10

$Cascade \ style \ sheets \ and \ JavaScript$

Min-Ruey You

Department of Computer Science and Engineering National Taiwan Ocean University

CONTENTS

10.1	CSS Introduction									
	10.1.1	.1 CSS Syntax								
10.2	CSS Selector									
	10.2.1	Categories	1							
	10.2.2	Simple Selectors								
		10.2.2.1 Element selector	1							
		10.2.2.2 ID selector	1							
		10.2.2.3 Class selector	2							
		10.2.2.4 Grouping selector	2							
		10.2.2.5 Universal selector	3							
10.3	BoxMo	99 del	3							
	10.3.1	The CSS Box Model	3							
	10.3.2	The Real Size of an Element	3							
10.4	CSS Pi	roperties	4							
	10.4.1	结語	4							
10.5	Javascr	ipt Introduction	4							
	10.5.1	0.5.1 Javascript 在網頁中的角色								
10.6	Hello World									
	10.6.1	Example	5							
10.7	Javascript Program									
	10.7.1	Variable Declaration	9							
	10.7.2	Data Type	9							
	10.7.3	變數的操作 Operator 99	9							
	10.7.4	Number Operator	9							
	10.7.5	String Operator 100	0							
10.8	Javascr	ipt Condition and Loop 100	0							
	10.8.1	FlowChart 100	0							
	10.8.2	Condition 102	2							
	10.8.3	Condition 102	2							
	10.8.4	金字塔練習102	2							

		10.8.4.1 參考答案	103
10.9	Javascı	ript Html Input	103
	10.9.1	HTML DOM Event	103
	10.9.2	從 Javaascript 中取得 Html Element	103
	10.9.3	聽取 Html 中發出的 Event	104
	10.9.4	更改 Html 的內容	104

\nobreak

10.1 CSS Introduction

CSS 是 Cascading Style Sheets 的缩寫,他可以幫助我們去定義 HTML 中每個 element 要如 何去顯示。舉例而言,可以定義每個文字的顏色、物件的外框、element 之間的距離,甚至撰寫 動畫等等。

由於我們要能透過 CSS 描述,"特定的 element"要用"特定的方式顯示",因此 CSS 的語法就可以間單的分為兩個重點:

- 1. selector 語法:用來指定 HTML 中,哪些 element 要套用這個樣式
- 2. 樣式描述:用來說明指定的 elements 要用什麼樣子去顯示

10.1.1 CSS Syntax

接下來我們來看看實際上的寫法吧! 說明圖是從 W3school 借來的。

一開始我們會先撰寫 selector ,指定我們之後定義的樣式要套用在那些 element 之上,接下來用大括號包覆地區域就是我們實際定義樣式的區域了。

樣式的定義上如範例所示,每種不同的樣式定義稱為一個 Declaration,每個 Declaration 之間用"分號"區隔。而每個 Declaration 又可以分為 Property 與 Value 的部分。



10.2 CSS Selector

CSS 用來選取 Html 中 element 的方式十分有趣,而且這個語法由於很好用,所以也常常被用 在其他的地方,例如用來撰寫網頁爬蟲的 BeautifulSoup 中也支援使用 CSS Selector 的語法來 取得想要拿到的 element (這個比傳統的 XPath 要好用超多的),另外 Javascript 中,也需要時 常透過更改 HTML element 的 attribute 來與使用者互動,因此 javascript 中,也有 jQuery 套件利用類似的方式來選取 HTML element,因此學會這方面的技術未來可以應用在很多不同 的地方。

10.2.1 Categories

基本上根據他們篩選 element 的方法,可以分為五類

- Simple selectors (select elements based on name, id, class)
- Combinator selectors (select elements based on a specific relationship between them)
- Pseudo-class selectors (select elements based on a certain state)
- Pseudo-elements selectors (select and style a part of an element)
- Attribute selectors (select elements based on an attribute or attribute value)

