Python 網路爬蟲簡介

Introduction to Web Crawler

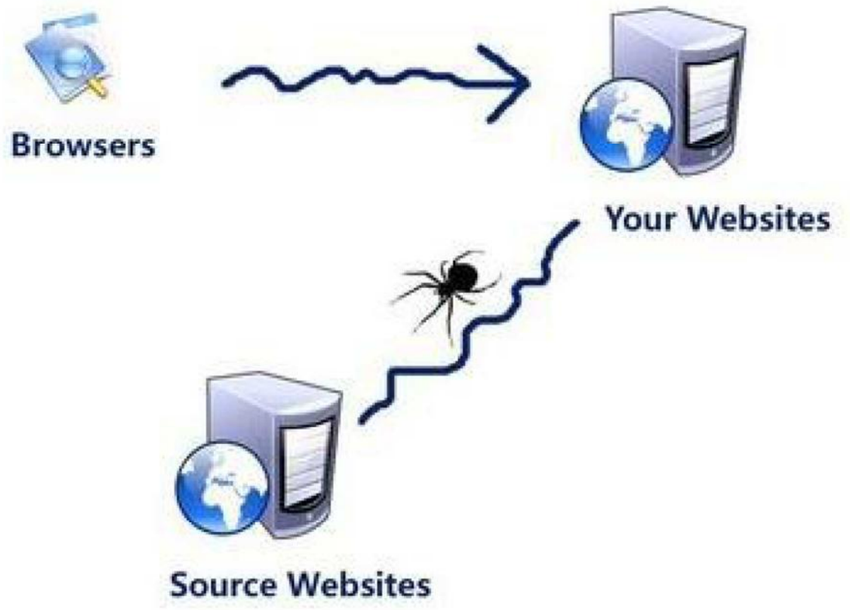
Python Web Crawler

網站資料擷取

- 資料分析的首要工作是取得正確的資料，並且保持乾淨的形式；取得資料的方式及類型很多，其中一種就是指到網站上抓取你要的東西。而Python擁有許多套件，可協助網站資料擷取。
- 過去我們瀏覽網頁是使用瀏覽器，現在透過Python就可以直接瀏覽網頁，也可以直接下載網頁。
- 一般我們將從網頁搜尋資源的程式稱之為「網頁爬蟲」，一些有名的搜尋引擎公司就是不斷地送出網路爬蟲搜尋網路最新訊息，以保持搜尋引擎的熱度。
- 近年來，由於開放資料的風氣盛行，政府和民間陸續將所蒐集到的資料釋出，成為取得資料相當方便的來源。
- 但有些資料並不在這類型的網站上，因此若能學會基本的網路爬蟲方法，則能將擷取網路上資料，加以運用。

網路爬蟲簡介

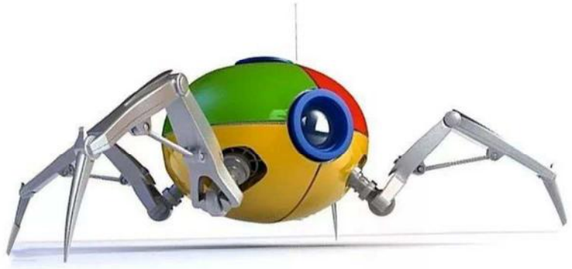
- 網路爬蟲，英文稱做Web Crawler 或 Web Scraping，是一個可以從網路獲取資料的技術，也可以說是一種按照一定的規則，自動的抓取網路資訊的程式或者腳本。
- 它可從全球資訊網上下載網頁，是搜索引擎的資料來源。
- 基本的網路爬蟲從一個初始頁面的網址開始，在抓取網頁的過程中，不斷從當前頁面上抽取新的網址放入佇列，直到滿足系統的停止條件才結束。



網路爬蟲的用途

網路爬蟲是收集相關網路資料的利器，因此需要大量網路資料的情形就很適合爬蟲。相關用途主要如下：

- 社群發文的留言分析
- 即時、非即時的輿情分析
- 資料收集、資料分析流程進行
- 做為網頁搜尋引擎 (Google、Yahoo)的網頁收集器
- 數據挖掘、社群演化、線上人類行為、人類動力學研究、計量社會學、複雜網路等科學研究領域



