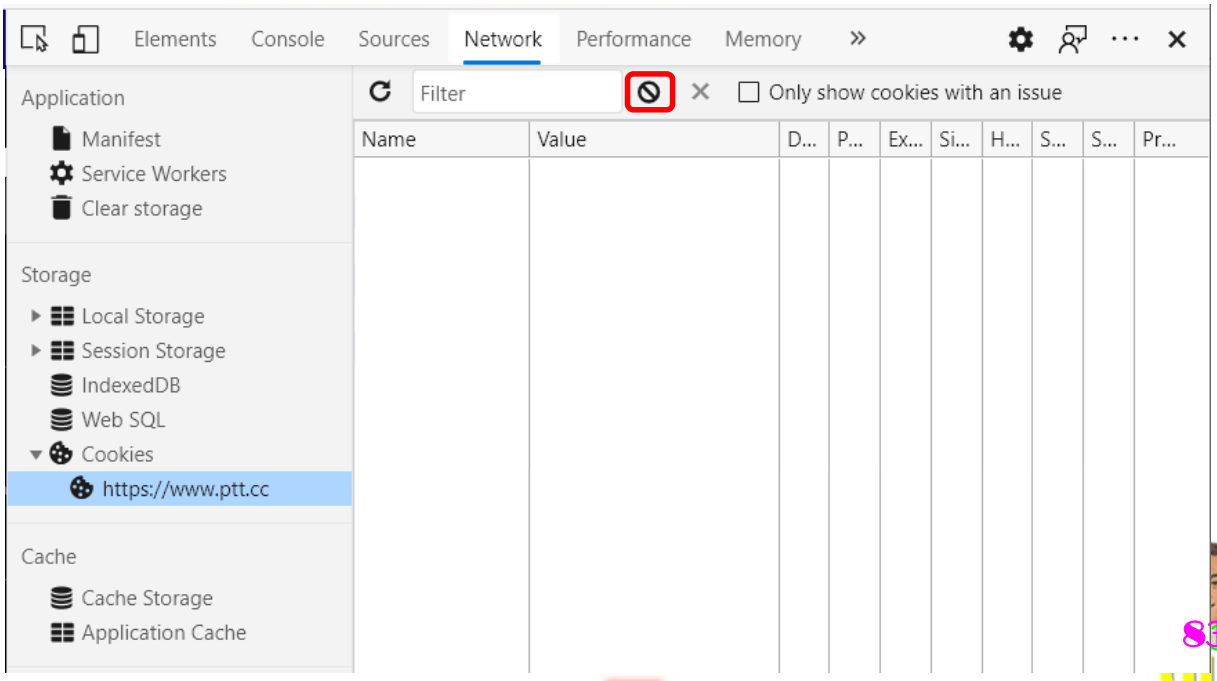
進入開發人員工具(F12)



