



Chapter 08

jQuery基礎與開發人員工具

中原大學 資訊管理學系

賴錦慧 老師

chlai@cycu.edu.tw

大綱

- jQuery的基礎
- jQuery的下載與使用
- 建立jQuery程式
- Google Chrome的開發人員工具

認識jQuery



- jQuery是一個JavaScript函式庫
 - 能更簡潔的方式來撰寫JavaScript程式碼
 - 擴充JavaScript的功能

- jQuery是在2006年1月由John Resig在BarCamp NYC發表的網頁技術，這是一種高效率和簡潔的JavaScript函數庫，目前是MIT和GPL授權的免費軟體，可供個人或商業專案使用。

- jQuery強調JavaScript與HTML之間的交互作用
 - 使用簡潔程式碼來處理DOM
 - 走訪網頁元素來更改外觀、新增特效、事件處理、動畫
 - 支援Ajax來加速Web應用程式開發

為什麼使用jQuery (1/3)

□ 讓JavaScript程式碼更簡潔

- 對於JavaScript可能需要數十行程式碼才能完成的工作，在jQuery只需幾行程式碼就可以完成相同的工作

□ 例如：建立HTML表格的斑馬紋效果

`$('table tr:nth-child(even)').addClass('zebra');`

- jQuery程式碼只有一行，就可以替HTML表格加上斑馬紋效果

JavaScript	Dynamic HTML	Ajax
JSP	PHP	Java Applete
VBScript	ASP	Java Servlet
ASP.NET	ActionScript	Flash

為什麼使用jQuery (2/3)

□jQuery改變你撰寫JavaScript的方式

JavaScript

```
divs = document.getElementsByTagName('div');  
for (i = 0; i < divs.length; i++) {  
    divs[i].style.display = 'none';  
}
```

jQuery

```
$("#div").hide();
```

Javascript

```
<input type="button" id="myBtn" value="click"  
      onclick="alert(this.id)" />
```

```
var btn = document.getElementById("myBtn");  
btn.onclick = function(){alert(this.id);};
```

```
var btn = document.getElementById("myBtn");  
btn.addEventListener("click", function(){alert(this.id);});
```

```
var btn = document.getElementById("myBtn");  
btn.attachEvent("onclick", function(){alert(this.id);});
```

jQuery

```
$("#myBtn").on("click",function(){alert(this.id);});
```

為什麼使用jQuery (3/3)



□ 支援跨瀏覽器

- 解決: 相同程式在不同瀏覽器上有不同的效果
- 自動依不同瀏覽器進行解析對應, 不需注意瀏覽器間的差異

□ 與其他JavaScript函式庫相容性高

- jQuery會自動避免與其他JS函式庫競合, 運作較穩定

□ 擴充性高

- jQuery有多種外掛, 功能豐富
- <http://plugins.jquery.com/>

jQuery的下載與使用

- 下載jQuery
- 在JavaScript程式使用jQuery

下載jQuery



- 在jQuery官方網站可以下載最新穩定版本的jQuery
 - <http://jquery.com/download/>
 - 請注意！2.x版不支援Internet Explorer 6、7和8版，舊版請使用1.x版
 - 程式碼檔案共有兩種版本

版本	說明
Compressed Production	壓縮版本提供最小檔案尺寸，適合使用在實際運作的Web應用程式或網站
Uncompressed Development	沒有壓縮版本，主要是用來除錯，和讓有興趣的使用者了解jQuery程式碼是如何完成這些神奇功能

在JavaScript程式使用jQuery-1



***jQuery教學 <http://www.w3schools.com/jquery/default.asp>

使用下載的jQuery程式碼檔案

- jQuery下載檔案是副檔名.js的JavaScript程式碼檔案
- 只需將檔案置於HTML網頁的同一資料夾，就可以在HTML網頁<head>標籤的<script>子標籤含括外部JavaScript程式碼檔案的jQuery函數庫

```
<script src="jquery-2.1.1.min.js"></script>
```

```
<script>
```

```
.....
```

```
</script>
```

在JavaScript程式使用jQuery-2



使用CDN(Content Delivery Network)

- **CDN**是將資料存放在網路系統的多個電腦節點，以加速資料存取
- 資料是儲存在全球多個不同位置的伺服器，當存取資料時，就會從最近的伺服器來取得資料，以加速資料存取
 - jQuery CDN: <https://code.jquery.com/>
 - Microsoft CDN:
https://www.asp.net/ajax/cdn#jQuery_Releases_on_the_CDN_0
 - Google Ajax API CDN:
<https://developers.google.com/speed/libraries/#jquery>
- 在JavaScript程式可以在HTML網頁<head>標籤的<script>子標籤含括CDN的jQuery函數庫：

```
<script  
  src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">  
</script>  
<script>  
.....  
</script>
```

建立jQuery程式

- 建立第一個jQuery程式
- jQuery程式架構

jQuery基本語法(1/3)



□ HTML元素的操作

- 透過選擇器設定目標元素, 再以方法(Method)操作該元素的內容與樣式
- **jQuery("選擇器").方法(參數);**
 - jQuery("a.article").css({"background-color":"#eee"});
- **\$("選擇器").方法(參數);**
 - \$("a.article").css({"background-color":"#eee"});

jQuery基本語法(2/3)



□ 語法開始執行的時間點

- 傳統JS: 瀏覽器完全載入HTML或圖片等網頁元件後才執行JS語法, 避免JS操作尚未載入的元素而導致錯誤發生
- Script載入結束前, 若使用者有任何操作, 也會造成非預期的結果
- `$(document).ready(處理內容);`

```
$(document).ready(function (){  
    //這裡寫處理程式  
});
```

```
$(function (){  
    //這裡寫處理程式  
});
```

jQuery基本語法(3/3)



□ 方法串聯(Method Chain)

- jQuery針對同一元素的多個操作,可用簡潔的方式表達,程式更容易閱讀
- Example
 - Script中要取得元素有幾個步驟要執行

```
$("#div#docs p").hide();  
$("#div#docs p").removeClass("article");  
$("#div#docs p").addClass("text");  
$("#div#docs p").show();
```

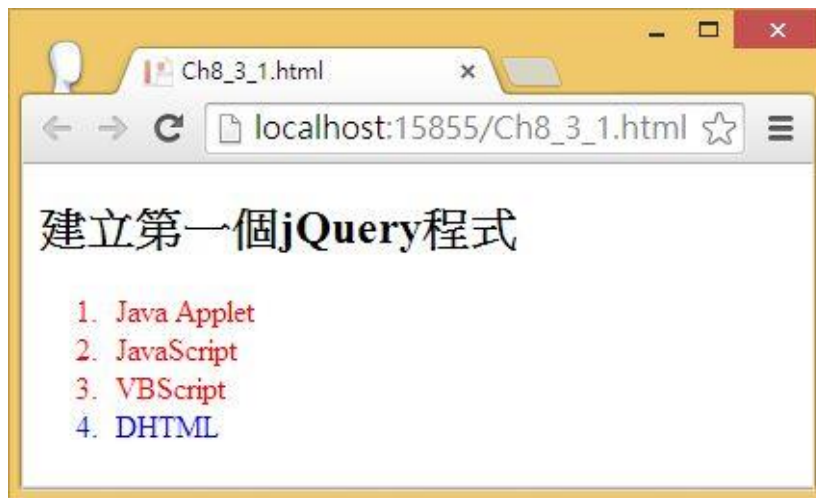


```
$("#div#docs p").hide().removeClass("article").addClass("text").show();
```

建立第一個jQuery程式

```
01: <!DOCTYPE html>
02: <html>
03: <head>
04: <meta charset="utf-8"/>
05: <title>Ch8_3_1.html</title>
06: <style type="text/css">
07: .blue { color: blue; }
08: .red { color: red; }
09: </style>
10: <script
src="jquery2.1.1.min.js"> </script>
11: <script>
12: $(document).ready(function() {
13:     $('#list').addClass('red');
14:     $('#list li:last').addClass('blue');
15: });
16: </script>
17: </head>
```

```
18: <body>
19: <h2>建立第一個jQuery程式</h2>
20: <ol id="list">
21:     <li>Java Applet</li>
22:     <li>JavaScript</li>
23:     <li>VBScript</li>
24:     <li>DHTML</li>
25: </ol>
26: </body>
27: </html>
```



jQuery基本程式結構



- 在HTML網頁的jQuery程式碼是使用內建ready()方法註冊ready事件，以便在瀏覽器完成DOM建立後執行此事件的處理函數：

```
$(document).ready(function() {  
    .....  
});
```

