

# 動態網頁

Instructor : 黃語昕

email : [silvia.huang@gmail.com](mailto:silvia.huang@gmail.com)

Time : 2018-03-14



# Course Objectives

## I、HTML 基礎語法

- (I). 環境介紹
- (II). HTML概述
- (III). 畫面描述
  - 1.文字與段落的處理
  - 2.圖片
  - 3.HyperLink
  - 4.畫面控制--List、Table
  - 5.視窗控制--iframe
  - 6.互動網頁的配置—Form

## II、CSS (Cascading Style Sheet)

- (I). 什麼是CSS？
- (II). 語法定義
  - 1.定義標籤(tag)的方式
  - 2.定義類別(class)的方式
  - 3.定義識別(id)的方式
- (III). 語法介紹
  - 1.文字控制
  - 2.顏色、背景控制
  - 3.版面配置
  - 4.其他
  - 5.補充
- (IV). CSS3 的歷史與新特色

# Course Objectives

## III、JavaScript

### (I). What is JavaScript ?

1. 歷史
2. 特色
3. 用在哪裡?

### (II). Syntax Basic

1. 變數、資料型態
2. 輸出與輸入
3. 運算子
4. 迴圈敘述
5. 選擇性敘述
6. 函數

### (III). Object-Oriented

1. 物件
2. 事件
3. window物件
4. 內建物件
5. 自訂物件

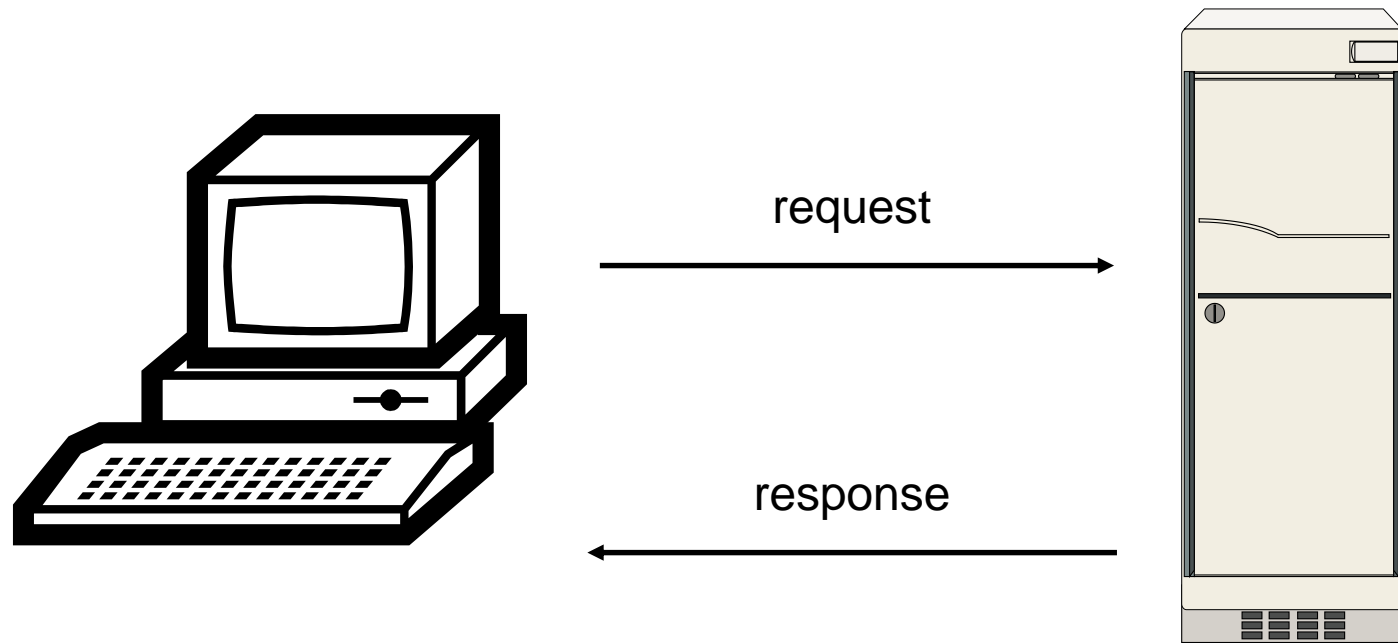
### (IV). 應用在HTML5上

# I. HTML

HyperText Markup Language

# (I). 環境介紹

## i. Internet環境



## ii. 教室環境

### 1. 軟體需求

- 瀏覽器(Browser) :

- Internet Explorer
- Mozilla Firefox
- Google Chrome
- Opera
- Safari

- 編寫工具(Text Editor) :

- Notepad++
- Sublime Text
- Visual Studio Code

### 2. JavaScript console

### 3. Web Server :

- Windows : 安裝Apache或IIS
- Mac或Linux已有內建Apache



## HTML5之後的 IE的處境



### iii. URL(Uniform Resources Locator)

◎ 在Internet上尋找資源的一種通用的方式

◎ <http://www.hinet.net/index.html>

包括了

- How : "http://" or "ftp://" ...

- Where : 網址(ex. [www.hinet.net](http://www.hinet.net))

- What : 網頁或其他資源(ex. [index.html](http://www.hinet.net/index.html))

### iv. HTML 4.01 Specification

<http://www.w3.org/TR/REC-html40/>

### v . W3C Working Draft, 6 February 2018

<https://www.w3.org/TR/html53/>

Editor's Draft, 5 February 2018

<http://w3c.github.io/html/>

HTML Tags + JavaScript APIs + CSS



HTML5

# 瀏覽器與支援現況

## • 支援的五大瀏覽器

- Internet Explorer
- Mozilla Firefox
- Google Chrome
- Opera
- Safari



## • 支援現況：

<http://caniuse.com/>

<http://fmbip.com/litmus/>

<http://html5test.com/>

## (II). HTML概述

◎ Hypertext Markup Language

◎ <http://www.w3.org>

◎ Tag (標籤)

• 分為一對標籤及單一標籤

一對標籤：<tag>...內容...</tag>

單一標籤：<tag> 或 <tag />

• **free-formate**：空白、大小寫、Enter皆不受影響

# HTML4--Skeleton (1/2)

```
<html>  
  <head>  
  </head>  
  <body>  
  </body>  
</html>
```

與Browser溝通的信息

網頁的內容

# HTML4--Skeleton (2/2)

```
<!DOCTYPE html PUBLIC
  "-//W3C/DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <meta http-equiv="content-type"
      content="text/html; charset=UTF-8" />
    <link rel="stylesheet" type="text/css"
      href="cssName.css" />
    <script language="javascript" src="xxx.js">
    </script>
  </head>
  <body>
    add the meat...
  </body>
</html>
```



# HTML5--Skeleton

```
<!DOCTYPE html>
<HTML>
  <HEAD>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="cssName.css">
    <script src="jsName.js"></script>
    <title>My Website</title>
  </HEAD>
  <BODY>
    add the meat...
  </BODY>
</HTML>
<!--template.html-->
```

# 標籤的屬性

為標籤加入更多的描述

[ex]：以**<hr>**為例

- 這條水平線有多寬？
- 多高(厚)？
- 什麼顏色？
- 什麼位置？(左、右or中)

```
<hr width ="屬性值" size ="屬性值"  
color ="屬性值" align ="屬性值">
```

### (III). 畫面描述

## { HTML4 . 01版本的重點 }

1. 文字與段落的處理
2. 圖片 -- **img**
3. **hyperlink**
4. 畫面控制 -- **list**、**table**
5. 視窗控制 -- **iframe**
6. 互動網頁的安排與配置 -- **form**

# 寫在前面：關於整個網頁

## **<body 屬性>.....</body>**

屬性	屬性值	說明	CSS
<b>bgcolor</b>	Color Name   rgb	背景顏色	取代
<b>background</b>	路徑 + 圖檔名稱	背景圖片	取代
bgproperties	fixed	浮水印	取代
text	Color Name   rgb	整頁文字的顏色	取代
link	Color Name   rgb	超連結的顏色	取代
vlink	Color Name   rgb	(visited) 連結過後的顏色	取代
alink	Color Name   rgb	(active) 按下連結的顏色	取代

