

# HTML5 & CSS3 講義

王振民

wangjenmin@gmail.com

2014.05.29

## 目錄

第 1 章、HTML 與網頁 .....	1
第 1 節、HTML 初步 .....	1
第 1 項、在 Windows 中顯示副檔名 .....	1
第 2 項、最陽春的網頁 .....	1
第 3 項、標準的 HTML 文件 .....	2
第 4 項、檢測 HTML5 的支援度 .....	2
第 5 項、HTML5 的基本樣版 .....	3
第 2 節、基本的標籤 .....	4
第 1 項、<P>...</P> 段落 .....	4
第 2 項、  換行 .....	5
第 3 項、<H?>...</H?> 標題文字 ?=1~6 .....	5
第 4 項、物件與屬性 .....	6
第 5 項、對齊方向 .....	6
第 6 項、<CENTER>...</CENTER> 置中對齊 .....	6
第 7 項、<HR> 分隔線 .....	7
第 8 項、特殊符號 .....	7
第 9 項、其它字型樣式 .....	7
第 2 章、CSS 串接樣式表 .....	8
第 1 節、在標籤內宣告樣式 .....	8
第 2 節、文字的樣式 .....	9
第 3 節、整頁套用的宣告 .....	10
第 4 節、自訂樣式 .....	10
第 5 節、指定 ID 的樣式 .....	11
第 6 節、多個標籤的樣式設定 .....	12
第 7 節、指定標籤裡的標籤樣式 .....	12
第 8 節、Pseudo-Classes .....	12
第 3 章、色彩、背景、游標 .....	14
第 1 節、色彩的表示 .....	14
第 2 節、背景 .....	14

第 3 節、 改變游標的型式.....	15
第 4 節、 觀念：樣式的繼承.....	16
第 4 章、 項目符號及編號.....	17
第 1 節、 項目符號及編號.....	17
第 1 項、 <UL>...</UL> 無序號列表，項目符號.....	17
第 2 項、 <OL>...</OL> 有序號列表，編號.....	17
第 3 項、 <LI> 項目標籤.....	17
第 2 節、 項目符號及編號的樣式.....	18
第 5 章、 超連結及圖片.....	19
第 1 節、 超連結.....	19
第 1 項、 <A href="URL">說明文字</A> 超連結 .....	19
第 2 項、 target 屬性.....	20
第 2 節、 插入圖片.....	20
第 1 項、 <IMG src=" 圖形檔的 URL"> 插入圖形檔 .....	20
第 2 項、 圖片的屬性.....	20
第 3 節、 圖片的超連結.....	21
第 6 章、 區塊.....	22
第 1 節、 建立表格.....	22
第 2 節、 表格的屬性.....	23
第 3 節、 表格的使用技巧.....	23
第 4 節、 區塊元件.....	24
第 5 節、 DIV.....	24
第 6 節、 SPAN .....	25
第 7 節、 FIELDSET .....	25
第 8 節、 版面配置.....	25
第 7 章、 位置及顯示隱藏.....	27
第 1 節、 絕對位置.....	27
第 2 節、 顯示及隱藏.....	28
第 3 節、 顯示區域.....	29
第 4 節、 文字的顯示區域.....	30
第 8 章、 其它標籤.....	31
第 1 節、 書籤.....	31
第 1 項、 <A name=#書籤名稱> 建立書籤 .....	31
第 2 節、 特殊效果.....	32
第 1 項、 <MARQUEE>...</MARQUEE> 跑馬燈 .....	32
第 2 項、 <EMBED src="URL"> 插入多媒體影片.....	32
第 3 項、 <BG SOUND src="URL"> 背景音樂.....	32
第 3 節、 自動讀取另一份文件.....	33

第 4 節、在網頁中插入另外一個網頁.....	33
第 9 章、視窗框架.....	34
第 1 節、框架頁.....	34
第 1 項、<FRAMESET>...</FRAMESET> 分割視窗 .....	34
第 2 項、<FRAME> 將 HTML 嵌入視窗 .....	34
第 2 節、FRAME 的進階應用 .....	35
第 1 項、巢狀結構.....	35
第 2 項、target 屬性 設定目的視窗.....	35
第 3 項、target 屬性的再應用 .....	36
第 4 項、預設目標框架.....	36
第 10 章、雙向互動的表單 .....	37
第 1 節、<FORM>...</FORM> 表單 .....	37
第 2 節、<INPUT> 表單中的輸入欄位 .....	37
第 1 項、TEXT 單行文字框.....	38
第 2 項、PASSWORD 密碼欄.....	38
第 3 項、SUBMIT 送出鈕.....	38
第 4 項、RESET 重設鈕.....	38
第 3 節、更多的輸入欄位.....	39
第 1 項、RADIO 單選鈕.....	39
第 2 項、CHECKBOX 多選欄.....	39
第 3 項、select 列表 下拉清單 .....	39
第 4 項、TEXTAREA 多列文字欄.....	39
第 11 章、樣式檔的連結 .....	41
第 1 節、樣式檔的建立及連結.....	41
第 2 節、局部宣告.....	41
第 3 節、宣告樣式注意事項.....	42
第 4 節、物件的命名.....	42
第 12 章、JavaScript .....	43
第 1 節、Java 與 JavaScript.....	43
第 1 項、Java .....	43
第 2 項、JavaScript .....	43
第 2 節、JavaScript 語言結構 .....	43
第 1 項、先來看看簡單的 JavaScript 範例 .....	43
第 2 項、結構.....	43
第 3 項、函數.....	44
第 4 項、變數.....	44
第 5 項、陣列.....	44
第 6 項、運算式.....	44

第 7 項、物件、屬性、方法.....	45
第 8 項、註解.....	45
第 9 項、特殊字元.....	45
第 3 節、JavaScript 與表格的實作練習 .....	45
第 4 節、條件式與迴圈.....	46
第 1 項、條件式.....	46
第 2 項、迴圈.....	47
第 3 項、break 中斷 .....	47
第 4 項、事件.....	47
第 5 節、date 物件 .....	48
第 6 節、navigator 物件 .....	49
第 7 節、window 物件.....	49
第 13 章、HTML5 新增的標籤 .....	51
第 1 節、基本介紹.....	51
第 2 節、網頁結構.....	52
第 1 項、舊的網頁結構.....	52
第 2 項、HTML5 的基本網頁架構.....	53
第 3 項、加上 hgroup.....	55
第 4 項、加上 figure 標籤 .....	55
第 5 項、使用 aside 標籤.....	56
第 6 項、nav 標籤.....	57
第 7 項、section 標籤.....	57
第 8 項、HTML5 網頁架構小結 .....	57
第 3 節、語義相關的新標籤.....	58
第 1 項、time 標籤 .....	58
第 2 項、mark 標籤 .....	58
第 4 節、表單相關的標籤.....	59
第 1 項、input 的 placeholder 屬性.....	59
第 2 項、input 的 focus 狀態 .....	59
第 3 項、Input 的 autofocus 屬性.....	59
第 4 項、Input 的 required 屬性.....	59
第 5 項、使用正規運算式驗證資料.....	60
第 6 項、Output 標籤.....	60
第 7 項、Input 的 number 型態.....	60
第 8 項、Input 的 range 型態.....	60
第 9 項、Input 的 email 型態.....	61
第 10 項、Input 的 url 型態 .....	61
第 11 項、Input 的 search 型態 .....	61