- 上述程式碼的「\$」符號是jQuery物件的別名，它是JavaScript特殊的變數名稱
- \$()函數是呼叫建構函數建立jQuery物件，也就是jQuery()函數
- 參數document是HTML網頁本身，然後註冊ready事件，在括號中是事件處理程式碼，使用的是匿名函數 (Anonymous Function)。

jQuery程式敘述的基本語法



- jQuery程式敘述的基本語法是呼叫jQuery()函數，或別名\$()函數開始：

\$(選擇器字串).method(參數列);

jQuery(選擇器字串).method(參數列);

- 上述語法是呼叫\$()函數，以參數的選擇器字串取出符合條件的DOM物件，然後呼叫之後的方法進行處理。

jQuery程式敘述的基本語法



□ jQuery程式敘述主要可以分成三大部分，如下表所示：

選擇器	動作	參數列
<code>\$('#list')</code>	<code>.addClass</code>	<code>('red');</code>
<code>jQuery('#list')</code>	<code>.addClass</code>	<code>('red');</code>

- 1) 從HTML網頁的DOM中找出符合選擇器條件的元素
- 2) 使用動作的方法來替元素套用樣式、註冊事件或建立動畫等
- 3) 參數列是執行動作所需的參數

jQuery的程式架構-串聯



- 因為jQuery方法的傳回值是jQuery物件本身，可以使用串聯（ Chaining ）來依序呼叫多個方法

```
$('#list li:last').addClass('blue').prev().removeClass('red');
```

- 程式碼依序呼叫addClass()、prev()和removeClass()共三個方法，稱為串聯

jQuery回撥函數(說明)



- JavaScript程式碼是一列接著一列依序的執行，下一列程式碼一定需要等到前一列程式碼執行後，才能執行
- 「回撥函數」 (Callback Functions) (回呼函數)
 - 對一些需要長時間執行的操作，例如：動畫效果，不希望在動畫結束後，才能執行下一列指令

jQuery回撥函數(語法)



- 回撥函數是等到目前操作結束後自動呼叫的函數

- 例如：在動畫效果結束後，就呼叫此函數

- jQuery動畫效果都是使用這種程式結構

- `$(選擇器字串).動畫方法(播放速度, 回撥函數);`

- 上述語法可以替選擇元素套用動畫方法的動畫，第2個參數是回撥函數(ch11)

jQuery回撥函數(範例)

□ 例如：在<p>標籤套用hide()方法的隱藏動畫：

```
$("#p").hide("slow",function() {  
    document.write("已經隱藏這一段文字內容...<br/>");  
});
```

```
document.write("準備隱藏這一段文字內容...<br/>");
```

■ 方法的第2個參數是**匿名回撥函數**，當完成動畫效果後，就會顯示一個訊息文字，因為使用回撥函數，所以不會影響第2個document.write()方法的執行。

練習1

- 請開啟ch08-ex1.html,利用CSS與jQuery程式碼,以產生如下圖之表格:

奇數列:

底色:#1E90FF

字體大小:30px; 字體顏色: white;

偶數列:

底色: :#AFEEEE

字體大小:30px;

JavaScript	Dynamic HTML	Ajax
JSP	PHP	Java Applet
VBScript	ASP	Java Servlet
ASP.NET	ActionScript	Flash

Google Chrome的開發人員工具(1/3)



- 檢視HTML元素
- 檢視CSS樣式
- 主控台標籤頁
- JavaScript程式碼的偵錯

Google Chrome的開發人員工具(2/3)

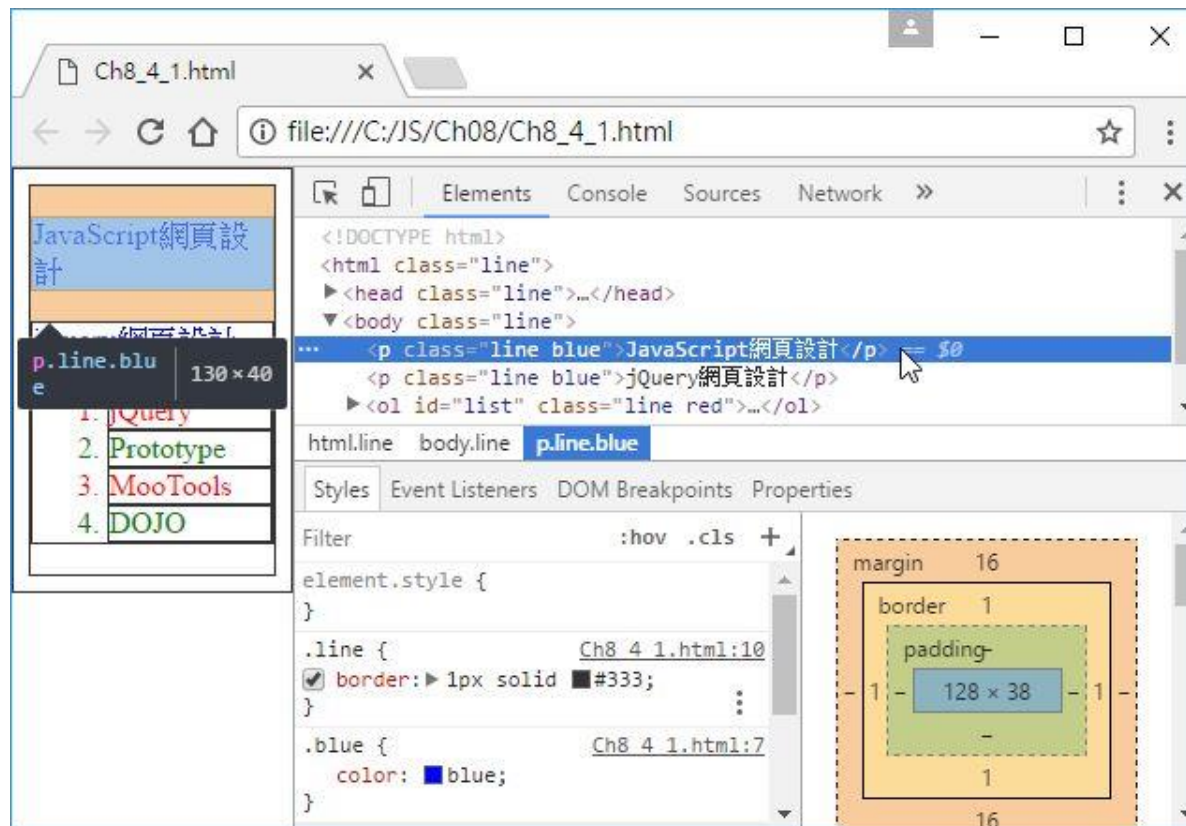


□ Google Chrome 瀏覽器內建功能強大的開發人員工具

- 執行程式除錯
- 即時檢視和編輯HTML元素與屬性
- 套用CSS樣式規則。

Google Chrome的開發人員工具(3/3)