網路爬蟲是否合法

- 網路爬蟲目前處於早期發展階段，「允許哪些行為」尚處於摸索之中。以目前的情況來看，如果抓取資料的行為僅限於個人使用， 則應不存在問題；而如果資料用於轉載，那麼抓取的資料類型就非常關鍵了。
- 世界各地法院的一些案件可以協助我們了解哪些網路爬蟲行為是允許的。
  - 在Feist Publications, Inc. 起訴Rural Tel寫phone Service Co. 的案件中，美國聯邦最高法院裁定抓取並轉載真實資料 （比如 電話清單〉是允許的。
  - 在澳大利亞， Telstra Corporation Limited 起訴Phone Directories Company Pty Ltd 這一類似案件中， 則裁定只有擁有明確作者的資料，才可以獲得版權。
  - 在歐盟的ofir.dk起訴home.dk一案中， 最終裁定定期抓取和深度連結是允許的。
  - 這些案件告訴我們， 當抓取的資料是現實生活中的真實資料（比如營業位址、 電話清單） 時，是允許轉載的。
  - 但是如果是原創資料（比如，意見和評論），通常就會受到版權限制，而不能轉載。

- 當要抓取某個網站的資料時，請記住自己是該網站的訪客，理應約束自己的抓取行為， 否則他們可能會封鎖IP，甚至採取更進一步的法律行動。

網路爬蟲是否必要

- 理想狀態下， 爬蟲並不是必須品。有些網站會提供 API 以(半)結構化的格式共用資料。(例如CSV格式、JSON格式)
- 然而，並不是所有的資料都能這麼方便地以CSV或者JSON載入工作環境，有時候資料散落在網路不同的角落裡。
- 此外，也並不是每一個網站都會建置 API（ Application Programming Interface ），讓我們方便的把資料下載下來。
- 現實情況中，只有部份網站提供了API，而且它們通常會限制可以抓取的資料，以及訪問這些資料的頻率。
- 而對於網站開發者而言，維護前端介面比維護後端API介面優先順序更高。
- 因此，我們除了了解使用 API 下載我們所需的資料之外， 學習一些爬蟲技術的相關知識更是有助於資料的擷取。

用什麼語言寫網路爬蟲？

Coding. Crawler

Java

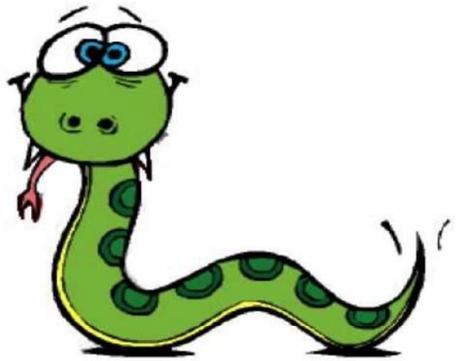
```
public class CrawlerExample {

    public static void main(String[] args) throws IOException {
        PrintWriter textFile = null;
        try {
            textFile = new PrintWriter("result.txt");
            System.out.println("Enter the URL you wish to crawl..");
            System.out.print("@> ");
            String myUrl = new Scanner(System.in).nextLine();

            String response = getContentByUrl(myUrl);

            Matcher matcher = Pattern
              .compile("href=[\"'](.[^\"']+)[\"']").matcher(response);
            while (matcher.find()) {
                String url = matcher.group(1);
                System.out.println(url);
                textFile.println(url);
            }
        } finally {
            if(textFile != null) {
                textFile.close();
            }
        }
    }

    private static String getContentByUrl(String myUrl)
            throws IOException {
        URL url = new URL(myUrl);
        URLConnection urlConnection = url.openConnection();
        BufferedReader in = null;
        StringBuilder response = new StringBuilder();
        try {
            in = new BufferedReader(new InputStreamReader
              (urlConnection.getInputStream()));
            String inputLine;
            while ((inputLine = in.readLine()) != null) {
                response.append(inputLine);
            }
        } finally {
            if(in != null) {
                in.close();
            }
        }
        return response.toString();
    }
}
```