第 12 項、 Input 的 tel 型態.....	61
第 13 項、 Input 的 date 型態.....	61
第 14 項、 Input 的 color 型態.....	62
第 15 項、 Progress 標籤.....	62
第 16 項、 Meter 標籤 .....	62
第 5 節、 影音.....	62
第 1 項、 HTML5 的音樂 .....	62
第 2 項、 直接使用 html5 的 video 播放影片 .....	63
第 3 項、 加上 JavaScript 控制 video 物件.....	64
第 6 節、 Canvas .....	65
第 1 項、 繪製矩形.....	65
第 2 項、 繪製矩形時加上透明度.....	65
第 3 項、 繪製三角形.....	66
第 4 項、 繪製弧形.....	67
第 5 項、 線性漸層色彩.....	67
第 6 項、 圓形漸層色彩.....	68
第 7 項、 不同圓心半徑的圓形漸層色彩.....	69
第 8 項、 基本繪圖程式.....	69
第 9 項、 描繪方格紙。.....	74
第 10 項、 在方格紙上繪製資料。.....	75
第 11 項、 皮球彈跳的動畫。.....	77
第 14 章、 CSS3 新增功能.....	79
第 1 節、 css3 網頁功能.....	79
第 1 項、 繪製圓角.....	79
第 2 項、 設定文字影陰.....	80
第 3 項、 多個背景檔案的設定.....	80
第 4 項、 Hover 虛擬類別.....	81
第 5 項、 轉換特效.....	81
第 6 項、 轉換特效之 2 .....	82
第 7 項、 轉換特效說明.....	83
第 8 項、 多欄版面.....	84
第 2 節、 段落或文字功能.....	84
第 1 項、 ::first-line 與::first-letter 的 pseudo-elements .....	84
第 2 項、 :focus.....	85
第 3 項、 ::before.....	86
第 4 項、 ::after.....	86
第 5 項、 ::selection.....	86
第 6 項、 Attribute Selector.....	86

第 7 項、 child selector.....	87
第 3 節、 Web Storage.....	89
第 1 項、 計數器.....	89
第 2 項、 記事本.....	89
第 4 節、 XMLHttpRequest.....	90
第 5 節、 地理位置.....	92
第 1 項、 取得地理資訊.....	92
第 2 項、 地理位置—連到 google.....	93
第 3 項、 google maps 的 javascript 函式庫.....	94
第 6 節、 串接樣式表.....	94
第 1 項、 螢幕和列印之間.....	94
第 2 項、 不同解析度設定不同樣式.....	95
第 3 項、 更強化的設定.....	96
第 4 項、 不同解析度套用不同的串接樣式檔.....	96
第 5 項、 識別裝置的螢幕解析度.....	97
第 6 項、 @import 方式匯入樣式檔.....	97

# 第 1 章、HTML 與網頁

HTML (HyperText Markup Language; 超文字檔案的標記語言) 其實就是我們平常在 Internet 上瀏覽網頁時, 看到的真正文件內容。平常瀏覽網頁時, 可以在瀏覽器中執行『檢視／原始檔』檢視網頁內容, 會發現原來網頁是單純的文字內容。

我們可以將網頁當做『內容』、『標籤』所描述出來的結果, 更進一步的學習後, 網頁就可以是『內容』、『標籤』、『樣式』三者組成的結果, 就可以讓網頁呈現出多采多姿的效果了。

會不會覺得我們到目前為止, 都還沒有提到網頁上常常出現的『圖片』? 那是因為網頁其實是個純文字的檔案, 『圖片』是透過『標籤』的描述, 告訴瀏覽器要將圖片檔案顯示在網頁的某個位置的。因此, 圖形檔案(或其它類似的檔案, 例如 flash 檔) 是存在獨立的圖檔中, 和網頁的純文字檔案是分別不同的檔案。

## 第 1 節、HTML 初步

### 第 1 項、在 Windows 中顯示副檔名

在視窗環境下操作時, 系統預設是隱藏副檔名的, 雖然可以讓一般使用者只要看看 icon 就知道它的文件格式而不用煩惱什麼副檔名是什麼格式, 但是卻由於無法更改副檔名, 相對的也不能變更檔案類型。因此, 我們進行以下動作讓系統顯示副檔名:

1. 執行『開始 / 設定 / 控制台 / 資料夾選項』, 或者也可以經由任何一個資料夾 (例如: 我的文件夾、檔案總管) 執行『工具 / 資料夾選項』。
2. 切換到『檢視』標籤頁, 將『隱藏已知檔案類型副檔名』選項的打勾去除。

### 第 2 項、最陽春的網頁

我們先來製作一個最簡單的網頁, 請依照以下方式建立:

1. 呼叫出文字編輯器 (Windows 可以執行記事本、Linux 可以使用 vi)
2. 輸入“一個陽春的 HTML 文件”
3. 另存新檔為 test.htm (請注意副檔名要改為.htm 或.html)
4. 直接由檔案總管 (或瀏覽器) 雙按開啟該文件

這樣子，應該可以在瀏覽器上看到這個網頁囉！這裡我們知道很重要的一點：就算是單純的文字也可以是網頁，關鍵在於副檔名。

### 第 3 項、標準的 HTML 文件

雖然純文字就可以在瀏覽器上顯示，但卻沒有辦法設定其它樣式（例如標題、色彩、大小…），因此，我們必須加上『標籤』，透過『文字』與『標籤』的組合，才算是基本的 HTML 文件。基本的 HTML 文件骨架像底下例子，可以先製作好儲存，隨時可以套用：

```
<HTML> //HTML 文件的開始
<HEAD> //標示文件資訊
  <TITLE>歡迎光臨</TITLE> //文件標題
</HEAD>
<BODY> //本文區開始
  一個陽春的文件 //文件內容顯示的區域
</BODY>
</HTML> //HTML 文件結束
```

### 第 4 項、檢測 HTML5 的支援度

Modernizr (<http://modernizr.com/>) 是個 Open Source 的 JavaScript 的程式庫，可以幫助我們檢測瀏覽對於 HTML5 的支援程度。在這個例子裡，我們加了一個 modernizr 的 JavaScript，它會建立一個 Modernizer 的全域物件，我們可以使用這個物件，得知瀏覽器對於 HTML5 某個功能的支援程度。

請開啟瀏覽器，瀏覽「<http://www.kingbig.idv.tw/class/html5/t00.htm>」。

若想知道您現在正在使用的瀏覽器對於 HTML5 支援程度的完整測試，可以連到「<http://html5test.com/>」這裡，立刻就知道囉！



## 第 5 項、HTML5 的基本樣版

直接來看看 HTML5 的基本樣版，以下內容，可以看出在 head 區塊以上的程式，精簡了很多。

```
<!DOCTYPE html>
<html lang="zh-TW">
<head>
  <title>網站標題</title>
  <meta charset="utf-8" />
  <meta name="viewport"
    content="width=device-width,
      user-scalable=yes,
      initial-scale=1.0,
      maximum-scale=3.0" />

  <script src=modernizr.custom.24002.js></script>
  <link href="styles.css" rel="stylesheet">

  <style>
    article, aside, figure, footer, header,
    hgroup, menu, nav, section
    {
      display: block;
    }
  </style>

</head>
<body>

  <p>這是 html5 的基本樣版。</p>
  <p>在 style 區塊宣告的，都是在 html5 所新增的標籤，但為了暫時讓不支援 html5
  的瀏覽器，也能正常的解讀 html5 的網頁，將這些標籤手動設定為區塊元件。

</body>
</html>
```