- ❑ 在瀏覽器切換顯示開發人員工具
- ❑ 請啟動Chrome瀏覽器載入HTML網頁Ch8_4_1.html後，按F12鍵切換顯示開發人員工具



檢視HTML元素- Elements標籤頁

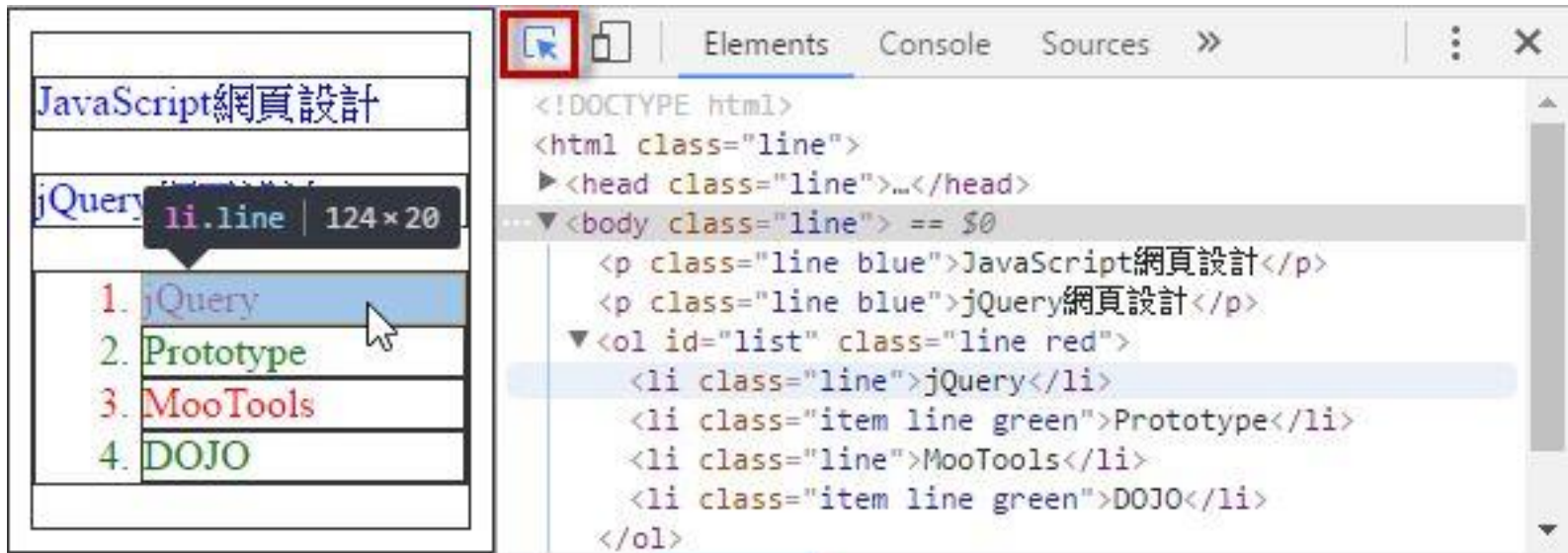
- 在開發人員工具選【Elements】標籤頁，可以顯示HTML元素的HTML標籤，我們可以在此標籤檢視HTML元素，如下圖所示：



檢視HTML元素- 選取HTML元素(1/2)

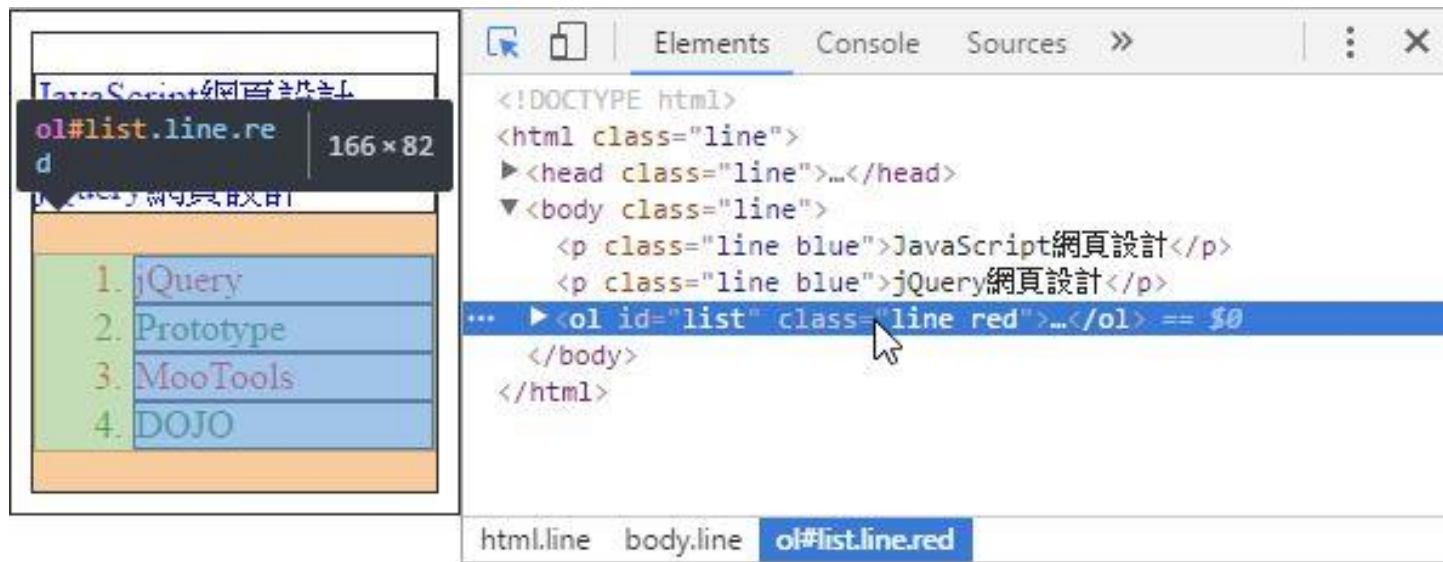


- 使用游標在網頁內容選取：請點選【Elements】標籤前方的箭頭鈕，就可以在網頁內容選取元素，當使用滑鼠移至欲選取元素的範圍時，就會在元素周圍顯示藍底，表示選取特定元素



檢視HTML元素- 選取HTML元素(2/2)

- 在Elements標籤選取：我們可以直接展開HTML標籤的節點來選取指定的HTML元素



檢視HTML元素- 更改HTML元素內容 (1/2)

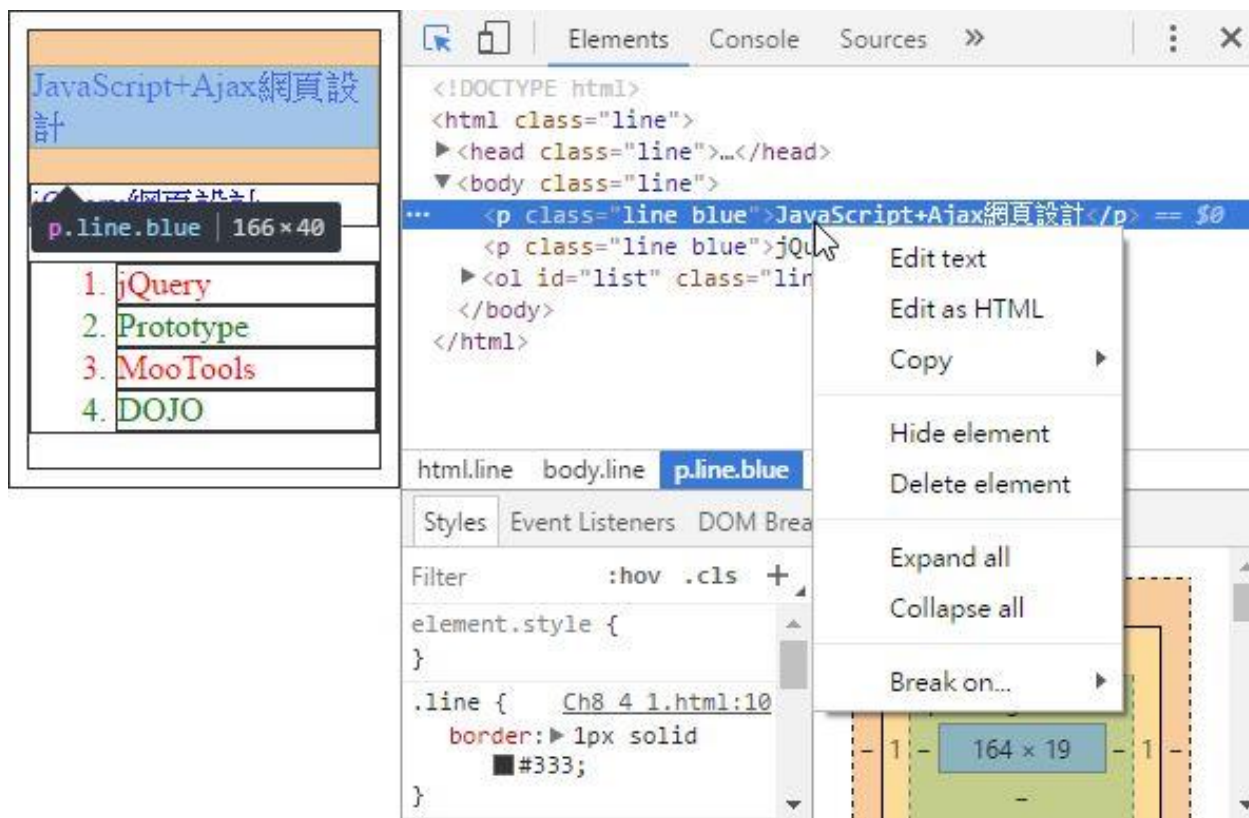


- 當選取HTML元素後，我們不只可以檢視標籤、屬性和其內容，還可以直接更改元素內容的文字節點

```
<!DOCTYPE html>
<html class="line">
  <head class="line">...</head>
  <body class="line">
    <p class="line blue">JavaScript+Ajax網頁設計</p>
    <p class="line blue">jQuery網頁設計</p>
    <ol id="list" class="line red">...</ol>
  </body>
</html>
```


