**與擷取相關的JavaScript**

**用戶端語言JavaScript，是網路上最常見與支援最好的用戶端腳本語言。它可用於蒐集資訊、提交表單、嵌入多媒體系統、執行線上遊戲。看起來簡單的網頁也經常帶有多段的JavaScript。你可以從原始碼Script標籤看到它。**

**<script>**

**alert(“This creats a pop-up using JacaScript”);**

**</script>**

**JavaScript簡介**

JavaScript是弱型別語言，雖然運算子、迴圈、陣列等語法與C++和Java很像。

舉例來說: 下面進行費氏及數遞回計算並輸出到瀏覽器的開發者控制台:

<script>

Function fibonacci(a, b){

var nextNum = a + b;

console.log(nextNum+” is in the Fibonacci sequence”);

if(nextNum < 100){

fibonacci(b, nextNum);

}

}

fibonacci(1, 1)

</script>

JavaScript也能將函式當作變數傳遞

<script>

var fibonacci = function() {

var a = 1;

var b = 1;

return function() {

var temp = b;

b = a+b;

a = temp;

return b;

}

}

var fibInstance = fibonacci();

console.log(fibInstance() +” is in the Fibonacci sequence”);

console.log(fibInstance ()+” is in the Fibonacci sequence”);

</script>

**常見JavaScript函式庫**

雖然JavaScript語言核心非常重要，但如今的網路更需要第三方函式庫。檢視網頁原始黨就會看到這些常用寒士庫。

使用Python執行JavaScript非常花時間界耗費處理能力。

1. **jQuery**

**……..**

1. **Google Analytics**

**………**

1. **Google Maps**

**………**

**Ajax與動態HTML**

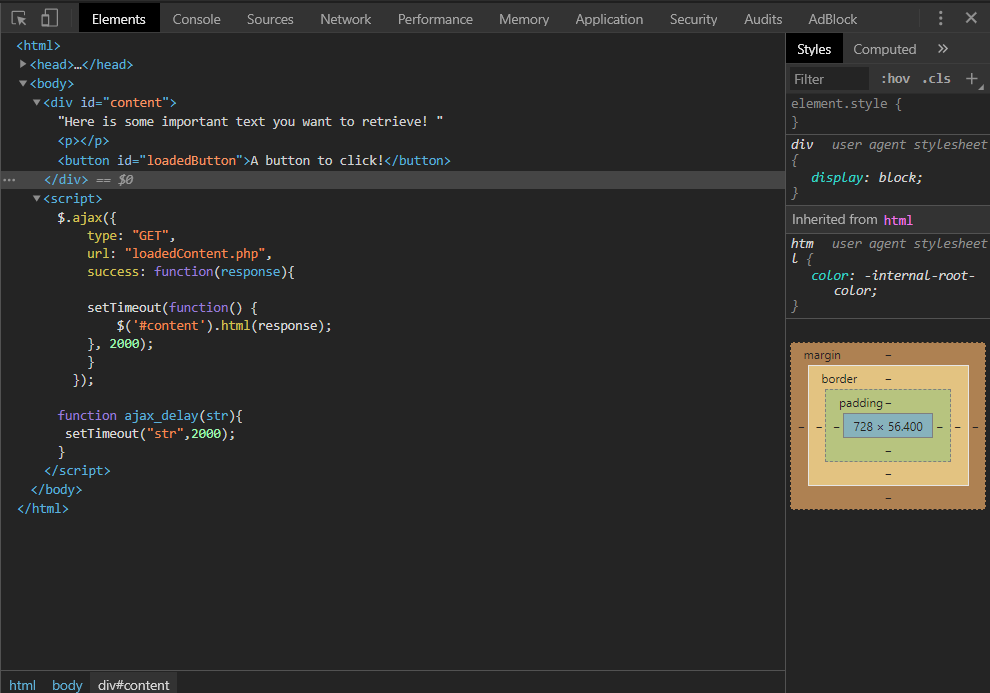
前面只討論發送某種HTTP請求 給 網頁伺服器。若能提交表單或從伺服器取得資訊而沒有重新載入網頁，該網站可能使用Ajax。

Ajax並非一種語言而是一組完成特定任務的技術(很像網路擷取)