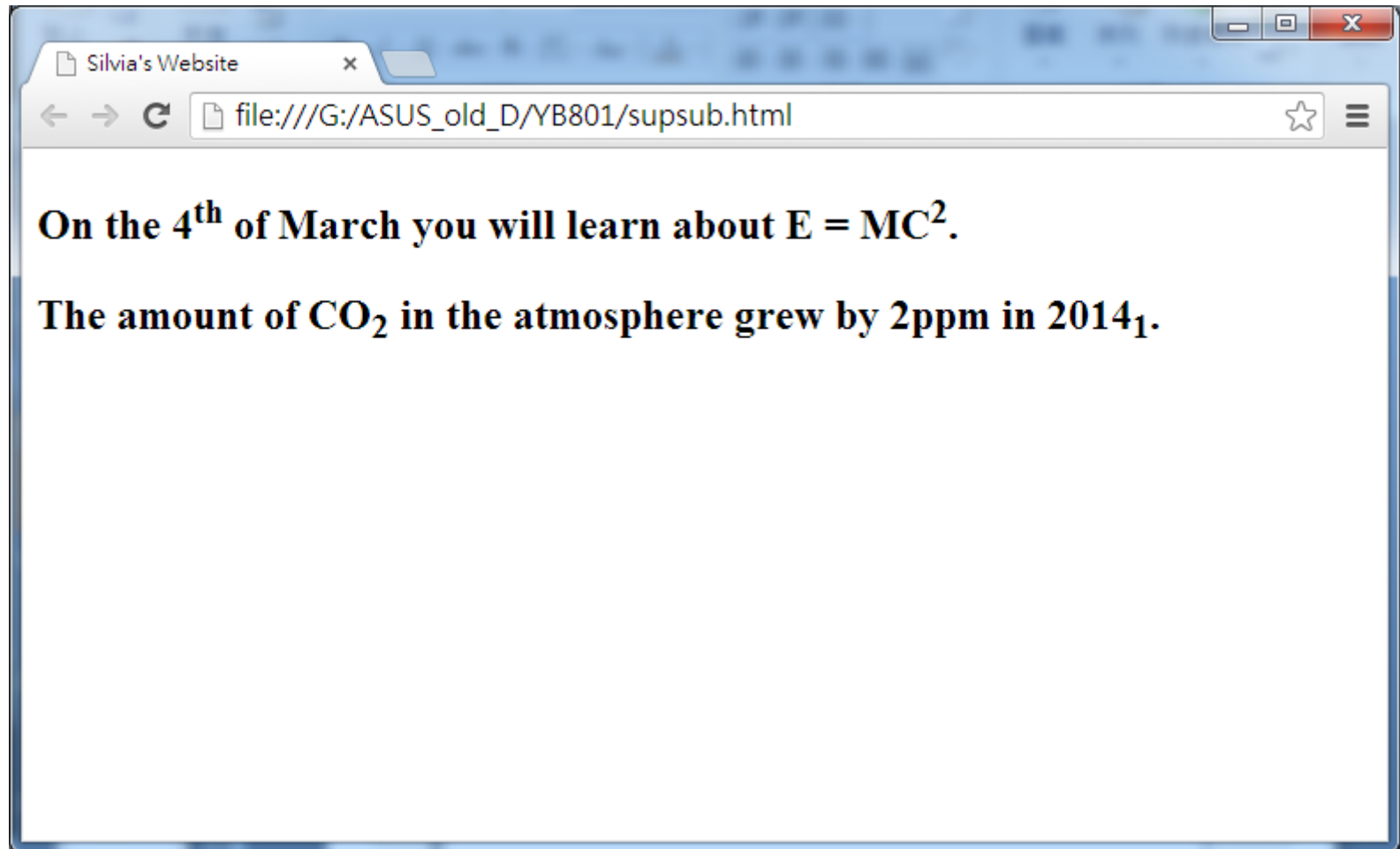
# 1.文字與段落的處理 (1 / 4)

<font 屬性>...文字內容...</font>			
屬性	屬性值	說明	CSS
<b>size</b>	1~7 (小→大) , default=3 +1~+4 , -1~-2	文字大小	取代
<b>color</b>	color Name   rgb	文字顏色	取代
<b>face</b>	標楷體、新細明體、Verdana、Tahoma...	字體	取代

# 1.文字與段落的處理 (2 / 4)

字型變化		
屬性	說明	CSS
<code>&lt;b&gt;...&lt;/b&gt;</code> <code>&lt;i&gt;...&lt;/i&gt;</code> <code>&lt;u&gt;...&lt;/u&gt;</code> <code>&lt;big&gt;...&lt;/big&gt;</code> <code>&lt;small&gt;...&lt;/small&gt;</code>	粗體 斜體 底線 字體相對變大 字體相對變小	取代
<code>&lt;sup&gt;...&lt;/sup&gt;</code> <code>&lt;sub&gt;...&lt;/sub&gt;</code>	上標 下標	取代
<code>&lt;strong&gt;...&lt;/strong&gt;</code> <code>&lt;em&gt;...&lt;/em&gt;</code> <code>&lt;blockquote&gt;...&lt;/blockquote&gt;</code> <code>&lt;abbr&gt;...&lt;/abbr&gt;</code> <code>&lt;cite&gt;...&lt;/cite&gt;</code> <code>&lt;dfn&gt;...&lt;/dfn&gt;</code> <code>&lt;ins&gt;...&lt;/ins&gt;</code> <code>&lt;del&gt;...&lt;/del&gt;</code> <code>&lt;s&gt;...&lt;/s&gt;</code>	加強 強調  縮寫 註解 定義	取代

## ◎ 做做看：supsub.html





# 1.文字與段落的處理 (3 / 4)

標籤	屬性	說明	CSS
<code>&lt;pre&gt;.....&lt;/pre&gt;</code>	<code>width</code>	原封不動的呈現	
<code>&lt;h1&gt;.....&lt;/h1&gt;</code>	<code>n=1~6</code> (大→小) <code>align</code>	標題文字Headings	取代
<code>&lt;p&gt;.....&lt;/p&gt;</code>	<code>align</code> 屬性值： <code>right left center justify</code>	分段 Paragraph	
<code>&lt;br&gt;</code>		換行 line Break	
<code>&lt;center&gt; ... &lt;/center&gt;</code>		置中對齊	取代
<code>&lt;!-- 註解 --&gt;</code>		註解	
<div><div><div>Tips</div><div><p><code>&lt;hr&gt;</code>通常用來做甚麼？</p><ol style="list-style-type: none"><li>1. 分隔頁面</li><li>2. 標示某些主題的結尾</li><li>3. 代表清單的開端</li><li>4. 分隔圖片</li></ol></div></div></div> <code>&lt;hr&gt;</code>	<code>align=right left center(default)</code> <code>width=percent pixels</code> <code>size=pixels(px)</code> <code>color=colorName rgb</code> <code>shade(default) noshade</code>	水平分隔線 Horizontal Rules	25

# 1.文字與段落的處理 (4 / 4) --特殊字元

畫面呈現	特殊字元	說明	畫面呈現	特殊字元	說明
[空白]	<b>&amp;nbsp;</b>	空白 (Non-Break Space)	€	<b>&amp;euro;</b>	歐元符號
<	<b>&amp;lt;</b>	小於 ( less than )	¥	<b>&amp;yen;</b>	日圓符號
>	<b>&amp;gt;</b>	大於 ( greater than )	¢	<b>&amp;cent;</b>	美元
&	<b>&amp;amp;</b>	And (ampersand)	£	<b>&amp;pound;</b>	英鎊
"	<b>&amp;quot;</b>	雙引號 (quotation mark )	♠	<b>&amp;spades;</b>	spade
©	<b>&amp;copy;</b>	copyright	♣	<b>&amp;clubs;</b>	club
®	<b>&amp;reg;</b>	registred trademark	♥	<b>&amp;hearts;</b>	heart
™	<b>&amp;trade;</b>	trademark	♦	<b>&amp;diams;</b>	diamond
§	<b>&amp;sect;</b>	section			

## 2. 圖片 (1 / 2)

### (1) . GIF、JPEG與PNG

	GIF (Graphics Interchange Format)	JPEG (Joint <b>P</b> hotographic <b>E</b> xperts <b>G</b> roup)	PNG (Portable Network Graphics)
透明背景	有	無	有
動畫效果	有	無	無
色彩量	256色	24bits	48bits (最高支援)
檔案大小	小	中	大
使用時機	·icon ·動畫 ·透明背景圖片	·照片 ·失真性壓縮 ·不支援透明度	·漸層又透明背景的圖片 ·非失真性壓縮格式 ·支援Alpha ·改善並取代GIF作為適合 網路傳輸的格式

## 2. 圖片 (2 / 2)

(2) . `<img src = "圖檔的路徑+檔名" >`

屬性	屬性值		說明	CSS
align	top middle bottom (default)	left right	對齊方式 (文繞圖)	
height	pixels (px)		圖片高度	
width	pixels (px)		圖片寬度	
border	pixels (px)		圖片邊框	
alt	說明文字		說明文字	
vspace	pixels (px)		(文繞圖)	
hspace	pixels (px)		(文繞圖)	