檢視HTML元素- 更改HTML元素內容 (2/2)

- 在標籤上按滑鼠【右】鍵，可以看到快顯功能表編輯功能的相關指令



檢視HTML元素- 更改HTML屬性

- 在選取HTML元素後，只需按二下屬性，可以馬上更改標籤屬性，包含屬性名稱和屬性值（同樣不會影響原始程式碼），例如：更改class屬性值，可以馬上看到清單項目成為綠色



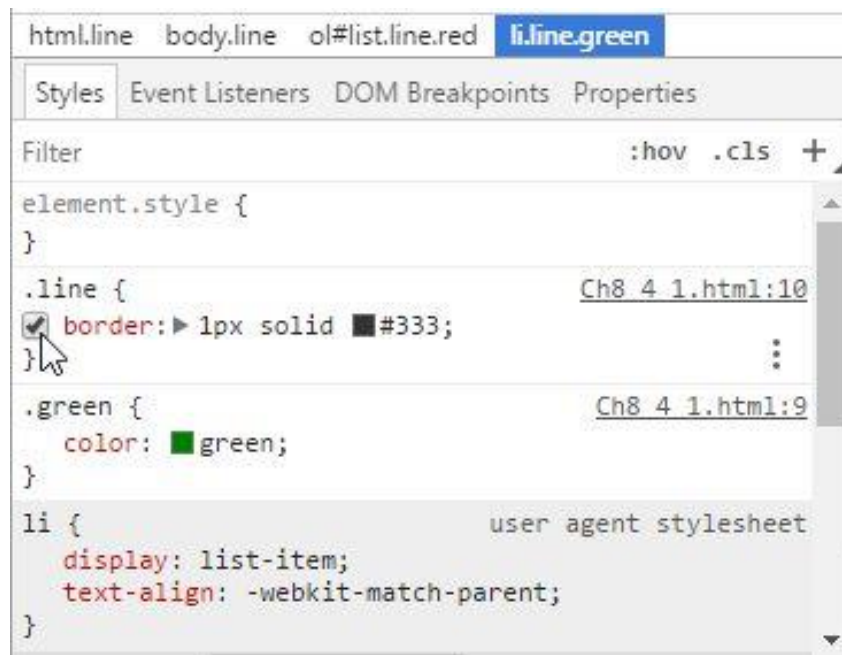
```
<!DOCTYPE html>
<html class="line">
  <head class="line">...</head>
  <body class="line">
    <p class="line blue">JavaScript+Ajax網頁設計</p>
    <p class="line blue">jQuery網頁設計</p>
    <ol id="list" class="line red">
      <li class="line green">jQuery</li>
      <li class="item line green">Prototype</li>
      <li class="line">MooTools</li>
      <li class="item line green">DOJO</li>
    </ol>
```

檢視CSS樣式- 說明

- 在Google Chrome開發人員工具選取HTML元素後，就可以在【**Styles**】標籤頁檢視樣式屬性，在此標籤頁可以顯示有哪些CSS樣式規則套用在選取的HTML元素，如果瀏覽器視窗夠寬，標籤是位在右邊；否則顯示在下方。

檢視CSS樣式- Styles標籤頁

- 在Styles標籤頁可以檢視標籤套用的樣式規則清單，只需選取HTML元素，就可以在下方看到套用樣式
- 樣式上如果有刪除線，表示樣式已經被覆寫，當移至樣式可以看到前方的核取方塊，取消勾選，即可關閉樣式（這些操作也不會影響原始程式碼）



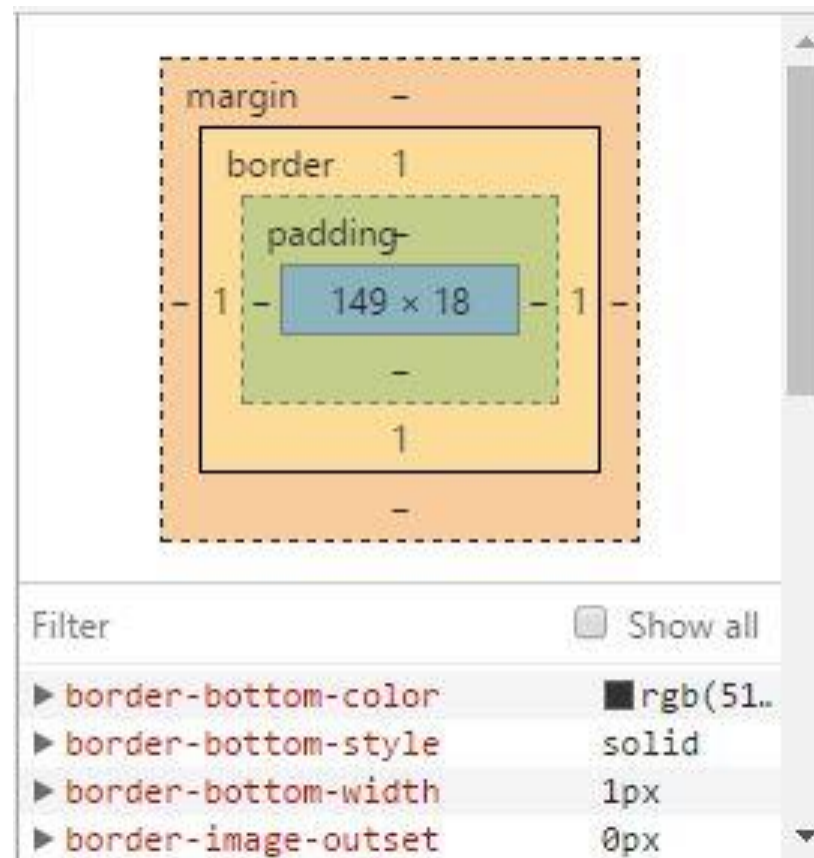
檢視CSS樣式- 檢視繼承的樣式規則

- 在【Styles】標籤頁捲動至下方，可以看到樣式套用繼承的上層元素樣式規則（Inherited from），幫助我們了解為什麼CSS樣式沒有作用
 - 例如：元素的色彩是綠色，而不是紅色，因為上一層元素的樣式被覆寫，



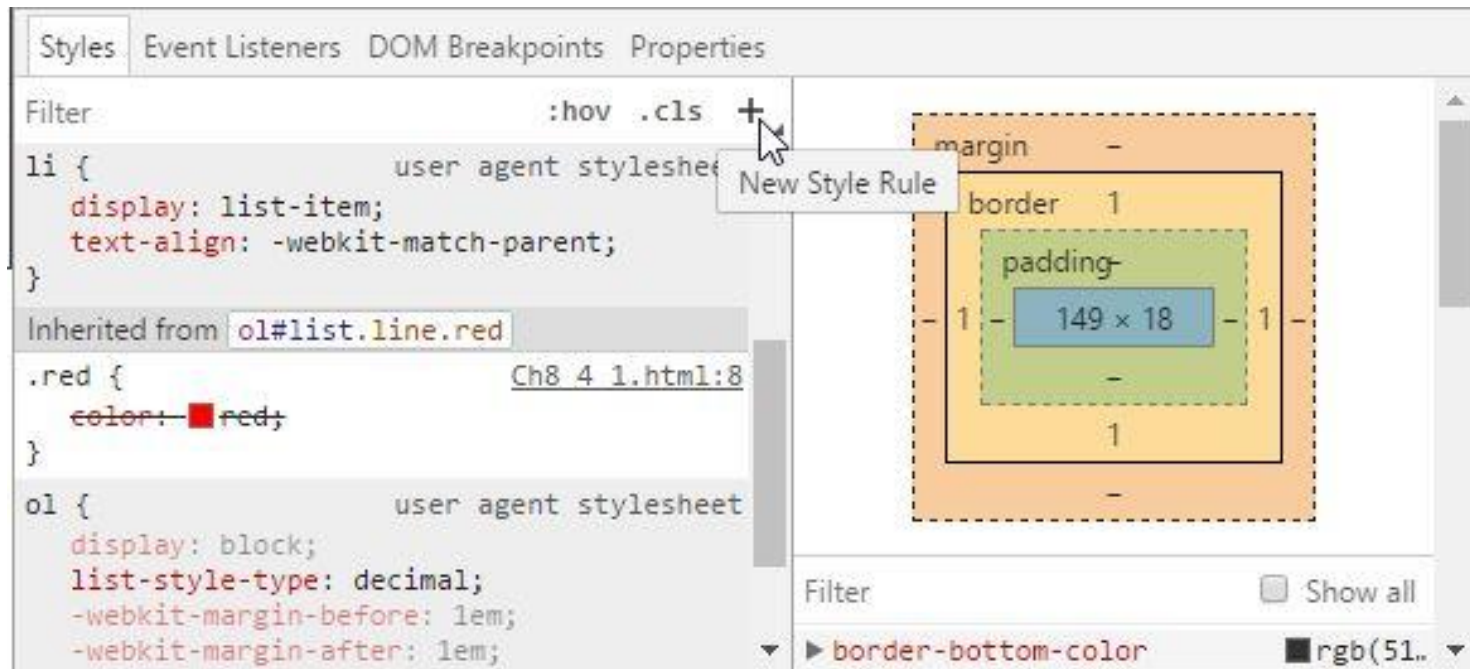
檢視CSS樣式- 元素樣式的方框模型 (Box Model)