這個基本的樣版，有幾個重要的修改：

1. doctype：文件型別，改的更簡單了！加上這一行，只要是支援 HTML5 的瀏覽器，就會認得它，不過，要特別注意的是，它必須是第一行，前方不能有空白或空行。
2. html 標籤：這是網頁的根元素，它也變得更簡單了。在這裡直接就指定 html 網頁內容的語系是什麼。

3. head 區塊：在 head 區塊裡，除了具有 title 標籤，最常見的還有 meta 標籤，它是 metadata 縮寫，代表一些解釋性的資訊，在 html5 裡，使用 meta 標示文件的編碼，也更簡單了。
4. 在 head 區塊裡要宣告讀取外部檔案的動作都變的比較簡單，例如，<link>讀取 css 的樣式檔案，以及<script>讀取 js 的檔案。
5. 這個範例裡，我們在 head 區塊多加了一些標籤，同時將它們都宣告為 block 元件，主要是為了相容舊版的瀏覽器，也能夠正常顯示 HTML5 的網頁內容。

## 第 2 節、基本的標籤

由上例 HTML 的基本架構中可以得知標籤大多數是由<標籤>...</標籤>所組成(只有少部份是獨立的標籤)，分別表示標籤開始、標籤結束，至於被<標籤>...</標籤>所包含的部份會被標籤所影響，具有相對應的效果或意義，例如：

<HTML>...</HTML>	//宣告 HTML 文件的起始與結束
<HEAD>...</HEAD>	//表示這一部份都是宣告文件相關資訊、定義
<BODY>...</BODY>	//真正會顯示在網頁上的內容

在 HTML 標籤中所輸入的英文是沒有分大小寫的！在剛開始的講義裡，可能會用英文大寫字，目的是為了能使文字更清楚，各位在練習時，可以不用在意英文大小寫的問題。

### 第 1 項、<P>...</P> 段落

使用任何一個文書編輯程式，一定是以『文字』與『段落』做為編排的依據，在 HTML 中也是一樣，所有的『一般文字』都建議使用<P>...</P>包圍，其中的內容就代表著一個段落。

<BODY>	
<P>大學</P>	//第一個段落
<P>大學之道，在明明德，在親民，在止於至善。知止而後有定，定而後能靜，靜而後能安，安而後能慮，慮而後能得。</P>	//第二個段落
</BODY>	

## 第 2 項、<BR> 換行

在 HTML 文件中直接按 ENTER 換行是沒有用的，例如：

```
<BODY>
  <P>大學                                     //按了 Enter 希望能夠換行？
    大學之道，在明明德，在親民，在止於至善。知止而後有定，定而後能靜，靜而後能
    安，安而後能慮，慮而後能得。</P>
</BODY>
```

實際上，會發現它們還是顯示成同一行，只是其中多了一個空格。請特別注意，在 HTML 中無論我們按了多少個 Enter 鍵、或是打了多少個空白鍵，在網頁上都只會顯示成一個空格。這些 Enter 或空白鍵主要是使我們閱讀 HTML 文件時能夠比較方便而已。

如果希望文字能強迫換到下一行而且仍是同一個段落（就像在 WORD 中使用 Shift + Enter 鍵一樣的意思），就可以使用<BR>標籤。因此，上面的網頁，應該要改成底下的方式：

```
<BODY>
  <P>大學                                     //加入 <br>
    <BR>大學之道，在明明德，在親民，在止於至善。知止而後有定，定而後能靜，靜而
    後能安，安而後能慮，慮而後能得。</P>
</BODY>
```

另外，再強調一次，我們在編輯 HTML 中所看到的格式只是讓我們閱讀 HTML 文件時能夠比較方便，並不會影響瀏覽器中看到的效果。也就是說，底下的例子和上面的例子呈現的效果是一樣的：

```
<BODY><P>大學<BR>大學之道，在明明德，在親民，在止於至善。知止而後有定，定而
後能靜，靜而後能安，安而後能慮，慮而後能得。</P></BODY>
```

## 第 3 項、<H?>...</H?> 標題文字 ?=1~6

<H1>...</H1>是標題一，層級最高，字型大小也最大，而<H6>...</H6>是標題六，在標題中的層級最低，字型大小也最小。將文字設定標題的樣式後，會自動在前後分段，因此，不需加上<P>...</P>的標籤，也沒有必要加上，因此被<H1>...</H1>包含的文字是屬於標題的層級，而不是段落文字（內文）的層級。

```
<BODY>
  <H1>李白的詩</H1>
  <H2>靜夜思</H2>
```

```

<HR>
  <P>床前明月光
  <BR>疑視地上霜
  <BR>舉頭望明月
  <BR>低頭思故鄉</P>
<HR>
  <P>取自唐詩三百首</P>
</BODY>

```

## 第 4 項、物件與屬性

在電腦的世界中，我們加入了很多『物件』的觀念，舉個例子，我們可以指定某個房間的長、寬、高，也可以指定牆壁的顏色要什麼色彩，這個觀念加入到電腦世界，可以使我們對物件的設定簡易很多。

實際使用時，我們是以『屬性＝屬性值』的方式來指定我們希望的長、寬（width=20）、高（height=120）、色彩（color=blue）…等，當然，不同的物件有不同的屬性名稱，有些是共通的，有些是特定某個物件才具備的屬性，在接下來的內容中，我們會陸續介紹不同的屬性，可以在不同的標籤中試試看。

若在同一個標籤中要使用多個屬性設定，記得使用空格間隔就可以了，例如：

```
<標籤名稱 屬性 1 = 屬性值 屬性 2 = 屬性值 ...>...</標籤名稱>
```

## 第 5 項、對齊方向

```
align 設定對齊方向；屬性值有 right, center, left
```

使用<P>…</P>設定段落文字，可以加上屬性設定，讓文字可以靠左、置中、靠右對齊。例如：

```

<P align=center>這一個段落的文字會置中對齊。</P>
<P>這一個段落的文字不會被影響。</P>

```

就如同前面所說的，ALIGN 的屬性也可以運用在<H1>之中，或其它的標籤。屬性的設定效果只發生在設定的標籤中（跟隨著標籤的起始與結束的），也就是說，我們設定為<P align= center>的置中對齊效果，在遇到</P>結束段落標籤時也同時結束了屬性設定的效果。

## 第 6 項、<CENTER>…</CENTER> 置中對齊

在<CENTER>…</CENTER>之中的內容都會置中對齊，通常是設定多個標籤要置中對齊使用，總必一個一個設定要簡單、方便多了。

## 第 7 項、<HR> 分隔線

使用分隔線可以使段落更分明。文件遇到<HR>標籤，會將<HR>前後自動分段。

## 第 8 項、特殊符號

若要在網頁上顯示特殊符號，必須輸入特殊的內容，才能在顯示時得到我們需要的特殊號。

符號	輸入文字
空白	&nbsp;
>	&gt;
<	&lt;
&	&amp;
“	&quot;
®	&reg;
©	&copy;

## 第 9 項、其它字型樣式

還有以下幾種字型效果：

<B>...</B>	粗體字
<I>...</I>	斜體字
<TT>...</TT>	打字機字體
<U>...</U>	加上底線
<STRIKE>...</STRIKE>	加刪除線
<SUB>...</SUB>	下標字
<SUP>...</SUP>	上標字

## 第 2 章、CSS 串接樣式表

前面學會了一些 HTML 的基本標籤設定，就像你學的，字型大小才七種等級，HTML 原有的標籤樣式太少，因此，CSS (Cascading Style Sheet：串接樣式表) 擴充了 HTML 的標籤樣式。使用 CSS 的好處可多著，慢慢的你就會發覺的。