由於篇幅上面的限制,這邊就先只介紹 Simple selectors 剩下的同學可以去 W3school 上自己學習。

10.2.2 Simple Selectors

雖然叫做 Simple Selectors , 我們依然可以根據用來選擇 element 的條件來簡單的為她分類為:

- Element selector
- ID selector
- Class selector
- Grouping selector
- Universal selector

10.2.2.1 Element selector

```
我們可以在 selector 中寫上所想要選擇的 element 名稱來設定該類 element 的顯示樣式。
範例:
```

```
1 p {
            color: yellow;
3 }
5 div {
            height: 50px;
7 width: 60px;
}
```

10.2.2.2 ID selector

92

除了使用 element 名稱來指定樣式之外,我們還可以幫 element 設定 id attribute, 然後利用 這個 id 來設定不同的顯示樣式。 HTML:

<<u>div</u> <u>id</u>="block"></<u>div</u>>

CSS: 1 #block { height: 50px; width: 50px; color: red; 5 }

10.2.2.3 Class selector

Class selector 的使用方法與 ID selector 的使用方法刚好相反, Class selector 是先在 css 中先 定義好一個種類的樣式, HTML 中則是透過 class attribute 來選擇要使用哪一個種類的樣式顯 示。

CSS: 1 .redBlock { height: 50px;

```
3 width: 50px;
color: red;
```

5 }

HTML:

1 <<u>div</u> <u>class</u>="redBlock"></<u>div</u>>

* 這個範例的效果與 Id selector 的效果是相同的喔,同學可以根據自己的使用需求來決定要怎麼定義樣式。

10.2.2.4 Grouping selector

這種 selector 只是為了讓我們更方便定義樣式而存在。有時候我們會發現,我們會常常重複定 義著相同的樣式,可是需要套用到同一種 element 上面,因此我們常常會出現如以下的寫法:

```
1 p {
    color: red;
3 }
5 input {
    color: red;
7 }
9 div {
    color: red;
11 }
```

這時候 group selector 的語法就可以幫助我們了,我們可以改為這樣寫:

```
1 p, input, div {
    color: red;
3 }
```

Cascade style sheets and JavaScript

透過 group selector 的寫法可以省略很多重複的樣式定義。

10.2.2.5 Universal selector

這個也不多做解釋了,如字面意思,他可以針對所有的 element 進行樣式的定義。

1 * { color: red; 3 }

這樣所有的 element 都會套用到這個樣式。

10.3 BoxModel

在 CSS 中,每個 element 都可以被劃分為很多不同的區域,而這樣的模型我們稱為 BoxModel, 了解 BoxModel 對於網頁中一個元素要怎麼顯示在特定的位置是十分重要的。

10.3.1 The CSS Box Model



基本上一個 Html 元件會被分為這四個部分,分別為 Margin, Border, Padding, Content。

- Content:實際 element 內容所顯示的區域
- Padding: 位於外框 (Border) 與內容 (Content) 之間的空間
- Border: 顧名思義, 他就是 element 的邊框囉
- Margin:這個範圍是 element 與 element 之間的空間

10.3.2 The Real Size of an Element

實際上,我們透過 CSS 可以去設定的 height, width 屬性只是 BoxModel 中 Content 中的 height, width。實際整個 element 的寬高還要去加上 Padding, Border, Margin 這些元素才是 真正的元素大小喔。 舉例來說:

1 div {
 height: 200px;
3 width: 100px;
padding: 15px;
5 border: 5px solid black;
margin: 10px;
7 }

以這個例子來說, <div> 的高度會是 realHeight = height + padding + border + margin realHeight = 200 + 15 + 5 + 10 px realHeight = 230 px

10.4 CSS Properties

有鑑於內容與時間的問題,這邊就不詳細一個一個說明每個 CSS Property 的作用,這邊 同學有興趣直接拿一個 Property 來嘗試應該就會知道他有什麼效果了。實習課這邊一個 一個示範可能也會非常無聊,因此這邊提供一個網頁,同學進去尋找想要嘗試的樣式。CSS Reference(https://www.w3schools.com/cssref/default.asp)

