取得網路資料：

建立Python網路爬蟲：

使用Python程式模擬瀏覽器送出HTTP請求。 完整瀏覽器會執行JavaScript程式碼，Python爬蟲程式則不會。

(Review 補充：JavaScript程式碼執行的動態效果、動畫等)

使用Quick JavaScript Switcher擴充功能：開啟功能可以直接”開關”網頁的 JavaScript 功能

關掉 JavaScript 後，發現哪個部分的內容不見了，那個內容很可能是 JavaScript 所產生。

如果想要爬取的資料是執行JavaScript之後才產生的內容

無法執行JavaScript程式碼，就無法產生該資料(後面也就無法爬取)，此時便需要其他工具進行網路爬蟲。

==============================================================

Python網路爬蟲所需工具＆函式庫

●網路爬蟲分析工具：

最常使用的是瀏覽器內建的”開發人員工具”，可以幫助定位HTML網頁中資料的位置，和找出資料的特徵，例如標籤名稱、屬性值。

(Review 補充：使用F12開啟)

●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，如果需要執行JavaScript程式碼，則需要”Selenium”自動瀏覽器。

(Review 補充：使用”Selenium”自動瀏覽器，會執行JavaScript、Python程式碼，所以爬取出來的資料可能包含一些JavaScript程式碼撰寫的動態效果，例如下圖中紅色的New)

網路爬蟲函式庫：

取得回應HTML網頁內容後，需要使用剖析HTML網頁，以取出所需資料。

◎擷取靜態網頁資料(HTML標籤的資料)：

對於使用HTML標籤建立的網頁內容，可使用BeautifulSoup、Ixml模組來擷取網頁內容。

◎擷取動態網頁資料(JavaScript產生的HTML標籤資料)：

如果Web網站是使用JavaScript產生的動態網頁內容，則需要”Selenium自動瀏覽器”來取得動態網頁內容，然後再使用BeautifulSoup、Ixml模組來擷取網頁內容。

(Review 補充：基本上都直接用”Selenium自動瀏覽器”來爬取資料，就不用額外分析是否為靜態網頁)

Python網路爬蟲的基本步驟：

1. 找出目標URL網址和參數

找出目標資料是位於哪一個Web網站，即網域名稱和埠號。

1. 判斷網頁內容如何產生

找出URL網址和參數後，使用前述”Quick JavaScript Switcher”擴充功能，切換執行JavaScript程式碼，以便判斷網頁內容是否有改變。

●如果網頁內容完全相同：

不論是否執行JavaScript程式碼，網頁內容皆相同，這是靜態HTML網頁，使用”Requests”模組取得網頁資料。

●網頁內容有差異，但是目標資料沒有改變：

表示JavaScript程式碼只影響非目標資料，例如動態效果、使用介面排版等，因為目標資料仍存在，所以使用”Requests”模組取得網頁資料。

●目標資料消失：

如果執行JavaScript程式碼後會影響到目標資料，表示該資料是透過JavaScript程式碼產生，使用”Requests”模組取得的網路資料並不會執行JavaScript程式碼，所以無法使用”Requests”模組取得網頁資料，須使用”Selenium”來取得網路資料。

(Review 補充：基本上都直接用”Selenium自動瀏覽器”來爬取資料，就不用額外分析是否為靜態網頁)

1. 擬定擷取資料的爬蟲策略

當判斷出網頁內容的產生方式，成功取回資料後，接著需擬定擷取資料的網路爬蟲策略，即”如何在HTML網頁定位目標資料”，常用技術如下：

●CSS選擇器：

CSS選擇器，是CSS層級式樣式表語法規則的一部分，可以定義哪些HTML標籤需要套用CSS樣式，可以使用CSS選擇器來定位網頁資料。

(Review 補充：”div”、”a”、”class”等等，開啟F12即可得到)

●XPath表達式：

XPath表達式是一種XML技術的查詢語言，可以在XML文件找出所需的節點，也適用HTML網頁，故可使用XPath表達式瀏覽HTML網頁，來找出指定的HTML標籤和屬性(”Selenium支援”)。

●正規表達式：

正規表達式是一種小型範本比對語言，可以使用範本字串進行字串比對，以便從文字內容中找出符合的內容，可配合CSS選擇器搜尋指定的標籤內容，例如：金額、地址、電話等。

1. 取得的資料存成檔案或存入資料庫

成功爬取和收集好資料後，整理或結構化資料，並儲存起來，一半而言會儲存成CSV、JSON檔案、或是存入資料庫。