# 3. HyperLink

`<a> ... </a>`

屬性	屬性值
<b>href</b>	<p><code>&lt;a href="相對路徑和絕對路徑"&gt;...&lt;/a&gt;</code></p> <ul style="list-style-type: none"><li>◦ 網站內連結： <code>&lt;a href=" ../sample/box.htm"&gt;某網頁&lt;/a&gt;</code></li><li>◦ 跨網站連結： <code>&lt;a href=<a href="http://tw.yahoo.com">http://tw.yahoo.com</a>&gt;奇摩&lt;/a&gt;</code></li><li>◦ 本頁連結： <code>&lt;a href="#連結名稱"&gt;...&lt;/a&gt;</code></li><li>◦ email： <code>&lt;a href="<a href="mailto:silvia.huang@gmail.com">mailto:silvia.huang@gmail.com</a>"&gt;email&lt;/a&gt;</code></li></ul>
<b>name</b>	連結名稱(自行命名)
<b>target</b>	<code>_blank</code>   <code>_self</code> (default)

## 4. 畫面控制--List、Table (1 / 5)

### (1) . 符號清單 <ul>.....</ul>

標籤	屬性	子標籤	說明	CSS
<code>&lt;ul type=""&gt;</code> ... <code>&lt;/ul&gt;</code>	disk square circle	<code>&lt;li&gt;...&lt;/li&gt;</code>	符號清單為 無序列排序 <b>unordered list</b>	取代

### (2) . 編號清單 <ol>.....</ol>

<code>&lt;ol type=""&gt;</code> ... <code>&lt;/ol&gt;</code>	(數字) 1 A a I i start="9"	<code>&lt;li&gt;...&lt;/li&gt;</code>	編號清單為 有序列排序 <b>ordered list</b>	取代
--	---	---------------------------------------	---------------------------------------	----

## 4. 畫面控制--List、Table (2 / 5)

[ex] .

```
<list>
  <ul type="屬性">
    <li>Silvia</li>
    <li>Peter</li>
    <li>Tina</li>
  </ul>
</list>
```

output :

- Silvia
- Peter
- Tina

Tips

最外圍<list>...</list>  
目前大多都省略了^^

## 4. 畫面控制--List、Table (3 / 5)

### (3) . 定義清單 <dl>.....</dl>

標籤	子標籤
<code>&lt;dl&gt;</code> ... <code>&lt;/dl&gt;</code>	<code>&lt;dt&gt;.....&lt;/dt&gt;</code> 用來定義名詞 <code>&lt;dd&gt;.....&lt;/dd&gt;</code> 用來容納定義

### (4) . 巢狀清單

```
<ul>
  <li>AAA</li>
  <li>BBB
    <ul>
      <li>aaa</li>
      <li>bbb</li>
      <li>ccc</li>
      <li>ddd</li>
    </ul>
  </li>
  <li>CCC</li>
</ul>
```

- AAA
- BBB
  - aaa
  - bbb
  - ccc
  - ddd
- CCC



## 4. 畫面控制--List、Table (4 / 5)

<table>.....</table> 表格		
屬性	屬性值	說明
align	left (default)   center   right	對齊方式
border	pixels (px)	邊框
bordercolor	color-name   rgb	邊框顏色
width	percent (%)   pixels (px)	寬度
height	pixels (px)	高度
<b>cellspacing</b>	percent (%)   pixels (px) default = 2	格子 (cell) 與格子之間的距離
<b>cellpadding</b>	percent (%)   pixels (px) default = 1	內容與邊框之間的距離
bgcolor	color-name   rgb	表格的背景色
background	路徑+檔名	表格的背景圖

## 4. 畫面控制--List、Table (5 / 5)

`<table>.....</table>` 表格

`<tr>.....</tr>` 列

`<td>.....</td>` 欄

`<th>.....</th>` 欄的標題列

屬性	屬性值	說明
align	left center right	
valign	top middle bottom	
width	percent (%)  pixels (px)	
bgcolor	color-name rgb	
background	路徑+檔名	
<b>colspan</b>	數字	合併儲存格 (欄)
<b>rowspan</b>	數字	合併儲存格 (列)
wrap	(default)	
nowrap		

## ◎ 做做看：verticalLine.html



## 5. 視窗控制--**iframe**

### **<iframe>.....</iframe>** 內置框架

屬性	屬性值	說明
<b>src</b>	路徑+檔名	
name		
height	pixels	
width	pixels	
frameborder	0   1	
sandbox		
srcdoc	HTML_code	<code>&lt;iframe src="" srcdoc="&lt;p&gt;Hello!&lt;/p&gt;"&gt; &lt;/iframe&gt;</code>
scrolling	no   yes   auto	

## 6. 表單--Form (1 / 3)

<form>...</form>		
屬性	屬性值	說明
<b>action</b>	url   mailto	
<b>method</b>	get   post	
enctype	text/plain	
name		
autocomplete	on   off	HTML5
novalidate	novalidate	HTML5

## 6. 表單--Form (2 / 3)

```
<form>
  <input type="欄位型態"
        name="欄位名稱"
        value="傳回值或按鈕上的名稱">
</form>
```

欄位型態	其他屬性	說明	
text	size   maxlength	文字框	
password	size   maxlength	文字密碼框	
radio		單選	
checkbox		多選	
button		按鈕	
submit		傳送按鈕	
reset		重寫按鈕	

## 6. 表單--Form (3 / 3)

```
<form>
  <子標籤>...</子標籤>
</form>
```

子標籤	屬性	說明
select	name="欄位名稱" size="數字" multiple	下拉式選單 (drop-down menu)
<pre>&lt;select name=" "&gt;   &lt;option value="傳送值"&gt;...&lt;/option&gt;   &lt;option value="傳送值"&gt;...&lt;/option&gt;   &lt;option value="傳送值"&gt;...&lt;/option&gt; &lt;/select&gt;</pre>		
textarea	name="欄位名稱" rows="數字" cols="數字" wrap="off virtual physical"	文字輸入欄 (textbox)

# 7. HTML5表單(Form)強化版本 (1 / 2)

## <input type="欄位型態">

欄位型態	屬性值	說明
search	字串	搜尋
tel	字串	電話
url	絕對位址	網址
email	email	email
number	整數值	數字
range	數值	範圍
<b>datetime</b>	UCT時區的日期時間	日期時間
<b>date</b>	日期	日期
<b>month</b>	月份	月份
<b>week</b>	星期	星期
<b>time</b>	時間	時間
<b>datetime-local</b>	日期時間	本地日期時間
<b>color</b>	RGB	顏色
<b>file</b>	0~N個檔案	檔案
<b>image</b>	座標值	圖檔



## 7. HTML5表單(Form)強化版本 (2 / 2)

### **<input type="欄位型態" 屬性>**

屬性	說明
checked   selected	預選
readonly   disabled	不能選
required	必選
min   max	最小值   最大值
step	>0的數字或any
size	控制項大小
list	使用者預設選項
dirname	file名稱
multiple	可指定一個以上的值
maxlength   minlength	可輸入的最大(或最小)字元長度
autocomplete	列舉型態：on [sticky]   off [每次都重新輸入]   default
pattern	正規表示法
placeholder	簡短的提示文字

## 8. 補充標籤

標籤	說明
<code>&lt;button&gt;&lt;/button&gt;</code>	按鈕
<code>&lt;fieldset&gt;</code> <code>&lt;legend&gt;標題文字&lt;/legend&gt;</code> <code>&lt;/fieldset&gt;</code>	外框線
<code>&lt;label for=""&gt;&lt;/label&gt;</code>	
與CSS有關的標籤	
<code>&lt;style&gt;&lt;/style&gt;</code>	放在<head>標籤內
<code>&lt;link&gt;</code>	放在<head>標籤內
<code>&lt;div&gt;&lt;/div&gt;</code>	
<code>&lt;span&gt;&lt;/span&gt;</code>	

# II. CSS

## Cascading Style Sheet

<https://www.w3.org/Style/CSS/>

## 1. CSS 的發展時程：

1.0(1997) -> 2.0(1998/5) / 2.1 (2004)-> ->  
3.0(2010至今)

## 2. 特點

- 網頁的化妝師
- 可將資料和外觀分開

# (I).語法定義

# 1. 定義標籤(tag)的方式：

## (i) .inline

在HTML標籤裡面賦予**style**屬性

```
<p style="background-color:red;font-size:24px;">
```

## (ii) .embedded

將標籤的CSS屬性，定義在<style>...</style>標籤內。

<style></style>放在<head>...</head>

```
<style>
```

```
  p{
```

```
    background-color:red;
```

```
    font-size:24px;
```

```
  }
```

```
</style>
```

## (iii) .linked

將標籤的CSS屬性，定義在一個副檔名為.css的檔案裡，再用

```
<link rel="stylesheet" href="xxx.css">
```

把檔案include進html檔案內。

## 2. 定義類別(class)的方式：

定義方式： `.className` (className 自行命名)