10.4.1 結語

到這邊 CSS 的基本語法與概念的介紹就完成了,同學到這邊應該就會使用 CSS 的語法,只 是大家還不知道什麼時候要用什麼 Property 來達到想要的效果,這邊很難透過實習課告訴大 家,畢竟這算是經驗的累積,所以希望大家可以自己去透過上面的網站去嘗試,或是多看看如 CSS-TRICKS(https://css-tricks.com/) 之類的網站,看看某些帥氣的效果,別人是怎麼達 成的,藉此獲得更多的經驗值。

10.5 Javascript Introduction

在學習這個語言之前,我們先想想我們想要拿 Javascript 來做什麼呢?

10.5.1 Javascript 在網頁中的角色

網頁中,通常除了單純顯示內容之外,我們也希望網頁可以根據使用者的一些行為給予一些互動。舉例來說,註冊帳號的網頁中,如果我們在輸入手機的欄位中輸入了除了數字之外的字母, 網頁中要馬上跳出提示這個欄位中不可以出現數字之外的內容。諸如此類的效果,其中都少不了 Javascript。

接下來,我們來看一下, Javascript 要做到這樣的效果需要經過哪幾個步驟。

Cascade style sheets and JavaScript



- 1. 接收使用者的輸入內容
- 2. 根據輸入內容來做判斷
- 3. 將結果顯示在網頁上

首先 Javascript 一定要能從網頁中獲得使用者的輸入,這樣我們才可以根據他的輸入來 做判斷,接著他要能描述邏輯,這樣才可以處理使用者不同的行為要顯示不同的內容,最後 Javascript 要能更改網頁中的內容與樣式,讓使用者覺得網頁會隨著他的輸入而變化。 因此,接下來的教學會根據以下這三個步驟來解說。

- 1. Javascript 基本邏輯的撰寫
- 2. Javascript Html Event Input
- 3. Javascript 對於 Html 元件的操作

10.6 Hello World

在學習任何語言前,照慣例一定要先看看他的 Hello World 的寫法,從中學習基礎環境的架設。

10.6.1 Example

[開啟範例](example/HelloJavascript.html)

```
1 <!DOCTYPE html>
  <html>
3   <head>
        <title>Hello Javascript</title>
5        <meta charset="utf-8">
```

在這個範例中,大部分的內容是與 HTML 的 Hello World 是相同的。不同的地方在於,我們加入了一個新的 script element 。在 HTML 中 Javascript 的程式碼需要撰寫在 script element 中。

接下來,我們看看一個最簡單的 Javascript 程式碼:

console.log("Hello World");

這行程式碼會將 Hello World 送到 console 中顯示。那....console 在網頁中的哪裡呢?(這 邊以 Chrome 為範例) 同學以 Chrome 打開範例程式碼後,按下 F12 會跳出 DevTools。



選擇上方的 Console 可以跳到 Console 頁面,就可以看到剛剛我們印出來的結果。 除了透過 Console 標籤切換之外,也可在右上方的設定選擇 show console drawer 來將 Console 頁面固定至下方,這樣比較方便去觀看。

 $Cascade\ style\ sheets\ and\ JavaScript$



0	De	ev⊺o	ols - file:///C	:/Users/Chu	pei/Des	sktop/	Web%20Pa	ge%20Design	n%20	Tutorial/e	kample/HelloJ	avascript.htn	h			-	6]	×
L,	È (1	Elements	Console	Sour	rces	Network	Performanc	e	Memory	Application	Security	Audits	AdBlock					1
	He	0	top World	,	O	Filt	er			Defa	ult levels 🔻				Dock side	ø			
>	TH2	110	101 10												Focus debu	iqqee			
															Show cons	ole dra	wer		Esc
															Search		С	trl + Sh	nift + F
															Run comm	and	C	trl + Sh	nift + P
															Open file			C	Ctrl + P
															More tools				
															Shortcuts				
															Settings				F1
															Help				
														_					



