

AP104

HTML5

Instructor: 黃語昕

E-mail: silvia.huang@gmail.com

Course Information:

Time : (see schedule)

Classroom : R105

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.互動網頁的安排與配置—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
 - 。圓角 -- Radius
 - 。顏色的更多詮釋 -- RBGA
 - 。陰影 -- Shadow Effects
 - 。漸層 -- Gradient

- 。變形 -- Transform
- 。動畫 -- Animation
- 。多欄位 -- Multiple Columns

III、JavaScript

(I). What is JavaScript ?

1. 歷史
2. 特色
3. 用在哪裡？

(II). Syntax Basic

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

(III). Object-Oriented

1. 物件的屬性與方法
2. 事件
3. window 物件
4. 內建物件
5. 自訂物件

IV、HTML5

(I). 環境介紹(前述)

(II). HTML5 概述

1. 新標籤
2. 新表單(前述)

(III). 繪圖 { Canvas }

1. 基本功能
2. 進階功能

(IV). 影音多媒體 { Video & Audio }

(V). 拖放操作 { Drag & Drop }

1. 基本功能
2. 搭配 file APIs

(VI). 地理定位 { Geolocation }

(VII). 資料儲存 { Web Storage }

Grades:	Program Assignments	70%
	Examinations (Midterm & Final)	30%

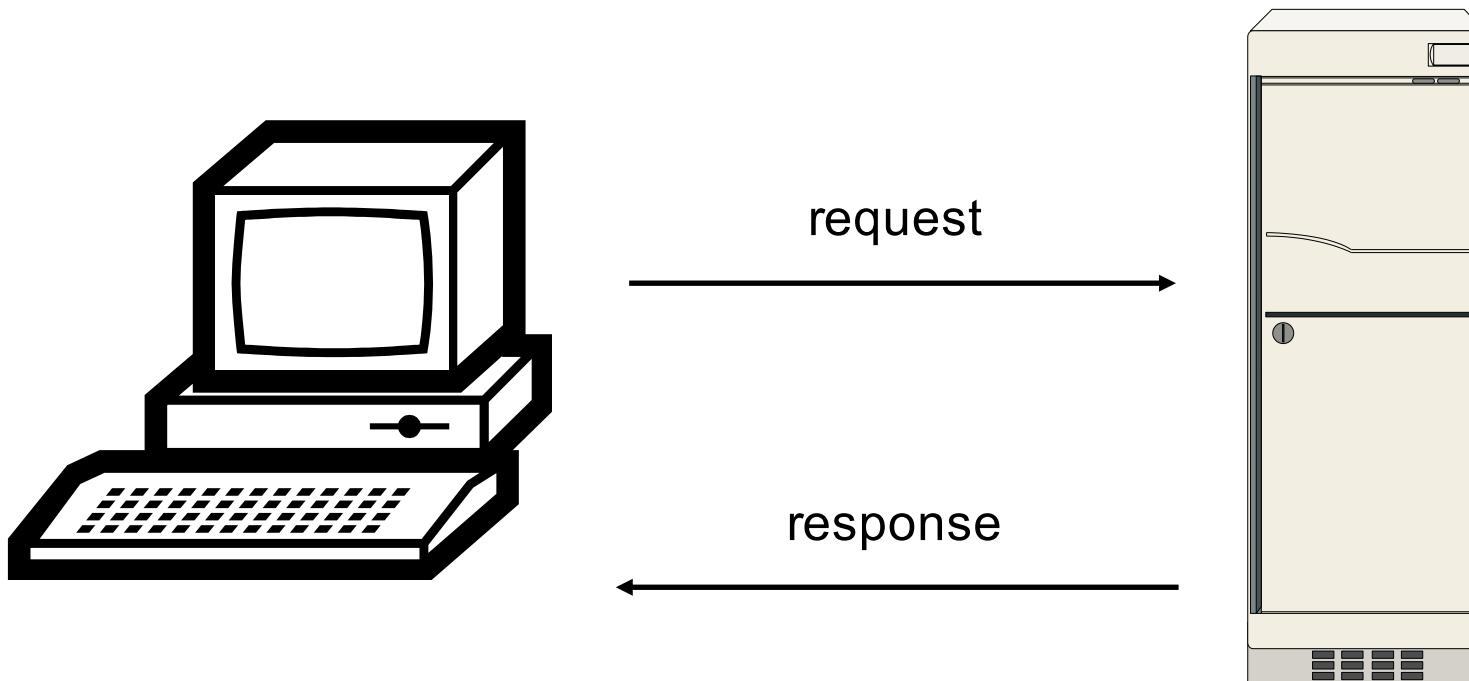
I. HTML

HyperText Markup Language

(I). 環境介紹

i. Internet環境

提供服務--E-mail、FTP、World Wide Web(WWW) ...



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++(no MAC)
 - Sublime Text
 - Eclipse
- ### 2. JavaScript console
- ### 3. Web Server：
- Windows：安裝Apache或IIS
 - Mac或Linux已有內建Apache

What are we?



BROWSERS!



BROWSERS! BROWSERS!



What do we want?



MORE SPEED!



MORE SPEED!



And when do we want it?



RIGHT NOW!!!



RIGHT NOW!!!



BROWSERS!



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.htm>

包括了

- How : "http://" or "ftp://" ...
- Where : 網址(ex. www.hinet.net)
- What : 網頁or其他資源(ex. index.htm)

iv. HTML 4.01 Specification

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

v. HTML 5.1 Lastest Published Version

<http://www.w3.org/TR/html51/>

(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>      } 與Browser溝通的information
  <body>
    </body>      } 網頁的內容
</html>●
```

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-->
```


（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
bgcolor	Color Name rgb	背景顏色	取代
background	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)

...文字內容...