引用方式： `<標籤 class="className">`

## 3. 定義識別(id)的方式：

定義方式： `#idName` (idName 自行命名)

引用方式： `<標籤 id="idName">`

## ◎小筆記：

- 何謂**selectors**
- 何謂**cascading**
- 同中求異
- <http://www.w3.org/Style/CSS/>



## (II). 語法介紹

## 0. 寫在前面--關於度量單位：

	CSS	說明
Pixel	<b>px</b>	像素(電腦畫面上兩點的距離)
Point	pt	一個point(點)是 1/72 英吋。 一個point是多少pixel，取決於螢幕解析度。 在72dpi的系統中，一個point等於一個pixel
Pica	pc	12個point
Em	<b>em</b>	字型的字體大小(font-size)。 若字型大小是16個pixel，則1 em = 16 pixel
En	en	½ em
X-height	ex	小寫*字母的高度
Inch	in	英吋(亨利八世的拇指長度)
Centimeter	cm	1/100 公尺
Millimeter	mm	1/1000 公尺

# 1. 文字控制：

## (i). 字型屬性

	屬性	屬性值
字型種類	font-family	字型名稱, .....
字型樣式	font-style	normal   <i>italic</i>   <i>oblique</i>
字型轉換	font-variant	normal   SMALL-CAPS
字型粗細	font-weight	normal   bold   bolder   lighter   100 (細) ~ 900 (粗)
字型大小	font-size	[長度] 24px [百分比] 110% [相對大小] larger   smaller [絕對大小] xx-small   x-small   small   medium   large   x-large   xx-large
總和屬性	<b>font</b>	所有字型屬性的設定值

## (ii).文字屬性

文字間隔	word-spacing	normal   [長度]
字母間隔	letter-spacing	normal   [長度]
修飾文字	<b>text-decoration</b>	none   underline   overline   line-through
垂直對齊排列	vertical-align	baseline   sub   super   top   text-top   middle   bottom   text-bottom   百分比
字母大小寫 轉換	text-transform	none   capitalize   UPPERCASE   lowercase
文字排列方式	<b>text-align</b>	left   right   center   justify
文字縮排	text-indent	[長度]   [百分比]
列高	line-height	normal   [數字]   [長度]   [百分比]

## 2. 顏色、背景控制：

前景顏色	color	[color-name] rgb(nr%, ng%, nb%)
背景顏色屬性	background-color	[顏色] transparent (透明)
背景圖案	background-image	[圖檔的 url] none
背景重複方式	background-repeat	repeat   repeat-x   repeat-y   no-repeat
固定背景	background-attachment (浮水印)	scroll   fixed
背景位置	background-position	[長度]   [百分比]   {[top/center/bottom] [left/center/right]}
背景總和屬性	<b>background</b>	所有背景屬性

### 3. 版面配置 -- 盒狀屬性：

#### (i) .邊界屬性 (outside border)

上邊界	margin-top	[長度]   [百分比]
右邊界	margin-right	[長度]   [百分比]
下邊界	margin-bottom	[長度]   [百分比]
左邊界	margin-left	[長度]   [百分比]
邊界總和屬性	<b>margin</b>	邊界的總和屬性

#### (ii) .填白屬性 (inside border)

上邊填白	padding-top	[長度]   [百分比]
右邊填白	padding-right	[長度]   [百分比]
下邊填白	padding-bottom	[長度]   [百分比]
左邊填白	padding-left	[長度]   [百分比]
填白總和屬性	<b>padding</b>	填白的總和屬性

### (iii). 邊框屬性 (border)

邊框樣式	border-style [補充]
	border-[ top   bottom   left  right ]-style
邊框顏色	border-color
	border-[ top   bottom   left  right ]-color
邊框寬度	border-width
	border-[ top   bottom   left  right ]-width

### (iv). 總和屬性

上邊框	border-top
下邊框	border-bottom
左邊框	border-left
右邊框	border-right
邊框總和屬性	<b>border</b>

### (iv). 補充

邊框樣式	none   solid   dotted   dashed   double   groove   ridge   inset   outset (3D凹線   3D凸線   嵌入線   隆起線)
------	---

## 4. 版面配置 -- 條列屬性：

項目類型	<code>list-style-type</code>	<code>none</code>   <code>decimal</code> <code>disc</code>   <code>circle</code>   <code>square</code> <code>lower-roman</code>   <code>upper-roman</code> <code>lower-alpha</code>   <code>upper-alpha</code>
項目圖像	<code>list-style-image:url()</code>	<code>url(圖檔的路徑+檔名)</code>   <code>none</code>
樣式位置	<code>list-style-position</code>	<code>inside</code>   <code>outside</code>
總和屬性	<b><code>list-style</code></b>	所有條列項目屬性設定



## 5. 版面配置 -- 定位屬性：

擺設位置	position	absolute   relative static(default)
視窗頂端距離	top	[長度]   [百分比]
視窗左端距離	left	[長度]   [百分比]
設定元件在網頁上的3d定位	z-index	[數字]

## 6. 其他：

顯示元件設定	<code>visibility</code>	<code>hidden</code>   <code>visible</code>
浮動元件	<b><code>float</code></b>	<code>left</code>   <code>right</code>   <code>none</code>
清除浮動元件	<b><code>clear</code></b>	<code>none</code>   <code>left</code>   <code>right</code>   <code>both</code>
元件寬度	<code>width</code>	[長度]   [百分比]
高度	<code>height</code>	[長度]   [百分比]
元件顯示	<b><code>display</code></b>	<b><code>block</code></b>   <code>inline</code>   <b><code>inline-block</code></b> <code>list-item</code>   <code>none</code>
空白	<code>white-space</code>	<code>normal</code>   <code>pre</code>   <code>nowrap</code>

# ◎ 做做看：fieldset.html

The screenshot shows a Windows Internet Explorer browser window titled "Access keys - Windows Internet Explorer". The address bar displays "C:\Users\Silvia\Desktop\accesskeys.html". The browser's menu bar includes "檔案(F)", "編輯(E)", "檢視(V)", "我的最愛(A)", "工具(T)", and "說明(H)". The toolbar shows several icons, including a search icon, a refresh icon, a home icon, a star icon, and a settings icon. The browser's address bar also shows several tabs: "HTML5 Forms Web Resourc...", "Online video tutorials & train...", "數位學習 - 教各種軟體使...", and "HTML語法教學".

The main content area displays a "User Registration Form". The form is divided into two sections, each enclosed in a dashed border:

- Personal Information**
  - Name:
  - Email Address:
  - Password:
  - Confirm Password:
- Address Details**
  - Address line one:
  - Address line two:
  - Town / City:
  - Zip / Post code:

Below the Address Details section is a "Sign Up!" button.

The Windows taskbar at the bottom shows several icons: a blue "S" icon, a purple icon, a red icon, a blue icon, an orange icon, a blue icon, and a green icon.

## ◎ 做做看：positionLimes.html



# (III). CSS3

# CSS3 的新特色

- 更多的選擇器 (**selectors**)
- 圓角 -- Radius
- 顏色的更多詮釋 -- RGB &RBGA
- 陰影 (盒狀陰影與文字陰影) -- Shadow Effects
- 漸層 -- Gradient
- 變形 -- Transformation
- 動畫 -- Animation
- 多欄位 -- Multiple Columns

## (IV). CSS3的語法

# CSS3：語法格式與套用方式

- 同CSS2

**<link rel="stylesheet" href=" ">**

- HTML+CSS+JavaScript 小工具：

<http://codepen.io/pen/>



# CSS3 進階--選擇器 { selectors } (1/4)

·What is selector(選擇器)?