10.7 Javascript Program

程式通常是接收一些輸入,然後根據程式的邏輯,返回一些輸出。因此在程式之中,我們會希望 能儲存我們輸入的數值,或是計算後的結果。而程式之中,用來儲存數值的元素,我們成為變數 (Variable)。

10.7.1 Variable Declaration

變數的宣告,也就是告訴電腦說,我想要產生一個變數來儲存數值,並且給他一個名子,讓我們 之後可以根據這個名子找到他。

```
1 \, var \, x = 5;
```

在這邊這樣的寫法,意思是:「建立一個變數,名子叫做 x ,並將數值 5 存入這個變數中」。

10.7.2 Data Type

變數根據裡面儲存數值型態的不同,可以分為很多不同的種類喔,Javascript 的變數,可以儲存:

- 整數 (integer)
- 浮點數 (float)(其實就是有小數點的數值啦)
- 字串 (String)
- 布林值 (Boolean)
- 物件 (object)

1	var	x	=	1;	//整數
	var	у	=	10.13;	// 浮 點 數
3	var	z	=	"Hello";	//字串
	var	р	=	true;	//布林值
5	var	0	=	<pre>{ firstName: 123 }</pre>	//物件

10.7.3 變數的操作 Operator

對於不同類型的變數,我們可以對他們進行的操作也不同喔! * 這邊要跟同學額外說明,剛剛上面我們常常看到 = 這個符號,這並不是數學上的等於 喔,他代表 Assignment,是專門把數值放入變數之中的符號。

10.7.4 Number Operator

對於數值類的變數,我們可以對他們套用加法 (+), 減法 (-), 乘法 (*), 除法 (/), 跟我們一般進 行的數值運算是相同的。

```
1 var x = 1;
3 x = x + 2;
   console.log(x);
                           // 3
5
   \mathbf{x} = \mathbf{x} - \mathbf{1};
   console.log(x);
                           // 2
7
9
   x = x * 12;
   console.log(x);
                           // 24
11
   x = x / 3;
13 console.log(x);
                           // 8
```

10.7.5 String Operator

字串的操作與數值上不同,操作上相較來說較為簡單,字串的操作主要是字串的串接(+);

```
1 var str = "Hello";
var blank = " ";
3 var world = "World";
5 var helloWorld = str + blank + world;
console.log(helloWorld); //Hello World
```

10.8 Javascript Condition and Loop

在學會將數值儲存起來之後,我們就可以來寫一些判斷啦!

10.8.1 FlowChart

相信應該有不少同學看過這種圖,我們稱為流程圖。其中主要是以方框來說明要做的事情,菱形 則是選擇條件。撰寫程式與撰寫流程圖是相同的,我們根據輸入的數值然後制定程式運行的流 程。

在程式的流程中,主要影響正個流程的走向有兩種不同的語法,分別稱為 Condition 跟 Loop。

 $Cascade\ style\ sheets\ and\ JavaScript$



10.8.2 Condition

在 Javascript 中,最常用的判斷語法為 if 判斷句,我們先來看看範例。

```
if (x == 0)
2 {
    console.log("x == 0");
4 }
else
6 {
    console.log("x != 0");
8 }
```

這邊的寫法代表了,如果 x 等於 0,就在 console 印出 "x == 0" 的字,其他的情況就在 console 印出 "x! = 0" 的字。

在這邊大家應該會注意到 == 這個比較運算,他在這邊代表了兩邊相等,除了等於之外, 還有大於與小於可以使用喔!