Ajax代表(Asynchrounous JavaScript and XML)，用於發送資訊給網頁伺服器並在沒有另一個網頁請求下取得資訊。

**不要說某個網站是用Ajax寫的，正確是:”此表單用Ajax溝通”**

實利: <http://pythonscraping.com/pages/javascript/ajaxDemo.html>



如同Ajax，動態HTML(DHTML)是一組技術，DHTML是HTML程式、CSS語言或兩者在用戶端腳本上改變網頁的HTML元素。按鈕在使用者移動滑鼠的時候出現、點擊後改變背景顏色、或觸發Ajax請求新的內容。

**什麼是ajax的動態載入？**

就是當使用者在頁面上觸發事件，javascript會對server發出一個請求，並把資料渲染回Client端。  
他的優點是觸發資料載入的當下，使用者還是可以繼續使用該程序。

**請注意: "動態”不一定是”移動”或 “改變”，會移動的圖片、影片不一定就是DHTML網頁，有些看起來進泰的網頁有DHTML在背後運行，以JavaScript操作HTML與CSS**

**舉例來說: 若擷取的網站夠多，會發現 擷取的內容 跟 瀏覽器上看到的內容 不一樣，看著擷取的內容卻找不到哪裡有瀏覽器上看到的內容**

**網頁還可能將你重新導向其他網頁但URL卻不變**

**這些現象是因為你的擷取程序沒有執行讓內容改變的JavaScript**

**沒有這些JavaScript，Htm不會動，所以才會不一樣。**

**在Python 中使用Selenium執行JavaScript**

Selenium(<http://www.seleniumhq.org/>)原本是用於網站測試的網頁擷取工具。他還被用於精確地描繪網站—**如同從瀏覽器檢視**。**Selenium用瀏覽器自動化載入網站，擷取必要的資料、甚至擷取畫面或執行動作。**

Selenium WebDriver API 支援 Java、C#、Ruby、Python 及 Perl 等多種語言，而且帶來跨越不同瀏覽器的自動化操作

Selenium自己沒有瀏覽器；需要第三方的瀏覽器整合。舉例來說，若使用Firefox與Selenium，你會看到螢幕上啟動Firefox、到指定網站、執行程式指定的動作。雖然可以觀賞，但有人偏好腳本在背景執行，因此使用了稱為**PhantomJS**(<http://www.phantomjs.org/>)的工具。

PhantomJS又稱**無頭瀏覽器**。它將網站載入記憶體並執行網頁上的JavaScript但不顯示給使用者。結合Selenium與PhantonJS可讓網站擷取程序輕鬆處裡Cookie、JavaScript、標頭或其他東西。

**Selenium**可由<http://pypi.python.org/pypi/selenium>或使用pip等工具安裝

PhantomJS 可由<http://.org/pypi/selenium>

**Selenium的selector**

**之前的章節是用BeautifulSoup的find與find\_all等selector選取網頁元素。**

**Selenium使用另一組selector搜尋WebDriver的DOM**

**此例使用find\_element\_by\_id這個selector，但下列selector也可行:**

**driver.find\_element\_by\_css\_selector(‘#content’)**

**driver.find\_element\_by\_tag\_name(‘div’)**

**選取多個元素，這謝selector可以用elements**

**driver.find\_elements\_by\_css\_selector(‘#content’)**

**driver.find\_elements\_by\_tag\_name(‘div’)**

**若還是要用回BeautifulSoup解析內容，你可以使用WebDriver的page\_source函式回傳網頁原始碼字串:**

**page\_source = driver. page\_source**

**bs = BeautifulSoup(page\_source, ‘html.parser’)**

**print(bs.find(id = ‘content’).get\_text())**

**其他Selenium WebDriver**

**處裡重新導向**

用戶端重新導向由瀏覽器的JavaScript而非伺服器在傳送網頁前進行。有時判斷網頁是否造訪過會有些麻煩。重新導向可能發生的很快，使你沒有注意道並以為是伺服器端重新導向。

但擷取網站時差別很明顯。伺服器重新導向可由Python的urllib函式庫處裡而無須Selenium幫忙(更多資訊見第3章)。**用戶端重新導向需要執行JavaScript**。