Basic selectors	
<b>tag</b> {.....}	<b>h1</b> {color:green;}
<b>.className</b> {.....}	<b>.myClass</b> {color:green;}
<b>#idName</b> {.....}	<b>#myId</b> {color:green;}
<b>*</b> {.....}	<b>*</b> {color:green;}
Attribute selectors (屬性選擇器)	
[語法] : <b>selector</b> [屬性]{CSS的屬性}	
<b>selector</b> [ <b>attribute</b> ]	
<b>selector</b> [ <b>attribute</b> = "value"]	
<b>selector</b> [ <b>attribute</b> *= "value"]	
<b>selector</b> [ <b>attribute</b> ^= "value"]	
<b>selector</b> [ <b>attribute</b> \$= "value"]	

# CSS3 進階--選擇器 { selectors } (2/4)

- pseudo-classes(虛擬類別選擇器)

[語法] : **selector:pseudo-class**

selector : <b>nth-child(n)</b>	
selector : <b>nth-child(odd)</b>	
selector : <b>nth-child(even)</b>	
selector : <b>first-child</b>	
selector : <b>last-child</b>	
selector : <b>nth-of-type(n)</b>	
selector : <b>only-child</b>	
selector : <b>only-of-type</b>	
selector : <b>empty</b>	

# CSS3 進階--選擇器 { selectors } (3/4)

- pseudo-classes(其他的虛擬類別)

<b>selector:pseudo-class</b>	意義
selector : <b>link</b> selector : <b>visited</b> selector : <b>active</b>	
selector : <b>hover</b>	
selector : <b>enabled</b> selector : <b>disabled</b>	
selector : <b>checked</b> selector : <b>selected</b>	
selector : <b>not ()</b>	

# CSS3 進階--選擇器 { selectors } (4/4)

## •Selectors API

- `querySelector()`
- `querySelectorAll()`

# CSS3的前置詞

- 目前的標準尚未完全底定，根據瀏覽器廠商研發功能的進度不同，所以需要加上前置詞

<http://www.w3.org/Style/CSS/current-work>

IE	Fifefox	Chrome	Opera	Safari
-ms-	-moz-	-webkit-	-o-	-webkit-

# CSS3 進階--背景、邊框、影像 (1/6)

## ·背景(background) 屬性：

屬性	屬性值
background-clip {根據指定的位置裁切背景圖}	padding-box {以padding為參考點進行裁切} border-box {以border為參考點進行裁切} {default} content-box {以文件內容為參考點進行裁切}
background-origin {設定背景圖片的起始點}	padding-box {default} border-box content-box
background-size	contain {調整圖片大小以放入容器的區域} cover {調整圖片大小以覆蓋容器的區域} length:寬度 高度; {若沒設定則為auto} percentage:50% 50%; {若沒設定則為auto}

# CSS3 進階--背景、邊框、影像 (2/6)

## ·邊框(borders) 屬性：

{圓角邊框}屬性	屬性值
border-radius	<b>length</b>   % {半徑參數} ex. border-radius: 25px; border- <b>top-left</b> -radius: 25px; 水平半徑 垂直半徑 ex. border-radius: 25px 45px;

{圖片邊框}屬性	意義[預設屬性]	
<b>border-image</b>	總和屬性	
border-image-source	來源[none]	url()
border-image-slice	裁切方式[100%]	n%、數字
border-image-width	寬度[1]	
border-image-outset	超出邊框的量 [0]	
border-image-repeat	重複模式[stretch]	round、repeat

# CSS3 進階--背景、邊框、影像 (3/6)

## ·邊框陰影與文字陰影(shadow)屬性：

屬性	屬性值
<code>box-shadow</code> {邊框陰影}	<b><code>box-shadow: x y blur spread color inset;</code></b> 水平偏移 垂直偏移 模糊 擴散 顏色 內外部陰影 -水平偏移:距離[正值為右,負值為左] -垂直偏移:距離[正值往下,負值往上] -模糊:陰影的模糊度,為 $\geq 0$ 的數,數字越大越模糊[可省略] -擴散:陰影的擴散度,正值往外擴散,負值往內[可省略] -顏色:陰影顏色 -內外部陰影: <code>inset</code> [將陰影切換為內部陰影] ex. <code>box-shadow:rgb(110,110,110)10px 10px 10px;</code> <code>box-shadow:rgba(110,110,110,.7)10px 10px 10px;</code> 多重陰影: <code>box-shadow:10px 10px 30px red,20px 20px 30px blue;</code> <code>box-shadow: 0px 0px 30px red, 0px 0px 30px yellow;</code>
<code>text-shadow</code> {文字陰影}	水平偏移 垂直偏移 模糊 擴散 顏色 內外部陰影 {同上}



# CSS3 進階--背景、邊框、影像 (4/6)

## ·鏡射陰影(reflect)屬性：

屬性	屬性值
<code>box-reflect</code> {鏡射}	方向 間距 遮罩 -方向: above   below   left   right -間距: 數值 -遮罩: url()   搭配漸層屬性
[ex]: <code>-webkit-box-reflect: below 10px;</code> (反射在下方距離10px)	
<code>box-reflect: below 0 -webkit-linear-gradient(top, rgba(250,250,250,0), rgba(250,250,250,.0) 30%, rgba(250,250,250,0.3));</code>	

## ·透明度(opacity)屬性：

屬性	屬性值
<code>opacity</code> {透明度}	0~1 [1:預設值,表示不透明]

# CSS3 進階--背景、邊框、影像 (5/6)

## ·線性漸層(linear-gradient)屬性：

屬性	屬性值
linear-gradient	<b>degree或direction,color-stop1,color-stop2</b> - <b>direction</b> : left (0deg) right (180deg) top (270deg和-90deg) (default) bottom (90deg和-270deg) top left   top right   bottom left   bottom right - <b>color-stop1</b> : color 0% - <b>color-stop2</b> : color 100%
repeating-linear-gradient	同上

# CSS3 進階--背景、邊框、影像 (6/6)

## ·放射狀漸層(radial-gradient)屬性：

屬性	屬性值
radial-gradient	<code>(center, shape, colorStop, ..., colorStop)</code> (水平, 垂直, 形狀, 半徑, <i>color-stop1</i> , <i>color-stop2</i> ) 水平: <code>left(0%)   center (50%)   right(100%)</code> 垂直: <code>top (0%)   center (50%)   bottom (100%)</code> - <b>shape</b> : <code>ellipse</code> (預設)   <code>circle</code> - <b>shape size</b> : <code>closest-side</code> 最近端 <code>closest-corner</code> 最近角 <code>farthest-side</code> 最遠端 <code>farthest-corner</code> 最遠角 <code>contain</code> 包含 <code>cover</code> 覆蓋
repeating-radial-gradient	同上

# CSS3 進階--特效(1/3)

## • 變形(transform)處理：

[語法]： `transform: translate(100px, 100px)`  
`rotate(30deg)`  
`scale(0.8);`

2D Transform	意義
<code>rotate(angle)</code>	旋轉
<code>translate(X,Y)   translateX()   translateY()</code>	移動
<code>scale(X,Y)   scaleX()   scaleY()</code>	縮放
<code>skew(X,Y)   skewX()   skewY()</code>	斜切
<code>none</code>	不作用

# CSS3 進階--特效(2/3)

## • 轉場(**transition**)效果：

[語法]：transition:播放時間 屬性 調整播放速度 延遲時間

transition	意義												
transition-duration	播放時間 [屬性   all   none]												
transition-property	屬性[時間]												
transition-timing-function	調整播放速度[ease   linear   ease-in   ease-out   ease-in-out   cubic-bezier() ] <table><tr><td></td><td>cubic-bezier()</td></tr><tr><td>ease</td><td>0.25, 0.1, 0.25, 1.0</td></tr><tr><td>linear</td><td>0, 0, 1, 1</td></tr><tr><td>ease-in</td><td>0.42, 0, 1, 1</td></tr><tr><td>ease-out</td><td>0, 0, 0.58, 1</td></tr><tr><td>ease-in-out</td><td>0.42, 0, 0.58, 1</td></tr></table>		cubic-bezier()	ease	0.25, 0.1, 0.25, 1.0	linear	0, 0, 1, 1	ease-in	0.42, 0, 1, 1	ease-out	0, 0, 0.58, 1	ease-in-out	0.42, 0, 0.58, 1
	cubic-bezier()												
ease	0.25, 0.1, 0.25, 1.0												
linear	0, 0, 1, 1												
ease-in	0.42, 0, 1, 1												
ease-out	0, 0, 0.58, 1												
ease-in-out	0.42, 0, 0.58, 1												
transition-delay	延遲時間[時間]												

# CSS3 進階--特效(3/3)

## • 動畫(animation)效果：

屬性	意義
@keyframes	Animation name
<b>animation</b>	總和屬性
animation-name	名稱
animation-duration	time
animation-timing-function	ease   linear   ease-in   ease-out   ease-in-out   cubic-bezier()
animation-delay	time
animation-iteration-count	infinite   次數
animation-direction	normal   reverse   alternate   alternate-reverse
animation-fill-mode	none   forwards   backwards   both
animation-play-state	paused   running

# CSS3 進階--多欄位排版

屬性	意義   語法
<code>column-count</code>	<code>number   auto</code>
<code>column-fill</code>	<code>balance   auto</code>
<code>column-gap</code>	<code>length   normal</code>
<b><code>column-rule</code></b>	<code>column-rule-width column-rule-style column-rule-color</code>
<b><code>column-rule-color</code></b>	
<b><code>column-rule-style</code></b>	<code>none   hidden   dotted   dashed   solid   double   groove   ridge   inset   outset</code>
<b><code>column-rule-width</code></b>	
<code>column-span</code>	<code>1   all</code>
<code>column-width</code>	<code>auto   length</code>
<code>columns</code>	<code>auto   column-width column-count</code>