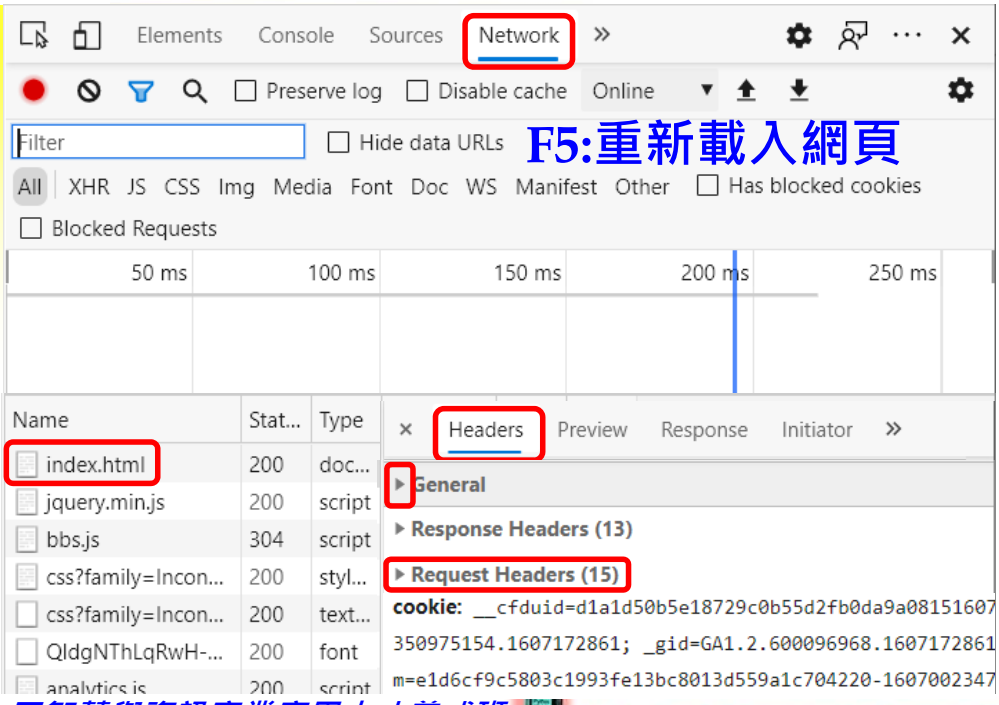
檢視Cookie



手動清除Cookie



取得 Cookie 資料



範例程式

```
import urllib.request as request
src="https://www.ptt.cc/bbs/Lifeismoney/index.html"
requestUA=request.Request(src, headers={
   "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.193 Safari/537.36 Edg/86.0.622.68"
})
with request.urlopen(requestUA) as response:
   data=response.read().decode("utf-8")
import bs4
root=bs4.BeautifulSoup(data, "html.parser")
titles=root.find_all("div", class_="title")
for title in titles:
   if title.a != None:
      print(title.a.string)
```

範例程式

```
import urllib.request as request
src="https://www.ptt.cc/bbs/sex/index.html"
requestUA=request.Request(src, headers={
   "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.193 Safari/537.36 Edg/86.0.622.68"
})
with request.urlopen(requestUA) as response:
   data=response.read().decode("utf-8")
import bs4
root=bs4.BeautifulSoup(data, "html.parser")
titles=root.find_all("div", class_="title")
for title in titles:
   if title.a != None:
      print(title.a.string)
```

範例程式

```
import urllib.request as request
src="https://www.ptt.cc/bbs/sex/index.html"
requestUA=request.Request(src, headers={
   "cookie":"over18=1",
   "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.193 Safari/537.36 Edg/86.0.622.68"
})
with request.urlopen(requestUA) as response:
   data=response.read().decode("utf-8")
import bs4
root=bs4.BeautifulSoup(data, "html.parser")
titles=root.find_all("div", class_="title")
for title in titles:
   if title.a != None:
      print(title.a.string)
```

長榮大學 Chang Jung Christian University

範例程式

```
#抓取上一頁的連結
nextLink=root.find("a", string="‹ 上頁")
print(nextLink)
print(nextLink["href"])
```

範例程式

```
import urllib.request as request
def getData(src):
    requestUA=request.Request(src, headers={
        "cookie":"over18=1",
        "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/86.0.4240.193 Safari/537.36 Edg/86.0.622.68"
    })
    with request.urlopen(requestUA) as response:
        data=response.read().decode("utf-8")
    import bs4
    root=bs4.BeautifulSoup(data, "html.parser")
    titles=root.find_all("div", class_="title")
    for title in titles:
        if title.a != None:
            print(title.a.string)
    nextLink=root.find("a", string="‹ 上頁")
    print(nextLink["href"])
#抓上一頁連結
src="https://www.ptt.cc/bbs/sex/index.html"
getData(src)
```

範例程式

```
import urllib.request as request
import bs4
def getData(src):
    requestUA=request.Request(src, headers={
        "cookie":"over18=1",
        "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.193
Safari/537.36 Edg/86.0.622.68"
    })
    with request.urlopen(requestUA) as response:
        data=response.read().decode("utf-8")
root=bs4.BeautifulSoup(data, "html.parser")
    titles=root.find_all("div", class_="title")
    for title in titles:
        if title.a != None:
            print(title.a.string) nextLink=root.find("a", string="‹ 上頁")
    return nextLink["href"]
#抓上一頁連結
src="https://www.ptt.cc/bbs/sex/index.html"
NextPage="https://www.ptt.cc/"+getData(src)
print(NextPage)
```