### 第 1 節、在標籤內宣告樣式

這種方式是和原本 HTML 類似，將它當做其中的一個屬性，透過屬性值的設定以得到不同的樣式。對我們而言，使用的習慣沒有改變，直接將樣式宣告在標籤之中，因此也只有這一部份的標籤會受到樣式宣告的影響，其它的標籤不會被影響。樣式的宣告方式如下：

```
style="屬性 1:屬性值; 屬性 2:屬性值; 屬性 3:屬性值;"
```

請注意，這一串是要放在標籤之中當成其中的一個屬性，而且，指定屬性值是用『:』而不是等號，二種以上的樣式是用『;』來區隔而不是空格。

例如：

```
<P style="color:red; font-size:48;">這一段文字是紅色大字</P>  
<P>這一段是原本的樣式，沒有改變</P>
```

CSS 樣式的設定和屬性一樣，有些樣式是每個標籤共通的，而有些樣式是某些標籤特有的，反正，你都可以試試看。

先來看看和文字設定有關的樣式吧！

## 第 2 節、文字的樣式

在 CSS 中，文字的樣式多了很多，但都很簡單，做了你就會了，參考如下：

功能	屬性	範例	
字型	Font-Family	font-family:'超研澤粗行楷','標楷體'	
字型大小	Font-Size	font-size:66	
字型樣式	Font-Style	font-style:normal	//正常字
		font-style:italic	//斜體字
轉為大寫	Font-Variant	font-variant:normal	//正常字
		font-variant:small-caps	//轉為大寫字
字距	Letter-Spacing	letter-spacing:10	//字距為 10
列高	Line-Height	line-height:3	//列高為 3 行高
首行文字縮排	Text-Indent	text-indent:60	//內縮 60 點
標示線	Text-Decoration	text-decoration:none	//沒有標示線
		text-decoration:underline	//底線
		text-decoration:overline	//上標線
		text-decoration:line-through	//刪除線
轉換大小寫	Text-Transform	text-transform:none	//不改變
		text-transform:capitalize	//首字大寫
		text-transform:uppercase	//全部大寫
		text-transform:lowercase	//全部小寫
對齊方式	Text-Align	text-align:right	//靠右對齊
		text-align:center	//置中對齊
		text-align:left	//靠左對齊
		text-align:justify	//左右對齊
上下對齊方式	Vertical-Align	vertical-align:super	//上標字

功能	屬性	範例
	vertical-align:sub	//下標字

作業：

請將 10boss.htm 開啟，並將網頁最佳化，也就是說將不必要的標籤移除，並自行設定其中的樣式。

## 第 3 節、整頁套用的宣告

在標籤內使用樣式設定，很簡單，你一學就會，但是現在你一定會覺得很麻煩了！為什麼？因為我們必須在每個標籤中都使用 style 來設定該標籤的樣式，有沒有方法可以只需要定義一次，就可以通用的呢？

有的，我們可以在<HEAD>...</HEAD>區段中定義一個<STYLE>...</STYLE>區段，在其中定義標籤的樣式（例如 P 標籤），則該網頁中只要有使用到這個標籤（使用到 P 標籤）的樣式都會依定義的樣式顯示，宣告方式如下：

```
<HEAD>
<style>                                //新增一個<style>區段
  P {                                    //指定 P 標籤，以 { } 啟始及結束
    color: red;                          //設定不同的樣式，記得要加上；
    font-size: 24;
  }
</style>
</HEAD>
```

上面這一段，我們也可以寫成這樣，比較不佔空間，但閱讀起來就比較難過一些了，也許，等你功力深厚些自然就不成問題囉！

```
<HEAD>
<style>                                //和上例的效果完全一樣
  P { color:red; font-size:24; }
</style>
</HEAD>
```

設定標籤樣式時，直接輸入樣式名稱即可，不需包含 <> 符號。

## 第 4 節、自訂樣式

HTML 原本提供的標籤名稱不多，而且，在<STYLE>...</STYLE>區段中設定的樣式，對該標籤而言，也只有一種呈現的效果，使用起來還不是不夠靈活，因此我們還可以自訂樣式。



在<STYLE>...</STYLE>中以『標籤.名稱 {屬性:屬性值;}』的形式定義，就是自訂樣式，在標籤中加入『Class="名稱"』就可以套用該樣式。若自訂樣式有指定標籤，例如『P.p1 {...}』，則表示該自訂樣式只能套用在 P 標籤裡，對於其他標籤不會產生作用。如果，自訂樣式要應用於任何標籤，可以使用萬用字元，例如『\*.p1 {...}』，也可以簡化為『.p1 {...}』這種型式。

參考以下例子：

```
<HEAD>
<style>
  .P1 {                                //新增自訂樣式為 P1
    color:red;                          //設定樣式
    font-size:24
  }
</style>
</HEAD>
<BODY>
  <P>原本的樣式</P>
  <P class="P1">套用自訂樣式</P>      //套用 P1 的樣式
</BODY>
```

在<STYLE>...</STYLE>中設定標籤的樣式，是直接使用標籤名稱來設定，而自訂樣式時，記得要在自訂樣式的名稱前加上『.』，而且，自訂樣式的名稱不要使用原有的標籤（例如 H1、P），但是可以將自訂樣式的名稱設定為 P1、P2...。

## 第 5 節、指定 ID 的樣式

我們可以將某個標籤指定 ID，設定的 ID 在整個網頁裡必須是唯一的、不能重覆，設定樣式的方式請參考如下：

```
<HEAD>
<style>
  #P01 {                                //新增 ID 為 P01 的樣式
    color:red;                          //設定樣式
    font-size:24
  }
</style>
</HEAD>
<BODY>
  <P>原本的樣式</P>
  <P id=P01>指定 ID 的樣式</P>      //指定 id
</BODY>
```

## 第 6 節、多個標籤的樣式設定

若有多個標籤的樣式相同，可以同時設定，不同標籤之間以『,』分隔即可。

```
<style>
  h1, h2, h3, h4 {
    color:red;
    font-size:24
  }
</style>
```

## 第 7 節、指定標籤裡的標籤樣式

若在樣式裡設定的標籤是『#div01 p』，彼此間只以空白分隔的，代表的是他們有上下層的關係，也就是說，p 標籤是要在 id 為 div01 的標籤內部，這個樣式才有作用。

```
<HEAD>
<style>
  #div01 p {
    color:red;
    font-size:24
  }
</style>
</HEAD>
<BODY>
  <div id=div01>
    <P>指定 ID 的樣式</P>
  </div>
</BODY>
```

## 第 8 節、Pseudo-Classes

以超連結 a 的標籤來看，還有以下四種 pseudo classes，分別表示『新的連結』、『已訪問過的連結』、『cursor 移上來時』、『按下左鍵的那個時間點』這四個作用，我們可以分別設定不同的效果。

```
<style>
  a:link {...}
```

```
a:visited {...}  
a:hover {...}  
a:active {...}  
</style>
```

另外，『:hover』也可以應用在其它標籤，如果設定『p:hover』，就可以在滑鼠游標移到 p 段落時，顯示另一種不同的樣式。

## 第 3 章、色彩、背景、游標

### 第 1 節、色彩的表示

色彩的表示方式和 HTML 的表示相同，可以使用色彩的英文字表示、也可以使用 #RRGGBB 的方式表示，不同的是，在 CSS 中是在 style 中使用 color 屬性來指定，若指定在 P 標籤中的話就是字型色彩。例如：

```
<P style=" color:#aacff;" >設定字型的色彩是淡藍色</P>
```