- 在Styles標籤的最後或左邊可以看到視覺化顯示元素尺寸的方框模型，包含邊界、填充和尺寸等資訊



檢視CSS樣式- 新增樣式規則 (New Style Rule)

- 我們可以按左上方【+】鈕新增元素的樣式規則來測試HTML標籤的顯示效果（同樣不會影響原始程式碼）



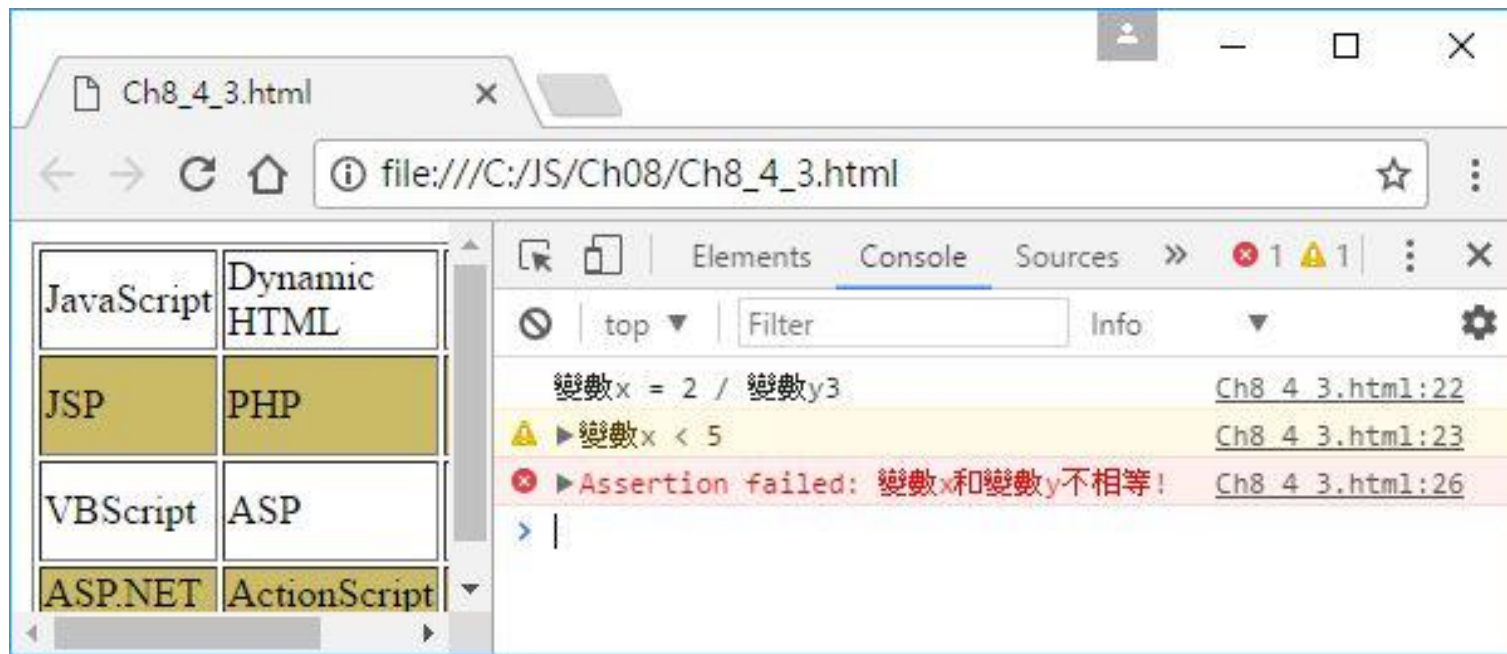


主控台標籤頁-console方法

- 如果在JavaScript程式中有使用console.log APIs的除錯程式碼，其輸入的內容就是在此標籤頁顯示，通常我們是用來顯示一些除錯資訊
 - console.log()
 - console.error()
 - console.warn()
 - console.info()
 - console.assert()
- 上述不同方法可以在主控台標籤頁顯示不同分類的訊息和圖示，方便我們進行程式碼的偵錯。

主控台標籤頁-圖例

- 例如：Ch8_4_3.html，訊息分別是使用 `console.log()`、`console.warn()`和 `console.assert()`方法顯示的除錯訊息



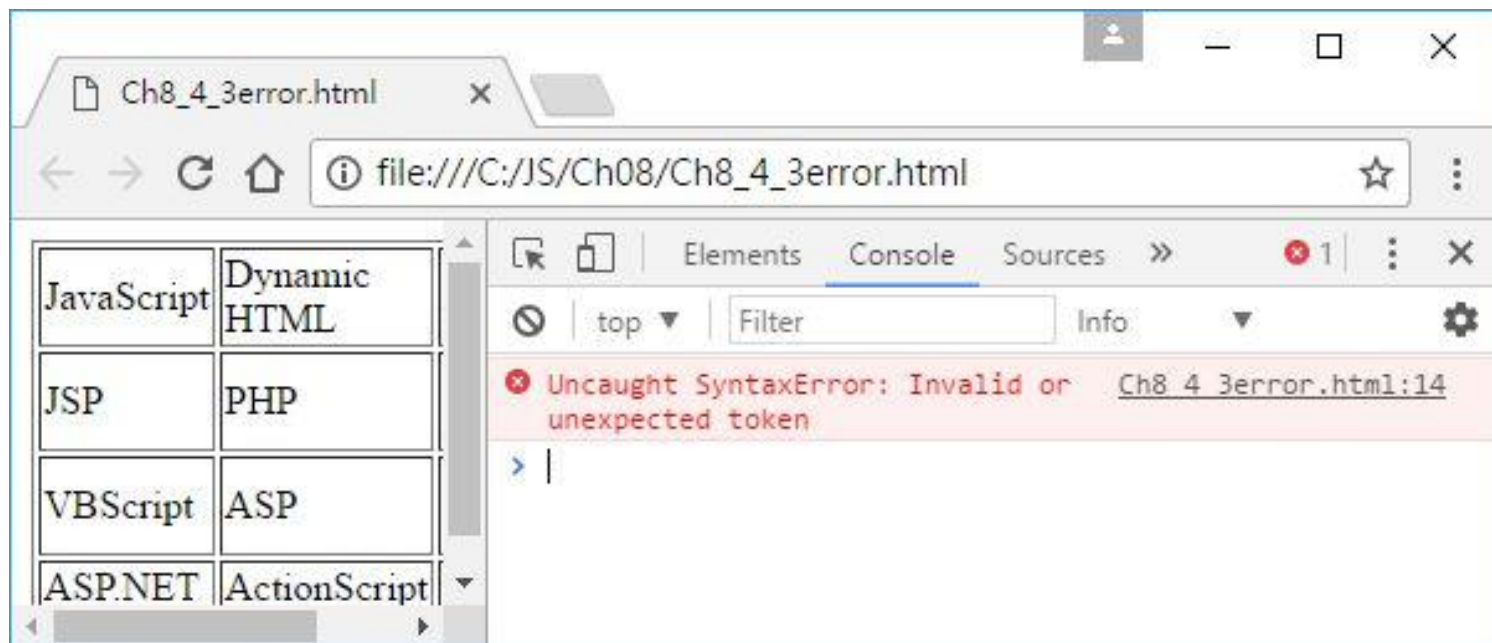


主控台標籤頁-程式碼

- `var x = 2;`
 - `var y = 3;`
 - `console.log("變數x = " + x + " / 變數y" + y);`
 - `console.warn("變數x < 5");`
 - `console.assert(x !== y, "變數x和變數y不相等!");`
 - `y = 2;`
 - `console.assert(x !== y, "變數x和變數y不相等!");`
-
- `console.log()`和`console.warn()`方法的參數就是顯示的訊息文字
 - `console.assert()`方法有2個參數，第1個參數是判斷條件，如果條件成立，才會顯示之後的訊息文字。