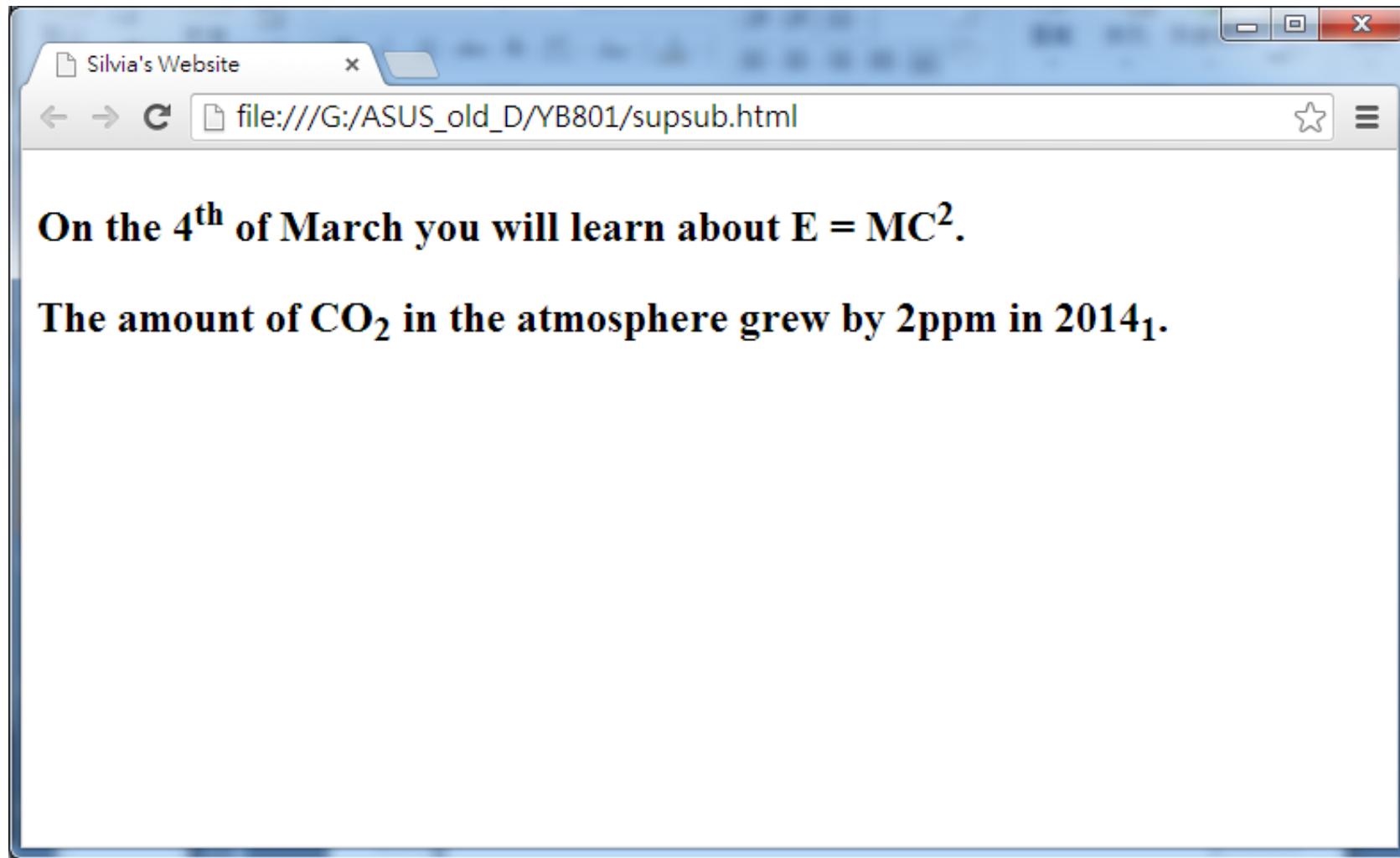
屬性	屬性值	說明	CSS
size	1~7 (小→大) , default=3 +1~+4 , -1~-2	文字大小	取代
color	color Name rgb	文字顏色	取代
face	標楷體、新細明體、Verdana、Tahoma...	字體	取代

1.文字與段落的處理 (2 / 4)

字型變化

屬性	說明	CSS
... <i>...</i> <u>...</u> <big>...</big> <small>...</small>	粗體 斜體 底線 字體相對變大 字體相對變小	取代
^{...} _{...}	上標 下標	取代
... ... <blockquote>...</blockquote> <abbr>...</abbr> <cite>...</cite> <dfn>...</dfn> <ins>...</ins> ... <s>...</s>	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
<pre>.....</pre>	width	原封不動的呈現	
<hn>.....</hn>	n=1~6 (大→小) align	標題文字Headings	取代
<p>.....</p>	align 屬性值：right left center justify	分段 Paragraph	
 		換行 line Break	
<center> ... </center>		置中對齊	取代
<!-- 註解 -->		註解	
<hr> Tips	align=right left center (default) width=percent pixels size=pixels(px) color=colorName rgb shade (default) noshade	水平分隔線 Horizontal Rules	

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]	 	空白 (Non-Break Space)	€	€	歐元符號
<	<	小於 (less than)	¥	¥	日圓符號
>	>	大於 (greater than)	¢	¢	美元
&	&	And (ampersand)	£	£	英鎊
"	"	雙引號 (quotation mark)	♠	♠	spade
©	©	copyright	♣	♣	club
®	®	registered trademark	♥	♥	heart
™	™	trademark	♦	♦	diamond
§	§	section			

2. 圖片(1 / 2)

(1). GIF、JPEG與PNG

	GIF (Graphics Interchange Format)	JPEG (Joint Photographic Experts Group)	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	<ul style="list-style-type: none">· icon· 動畫· 透明背景圖片	<ul style="list-style-type: none">· 照片· 失真性壓縮· 不支援透明度	<ul style="list-style-type: none">· 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).