色彩可以使用基本色彩，例如 red, blue, green... 等，也可以使用『#RRGGBB』方式設定色彩，其中的 RR,GG,BB 分別代表紅色、綠色、藍色三原色，它的值是以 16 進位表示，最小為 00，最大為 FF（換算成十進位就是 0~255），例如：

```
#000000 //黑色
#FFFFFF //白色
#FF0000 //只有紅色，所以就是紅色
#00FF00 //只有綠色，所以就是綠色
#0000FF //只有藍色，所以就是藍色
#FFFF00 //紅色加綠色，是中間色黃色或橙色
#FF00FF //紅色加藍色，是紫色
#AAAAFF //接近白色，藍色最多，所以是淡藍色
```

### 第 2 節、背景

CSS 的樣式設定，參考如下：

功能	屬性	範例
背景顏色	Background-color	Background-color:#ffc000
背景圖片	Background-image	Background-image:URL(img1.jpg) //別忘了要加上 URL()
重複的背景	Background-repeat	Background-repeat:repeat //重覆排列 Background-repeat:repeat-x

功能	屬性	範例
		//水平方向重覆 Background-repeat:repeat-y //垂直方向重覆 Background-repeat:no-repeat //不重覆
捲動的背景	Background-attachment	Background-attachment:scroll //捲動 Background-attachment:fixed //固定
背景的位置	Background-position	Background-position: x% y% //相對位置 Background-position: x y //絕對位置，x 和 y 都是任意數值 Background-position:X Y //預設位置，選擇要放置在右上或右下，使用以下文字： X 為 left, center, right Y 為 top, center, bottom

背景色彩或背景圖片的指定，並不是只能設定在<BODY>的標籤，實際上它可以用在任何標籤，包含稍後我們介紹的區塊元件。

### 第 3 節、改變游標的型式

Cursor 樣式可以設定游標的外觀，目前可以選用的游標有十四種：

屬性	範例
Cursor	Cursor:default //改回預設型式
	Cursor:crosshair //十字
	Cursor:hand //手
	Cursor:text //輸入文字的 I 型式
	Cursor:wait //漏斗型式
	Cursor:help //箭頭旁有？號型式
	Cursor:move //移動物件的型式

屬性	範例
Cursor:e-resize	//箭頭，東方
Cursor:ne-resize	//箭頭，東北方
Cursor:nw-resize	//箭頭，西北方
Cursor:n-resize	//箭頭，北方
Cursor:se-resize	//箭頭，東南方
Cursor:sw-resize	//箭頭，西南方
Cursor:s-resize	//箭頭，南方
Cursor:w-resize	//箭頭，西方

## 第 4 節、觀念：樣式的繼承

剛剛的例子『<P class=P1>』，是先套用了 P 的樣式，然後在原有的 P 樣式中再套用 P1 的樣式，因此，如果在 P 設定了字型大小是 66，而 P1 中沒有特別設定，那麼字型大小的部份，仍然會是以 P 設定的樣式呈現。簡單的說，以原來的樣式為基礎，再加上額外的樣式設定。

## 第 4 章、項目符號及編號

### 第 1 節、項目符號及編號

#### 第 1 項、<UL>...</UL> 無序號列表，項目符號

<UL>...</UL>這組標籤可以建立項目符號。

#### 第 2 項、<OL>...</OL> 有序號列表，編號

<OL>...</OL>標籤可以建立編號。

#### 第 3 項、<LI> 項目標籤

無論使用<OL>...</OL>建立編號、或<UL>...</UL>建立項目符號，都必須配合<LI>建立其中的項目，例如：

```
<BODY>
  <P>本餐廳菜色：</P>
  <UL>                                     //第一層級的項目符號
    <LI>天上飛的                           //其中的項目，以項目符號方式顯示
    <LI>地上爬的
    <OL>                                     //第二層級的編號
      <LI>牛肉麵                             //其中的項目，以編號方式顯示
      <LI>排骨麵
      <LI>什錦湯
    </OL>
    <LI>水裡游的
  </UL>
</BODY>
```

因此，可以知道<LI>在不同的標籤中具不同的功能，反過來說，它也都只是項目而已。

type 屬性可指定編號或項目符號的樣式；屬性值有 1, A, a, l, i

## 第 2 節、項目符號及編號的樣式

項目符號的樣式如下：

功能	屬性	範例
項目清單樣式	List-Style-Type	List-Style-Type:Disc //實心圓
		List-Style-Type:Circle //空心圓
		List-Style-Type:Square //實心矩形
編號樣式	List-Style-Type	List-Style-Type:Decimal //阿拉伯數字
		List-Style-Type:Lower-Roman //小寫的羅馬數字
		List-Style-Type:Upper-Roman //大寫的羅馬數字
		List-Style-Type:Lower-Alpha //小寫的英文字母
		List-Style-Type:Upper-Alpha //大寫的英文字母
圖片代替符號	List-Style-Image	List-Style-Image:url(位址) //在括弧中指定圖片的檔名。若同時設定 List-Style-Type 可在找不到圖片時替代顯示
項目符號位置	List-Style-Position	List-Style-Position:Inside //內縮
		List-Style-Position:Outside //外突
混合設定	List-Style	List-Style: Disc Inside //可混合設定，在項目和項目之間要空格



## 第 5 章、超連結及圖片

### 第 1 節、超連結

#### 第 1 項、<A href=" URL" >說明文字</A> 超連結

建立超連結的方式其實使用 A 標籤，再加上 href 的屬性，才能製作出超連結，超連結可運用的地方有很多，最常使用的方式有：連結站內的網頁、其它網站、電子郵件等，來看看底下的例子：

```
<BODY>
  <P><A href=test.htm>李白 靜夜思</A> //連結站內的網頁
  <P><A href=http://www.kingbig.idv.tw>山賊老巢</A> //連到網站
  <P><A href=mailto:kingbig@kingbig.idv.tw>電子郵件</A> //傳送電子郵件
  <P><A href=ftp://ftp.nctu.edu.tw>連結至 FTP</A> //使用 ftp 協定
</BODY>
```

如果要連結到其它的網站，需使用完整的 URL，記得要包含通訊協定，若連結的是站內的檔案，使用相對位址比較適合，至於電子郵件的超連結，在通訊協定部份輸入『mailto:』，不需要輸入『//』。

另外，電子郵件部份，還可以額外設定，在點選電子郵件超連結的時候，能同時填入信件的主旨及內文，例如：

```
<P><A href=mailto:kingbig@kingbig.idv.tw?subject=test>電子郵件</A>
  //包含主旨的電子郵件
<P><A href=mailto:kingbig@kingbig.idv.tw?subject=test&body=測試郵件>電子郵件</A>
  //包含主旨及內文的電子郵件
```

我們在 Outlook Explorer 中發送信件給通訊錄中的某個朋友時，應該是只看的到名字，而看不到電子郵件，實際上它的格式是『王大<kingbig@kingbig.idv.tw>』，因此，底下的方式，可以做到這個功能，但是，記得要加上雙引號：

```
<P><A href="mailto:王大<kingbig@kingbig.idv.tw>">發信給我</A>
```