# 補充：CSS3 的規則建立

·引用其他css檔案：

```
@import url(style/XXX.css);
```

·下載字型：

```
@font-face{  
    font-family: 自訂字型名稱;  
    src: url(fonts/XXX.ttf);  
}
```

·動畫：

```
@keyframes 自訂動畫影格名稱{  
    .....  
}
```

·Media Queries：

```
@media media_type and (media_feature){  
    .....  
}
```



# III. JavaScript

# (I). What is JavaScript ?

- **JavaScript**是一個能在**Browser(client-side)**執行的應用程式，目的是為了彌補**HTML**互動性不足的問題
- 可內嵌於網頁內，**JavaScript**可以是**HTML**的一部分
- **JavaScript**是 **client-side language**，由**Browser**來解譯和執行，不需要網路也可以執行。
- 可在支援**JavaScript**的**browser**上執行。
- **JavaScript**易學難除錯
  - 以物件為基礎(**Object-based**)的語言

# 1. JavaScript的優勢

- 為互動式網頁，讓網頁活起來
- 在client 端(Browser)執行即可
- 使用者資料驗證，減少server 端的負擔，減少網路壅塞
- 何時使用？
  - 增加網頁的可看性與互動性
  - 動態的資料呈現
  - 偵測user在網頁上的行為
  - 要求user輸入資料並驗證 (ex.加入會員)
  - 記錄使用者的選擇 (ex.購物車)
- **HTML5的Javascript APIs**

## 2. 開發工具

### ·簡易工具

- Notepad++
- Sublime text
- TextWrangler
- Eclipse
- W3schools

### ·參考網站：

<http://www.ecma-international.org/publications/standards/Ecma-262.htm>  
<http://www.w3schools.com/js/default.asp>

### 3. 撰寫格式

- 放在HTML的<body>中，將JS當HTML的element來使用

```
<body>
```

```
...
```

```
</body>
```

- 放在HTML的<body>中，透過事件驅動來執行JS

```
<body>
```

```
  <script>...</script>
```

```
</body>
```

- 寫成外部檔案 (xx.js)

```
<head>
```

```
  <script src="xx.js"></script>
```

```
</head>
```

## (II). Syntax Basic

撰寫任何程式的原則：

- 易於維護
- 具可攜性(**portable**)
- 最好程式碼本身就能代表其意義
- 並適時加入註解(**comment**)，增加程式的可讀性

- 單行註解                `//...`

- 多行註解                `/* ...`  
                              `*/`

- 常見錯誤：

- 大小寫字母、拼錯字、漏寫 `"` 或 `'` ...



# 1. 變數

·意義：

- 代表記憶體空間，可存放任何數值或物件
- 其所代表的值可任意改變

·命名原則：

- 由字母大小寫、數字、\$及底線組成
- 數字不能在開頭
- 大小寫視為不同(case-sensitive)
- 不能與**關鍵字(以及未來保留字)**相同(下一頁)

·宣告時，可使用關鍵字 var

·敘述(statement)結束，請以『 ； 』結尾

## 關鍵字或保留字：

break	delete	function	return	typeof
case	do	if	switch	var
catch	else	in	this	void
continue	finally	instanceof	throw	while
default	for	new	try	with

## 未來保留字：

abstract	double	implements	private	throws
boolean	enum	import	protected	transient
byte	export	int	public	volatile
char	extends	interface	short	
class	final	long	static	
const	float	native	super	
debugger	goto	package	synchronized	

## 2. 資料型態

• Javascript 有幾個重要的資料型態：

- 數字 **Number**
- 陣列 **Array**
- 字串 **String**
- 布林 **Boolean**
- 日期時間 **Date**
- 數學 **Math**
- 正規表示式 **RegExp**
- 物件

• 也是資料型態：

- `null` (空值)
- `undefined` (未定義)

宣告一個變數，沒立刻給值

使用一個物件，但未聲明屬性

定義了函數的參數，但並未傳值給該函數



- 宣告變數時，無須宣告資料型態，JS會在執行時自動轉換。

- 與變數有關的運算子：typeof

```
typeof variable;
```

```
//顯示此variable是何種資料型態
```

```
//若typeof 某函數;
```

```
//會顯示function，函數實際上就是物件
```

```
/*若結果為null，常常是表示物件不存在，所以null會被當成物件*/
```

- 資料型態轉換

```
parser會視程式的前後文決定資料型態的轉換。
```

```
或使用強制轉型的語法：
```

```
Number(object)
```

```
parseInt(object)
```

```
object.toString()
```

## 變數的關鍵字 `var`

宣告變數時，不一定要加上此關鍵字。

宣告變數時，盡可能立刻給初值，否則是`undefined`

每個變數都有其`scope`和 `life cycle`

其`scope`可大概分成區域和全域

較好的程式，盡可能使用區域變數

### • 定義區域變數：

- 在函數中使用`var`關鍵字
- 定義函數時的參數

### • 定義全域變數：

- 在函數中**不要**使用`var`關鍵字
- 寫在函數之外

舉例來說：

•meg在此為區域變數

```
function doFirst() {  
    var meg = 'error';  
    alert(meg);  
}
```

•meg在此為全域變數

```
var meg = 'error';  
function doFirst() {  
    alert(meg);  
}
```

•meg在此為全域變數

```
function doFirst() {  
    meg = 'error';  
    alert(meg);  
}
```

•meg在此為區域變數

```
doFirst('error');  
function doFirst(meg) {  
    alert(meg);  
}
```

## 3. 輸出 / 輸入

- 輸出到畫面 (少用)

```
document.write('Hello World!');
```

- 輸出到狀態列，目前 (非常少用)

```
window.status = "出現在狀態列";
```

- 跳出視窗

- alert 視窗
- confirm 視窗
- prompt 視窗

- Javascript console

```
console.log('Hello World!');
```



## 4. 運算子、運算式與敘述

### ·JS的運算子

運算子優先順序	描述	結合性
()、[]		從左到右
++、--、+(正)、-(負)、!	一元運算子	從右到左
*、/、%	算術運算子	從左到右
+(加)、-(減)	[同上]	從左到右
>、>=、<、<=	關係運算子	從左到右
==、!=	[同上]	從左到右
&&	邏輯運算子	從左到右
	[同上]	從左到右
? :	條件運算子	從左到右
=、+=、-=、*=、/=、%=、...	指定運算子	從右到左

·字串運算子：字串可用"+"將字串串接

## 5. 迴圈敘述

- 結構化的程式設計有三種結構：
  - 循序式：即逐行執行
  - 迴圈式：反覆執行到不想執行為止
  - 選擇式：多重選項，擇一而行
- 常見的迴圈形式：

迴圈	執行次數	使用時機
<code>while (條件判斷) {敘述;}</code>	0~N	不固定的執行次數與狀態
<code>for (初值; 條件判斷; 計次) {敘述;}</code> <code>for (...in...) {敘述;}</code>	0~N	固定的執行狀態
<code>do {敘述;} while (條件判斷);</code>	1~N	執行起碼一次

- 迴圈的好幫手**break**和**continue**
  - break**;敘述除了在迴圈有作用以外，還可使用在**switch**
  - continue**;敘述只在迴圈有作用

## 6. 選擇性敘述

選擇性敘述	說明	執行
<code>if (條件) { ... }</code>	·執行0或1次	符合條件就執行
<code>if (條件) {     (1) } else {     (2) }</code>	·條件大多是在一個數值範圍內；且可以放浮點數的比較	符合條件就執行(1)， 不符合就執行(2)
<code>switch (比對條件) {     case XX:         ...         break;     case OO:         ...         break;     ...     default; } //end of switch</code>	·比對條件控制在整數形態或字元形態  ·不適合使用條件在一個範圍內，例如1~1000之間	多重選擇

•雖然JavaScript的語法與C語言十分類似，但以下卻有些不同。

•try / catch

- try / catch不是迴圈
- 目的：希望user不會看到錯誤訊息 (做錯誤處理)
- 語法：

```
try{  
    //這裡的程式碼可能會導致錯誤  
}  
catch (e) {  
    //如果以上有錯誤就執行此處  
}  
finally{    //不一定要有finally  
    //不論是否有錯誤，此處都會被執行  
}
```

•for...in

- 用在物件的Traversal

## 7. 函數

·內建函數--直接使用即可，不需要宣告

內建函數	說明
<b>parseInt</b> (value,base)	傳回 string 轉成 10 進制後的整數；base 不一定要寫，default 是10 進制。
parseFloat(value)	傳回 string 轉成浮點數後的數目
<b>isNaN</b> (testValue)	檢查 testValue 是否為非數字型態 (is Not a Number)
escape(string)	傳回string 的 16 進制編碼得出來的字串
unescape(string)	傳回以 16 進制編碼字串的原本字串
eval(string)	把 string 當作 JavaScript 的敘述來執行
Number(object)	將物件轉換成數值，遇到不合法傳回NaN