屬性	屬性值		說明	CSS
<td>top middle bottom (default)</td> <td>left center X right</td> <td>對齊方式 (文繞圖)</td> <td></td>	top middle bottom (default)	left center X 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> ... `

屬性	屬性值
href	<code>...</code> ◦ 網站內連結： <code>某網頁</code> ◦ 跨網站連結： <code><a href="<u>http://tw.yahoo.com</u>">奇摩</code> ◦ 本頁連結： <code>...</code> ◦ email： <code><a href="<u>mailto:silvia.huang@gmail.com</u>">email</code>
name	自行命名
target	_blank _new _self

4. 畫面控制--List、Table (1 / 5)

(1) . 符號清單

標籤	屬性	子標籤	說明	CSS
<ul type=""> ... 	disk square circle	...	符號清單為無序列排序 unordered list	取代

(2) . 編號清單

<ol type=""> ... 	(數字) 1 A a I i start="9"	...	編號清單為有序列排序 ordered list	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
<dl> ... </dl>	<dt>.....</dt> 用來定義名詞 <dd>.....</dd> 用來容納定義

(4) . 巢狀清單

```
<ul>
  <li>Mousses</li>
  <li>Pastries
    <ul>
      <li>Crossant</li>
      <li>Mille-feuille</li>
      <li>Palmier</li>
      <li>Profiterole</li>
    </ul>
  </li>
  <li>Tarts</li>
</ul>
```

- Mousses
- Pastries
 - Crossant
 - Mille-feuille
 - Palmier
 - Profiterole
- Tarts

4. 畫面控制--List、Table (4 / 5)

<table>.....</table> 表格

屬性	屬性值	說明
align	left 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	percent (%) pixels (px)	高度
cellspacing	percent (%) pixels (px) default = 2	邊框與邊框之間的距離
cellpadding	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	
colspan	數字	合併儲存格 (欄)
rowspan	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
src	路徑+檔名	
name		
height	pixels	
width	pixels	
frameborder	0 1	
sandbox		
srcdoc	HTML_code	<pre><iframe src="" srcdoc=<p>Hello!</p>> </iframe></pre>
scrolling	no yes auto	

6. 表單--Form (1 / 3)

<form>...</form>

屬性	屬性值	說明
action	url mailto	
method	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)
	<select name=""> <option value="傳送值">...</option> <option value="傳送值">...</option> <option value="傳送值">...</option> </select>	
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	數值	數字
range	數值	範圍
datetime	UCT時區的日期時間	日期時間
date	日期	日期
month	月份	月份
week	星期	星期
time	時間	時間
datetime-local	日期時間	本地日期時間
color	RGB	顏色
file	0~N個檔案	檔案
image	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
<button></button>	按鈕
<fieldset> <legend>標題文字</legend> </fieldset>	外框線
<label for=""></label>	
與CSS有關的標籤	
<style></style>	放在<head>標籤內
<link>	放在<head>標籤內
<div></div>	
	

II. CSS

Cascading Style Sheet

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	px	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	em	字型的字體大小(font-size)。 若字型大小是16個pixel，則 $1\text{ em} = 16\text{ pixel}$
En	en	$\frac{1}{2}\text{ em}$
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 italic oblique
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	font	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	text-decoration	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	text-align	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 (浮水印)	background-attachment	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	background	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	margin	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	padding	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	border

(iv). 總和屬性

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	list-style-type	none decimal disc circle square lower-roman upper-roman lower-alpha upper-alpha
項目圖像	list-style-image:url()	url(圖檔的路徑+檔名) none
樣式位置	list-style-position	inside outside
總和屬性	list-style	所有條類項目屬性設定

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

◎ 做做看 : fieldset.html

Access keys - Windows Internet Explorer
C:\Users\Silvia\Desktop\accesskeys.html

Access keys

檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)

HTML5 Forms Web Resource... Online video tutorials & train... 數位學習 - 教各種軟體使... HTML語法教學

User Registration Form

Personal Information

Name:

Email Address:

Password:

Confirm Password:

Address Details

Address line one:

Address line two:

Town / City:

Zip / Post code:

Sign Up!

◎ 做做看 : positionLimes.html



(III). CSS3

CSS3 的新特色

- 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	
tag {.....}	h1 {color:green; }
.className {.....}	.myClass {color:green; }
#idName {.....}	#myId {color:green; }
* {.....}	* {color:green; }
Attribute selectors (屬性選擇器)	
[語法] : selector [屬性] {CSS的屬性}	
selector [attribute]	
selector [attribute = "value"]	
selector [attribute *= "value"]	
selector [attribute ^= "value"]	
selector [attribute \$= "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 : nth-child(n)	
selector : nth-child(odd)	
selector : nth-child(even)	
selector : first-child	
selector : last-child	
selector : nth-of-type(n)	
selector : only-child	
selector : only-of-type	
selector : empty	

CSS3 進階--選擇器 { selectors } (3/4)

- pseudo-classes(其他的虛擬類別)

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

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	length % {半徑參數} ex. border-radius: 25px; border-top-left-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】	
border-image	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
box-shadow {邊框陰影}	box-shadow: x y blur spread color inset; 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 $>=0$ 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:inset [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>
text-shadow {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
box-reflect {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]: -webkit-box-reflect: below 10px; (反射在下方距離10px)	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));

• 透明度(opacity)屬性：

屬性	屬性值
opacity {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	degree 或 direction , color-stop1 , color-stop2 - direction : left (0deg) right (180deg) top (270deg和-90deg) (default) bottom (90deg和-270deg) top left top right bottom left bottom right - color-stop1 : color 0% - color-stop2 : 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	(center, shape, colorStop, ..., colorStop) (水平, 垂直, 形狀, 半徑, <i>color-stop1</i> , <i>color-stop2</i>) 水平: left(0%) center(50%) right(100%) 垂直: top (0%) center (50%) bottom (100%) - shape : ellipse (預設) circle - shape size : closest-side 最近端 closest-corner 最近角 farthest-side 最遠端 farthest-corner 最遠角 contain 包含 cover 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
rotate(angle)	旋轉
translate(X, Y) translateX() translateY()	移動
scale(X, Y) scaleX() scaleY()	縮放
skew(X, Y) skewX() skewY()	斜切
none	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 border="1"><tbody><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></tbody></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
animation	
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 alte rnate-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
column-count	number auto
column-fill	balance auto
column-gap	length normal
column-rule	column-rule-width column-rule-style column-rule-color
column-rule-color	
column-rule-style	none hidden dotted dashed solid double groove ridge inset outset
column-rule-width	
column-span	1 all
column-width	auto length
columns	auto column-width column-count

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

<http://webdesign.about.com/od/macintoshhtmleditors/tip/free-macintosh-editors.htm>

• 簡易工具

- Notepad++
- Sublime text 2 / 3
- 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

```
<head>
```