主控台標籤頁-顯示程式碼錯誤

- 在主控台標籤頁也可以顯示JavaScript或jQuery程式碼的錯誤，例如：載入Ch8_4_3error.html



JavaScript程式碼的偵錯-說明

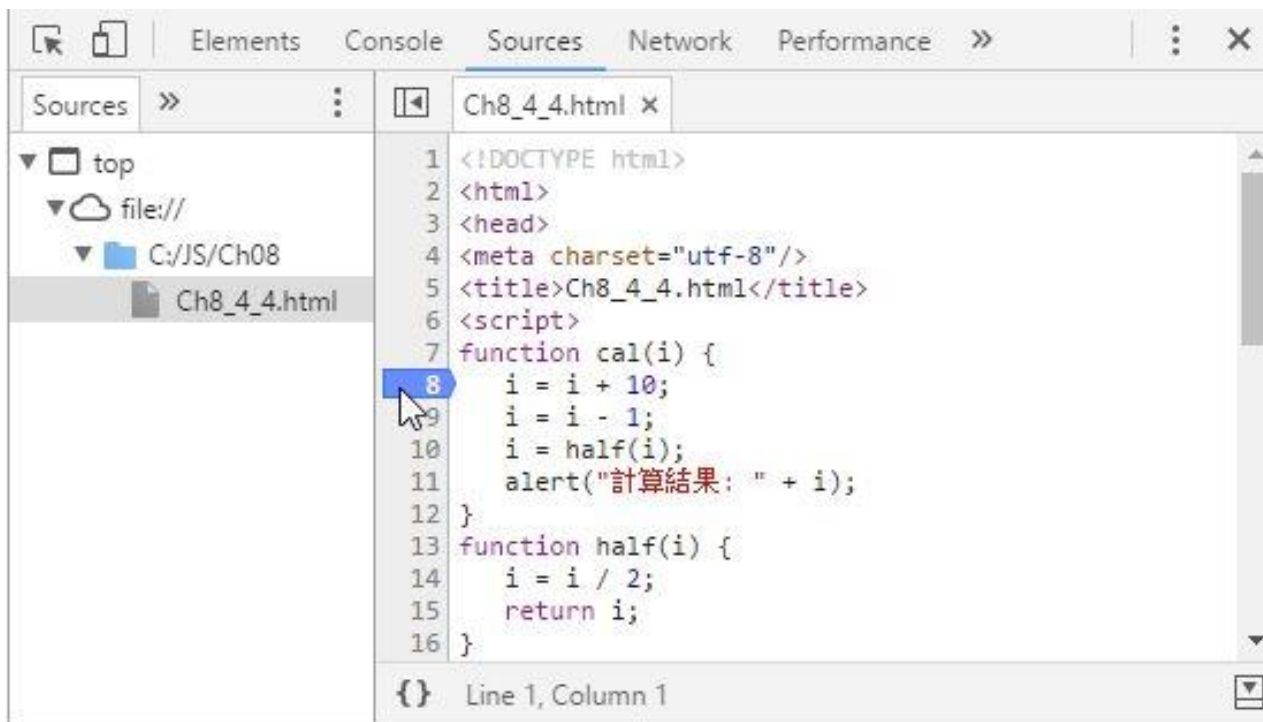


- Google Chrome的開發人員工具提供偵錯功能，
可以讓我們在程式碼新增中斷點（ Break Point ）
來執行JavaScript程式碼偵錯。

JavaScript程式碼的偵錯

□ 步驟一：在JavaScript程式碼新增中斷點

- 請啟動Google Chrome載入Ch8_4_4.html後，按F12鍵切換顯示開發人員工具。點選第8列的行號前方來新增中斷點，可以看到前方的藍色箭頭

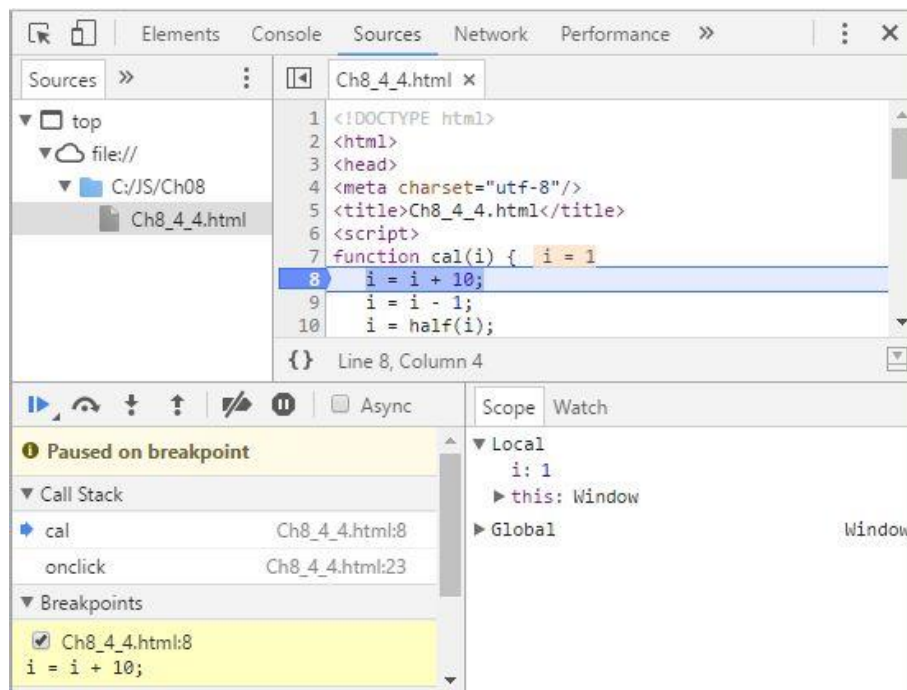


JavaScript程式碼的偵錯 (Step 1-2)



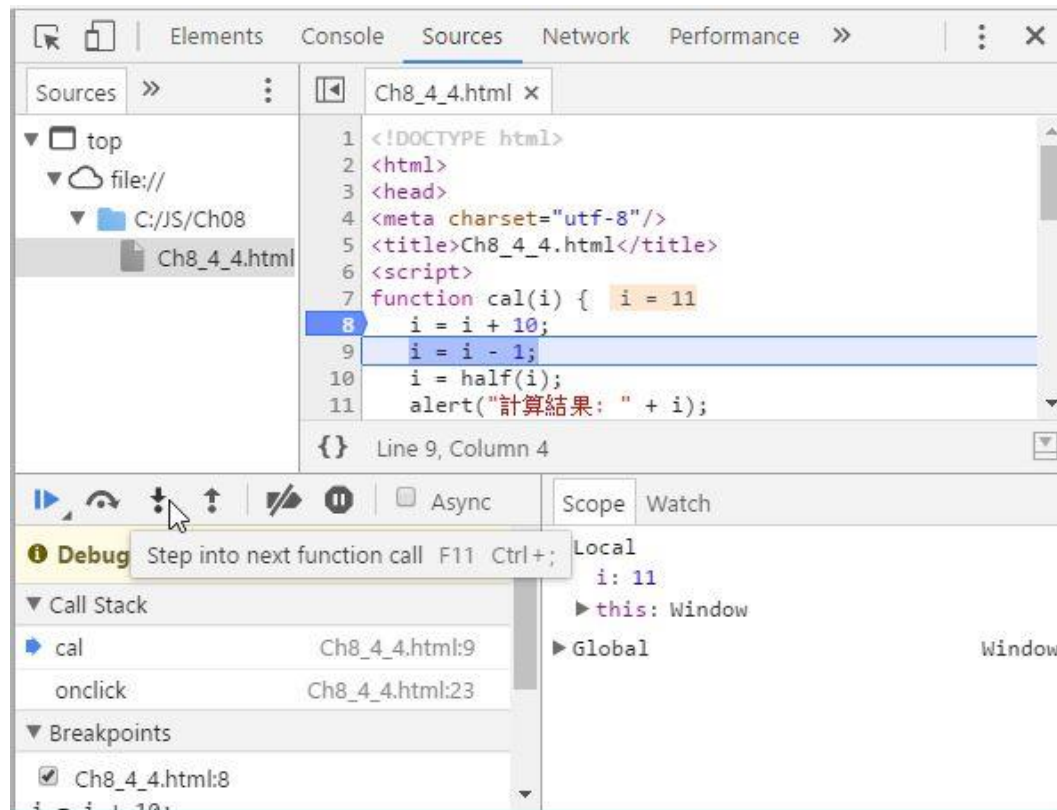
❑ 步驟二：執行JavaScript程式碼偵錯

- Step 1: 在Google Chrome 瀏覽器按上方工具列的【重新載入這個網頁】鈕重新載入網頁內容，且在網頁按【計算】鈕，可以執行JavaScript程式碼呼叫cal() 函數，參數值為1，因為有指定中斷點，所以執行程式是停止在中斷點的第8列
- Step 2: 在右下方【Scope】標籤顯示local區域變數i的值是1（變數i是cal()函數的參數）