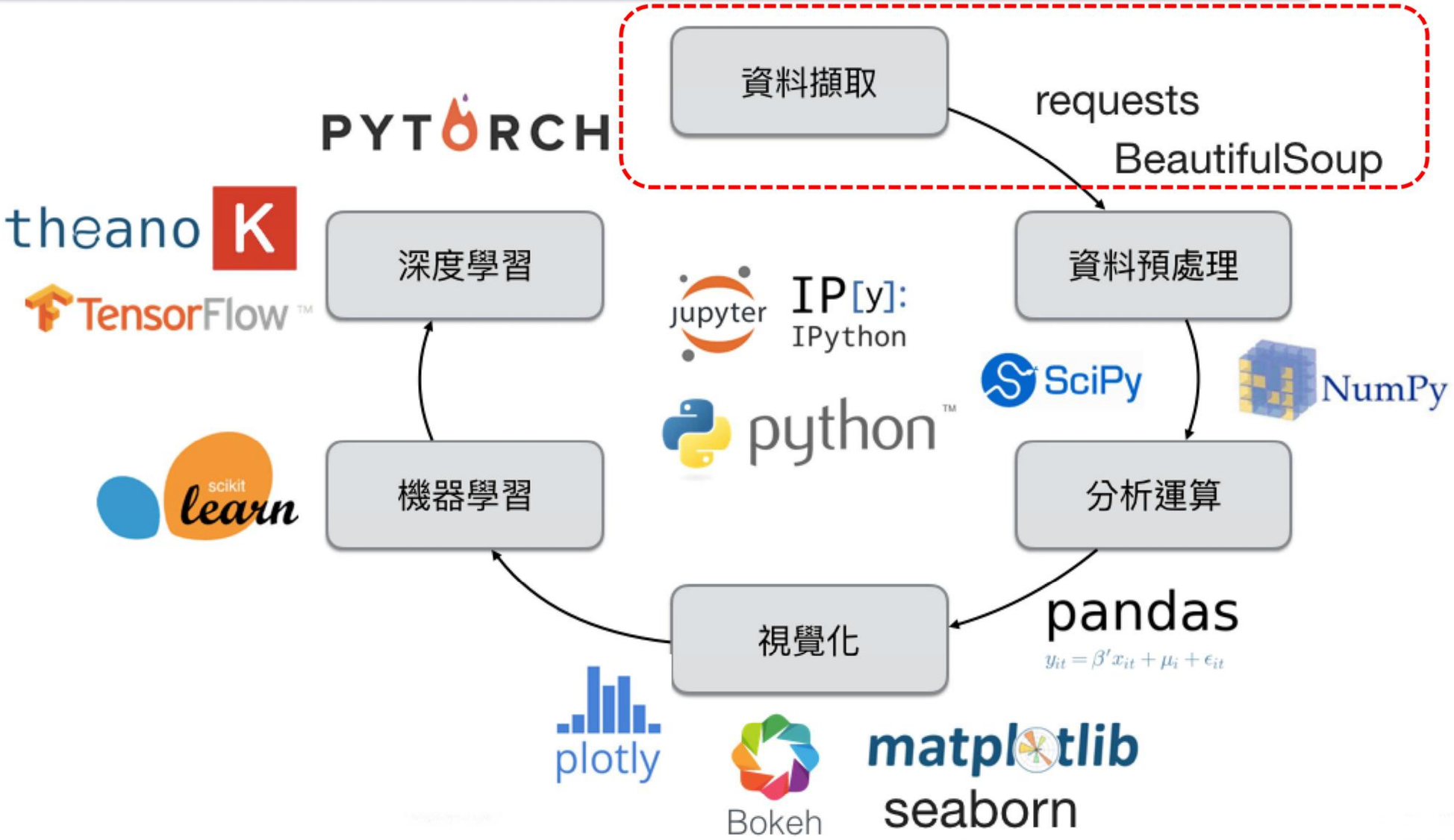
Python

```
if __name__ == '__main__':
    with open("result.txt","wt") as textFile:
        print("Enter the URL you wish to crawl..")
        myUrl = input("@> ")
        for i in re.findall("href=[\"'](.[^\"']+)[\"']",
            urllib.request.urlopen(myUrl).read().decode(), re.I):
            print(i)
            textFile.write(i+'\n')
```