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

```
</head>
```

- 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：
 - 由英文數字大小寫、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(保留字)相同
- 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
 - 日期時間 Date
 - 陣列 Array
 - 數學 Math
 - 字串 String
 - 正規表示式 RegExp
 - 布林 Boolean
 - 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

數字型態--可以進行加減乘除等運算

```
var num = 25;
```

字串型態--加上'...'或"..."，讓JS知道正在處理的是字串

```
var str = 'I am a simple string.';  
var str = '25';
```

布林型態--只有兩個值：true和false

```
var boo = true;
```

物件型態--除了以上型態的其他型態

```
var id = document.getElementById('context');
```

·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

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>	$0 \sim N$	不固定的執行次數與狀態
<code>for (初值; 條件判斷; 計次) {敘述;}</code>	$0 \sim N$	固定的執行狀態
<code>for (...in...) {敘述;}</code>		
<code>do {敘述;} while (條件判斷);</code>	$1 \sim N$	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>	<ul style="list-style-type: none">· 執行0或1次	符合條件就執行
<code>if(條件) { (1) } else { (2) }</code>	<ul style="list-style-type: none">· 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>	<ul style="list-style-type: none">· 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
parseInt (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
isNaN (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();
```

帶有參數的函數：

·定義

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

·呼叫

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

有傳回值的函數：

·定義

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

·呼叫

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

(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
navigator	
location	
history	
document	
status	
name	視窗物件名稱
closed	視窗是否關閉
defaultStatus	預設的狀態列訊息
opener	用 <code>window.open()</code> 開啟的那個視窗
[event]	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
<code>alert()</code>	警告訊息	
<code>confirm()</code>	確認訊息	
<code>prompt()</code>	提示訊息	
<code>open()</code>	開啟新視窗	
<code>close()</code>	關閉視窗	
<code>focus()</code>	成為焦點	
<code>blur()</code>	移開焦點	
<code>setTimeout()</code>	設定計時器	
<code>clearTimeout()</code>	取消計時器	
<code>setInterval()</code>	設定計時器	
<code>clearInterval()</code>	取消計時器	

視窗物件--window.navigator

- 提供瀏覽器的訊息

屬性	Result
appCodeName	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
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	location.search=querystring
方法	說明	
assign()		
reload()		
replace()		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()	
getElementsByName()	
getElementsByNameTagName()	
getElementsByNameClassName()	
修改HTML的elements的 屬性與方法	
element.innerHTML =	
element.attribute =	
element.setAttribute(attribute,value)	
element.style.property =	

視窗物件--window.document--3/3

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

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

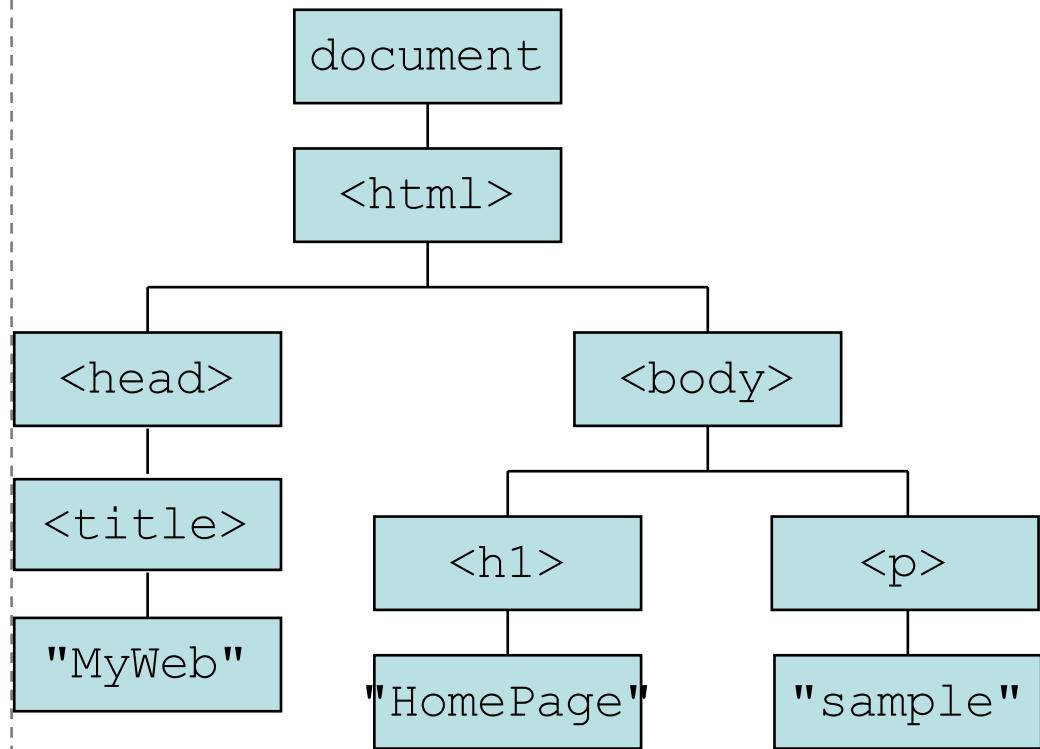
<p>sample</p>

<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

- 每個node都有一些屬性，並分別包含該節點的訊息：

屬性	
nodeName	<ul style="list-style-type: none">- element node就是標籤名稱，一定要大寫， nodeName=='BODY'- 屬性節點的nodeName就是屬性名稱
nodeValue	
nodeType	<ul style="list-style-type: none">-代表節點類型，用數字來表示-element(1)、attribute(2)、text(3)、comment(8)、document(9)
tagName	<ul style="list-style-type: none">- 很少用
innerHTML	
parentNode	
firstChild	
lastChild	
nextSibling	

文件物件模型 (DOM) -- 4 / 4

方法	
insertBefore()	
appendChild()	
replaceChild()	
createTextNode()	
cloneNode()	
removeChild()	
hasAttributes()	

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
<code>name=value</code>	<code>name</code> 是 Cookie 的名稱， <code>value</code> 是該 Cookie 的值，這個字串會以 16 進制編碼。必需要有的參數。
<code>expires=date</code>	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； <code>date</code> 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
<code>path=path</code>	指定可以存取該 Cookie 的路徑。如果不指定這個參數， <code>path</code> 就為設定該 Cookie 的網頁所在的路徑
<code>domain=domain</code>	指定可以存取該 Cookie 的網域。如果不指定這個參數， <code>domain</code> 就指定為設定該 Cookie 的網頁所在的網域
<code>secure</code>	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 > 3);</pre>

內建物件--Number

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

內建物件--String--1/2

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

內建物件--String--2/2

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

內建物件--Array

方法	說明
indexOf() lastIndexOf()	
pop() push()	
shift() unshift()	
concat()	
splice()	
slice()	
concat()	
join()	
reverse()	
sort()	
toString()	
屬性	
length	

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.LN1
 - Math.LOG2E
 - Math.LOG10E
 - Math.SQRT1_2
 - Math.SQRT2

- 方法

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

內建物件--Date--1/3

方法	說明
getDate()	
getDay()	
getFullYear()	
getHours()	
getMilliseconds()	
getMinutes()	
getMonth()	
getSeconds()	
getTime()	

內建物件--Date--2/3

方法	說明
toString()	
toTimeString()	
toLocaleTimeString()	
toLocaleDateString()	
toLocaleString()	
toJSON()	
toISOString()	
toDateString()	

內建物件--Date--3/3

方法	說明
setDate()	
setFullYear()	
setHours()	
setMilliseconds()	
setMinutes()	
setMonth()	
setSeconds()	

內建物件--RegExp

方法	說明
compile()	
exec()	
test()	
屬性	
global	
ignoreCase	
lastIndex	
multiline	
source	

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()  
};
```