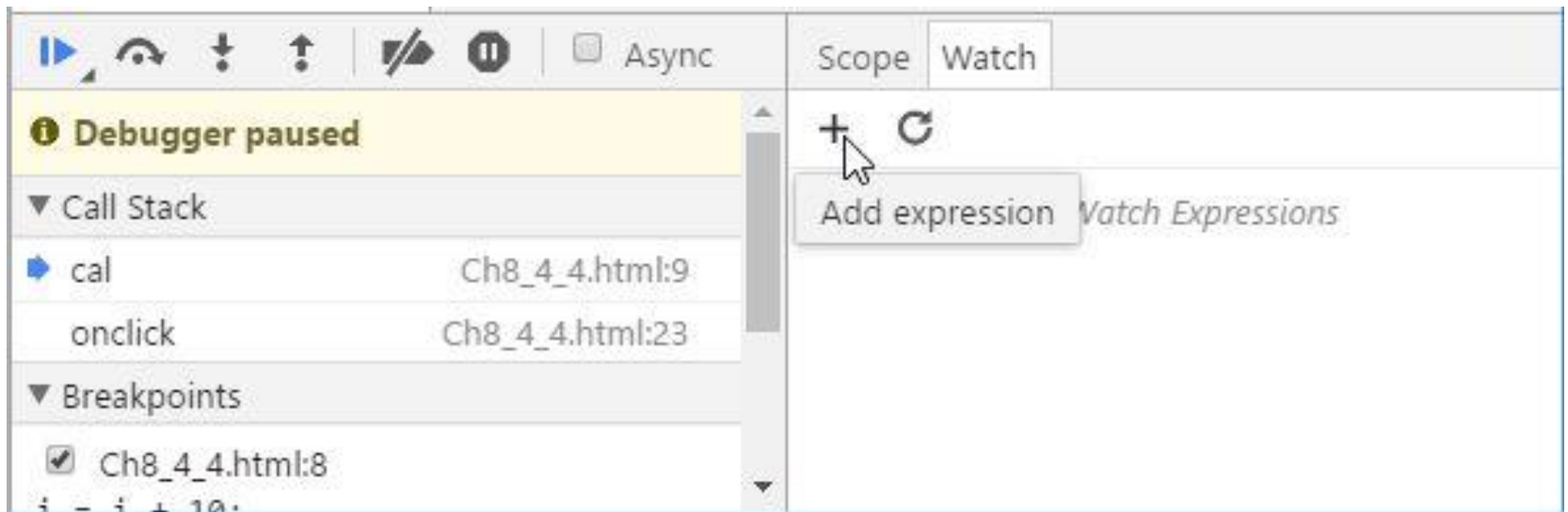
JavaScript程式碼的偵錯(Step 3)

- Step 3: 按F11鍵執行下一步至第9列，可以看到變數i的值成為11 (因為 $i + 10$)



JavaScript程式碼的偵錯(Step 4)

- Step 4: 在右下方選【Watch】標籤，可以新增監看的變數或運算式，位在其左邊上方是偵錯工具列的相關按鈕，如下圖所示：

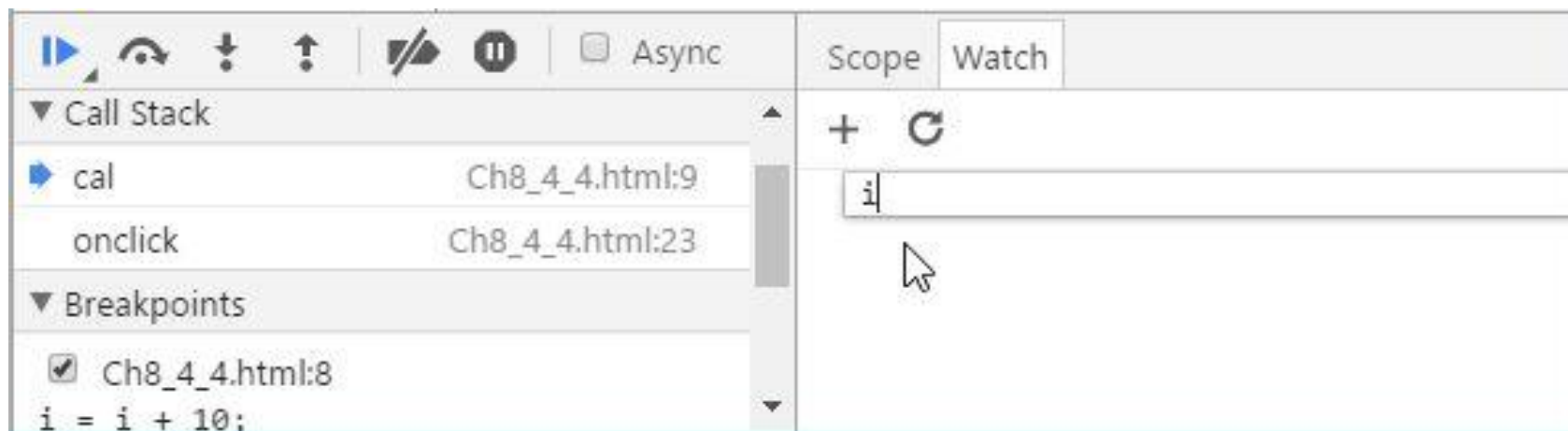


JavaScript程式碼的偵錯(Step 5)



□ Step 5: 執行JavaScript程式碼偵錯

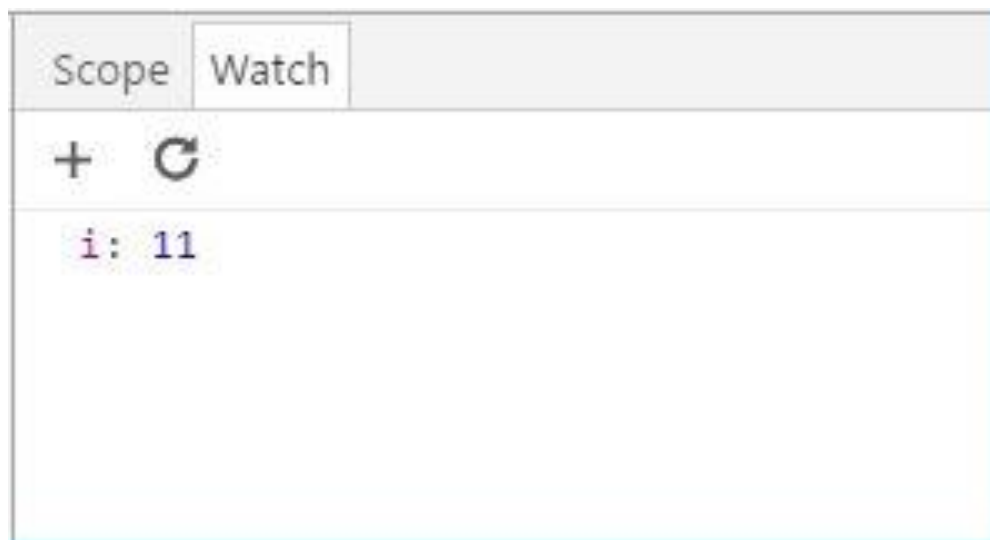
- 按【+】鈕新增監看的變數或運算式



JavaScript程式碼的偵錯(Step 6)