## ·自訂函數--

◦ 宣告 ◦ 定義 ◦ 呼叫

·**JavaScript**不用明顯的宣告，直接會寫『定義』，實際上也有『宣告』的意味

### ·定義

```
function functionName() {  
    // 敘述  
}
```

### ·呼叫

```
functionName();
```

有傳回值的函數：

### ·定義

```
function functionName()  
    // 敘述  
    return value;  
}
```

### ·呼叫

```
variable = functionName();
```

帶有參數的函數：

### ·定義

```
function functionName(arg1, arg2, ...) {  
    // 敘述  
}
```

### ·呼叫

```
functionName(value1, value2, ...);
```

# (III). Object-Oriented

# 1. 物件的屬性與方法

- 所有能描述的東西都是由物件組成
  - 大的物件裡面可以包含小物件
  - 許多的小物件可以組成大物件
- 每一個物件都有其特性與其使用方法
  - 物件的特性即為屬性 (**attributes**) -- 名詞&形容詞
  - 物件的使用方法即為方法 (**methods**) -- 動詞
- 物件有層級
- 事件 (**events**) 的發生可以啟動物件的方法或屬性



## 2. 事件 (events) 處理--1/4

Javascript 是 event-driven language

### (1).何謂事件

- 系統會將每個發生的動作轉成事件，然後送到程式處理  
user的動作包括，按下滑鼠或載入某一頁等等。  
系統的回應可能是：當某一頁載入完。
- 我們就根據不同的事件來設計不同的工作，  
負責處理事件的就叫做事件處理程序 (Event Handler)。
- **JavaScript** 的事件通常與物件有關，所以不同的物件就支援不同的事件處理程序

## 2. 事件 (events) 聆聽功能--2/4

### (2). 建立事件聆聽功能

包括一個物件、一個事件、一個處理函數

#### 1. 嵌入式事件處理 (.html)

```
<body onload="doFirst()">
```

#### 2. 過去的事件處理方法 (.js)

```
window.onload=doFirst;  
或 window.onload=function(){...}
```

#### 3. `window.addEventListener('load', doFirst, false);`

#### 4. for IE

```
window.attachEvent('onload', doFirst);
```

## 2. 事件 (events) 分類--3/4

### (3) . 事件分類

· 輸入裝置

· 鍵盤

· Browser

· Form

## 2. 事件 (events) 分類--4/4

### (4) . 事件物件的屬性和方法

屬性：

方法：

`preventDefault()`

### (5) . 引用事件

### 3. 視窗(window)物件 -- 屬性(1/2)

屬性	說明
<code>navigator</code>	
<code>location</code>	
<code>history</code>	
<code>document</code>	
<code>status</code>	
<code>name</code>	視窗物件名稱
<code>closed</code>	視窗是否關閉
<code>defaultStatus</code>	預設的狀態列訊息
<code>opener</code>	用 <code>window.open()</code> 開啟的那個視窗
<b>[event]</b>	物件的屬性本身也可以是物件

## 3.視窗(window)物件 -- 方法(2/2)

方法	說明	語法
<b>alert()</b>	警告訊息	
<b>confirm()</b>	確認訊息	
<b>prompt()</b>	提示訊息	
<b>open()</b>	開啟新視窗	
<b>close()</b>	關閉視窗	
<b>focus()</b>	成為焦點	
<b>blur()</b>	移開焦點	
<b>setTimeout()</b>	設定計時器	
<b>clearTimeout()</b>	取消計時器	
<b>setInterval()</b>	設定計時器	
<b>clearInterval()</b>	取消計時器	

# 視窗物件--window.navigator

·提供瀏覽器的訊息

屬性	Result
appName	Mozilla
appName	Netscape
appVersion	5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.111 Safari/537.36
cookieEnabled	true
language	zh-tw
onLine	true
userAgent	Win32
geolocation	
方法	說明
javaEnabled()	判斷是否支援JAVA

# 視窗物件--window.history

•儲存該網頁曾到訪過的URL資料

屬性	說明	
length	到訪過URL的個數	history.length
方法	說明	
go()	載入指定的URL	history.go(number   URL)
back()	載入上一個URL	history.back()
forward()	載入下一個URL	history.forward()



# 視窗物件--`window.location`

·提供了很多讀取所在網址的相關資訊

屬性	說明	
hash	取得anchor	
host		
hostname		
href		
origin		
pathname		
port		
protocol	通訊協定，包括冒號	
search	頁面跳轉	<code>location.search=querystring</code>
方法	說明	
<code>assign()</code>		
<code>reload()</code>		
<code>replace()</code>		載入新的 URL

# 視窗物件--window.document--1/3

屬性	說明
lastModified	
domain	
URL	
cookie	
documentElement	
head	
scripts	
title	
body	

屬性	說明
anchors	
forms	
images	
links	
bgColor	
fgColor	
linkColor	
alinkColor	
vlinkColor	

# 視窗物件--window.document--2/3

方法 (JS與HTML的elements產生關聯)	說明
getElementById()	
getElement <b>s</b> ByName()	
getElement <b>s</b> ByTagName()	
getElement <b>s</b> ByClassName()	
修改HTML的elements的 屬性與方法	
element. <b>innerHTML</b> =	
element.attribute =	
element.setAttribute(attribute,value)	
element.style.property =	115

# 視窗物件--window.document--3/3

方法 (新增或刪除HTML的elements)	說明
<code>createElement()</code>	
<code>removeChild()</code>	
<code>appendChild()</code>	
<code>replaceChild()</code>	
<code>write(text)</code>	
增加事件處理	
<code>document.getElementById(id).onclick=function() {}</code>	

# 文件物件模型 (DOM) -- 1 / 4

## •何為DOM?

- DOM == Document Object Model
- 此為W3C制定的標準
- 基本上，DOM屬於JavaScript的介面標準，各家瀏覽器依此標準建置實際的物件，使JavaScript可以直接使用

## •基本概念

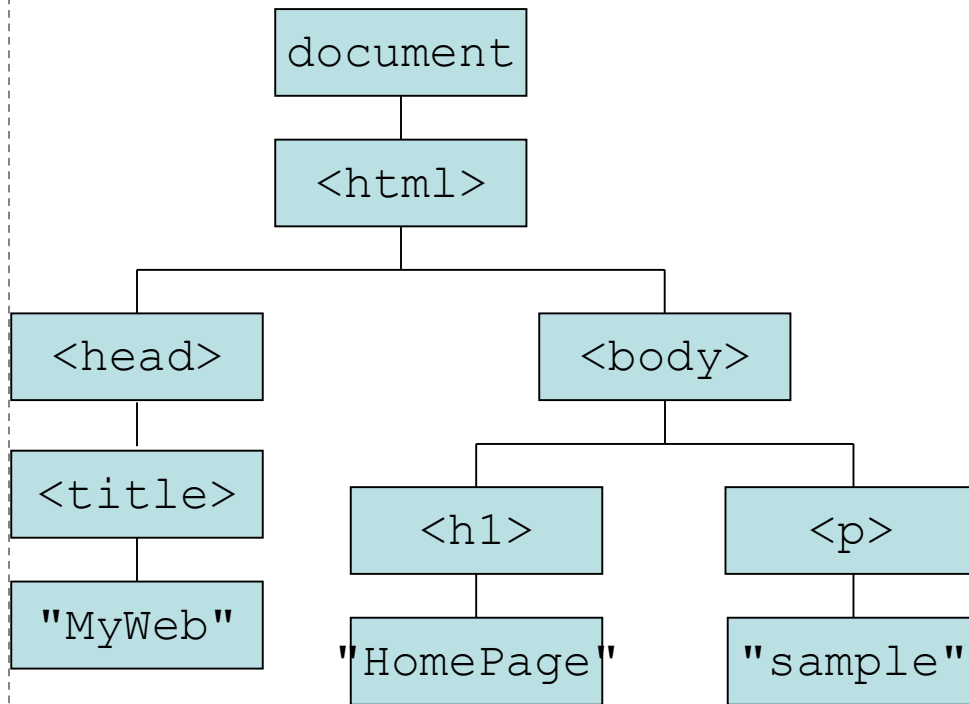
- 節點node：文件中任何一樣東西都是一個節點
- document node：整篇文件
- 層次：節點彼此之間存在著層次的關係

<p>sample</p>

<p>標籤為一個元素節點 (element node) ，  
並且包含文字節點 (text node) sample

<p>節點則為sample節點的parent node

# 文件物件模型 (DOM) -- 2 / 4



說明：

- 左側為HTML文件的部分
- 左邊10個框框都是節點
- 三個文字節點 (text node) ，都沒有後續節點稱為leaf
- 六個標籤皆為element node
- document為此family tree的根節點 (root)
- <head>和<body>有同個父節點 (parent node) ，所以他們彼此之間為兄弟節點 (sibling node)