## 第 2 項、target 屬性

在超連結中可以加入 target=\_blank 屬性，使用超連結在點選之後，會以開啟新視窗的方式顯示超連結的內容。

## 第 2 節、插入圖片

### 第 1 項、<IMG src=" 圖形檔的 URL" > 插入圖形檔

要在網頁中顯示圖片，需要使用 IMG 標籤，配合 src 屬性，才能在 HTML 文件中插入圖片。

如果圖片是在 Internet 其它網站上的，只需要在 src 屬性部份輸入完整的 URL 位址就可以了。如果圖片是在自己的網站中，建議使用相對路徑的方式，會比較恰當，而且，上傳網頁時，別忘了圖片也得同時上傳哦！

來看看這個例子，圖片和文字是放置於同一個段落，使用<BR>強迫換行：

```
<P><IMG src=最新的進化論.jpg>  
<BR>人類是由猩猩所演變而成的。</P>
```

### 第 2 項、圖片的屬性

圖片還有其它的屬性可以同時運用，如下：

border 屬性是設定邊框的大小；屬性值是像素（任意整數），0 表無邊框  
width 屬性可設定圖片寬度；屬性值可以是百分比、像素  
height 屬性設定圖片高度；屬性值可以是百分比、像素  
vspace 屬性設定圖片上下的間距；屬性值是像素  
hspace 屬性設定圖片左右的間距；屬性值是像素  
align 屬性設定圖片對齊方式，也是文繞圖的設定；屬性值是 left、right

底下有一些例子，參考看看：

```
<IMG src=最新的進化論.JPG border=5> //指定圖片邊框厚度為 5  
<IMG src=最新的進化論.JPG border=0 width=586 height=245>  
//指定圖片無邊框、寬度=586、高度=245  
<IMG src=最新的進化論.JPG border=0 width=293 height=120 vspace=20 hspace=20>  
//同時指定圖片邊框、寬、高、間距
```

另外有一些事項，參閱如下：

- 使用 width、height 屬性可以控制影像在瀏覽器中顯示的大小，但是並不會改變影像真正的大小，傳輸所花費的時間自然也沒有改變，除此之外，還可以使瀏覽器在影像尚未傳送過來前就會先將圖片所需的版面預留下來。
- 如果，我們不知道圖片原本的大小時，除了在影像軟體中得知，還可以在瀏覽器的圖片上按右鍵，點選內容，也可以知道圖片的大小。
- align 屬性除了可以設定圖片靠左、或靠右（其實也可以置中）對齊之外，同時也是設定成文繞圖片的效果。另外，還有一些不同的屬性值可以設定：

baseline, bottom, absbottom, top, texttop, middle, absmiddle

```
<BODY>
  <H1>人類的進化</H1>
  <P><IMG src=最新的進化論.JPG align=left> //設定成靠左的文繞圖效果
  <P>人類是由猩猩所演變而成的。
  <P>科技文明的腳步不斷地向前邁進，長久以來人類對於科技將帶天涯若比鄰的世界，一直存在著無限的憧憬。前幾天前才有人在慶祝電腦問世五十週年。而 INTERNET 的蓬勃發展，也不過是這幾年的事。
</BODY>
```

### 第 3 節、圖片的超連結

我們建立超連結時，是使用：

```
<A href="URL">說明文字</A>
```

可以稍加變化如下，就可以替圖片加上超連結了，如下：

```
<A href="URL"><IMG src="圖形檔的 URL"></A> 或者
<A href="URL"><IMG src="圖形檔的 URL">還可以包含文字</A>
```

我們可以將之前的網頁內容修改如下，將圖片也加上超連結：

```
<BODY>
  <H1>人類的進化</H1>
  <A href=http://www.discovery.com><IMG src=最新的進化論.jpg></A>
  <BR>人類是由猩猩所演變而成的。
</BODY>
```

加上超連結的圖片，會主動加上外框，可以用 border 屬性來取消外框。

## 第 6 章、區塊

### 第 1 節、建立表格

建立表格所需使用的標籤分別是：

<code>&lt;TABLE&gt;...&lt;/TABLE&gt;</code>	//表格
<code>&lt;TR&gt;...&lt;/TR&gt;</code>	//列
<code>&lt;TD&gt;...&lt;/TD&gt;</code>	//儲存格

表格的基本架構如下：

<code>&lt;TABLE&gt;</code>	//產生表格
<code>&lt;CAPTION&gt;表格標題&lt;/CAPTION&gt;</code>	//表格的標題
<code>&lt;TR&gt;</code>	//定義一行
<code>&lt;TD&gt;儲存格&lt;/TD&gt;</code>	//定義一個儲存格
<code>&lt;TD&gt;儲存格&lt;/TD&gt;</code>	
<code>&lt;/TR&gt;</code>	
<code>&lt;TR&gt;</code>	
<code>&lt;TD&gt;儲存格&lt;/TD&gt;</code>	
<code>&lt;TD&gt;儲存格&lt;/TD&gt;</code>	
<code>&lt;/TR&gt;</code>	
<code>&lt;/TABLE&gt;</code>	

也就是說，我們使用『建立表格 建立列 建立欄』的順序建立成一個完整的表格，先使用`<TABLE>...</TABLE>`建立表格的起始及結束，再使用`<TR>...</TR>`的方式建立表格之中要有多少列，最後，在每一列(`<TR>...</TR>`)之中使用`<TD>...</TD>`的方式建立這一行之中要有多少個儲存格。

另外有一些事項需要注意：

- 在不同列之中，可以設定不同數量的儲存格，但是這樣一來，就會有一些儲存格是突出的情況。
- 在『`<TD>儲存格之中的文字</TD>`』的設定中，文字並不會被`<P>`標籤的樣式影響，那是因為在文字的前後並未使用`<P>...</P>`標籤的原故，或者，可以寫成『`<TD><P>儲存格之中的文字</P></TD>`』。

## 第 2 節、表格的屬性

有一些屬性是之前學過的，可以拿來運用，但要注意放置的位置，例如：

- width 屬性置於<TABLE>中是指定表格寬度，在<TD>中則為儲存格寬度
- align 屬性放在<TABLE>中是表格的對齊方式，在<TR>中則是整列，<TD>則是指定該儲存格

border 屬性可以指定表格框線的粗細；屬性值是任意整數，0 表無框線表格

valign 屬性指定垂直對齊方式；屬性值有 top, middle, bottom

rowspan 屬性是合併縱向欄位，合併儲存格的功能

colspan 屬性是合併橫向欄位，合併儲存格的功能

我們來看看二個例子：

```
<TABLE border=2 width=80%>
  <CAPTION>表格範例</CAPTION>
  <TR><TD>第 1-1 區<TD>第 1-2 區<TD>第 1-3 區</TR>
  <TR><TD>第 2-1 區<TD>第 2-2 區<TD>第 2-3 區</TR>
  <TR><TD>第 3-1 區<TD>第 3-2 區<TD>第 3-3 區</TR>
</TABLE>
```

上面的例子如果加上合併儲存格的動作，修改如下：

```
<TABLE border=2 width=80%>
  <CAPTION>表格範例</CAPTION>
  <TR><TD COLSPAN=2>第 1-1 區          <TD>第 1-2 區 //橫向合併
  <TR><TD ROWSPAN=2>第 2-1 區<TD>第 2-2 區<TD>第 2-3 區 //縱向合併
  <TR>
          <TD>第 3-2 區<TD>第 3-3 區
</TABLE>
```