I. HTML5

HyperText Markup Language

||

||(I). 環境介紹 -- 如前述

||

(II). HTML概述 -- 如前述

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-->
```

HTML5 新詮釋

- HTML 4.01的超集合
- 移除了對於外掛程式的需要
- 標籤更具有描述性
- 能做更多事的CSS
- HTML5就是一組技術：
 - 多媒體的支援
 - 畫布加上變形特效，可製作出很棒的介面和動畫效果
 - 地理資訊
 - 拖曳功能
 - 離線存取
 - 讓JS更有效率的Web Workers
 - Web Storage
 - 還有...

認識JavaScript APIs

- 影音多媒體 (video & audio)
- 繪圖平台 (canvas)
- 拖曳操作 (drag & drop)
- 地理定位 (geolocation)
- 離線快取 (offline)
- 離線儲存 (web storage)
- 表單功能 (form)
- web workers
- web sockets
-

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/>

HTML5 -- New Tags

- 文件結構

- header
- section
- article
- aside
- nav
- footer
- hgroup

- 內嵌外部內容

- video
- audio
- source
- canvas
- figure
- figcaption
- embeded

- 表單

- keygen
- output
- progress
- meter
- ...

- 文字及其他

- mark
- ruby/rt/rp
- time
- command
- details
- datalist

2014年10月底已完全底定。

主要結構：

- header
- nav
- section
- aside
- footer



header
nav(navigator)
section
aside
footer

- header
- nav
- section
 - article 1
 - article 2
- footer



header
nav
article 1
article 2
footer

III、繪圖 | 畫布 { Canvas }

簡介Canvas

- 是HTML5中最重要的應用程式工具。
- 在一個固定的長寬裡，自由的繪製任何形狀的圖，像是把 browser當成簡易版的小畫家。
- 基本功能：
矩形 | 線條 | 繪製文字 | 繪製圖片 | 陰影 | 上色 | 漸層 | 曲線
- 進階功能：
動態圖表 | 小畫家 | 基本動畫

Canvas : 基本的屬性與方法

寫在 **.html** 中

width	default 為 300px
-------	-----------------

height	default 為 150px
--------	-----------------

寫在 **.js** 中

getContext ()	取得繪圖環境
---------------	--------

toDataURL ()	取得 canvas 內容
--------------	--------------

Canvas : 2D繪圖的屬性與方法

```
canvas物件.getContext("2d");
```

- 如何開始?

- 先與HTML的canvas取得聯繫

```
var x = document.getElementById("canvas");
```

- 規劃2D繪圖環境

```
var canvas = x.getContext("2d");
```

Canvas API : 基本方法

方法	
beginPath()	開始路徑；重設路徑
closePath()	關閉路徑
moveTo(x, y)	移到某個位置(座標)
lineTo(x, y)	畫線
stroke()	輸出外框
fill()	輸出填滿的內容
rect(x, y, width, height)	矩形
strokeRect(x, y, width, height)	輸出矩形--框
fillRect(x, y, width, height)	輸出矩形--填滿
clearRect(x, y, width, height)	
clip()	路徑內部為裁切的範圍

Canvas API : 文字 | 圖片

繪製文字的方法與屬性	
fillText('文字', x, y);	實心文字
strokeText();	空心文字(描邊)
fillStyle strokeStyle	顏色
font	與CSS寫法相同
textAlign	文字對齊的位置， left right center start end
textBaseline	設定基準線， top hanging middle bottom alphabetic

圖片的方法
drawImage(img, x, y[, width, height]);
createPattern(image, 'repeat repeat-x repeat-y');

Canvas API : 線條 | 顏色屬性

線條屬性	
lineWidth	粗細 (1px by default)
lineCap	線條的開始與結尾的樣子 butt (default) round square
lineJoin	線條接合的樣子 miter (default) round bevel

顏色屬性	
strokeStyle	如前述
fillStyle	如前述

Canvas API : 曲線方法(1/2)

方法