範例程式

```
count=0
src="https://www.ptt.cc/bbs/sex/index.html"
while count<5:
    src=NextPage="https://www.ptt.cc/"+getData(src)
    count+=1

count=0
src="https://www.ptt.cc/bbs/sex/index.html"
while count<15:
    src=NextPage="https://www.ptt.cc/"+getData(src)
    count+=1
    print("第",count,"頁\n==========================\n")
```

網頁爬蟲

Web Crawler - AJAX

基本流程

1. 連線到特定網址, 抓取資料
2. 解析資料, 取得實際想要的部份

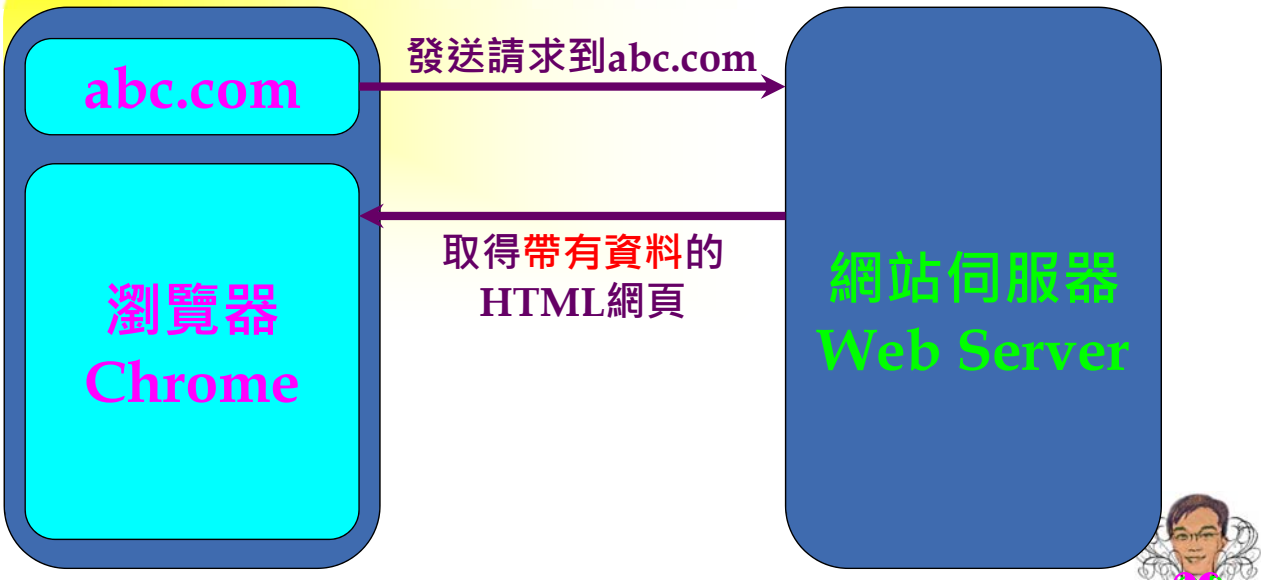
Web Crawler - AJAX

AJAX

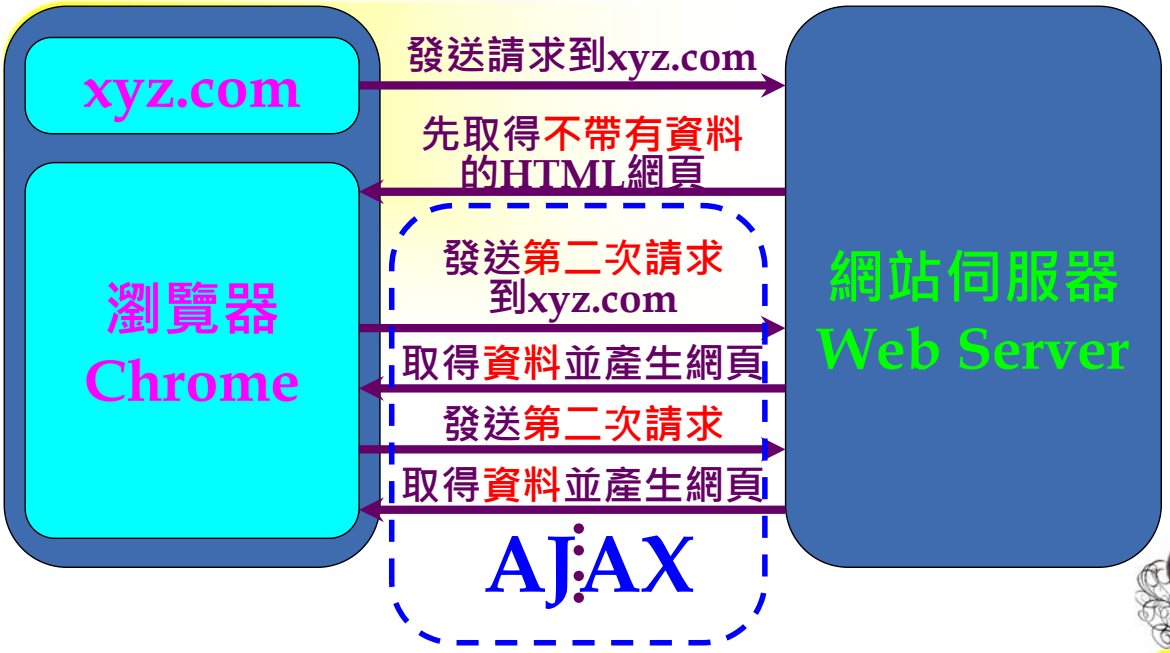
什麼是AJAX

網頁前端的JavaScript程式技術

傳統、一般的網頁



使用AJAX技術網頁



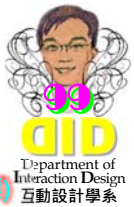
Web Crawler - AJAX

實務操作

kkday

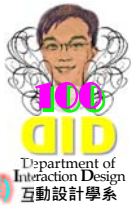
https://www.learncodewithmike.com/2020/10/scraping-ajax-websites-using-python.html

抓取https://www.kkday.com/的文章



關鍵問題

認出網站運作模式
找出真正能抓到資料的網址

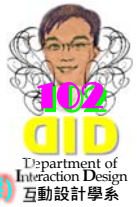


取得網頁資料

```
import urllib.request as req
url="https://medium.com/_/api/home-feed"
request=req.Request(url, headers={ "User-Agent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.193 Safari/537.36 Edg/86.0.622.68"
})
with req.urlopen(request) as response:
    data=response.read().decode("utf-8")
```

Json格式擷取

```
import json
data=json.loads(data)