## 第 3 節、表格的使用技巧

1. 如果將<TD>...</TD>分成不同行來撰寫 HTML，會在顯示時多出一個空白，若有影響瀏覽時呈現的效果，可以將<TD>...</TD>寫在同一行，避免多出這個空白。
2. 平常建立表格時，可以省略</TD>這個標籤。
3. 我們在可以在<TD>...</TD>之中，再建立一個<TABLE>...</TABLE>的標籤，可以使得網頁的變化更豐富，但是，如果要在表格中再插入表格的話，就不能省略</TD>標籤。

4. HTML 編排的功能並不像 WORD 或其它文書軟體、排版軟體那麼豐富，所以，表格（無框線表格）在網頁的編排上佔了重要的功能。

## 第 4 節、區塊元件

區塊元件主要是將某部份的標籤、文字、圖片…的資料給打包、封裝成一個區塊，那麼我們就可以針對這個區塊設定背景、位置、大小…等不同的設定。

表格就是一個標準的區塊元件，每一個儲存格中都可以存放不同性質的資料內容，例如存放文字、圖片、還可以再放個表格。

另外，CSS 對於表格也有一些擴充的樣式，參考下表：

功能	屬性	範例
框線	Frame	Frame:above //上方
		Frame:below //下方
		Frame:hsides //上下
		Frame:lhs //左邊
		Frame:rhs //右邊
		Frame:vsides //左右
		Frame:border //四邊
		Frame:box //也是四邊
		Frame:void //無外框
框線色彩	BorderColor	BorderColor:"#ffc000" //設定框線色彩
		BorderColorLight:red //左、上方色彩
		BorderColorDark:blue //右、下方色彩

## 第 5 節、DIV

DIV 是一個容器標籤，在<DIV>…</DIV>之中可以包含任何其它的標籤，例如文字、圖片、表格等等，使用上和平常沒有二樣，但是，若我們在<DIV>…</DIV>中設定了 width 和 height 屬性，那就將它設定了範圍了，超出範圍的就不會出現了！而且，還可以將整個 DIV 設成絕對位置（後述），放置在任何位置。

## 第 6 節、SPAN

SPAN 和 DIV 標籤一樣是個容器標籤，不同的是 SPAN 是個行內 (In-Line) 標籤，可以將<SPAN>...</SPAN>放置在同一列之中，和 DIV 標籤不同的是，使用<DIV>...</DIV>會在前後自動分段，建立成不同的段落。

## 第 7 節、FIELDSET

<FIELDSET>...</FIELDSET>標籤會在區塊的外圍，加上矩形的邊框，若加上 Background-color 的屬性，可以使區塊的設定更為明顯！

在<FIELDSET>...</FIELDSET>標籤之中使用<LEGEND>...</LEGEND>標籤，可以在框線的左上方加上標題文字。

## 第 8 節、版面配置

以上所介紹的各種區塊元件都可以設定版面，包括邊界、邊框色彩等等。

功能	屬性	範例	
邊界	Margin	Margin-top:10	//上邊界為 10
		Margin-right:10	//右邊界為 10
		Margin-bottom:10	//下邊界為 10
		Margin-left:10	//左邊界為 10
		Margin:8 6 4 2	//分別設定邊界上、右、下、左為 8, 6, 4, 2，若設定不足四個，則以對邊為設定值

**注意：**這裡所說的邊界是指該元件（或容器）與上層的元件（或容器）之間的距離

邊框樣式	Border-Style	範例	
	Border-Style:solid		//實線
	Border-Style:double		//雙實線
	Border-Style:groove		//立體內凹
	Border-Style:ridge		//立體浮凸
	Border-Style:inset		//內凹
	Border-Style:outset		//外凸

甚至，還可以分別設定四邊使用不同的樣式：

功能	屬性	範例
		Border-Top-Style:solid Border-Right-Style:double Border-Bottom-Style:groove Border-Left-Style:ridge  當然，還可以這樣設定： Border-Style: 上 右 下 左
邊框色彩	Border-Color	Border-Color:red //將邊框設為紅色  還可以這樣設定： Border-Top-Color:green //上為綠色 Border-Right-Color:blue //右為藍色 Border-Bottom-Color:red //下為紅色 Border-Left-Color:yellow //左為黑色  也可以這樣設定： Border-Color: 上 右 下 左
邊框粗細	Border-Width	Border-Width:12 //將邊框設為 12  可以這樣設定： Border-Top-Width:12 Border-Right-Width:10 Border-Bottom-Width:8 Border-Left-Width:6  當然也可以這樣設定： Border-Width: 上 右 下 左
垂直對齊方式	Vertical-Align	Vertical-Align:top //靠上對齊 Vertical-Align:middle //垂直置中 Vertical-Align:bottom //靠下對齊



## 第 7 章、位置及顯示隱藏

### 第 1 節、絕對位置

利用 CSS 可以設定某個物件的位置，可以讓我們將物件重疊在一起，或者放置於特定的位置。通常設定絕對位置時，都會配合區塊元件運用，位置的起點是以容器（物件的上層容器）的左上角為原點，越往右越大、越往下越大。

功能	屬性	範例
設定位置	position	position:absolute //設為絕對位置 position:relative //相對位置
垂直位置	top	top:100 //垂直座標為 100 點
水平座標	left	left:300 //水平座標為 200 點
物件的層次	z-index	z-index:1 //數值越大越上層，若數值為負值，則在文字之後

程式範例

```
<HEAD>
<style>
.type1 {
    font-size:128;    font-family:標楷體;    color:#ccffcc;
    position:absolute; //指定為絕對位置
    top:100; left:80; //指定起點座標
}
.type2 {
    font-size:72;    font-family:標楷體;    color:#ffccaa;
    position:absolute; //指定絕對位置
    top:100; left:200; //指定起點座標
}
</style>
</HEAD>
<BODY>
<P class="type1">第一行文字</p>
```

```
<P class="type2">第二行文字</p>
</BODY>
```

## 第 2 節、顯示及隱藏

我們可以設定物件是否要顯示：

功能	屬性	範例	
顯示及隱藏	Visibility	visibility:visible	//顯示物件
		visibility:hidden	//隱藏物件
顯示及隱藏	Display	Display:""	//顯示物件
		Display:"none"	//隱藏物件

Display 和 Visibility 的顯示隱藏不同，Display 並不會保留位置，而 Visibility 屬性是會將物件隱藏，但空間仍然會空著，不會取消。

### 程式範例

底下範例中的二張圖片都不會顯示出來，但有些差別，一張圖片的空間會保留，另一張的空間不會保留，如果沒有註明或標示的話，不會知道這裡原本有一張圖片的。

```
<HEAD>
<style>
  .img_type1 { visibility:hidden;}
  img_type2 { display:"none";}
  p { font-size:24; font-family:標楷體; color:#aaaaaa; }
</style>
</HEAD>
<BODY>
  <P>底下有一張圖片！</p>
  <P><IMG src="我好喜歡你.jpg" class="img_type1"></p>
  <P>猜猜看上面這張圖片是什麼意思？</p>

  <P>底下也有一張圖片！</p>
  <P><IMG src="我好喜歡你.jpg" class="img_type2"></p>
  <P>但是它的空間沒有空下來！</p>
</BODY>
```

這是以圖片為例說明，實際運用時可以是任何物件。

## 第 3 節、顯示區域

Clip 可以設定一個矩形的顯示區域，只有在範圍內的資料才能顯示出來，有些類似前面所說的區塊元件，但是 Clip 的樣式只針對該標籤有效，不像區塊元件可以包含不同的標籤：