- ==(等於)
- >(大於)
- < (小於)
- >= (大於等於)
- <= (小於等於)

10.8.3 Condition

```
Loop 我們在程式中稱為迴圈,在條件達成的情況下,他就會不斷的做同一件事情。
舉例來說:
```

```
var x = 0;

while(x < 5)

4 {
            console.log(x);

6            x = x + 1;

}

8 // 0 1 2 3 4
```

一開始 x 是 0,如果 x 小於 5,我就會執行大括號內的指令。這樣他就會一直重複地做到
 印出 x 的數值,然後將 x + 1,直到 x 不小於 5。
 * 到這邊同學可以消化一下,然後我們再開始下面的練習。

10.8.4 金字塔練習

嘗試設計一個程式,當x == 1的時候,在 console 印出

1 *

當 x == 2 的時候,在 console 印出

10.8.4.1 參考答案

```
var x = 3;
 1
   var i = 0;
3 \text{ var } j = 0;
 5 while(i < x)
   {
 7
        j = 0;
9
        var output = "";
        while(j <= <u>i</u>)
11
        {
             output = output + "*";
13
             j = j + 1;
        }
15
17
        console.log(output);
19
        \underline{i} = \underline{i} + 1;
   }
```

10.9 Javascript Html Input

學習完如何撰寫基本的程式流程之後,我們來看看 Javascript 要怎麼知道使用者輸入了甚麼呢?

10.9.1 HTML DOM Event

在 HTML 中,當使用者使用鍵盤或是滑鼠對 element 進行操作的時候,element 其實會發出一個訊號,這個訊號中會記錄了使用者正在對 element 進行甚麼樣的操作。例如,當我們用滑鼠 點擊 element ,他會發出一個訊號說使用者使用滑鼠左鍵進行點擊。因此在獲取使用者的輸入 之前,我們要先取得我們要關注的 Html element 。

10.9.2 從 Javaascript 中取得 Html Element

在 Javascript 中,我們可以透過 Html Element 的 ID attribute 來取得 html 中的 element 喔。

HTML:

```
<div id="block" class="a"></div>
javascript:

var x = document.getElementById("block");

console.log(x.id); //block
console.log(x.className); //a
```

10.9.3 聽取 Html 中發出的 Event

104

在 javascript 中,我們可以利用 addEventListener 這個方法來取得 Html 中送出來的 Event。

```
<! DOCTYPE html>
   <html>
2
        <<u>head</u>>
            <<u>title</u>>Chupei</<u>title</u>>
 4
            <<u>meta</u> <u>charset</u>="utf-8">
        </<u>head</u>>
6
        <body>
             <<u>input id="btn" type="button" value="Click!!!"></u>
 8
10
             <<u>script</u>>
                 function clickHandler()
                  {
12
                       console.log("Click!!");
                  }
14
                  var button = document.getElementById("btn");
16
                  button.addEventListener("click", clickHandler, false);
            </<u>script</u>>
18
        </body>
20 </html>
```

* 這邊有幾點需要特別注意喔!! * 由於 Html 是直譯式語言,因此我們 getElementById 一 定要寫在要取得的 element 之後,這也是為什麼這邊的 script element 寫在 body 最後的原因。

10.9.4 更改 Html 的內容

我們透過 getElementById 取得 Html element 之後,我們可以透過他更改 element 的各種 attriubte 以及 element content 喔!!

```
舉例來說:
```

```
2 <!DOCTYPE html>
2 <html>
4 <fitle>Chupei</fitle>
4 <fitle>Chupei</fitle>
6 </head>
6 </head>
8 <finput id="btn" type="button" value="Click!!!">
10 <div id="board">
6 </div</pre>
```

 $Cascade\ style\ sheets\ and\ JavaScript$

```
12
             <<u>script</u>>
14
                  function clickHandler()
                  {
                       var board = document.getElementById("board");
16
                       board.innerHTML = "Click!!";
board.style = "color: red;";
18
                  }
20
                 var button = document.getElementById("btn");
22
                  button.addEventListener("click", clickHandler, false);
24
             </<u>script</u>>
        </body>
26 </<u>html</u>>
```

這邊我們透過 innerHTML 來改變 div element 中的內容,透過 style 改變這個 element 的顯示樣式 (style attribute)。