```
arc(x, y, r, startAngle, endAngle, clockwise | anticlockwise);
```

```
moveTo(x0, y0);
```

```
arcTo(x1, y1, x2, y2, r);
```

Canvas API : 曲線方法(2/2)

方法

```
moveTo(x0, y0);  
quadraticCurveTo(cx, cy, x1, y1);
```

```
moveTo(x0, y0);  
bezierCurveTo(cx0, cy0, cx1, cy1, x1, y1);
```

Canvas API : 漸層

方法

--設定[線性|放射狀]漸層

createLinearGradient(x0, y0, x1, y1);

createRadialGradient(x0, y0, r0, x1, y1, r1);

--設定顏色

addColorStop(offset,color,...);

Canvas API : 其他屬性與方法

陰影屬性	
shadowColor	default:透明
shadowOffsetX	default:0
shadowOffsetY	default:0
shadowBlur	default:0

方法	
toDataURL();	把圖加進來，圖檔品質 (0~1)
setTimeout()	設定固定時間重複
setInterval()	設定固定時間重複

IV、影音多媒體 { Video & Audio }

Video & Audio

- HTML5新增了<video>和<audio>來解決過去大多的網路使用者，都得仰賴外掛程式(像是Adobe Flash 和 MS Silverlight)才能支援視訊和音訊的問題。
- 但在2010-01，Chrome3、Firefox3.5、Safari4即開始支援。
- HTML5允許任何影片格式，支援與否卻取決於瀏覽器，因瀏覽器的不同，可以播放的影片格式(video format)也有些不同。
- video格式：Ogg Theora(.ogv)、H.264(.mp4)、WebM(.webm)
audio格式：Ogg Vorbis(.ogg)、MP3(.mp3)、WAV(.wav)
- <http://www.mirovideoconverter.com>

Video 語法

寫法1：`<video src="屬性值" 屬性="屬性值">...</video>`

寫法2：`<video 屬性="屬性值" 屬性="屬性值">
 <source src="檔案的路徑+檔名">
 <source src="檔案的路徑+檔名">
</video>`

屬性	說明	example
src	影片來源	src = "video/temp.mp4"
controls	顯示控制項	controls controls="controls"
width	播放器寬度	width = "640"
height	播放器高度	height = "360"
loop	循環撥放	loop 不寫
autoplay	自動撥放	autoplay 不寫
poster	顯示代表圖	poster = "images/pink.gif"

Video+CSS+JavaScript： Programming the Video Player

- 將HTML檔案加上video標籤
 把video的功能鑲嵌到網頁中(架構+內容)
- 接下來使用CSS強化屬性
 根據網頁的主題、色調，量身定做一個video(呈現)
- 最後加上JavaScript語言
 讓user能更方便的操控畫面(行為)

Audio 語法

寫法1：`<audio src="屬性值" 屬性="屬性值">...</audio>`

寫法2：`<audio 屬性="屬性值" 屬性="屬性值">
 <source src="檔案的路徑+檔名">
 <source src="檔案的路徑+檔名">
</audio>`

屬性	說明	example
src	音樂來源	src = "audio/temp.mp3"
controls	顯示控制項	controls
loop	循環撥放	loop 不寫
autoplay	自動撥放	autoplay 不寫
preload	頁面載入即撥放	preload 不寫

JavaScript API：對於Video的控制

屬性與方法	說明
src	來源[URL]
currentSrc	實際讀取的來源[URL]
currentTime	目前播放位置的時間[秒數]
startTime	可以開始播放的時間[秒數]
duration	顯示資料長度[秒數]
paused	暫停中嗎？
defaultPlaybackRate	預設播放速度[0 by default]
playbackRate	播放速度[0 by default]，正數往前播，負數往後播
ended	結束了嗎？
muted	靜音嗎？
volume	音量[介於0.0~1.0]
load()	讀取影片
play()	播放
pause()	暫停

讀取與播放時，可能發生的事件

讀取時	播放時
emptied	play
loadstart	playing
progress	timeupdate
loadmetadata	waiting
loadeddata	ended
canplay	
canplaythrough	
load	
stalled	
suspend	
abort	
error	
loadend	

視訊字幕

- WebVTT格式(Web Video Text Tracks Format) ,
副檔名 : .vtt

```
hour:minutes:seconds-->hour:minutes:seconds  
字幕.....
```

- 字幕格式還有SRT
 - 使用<track>加入字幕
- ```
<track src="sample.vtt"
 srclang="en"
 kind="subtitles"
 label="English"
 default>
```

# V、拖放操作 { Drag & Drop }

# Drag & Drop

- 最早是IE5的正式標準
- 在使用拖放功能之前，必須先經由draggable屬性指定要拖曳的元素。
- 放下的地點沒有特別的限制，但拖放過程要透過事件來處理。
- 操作方法：
  1. 首先要指名開始處理拖曳事件
    - 寫成拖曳事件的function
  2. 接著設定該物件是否可以被拖曳
    - draggable設定為true | false
  3. 置放物件：default都是不接受該項操作，所以要先取消預設操作
    - event.preventDefault()
  4. 拖曳時會產生的事件
    - dragstart|drag|dragend
  - 置放物件會產生的事件
    - dragenter|dragover|dragleave|drop

# Drag & Drop : 對於Video的控制

事件	說明	處理拖曳事件的物件為 : <b>e.dataTransfer</b>
dragstart	拖曳開始	dropEffect 指定置放時的游標形狀 copy link move none
drag	拖曳中	effectAllowed 指定拖曳時的游標形狀 copy link move none copyLink copyMove  linkMove all
dragend	拖曳結束	
dragenter	拖進該範圍	
<b>dragover</b>	停在裡面	types
dragleave	脫離該範圍	clearData (type) 開始拖曳時會呼叫此方法
<b>drop</b>	放下	<b>setData (type, data)</b> <b>getData (type)</b> <b>files</b> 置放時會呼叫此方法
		setDragImage (img, x, y) 指定要做為圖示的圖片
		addElement (target) 指定非圖片的物件當作圖示



## **VI、檔案處理 { File API }**

# File API

- 何謂檔案？
- 讀取檔案資訊
- 讀取檔案內容
  - 1. 搭配<input type="file">選擇開啟的檔案  
常用事件：change  
文字檔案 | 圖檔 | 影片
  - 2. 直接從檔案總管將檔案拖曳到瀏覽器  
文字檔案 | 圖檔 | 影片

# 檔案

## 檔案的內建物件

FileList	選取多重檔案 <code>&lt;input type="file" multiple&gt;</code>
File	
<b>FileReader</b>	讀取File和Blob的資訊
Blob	Binary Large Object 二進位檔案

# 讀取檔案資訊

name	檔案名稱
type	MIME類型，若無法對應則出現空白
size	檔案大小(單位：byte)
lastModifiedDate	最後修改日期時間
root fullPath	