print(data)
```



Json格式特定欄位

```
posts=data["data"]
#取得Json格式下的data欄位資料

for key in posts:
#取得Json格式標籤資料
        title = key["name"]
        #每筆標籤的name欄位資料
```

範例程式

```
import urllib.request as req
url="https://www.kkday.com/zh-
tw/product/ajax_productlist/?country=&city=&keyword=%E5
%8F%B0%E5%8D%97%E5%B8%82&availstartdate=&availend
date=&cat=TAG_4_4&time=&glang=&sort=rdesc&page=1&ro
w=10&fprice=*&eprice=*&precurrency=TWD&csrf_token_na
me=d840df7741e3cb9df1302c3b8231afeb"
request=req.Request(url, headers={ "User-Agent":"Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/86.0.4240.193 Safari/537.36
Edg/86.0.622.68"
})
with req.urlopen(request) as response:
   data=response.read().decode("utf-8")
print(data)
```

範例程式

```
import json
data=json.loads(data)
print(data)

posts=data["data"]
print("票券名稱:")
for key in posts:
    title = key["name"]
    print(title)

print("票券詳細內容連結:")
for key in posts:
    title = key["url"]
    print(title)
```

範例程式

```
print("票券價格:")
for key in posts:
    title = key["price"]
    print(title)

print("最早可使用日期:")
for key in posts:
    title = key["earliest_sale_date"]
    print(title)

print("評價:")
for key in posts:
    title = key["rating_star"]
    print(title)
```


Q&A