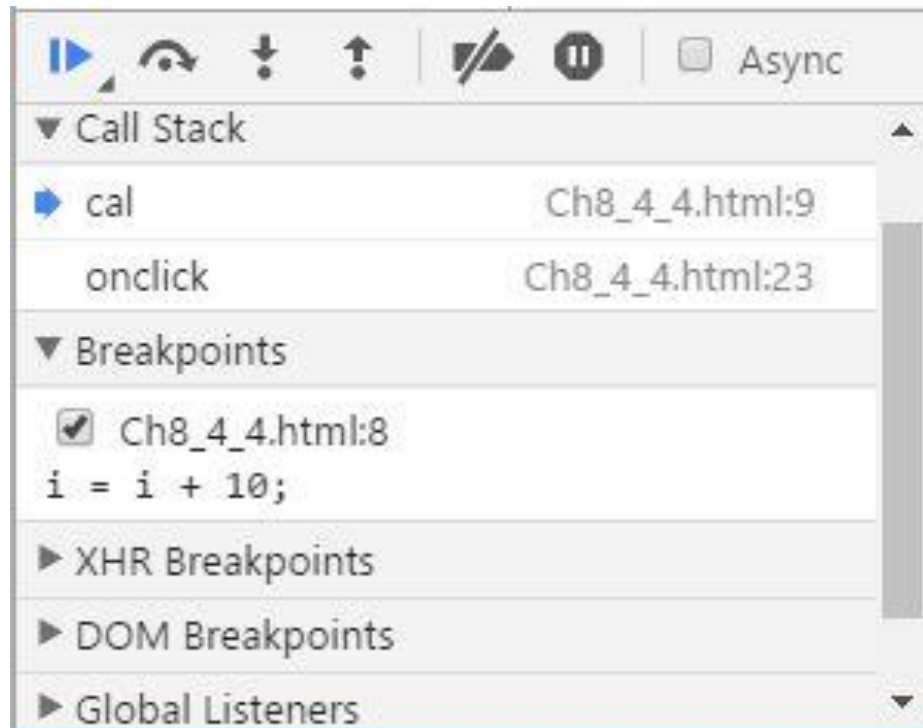
- Step 6：請輸入變數【i】，按Enter鍵，可以看到新增的監看變數i和目前值



JavaScript程式碼的偵錯(Step 7)

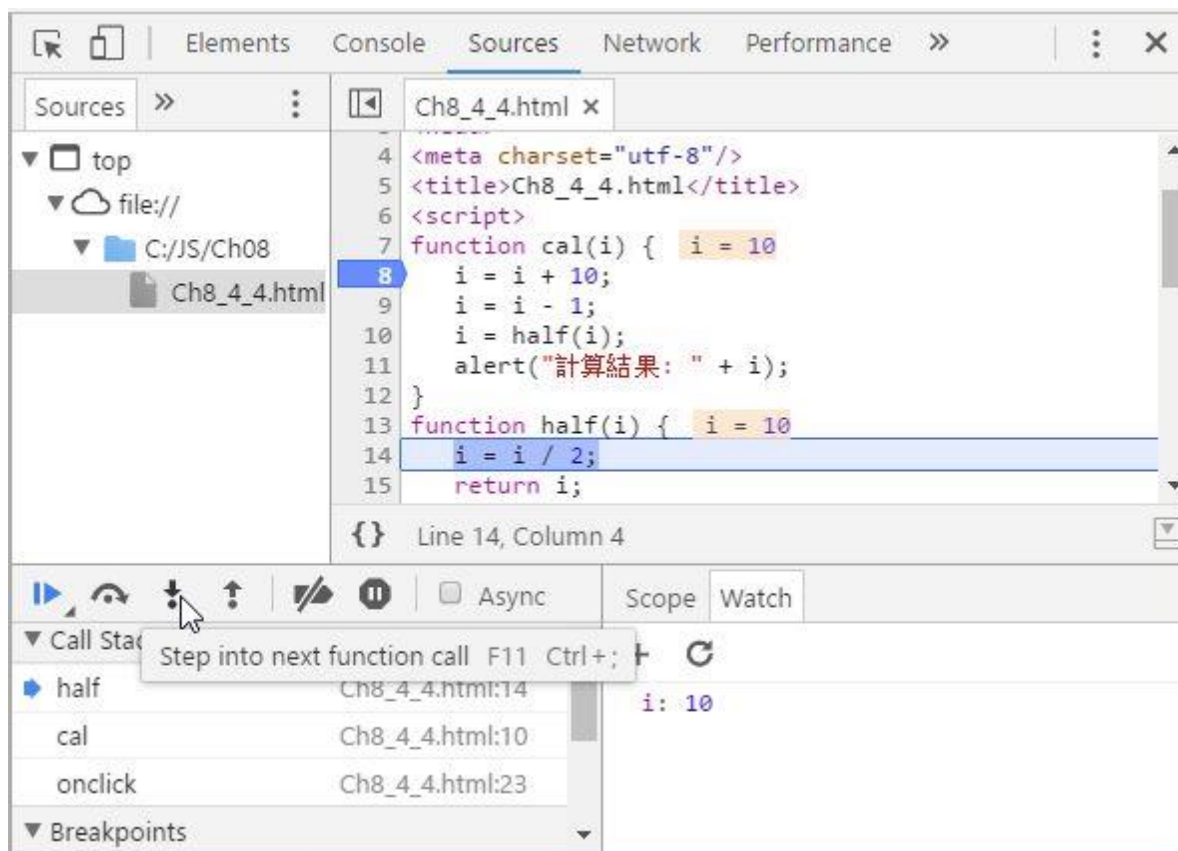


- Step 7：在左邊Call Stack（呼叫堆疊），可以看到呼叫cal()函數



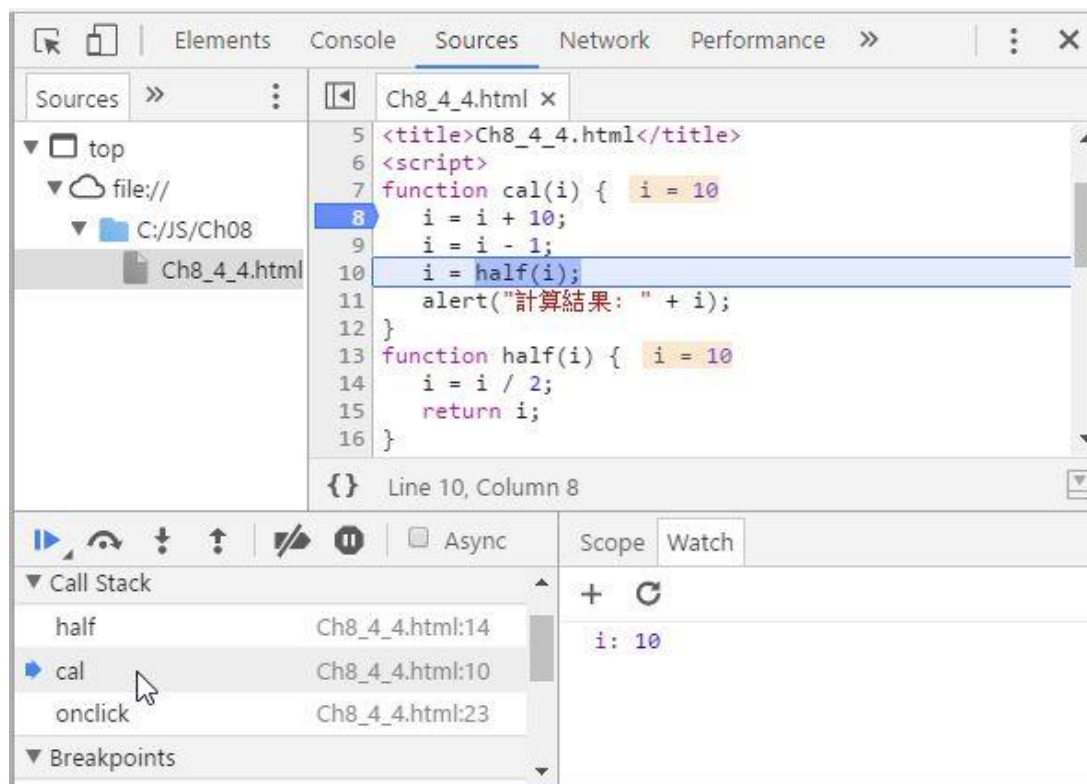
JavaScript程式碼的偵錯(Step 8)

- Step 8：按2次F11鍵執行第10列的函數呼叫，就會進入half()函數的第14列



JavaScript程式碼的偵錯(Step 9)

- Step 9：在左邊Call Stack標籤，可以看到cal() 函數再呼叫half()函數的函數呼叫過程，點選【cal】，可以看到反白顯示呼叫half()函數的哪一列



JavaScript程式碼的偵錯(Step 10)

- Step 10：請繼續按F11鍵執行下一列，可以逐步檢視或監看變數值，直到完成整個JavaScript程式碼的執行，最後可以看到顯示的訊息視窗