```
var file = document.getElementById('myFile').files[0];
.....
var message = 'File Name : ' + file.name + '\n';
message += 'File Size : ' + file.size + '\n';
message += 'File Type : ' + file.type + '\n';
message += 'Last Modified : ' + file.lastModifiedDate;
```

## 讀取檔案內容：`<input type="file">`

```
fileReader = new FileReader();
```

方法	
<b>readAsText()</b>	以純文字格式回傳結果
<b>readAsDataURL()</b>	以DataURL格式回傳結果，讀取圖檔和影片都使用此方法
abort()	中斷檔案讀取
屬性	
result	讀取到的內容
total	讀取的檔案大小
loaded	已讀到的檔案大小
事件	
loadstart   progress   load   abort   error   loadend	

# 讀取檔案內容：直接拖曳

## 使用物件：**e.dataTransfer**

事件物件的方法	
preventDefault()	
拖曳事件的屬性	
draggable	[optional] <pre>&lt;div id="fileContent" draggable="true"&gt; &lt;/div&gt;</pre>
置放 (drop) 事件	
dragenter	拖進該範圍
<b>dragover</b>	停在裡面
<b>drop</b>	放下

## VII、地理定位 { Geolocation }

# Geolocation API

· Geolocation 是藉由多種類型的資料收集機制，識別使用者或運算裝置的地理位置。一般而言，大部分的地理定位服務利用網路路線規劃位置或利用內部 GPS 裝置，來判斷位置。

Geolocation API 是否可使用需視裝置類型而定，部分瀏覽器/裝置支援此功能，其餘則不支援，因此，請記得並非所有的網路應用程式皆可使用地理定位。

· Geolocation 提供的方法：

1. **getCurrentPosition()**--單次擷取目前的位置
2. **watchPosition()**--能持續監控使用者位置，並定期確認是否有移動

以上兩種方法都是以非同步方式確認使用者所在位置。

※若使用者第一次連到此網站，瀏覽器一定會強制限制固定交談窗，詢問是否願意公開位置。

3. **clearWatch()**--清除監控

- `getCurrentPosition()` 與 `watchPosition()` 共用功能，且接收參數也相同

`getCurrentPosition()` 與 `watchPosition()`的第一個參數

### 1. 成功時的處理函數：

物件：事件物件 (`position` | `e` | `evt` | `event`)

內含的兩個屬性：`coords` 物件和 `timestamp` (時間戳記)

`coords` 物件的屬性--第一層級

- 緯度：`latitude` (單位：`double`)
- 經度：`longitude` (單位：`double`)
- 準確度：`accuracy` (單位：公尺)

`coords` 物件的屬性--第二層級

- 海拔高度：`altitude` (單位：`double`)
- 準確度：`altitudeAccuracy` (單位：`double`)
- 方向：`heading` (單位：`double`)
- 速度：`speed` (單位：`double`)

`getCurrentPosition()` 與 `watchPosition()` 的第二個參數，  
非必要操作

## 2. 錯誤時的處理函數：

物件：事件物件 (`error` | `e` | `evt` | `event`)

屬性：

◦ 錯誤碼：`code` (單位：`unsigned short`)

code的屬性值		
UNKNOWN_ERROR	0	未知錯誤
PERMISSION_DENIED	1	使用者不同意公開位置
POSITION_UNAVAILABLE	2	找不到使用者位置
TIMEOUT	3	逾時

◦ 錯誤訊息：`message` (單位：`DOMString`)

`getCurrentPosition()` 與 `watchPosition()`的第三個參數，  
非必要

### 3. 設定地裡位置：

屬性：

- `enableHighAccuracy` (單位：`bool` , `false` by default)  
//是否啟用高精準度功能
- `timeout` (單位：毫秒, `infinity/0` by default)  
//指定逾時的時間
- `maximumAge` (單位：毫秒, `0` by default)  
//可接受多久以前的資料

例如：若要取得高精準度功能，並設定10秒後逾時，且不使用舊的位置資料。

```
navigator.geolocation.getCurrentPosition(success,
 error, {
 enableHighAccuracy: true,
 timeout: 10000,
 maximumAge: 0
 });
```



# Google Maps API

參考網址：

<https://developers.google.com/maps/documentation/javascript/?hl=zh-TW>

寫在.html

```
<div id="message"
 style="width:1000px; height:800px;">
</div>
```

載入Google Maps API

```
<script src="http://maps.google.com/maps/api/js?sensor=false">
</script>
```

# 建立一個地圖物件

```
var map = new google.maps.Map(area,option);
```

//area：網頁上呈現出的地圖區塊範圍  
//option：地圖資訊

地圖資訊	
zoom	地圖的比例，數字越大顯示的區域就越大
center	地圖的中心點，使用 google.maps.LatLng 物件來表示。 <pre>var latlng=new google.maps.LatLng(latitude,longitude);</pre>
mapTypeId	地圖形式，衛星圖或街道圖 <pre>google.maps.MapTypeId.ROADMAP</pre> 地圖樣式的常數： ROADMAP：平常看到的那樣 SATELLITE：地圖方塊 HYBRID：以上兩者的混合圖 TERRAIN：顯示實際起伏

# 目前位置：Marker

```
var marker = new google.maps.Marker({position:經緯度,
map:地圖內容});
```

position	使用 google.maps.LatLng 物件來表示
map	使用 google.maps.Map 物件來表示
title	變換文字
icon	變換圖檔

```
var latlng=new google.maps.LatLng(lati, longi);
var map=new google.maps.Map(document.getElementById('message'), {
 zoom: 14,
 center: latlng,
 mapTypeId: google.maps.MapTypeId.ROADMAP
});
var image='../images/flag.png';
var marker=new google.maps.Marker({
 position: latlng,
 map: map,
 icon: image
});
```

# 地理位置編碼：Geocoder

```
var geocoder= new google.maps.Geocoder();
```

```
geocode(GeocoderRequest,callback(GeocoderResult,GeocoderStatus));
```

GeocoderRequest	建立編碼的相關資訊
GeocoderResult	編碼的回傳值 results[0].geometry.location
GeocoderStatus	編碼的回傳狀態