Python為網路爬蟲主流之因 - 相關支援套件眾多

- 對Linux和Windows跨平台都有不錯的支援
- 可使用Numpy，Pandas等進行科學計算與數值分析整合
- 可使用Matplotlib與Mayavi2進行2d與3d的視覺化處理
- 可使用Networkx進行複雜網路的處理
- 可使用Rpy與R語言介面進行統計相關的處理
- 可使用Django進行網站的快速開發

資料分析流程


資料擷取
requests
BeautifulSoup
資料預處理
分析運算
pandas
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$
視覺化
matplotlib
seaborn
Bokeh
plotly
機器學習
深度學習
PYTORCH
theano
K
TensorFlow
scikit
learn
jupyter
IP[y]:
IPython
python
SciPy
NumPy

資料來源

- 資料就像是原料一樣，而資訊就像是經過主廚精心烹調的料理。有適合的資料，才有好的資料價值。
- 跨資料分析能夠結合不同的知識，進而找出資料間無法明顯發覺的關連性。但這牽涉到不同領域的背景，也需要處理、串聯不同來源的資料集，是一個具有挑戰的工作。尤其當資料來自許多不同的單位，用不同的格式發布，造成資料搜集更加費時費工，有效地善用工具將可以節省人工成本的損耗。
- 「找資料」→「整理資料」→「使用資料」，是在分析前的標準程序。仰賴於網路科技的普及，搜尋引擎已經覆蓋了大量的網路資源。有效地使用搜尋引擎是尋找資料的基本功。
- 除此之外，隨著開放資料(Open data)的議題興起，現在也有越來越多的官方或非官方組織將其資料公開讓大家使用。

資料來源

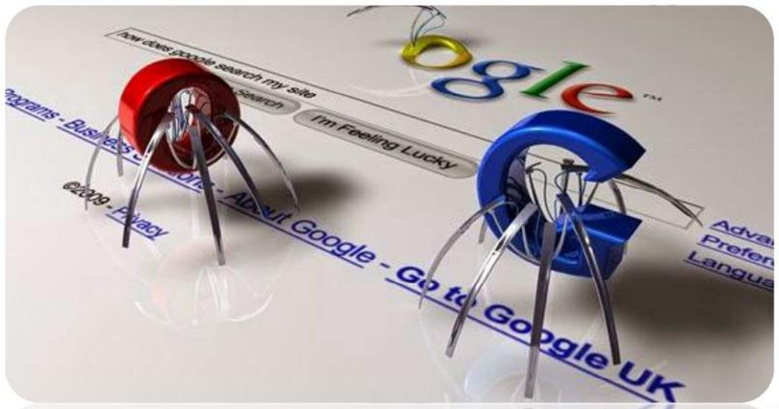
◆ **幾個常見的公開資料單位：**

- 政府資料開放平台
- 行政院主計總處
- 中選會選舉資料庫
- Google Public Data
- World Bank 世界銀行

資料來源 → 資料取得

- 知道資料的來源之後，接下來就是如何取得資料。
- 一般來說，常見的資料取得方式有以下三種：

1. **直接下載開放檔案格式**
2. **透過API取得開放檔案格式**
3. **擷取網頁資料**



資料取得

◆ 直接下載開放檔案格式：

- 資料存成檔案提供下載，主要的檔案格式為「CSV」、「JSON」、「XML」。
- 如果有提供上述格式的話，一般的程式語言或是商業軟體應都具備存取的功能，相對容易處理。

◆ 透過API取得開放檔案格式：

- API（Application Program Interface，應用程序介面），是一種提供程式化連接的介面，讓工程師/分析師可以選擇資料中要讀取的特定部分，而不需要把整批資料事先完整下載回來。
- API 一般都是直接連接到一個資料庫，而資料庫內儲存的都是即時更新最新版本的資料。
- API 主要分為兩個動作：**1. 使用者呼叫查詢； 2. 伺服器根據需求回傳。**
- 呼叫的方式有 POST 或是 GET；一般而言，回傳的資料格式會是 **JSON** 的格式。

◆ 擷取網頁資料：

- 除了上述二者以外，我們常常會發現所需的資料並不是一個特定的檔案，也沒有 API 可以使用，就是直接呈現在網頁上，可能一頁，也可能數頁。這樣的話，就只能自己寫爬蟲，把自己想用的資料從網頁上爬下來。

Q & A