# 文件物件模型 (DOM) -- 3 / 4

## •DOM的屬性

描述節點本身的訊息	
nodeName	<ul style="list-style-type: none"><li>- element node就是標籤名稱，一定要大寫， nodeName=='BODY'</li><li>- 屬性節點的nodeName就是屬性名稱</li></ul>
nodeValue	
nodeType	<ul style="list-style-type: none"><li>-代表節點類型，用數字來表示</li><li>-element(1)、attribute(2)、text(3)、comment(8)、document(9)</li></ul>
描述彼此之間的關係	
parentNode   childNodes	
firstChild   lastChild	
previousSibling   nextSibling	
描述文字節點	
innerHTML   innerText   textContent	

# 文件物件模型 (DOM) -- 4 / 4

## •DOM的方法

新增	
<code>createElement('標籤')</code>	
<code>appendChild()</code>	
<code>insertBefore()</code>	
<code>replaceChild()</code>	
<code>cloneNode()</code>	
刪除	
<code>removeChild()</code>	
判斷	
<code>hasChildNodes()</code>	判斷是否有子節點
<code>hasAttributes()</code>	判斷是否有該屬性

# document.cookie--1/3

- HTTP是一種無狀態的協定，所以web server不會記住任何東西
- 假設是FB的serve，有些資訊應該記住，例如會員是否已登入？是否投過票？是否玩過Bubble Witch Saga？就需要引入cookie
- cookie是儲存在user端的文字檔，它會與server端的目錄相關聯，當user發出request的時候，cookie會隨著HTTP的request傳送出去
- 而server端會決定如何處理cookie傳來的資訊
- 何處使用 Cookie：
  - 月曆記事簿
  - 登入，sticky form
  - 個人化網站

# document.cookie--2/3

## ·設定cookie

- `document.cookie = 'cookie=yes; expires=Tue, 16 Oct 2012 08:30:20 UTC; path=/; domain=iii.org';`
- `document.cookie = 'testcookie=maybe; expires=Mon, 22 Oct 2012 16:30:20 UTC; path=/; domain=iii.org';`

參數	說明
name=value	name 是 Cookie 的名稱，value 是該 Cookie 的值，這個字串會以 16 進制編碼。 必需要有的參數。
expires=date	指定Cookie 的有效日期，當過了有效日期後，此Cookie 就不會再儲存在瀏覽器；date是GMT的格式。如果未指定這個參數，此Cookie的有效日期就是使用者退出瀏覽器時
path=path	指定可以存取該 Cookie 的路徑。如果不指定這個參數，path就為設定該 Cookie 的網頁所在的路徑
domain=domain	指定可以存取該 Cookie 的網域。 如果不指定這個參數，domain 就指定為設定該 Cookie 的網頁所在的網域
secure	指定 Cookie 只可以傳送給 HTTPS 伺服器

# document.cookie--3/3

## •讀取cookie

讀取先前設定的cookie，可用for迴圈將前述的字串，用『;』來當成分隔的符號，可將cookie的4或5個參數一一讀取出來

## •刪除cookie

新增一個有效期間為過去時間的同名cookie，用來刪除cookie

## •可用屬性navigator.cookieEnabled來看看瀏覽器是否支援cookie

```
if (navigator.cookieEnabled) {  
    //瀏覽器有支援，可以設定或讀取cookie了  
}
```



## 4. 內建物件

•JavaScript 的內建物件：

- 1) Boolean
- 2) Number
- 3) String
- 4) Array
- 5) Math
- 6) Date
- 7) RegExp
- 8) ...

# 內建物件--Boolean

方法	說明
toString()	以字串表示布林值
valueOf()	取得物件原來的布林值
	<pre>var bool = true; var bool = new Boolean(); var bool = new Boolean(true); var bool = new Boolean(5 &gt; 3);</pre>



# 內建物件--Number

方法	說明
<code>toExponential(x)</code>	
<code>toFixed(x)</code>	
<code>toPrecision(x)</code>	
<code>toString()</code>	以字串表示布林值
<code>valueOf()</code>	取得物件原來的布林值
屬性	
<code>MAX_VALUE</code>	
<code>MIN_VALUE</code>	
<code>POSITIVE_INFINITY</code>	
<code>NEGATIVE_INFINITY</code>	
<code>NaN</code>	



# 內建物件--String--1/2

方法	說明
<code>charAt()</code>	
<code>indexOf()</code>   <code>lastIndexOf()</code>	
<code>slice()</code>	
<code>split()</code>	
<code>substr()</code>	
<code>toLowerCase()</code>   <code>toUpperCase()</code>	
<code>concat()</code>	
<code>replace()</code>	
<code>toString()</code>	
<code>trim()</code>	
屬性	
<code>length</code>	



# 內建物件--String--2/2

處理HTML物件的方法	說明
<code>anchor()</code>	
<code>big()</code>	
<code>blink()</code>	
<code>bold()</code>	
<code>fixed()</code>	
<code>fontcolor()</code>	
<code>fontsize()</code>	
<code>italics()</code>	
<code>link()</code>	
<code>small()</code>	
<code>strike()</code>	
<code>sub()</code>   <code>sup()</code>	



# 內建物件--Array

方法	說明
<code>indexOf()</code>   <code>lastIndexOf()</code>	
<code>pop()</code>   <code>push()</code>	
<code>shift()</code>   <code>unshift()</code>	
<code>concat()</code>	
<code>splice()</code>	
<code>slice()</code>	
<code>concat()</code>	
<code>join()</code>	
<code>reverse()</code>	
<code>sort()</code>	
<code>toString()</code>	
屬性	
<code>length</code>	135





# 內建物件--Math

·目的：提供複雜或特殊的數學運算

·屬性 (所有數學屬性皆不可修改)

- `Math.PI`
- `Math.E`
- `Math.LN`
- `Math.LN10`
- `Math.LOG2E`
- `Math.LOG10E`
- `Math.SQRT1_2`
- `Math.SQRT2`

·方法

- |                              |                               |
|------------------------------|-------------------------------|
| - <code>Math.abs(x)</code>   | - <code>Math.max(...)</code>  |
| - <code>Math.acos(x)</code>  | - <code>Math.min(...)</code>  |
| - <code>Math.asin(x)</code>  | - <code>Math.pow(x, y)</code> |
| - <code>...</code>           |                               |
| - <code>Math.ceil(x)</code>  | - <code>Math.random()</code>  |
| - <code>Math.cos(x)</code>   | - <code>Math.round(x)</code>  |
| - <code>Math.exp(x)</code>   | - <code>Math.sqrt(x)</code>   |
| - <code>Math.floor(x)</code> |                               |
| - <code>Math.log(x)</code>   |                               |

# 內建物件--Date--1/3

方法	說明
<code>getDate()</code>	
<code>getDay()</code>	
<code>getFullYear()</code>	
<code>getHours()</code>	
<code>getMilliseconds()</code>	
<code>getMinutes()</code>	
<code>getMonth()</code>	
<code>getSeconds()</code>	
<code>getTime()</code>	

# 內建物件--Date--2/3

方法	說明
<code>toDateString()</code>	
<code>toISOString()</code>	
<code>toJSON()</code>	
<code>toLocaleDateString()</code>	
<code>toLocaleString()</code>	
<code>toLocaleTimeString()</code>	
<code>toString()</code>	
<code>getTimeString()</code>	



# 內建物件--Date--3/3

方法	說明
<code>setDate()</code>	
<code>setFullYear()</code>	
<code>setHours()</code>	
<code>setMilliseconds()</code>	
<code>setMinutes()</code>	
<code>setMonth()</code>	
<code>setSeconds()</code>	

# 內建物件--RegExp

方法	說明
<code>compile()</code>	
<code>exec()</code>	
<code>test()</code>	
屬性	
<code>global</code>	
<code>ignoreCase</code>	
<code>lastIndex</code>	
<code>multiline</code>	
<code>source</code>	

## 5. 自訂物件

- 目的：產生與定義自訂物件

- 建立物件1：

```
var myObj = new Object();
```

```
var myObj = {};
```

- 建立物件2：

```
var myObj = new Object(name:'Anan',age:10);
```

```
var myObj = {name:'Anan',age:10};
```

- e.g.

```
var myObj = {
```

```
  name:'Anan',
```

```
  age:10,
```

```
  favoriteColors: {'blue', 'green', 'gray'},
```

```
  gender:'M',
```

```
  student:true,
```

```
  today:new Date()
```

```
};
```

# 補充: JSON

物件 (object) 和陣列 (array) 的結合