```
function geocodeAddress(geocoder, resultsMap) {
 var address = document.getElementById('address').value;
 geocoder.geocode({'address': address}, function(results, status) {
 if (status === google.maps.GeocoderStatus.OK) {
 resultsMap.setCenter(results[0].geometry.location);
 var marker = new google.maps.Marker({
 map: resultsMap,
 position: results[0].geometry.location
 });
 } else {
 alert('Geocode was not successful for the following reason: ' + status);
 }
 });
}
```

## **VIII 、資料儲存 { Web Storage }**

# Web Storage

- 網頁儲存區：cookie | web storage | IndexedDB
- 網頁儲存區是為了在client的磁碟上保存少量資料的儲存區。之前都是用cookie來處理。
- W3C將Web Storage定義為client端的 Javascript 環境中的一種實作的 Storage 介面的實體。
- 此介面提供了一組基於 key / value 的操作方法，隱藏了資料存續細節。也就是一個 storage 就是一個 hash table，你只需要按照 hash table 的方式存取資料，client端的底層會幫忙處理資料存續的工作，完全不必知道資料如何存取資料庫。
- Web Storage 目前有兩種型態的儲存體：localStorage和sessionStorage
- 就 Javascript 來說，就是兩個全域變數：localStorage 和 sessionStorage

## · Local Storage(本機儲存區)

- localStorage 的持續時間與存在範圍與 Cookie 類似。

它的持續時間由撰寫者指定，不會隨著瀏覽器關閉而自動終止。

它的存在範圍，同一個網站的所有網頁都會使用同一個 localStorage

- 把值放到儲存區中

```
localStorage.settings = 'ABC';
```

```
//或 localStorage["settings"] = 'ABC';
```

```
//或 localStorage.setItem("settings", 'ABC');
```

- 取出值

```
var value = localStorage.settings;
```

```
var value = localStorage["settings"];
```

```
var value = localStorage.getItem("settings");
```

- 刪除值

```
delete localStorage.settings;
```

```
delete localStorage["settings"];
```

```
localStorage.removeItem("settings");
```

- 刪除所有資料

```
localStorage.clear();
```

## · Session Storage(區段儲存區)

- 在新分頁或新視窗中開啟連結時，client端程式將會為新開啟的視窗建立一個新的session，每一個session代表一組獨立的可用資源。
  - 而 sessionStorage 就是屬於session管理的資料項目。代表每個視窗都會有自己的 sessionStorage；不同視窗的 sessionStorage 就是不同的內容。
  - 當網頁關閉時，表示此session結束了，所以此 sessionStorage 會被刪除。等到下次再開啟此網頁時，sessionStorage 的內容將會重新開始。
  - 因此，sessionStorage 只適合用於儲存暫時的資料。
- 結論： sessionStorage 物件是個暫存性的儲存區域，其中的資料以 session 為基礎，每個 session 有專屬的儲存區域，除了使用者對其進行異動之外，其間的資料會隨著 session 結束而結束。
- localStorage 和 sessionStorage 的傳回值皆為 Storage 物件，用法完全相同

## • 使用Javascript對WebStorage的控制(方法)

方法	傳回值	
length	int	回傳儲存的資料數目
key(index)	string	回傳index
getItem(key)	value	回傳index對應的資料
setItem(key, value)	void	儲存index的資料值
removeItem(key)	void	刪除key所對應的資料
clear()	void	刪除所有的資料

- 使用Javascript對WebStorage的控制(屬性)

屬性	
key	要變更的資料的 <b>key</b>
oldValue	變更前的資料(複本)
newValue	變更後的資料(複本)
url	事件發生的來源
storageArea	要變更的儲存區的 <b>reference</b>

```
/*customizeVideo.html*/
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8"/>
 <title>Video</title>
 <link rel="stylesheet" href="main.css">
 <script src="new.js"></script>
</head>
<body>
 <section id="skin">
 <video id="myMovie" width="640" height="360">
 <source src="videos/videoplayback.mp4">
 </video>
 <nav>
 <div id="buttons">
 <button id="playButton">Play</button>
 </div>
 <div id="defaultBar">
 <div id="progressBar"></div>
 </div>
 <div style="clear:both;"></div>
 </nav>
 </section>
</body>
</html>
```

```
// main.css
body{
 text-align:center;
}
section,nav{
 display:block;
}
#skin{
 width:700px;
 margin:10px auto;
 padding:5px;
 background:red;
 border: 3px dashed black;
 border-radius:10px;
}
nav{
 margin:5px auto;
}
#buttons{
 float:left;
 width:70px;
 height:22px;
}

}
#defaultBar{
 float:left;
 width:600px;
 height:16px;
 padding:4px;
 background:yellow;
 border: 2px solid black;
}
#progressBar{
 width:0px;
 height:16px;
 background:blue;
}
```

```

// new.js

function doFirst() {
 barSize=600;
 myMovie = document.getElementById('myMovie');
 playButton = document.getElementById('playButton');
 defaultBar = document.getElementById('defaultBar');
 progressBar = document.getElementById('progressBar');

 playButton.addEventListener('click',playOrPause,false);
 defaultBar.addEventListener('click',clickedBar,false);

}

function playOrPause() {
 if(!myMovie.paused && !myMovie.ended) {
 myMovie.pause();
 playButton.innerHTML = 'Play';
 window.clearInterval(updateBar);
 }else{
 myMovie.play();
 playButton.innerHTML = 'Pause';
 updateBar = setInterval(update,500);
 }
}

function update() {
 if(!myMovie.ended) {
 var size = parseInt(barSize/myMovie.duration*
myMovie.currentTime);
 progressBar.style.width = size + 'px';
 }else{
 progressBar.style.width = '0px';
 playButton.innerHTML = 'Play';
 window.clearInterval(updateBar);
 }
}

```

```
function clickedBar(e) {
 if (!myMovie.paused && !myMovie.ended) {
 var mouseX = e.clientX - bar.offsetLeft;
 var newTime = mouseX * myMovie.duration / barSize;
 myMovie.currentTime = newTime;
 progressBar.style.width = mouseX + 'px';
 }
}
window.addEventListener("load", doFirst, false);
```