功能	屬性	範例
調整顯示區域	Clip:Rect( )	Clip:Rect(Top Right Bottom Left) //Top：左上角的垂直座標 //Right：右下角的水平座標 //Bottom：右下角的垂直座標 //Left：左上角的水平座標

例如，左上角的座標為(50,30)，而右下角的座標為(120,100)，則設定為：

```
Clip:Rect(30 120 100 50)
```

要注意的是，這裡所設定的座標是相對於物件上層的容器而言。如果不指定顯示的邊界，可以直接使用 auto 來取代其中座標。

程式範例

```
<HEAD>
<style>
  .img_type1 {
    position:absolute;
    clip:rect(100 586 245 0);
  }
</style>
</HEAD>
<BODY>
  <P><IMG src=最新的進化論.jpg class=img_type1></p>
  <P>圖片的上半部份沒有顯示出來！</p>
</BODY>
```

## 第 4 節、文字的顯示區域

使用 Overflow 屬性配合 width 及 height 來指定其寬度和高度，可以指定文字的內容超出顯示範圍部份時，應該如何處理，參閱如下：

功能	屬性	範例
文字的顯示區域	Overflow:visible	//全部顯示，預設值
	Overflow:hidden	//超出部份隱藏
	Overflow:scroll	//顯示右、下方捲軸
	Overflow:auto	//超出時才顯示捲軸

程式範例

```
<HEAD>
<style>
.type1 {
    position:absolute;
    top:100;    left:200;
    width:200;    height:100;
    overflow:auto;
}
</style>
</HEAD>
<BODY>
<P class=type1>
水調歌頭    蘇軾<br>
明月幾時有，把酒問晴天。不知天上宮闕，今夕是何年？<br>
我欲乘風歸去，又恐瓊樓玉宇，高處不勝寒。起舞弄清影，何似在人間？<br>
轉朱閣，紙綺戶，照無眠。不應有恨，何事長向別時圓？<br>
人有悲歡離合，月有陰晴圓缺，此事古難全。但願人長久，千里共嬋娟。<br>
</p>
</BODY>
```

## 第 8 章、其它標籤

### 第 1 節、書籤

#### 第 1 項、<A name=#書籤名稱> 建立書籤

書籤建立之後，使用超連結方式為<A href=#書籤名稱>，若在別的檔案中，則為<A href=URL#書籤名稱>

建立 TEST6.HTM 文件：

```
<HTML>
<HEAD>
  <TITLE>書籤</TITLE>
</HEAD>
<BODY>
  <UL>李白的詩
    <LI><A href=#poem1>靜夜思</A> //超連結 1
    <LI><A href=#poem2>玉階怨</A> //超連結 2
  </UL>
  <BR><BR><BR><BR><BR><BR><BR><BR><BR><BR> //故意設定的間距
  <HR>
  <A name=#poem1> //書籤 1
  <H2>靜夜思</H2>
  <P>床前明月光
  <BR>疑視地上霜
  <BR>舉頭望明月
  <BR>低頭思故鄉
  <BR><BR><BR><BR><BR><BR><BR><BR><BR><BR> //故意設定的間距
  <HR>
  <A name=#poem2> //書籤 2
  <H2>玉階怨</H2>
  <P>玉階生白露
  <BR>夜久侵白襪
  <BR>卻下水晶廉
  <BR>玲瓏望秋月
  <BR><BR><BR><BR><BR><BR><BR><BR><BR><BR> //故意設定的間距
</BODY>
</HTML>
```

## 第 2 節、特殊效果

### 第 1 項、<MARQUEE>...</MARQUEE> 跑馬燈

跑馬燈不只是可以使用在文字上，還可以使用在圖片上。中間所包含的文字可以不只一行，可以加上<BR>分行，也可以加上 ALIGN 屬性設定對齊方式。

behavior 屬性設定捲動效果；屬性值有 scroll, slide, alternate

scroll	//一般的捲動效果
slide	//碰到邊界後就不動了
alternate	//左右彈來彈去

direction 屬性設定捲動方向；屬性值有 left, right, up, down

width, height 屬性設定跑馬燈的寬度與高度；屬性值可以是像素或百分比

loop 屬性設定捲動的次數；屬性值可以是任意整數，不指定表連續

scrollamount 設定每次移動的間距；屬性值是任意整數

scrolldelay 設定多久時間移動一次(1/100 秒)；屬性值是任意整數

### 第 2 項、<EMBED src=" URL" > 插入多媒體影片

其實不只是影片，很多不同的多媒體檔案，你都可以試試看。

```
<P><EMBED src=MYGOD.MPE> //插入一個影片
```

### 第 3 項、<BGSOUND src=" URL" > 背景音樂

音樂也可以使用<EMBED>標籤方式在網頁上產生，和<BGSOUND>的差別在於前者會出現『控制面板』可以由訪客決定播放（載入網頁時會自動播放）、暫停、中止，而後者沒有控制面板，有些強迫推銷的意思，訪客一定得聽音樂。

其實按下『ESC』也可以停止音樂的播放，記得要在網頁載入結束後再按，否則網頁內容也會一起停止載入。

## 第 3 節、自動讀取另一份文件

<META HTTP-EQUIV="refresh" CONTENT="秒數;URL=文件的 URL" >

首先，別忘了要加上『URL=』的字串，後面才是輸入真正的網址。我自己老是忘了這一點。

在<HEAD>...</HEAD>之間加入上面一行，就可以讓文件在時間到達時，自動讀取另一份文件。最簡易的聊天室，其實就是將自動讀取的文件設定為同一份文件，就可以在固定時間更新為最新的資料內容。

```
<HTML>
<HEAD>
  <META HTTP-EQUIV="refresh" CONTENT="2;URL=http://www.kingbig.idv.tw">
</HEAD>
<BODY>
  <P>二秒後就會自動讀取 http://www.kingbig.idv.tw 的網頁</P>
</BODY>
</HTML>
```

## 第 4 節、在網頁中插入另外一個網頁

## 第 9 章、視窗框架

### 第 1 節、框架頁

要分割視窗，框架頁是最重要的。由於框架頁最主要的功能是在定義分割視窗，並沒有包含內文，因此，可以沒有<BODY>...</BODY>標籤。

設定框架的方式，和設定表格的方式有些類似，如下：

```
<HEAD> //必須放在<HEAD>區段中宣告
┌<FRAMESET> //定義框架頁起始分割
│   <FRAME> //框架 1
│   <FRAME> //框架 2
└</FRAMESET>
</HEAD>
```

#### 第 1 項、<FRAMESET>...</FRAMESET> 分割視窗

這組標籤必須放在<HEAD>...</HEAD>之中，還必須配合底下的屬性一起運用：

cols, rows 屬性設定分割為欄或列；屬性值可以是點數、百分比、星號

我們來看看實際的例子：

```
<HEAD>
  <TITLE>分割視窗</TITLE>
  <FRAMESET cols=*, 2*, *> //將視窗分割為左、中、右三個框架
    <FRAME> //左邊的框架，比例為 1
    <FRAME> //中間的框架，比例為 2
    <FRAME> //右邊的框架，比例為 1
  </FRAMESET>
</HEAD>
```

在<FRAMESET>中可以使用 border 屬性設定框架分割的框線粗細。

#### 第 2 項、<FRAME> 將 HTML 嵌入視窗

還需要配合以下的屬性一起運用才有作用。



src 屬性設定該框架的初始網頁；屬性值是網頁的 URL  
 name 屬性設定該框架的名稱；屬性值為自訂的名稱  
 scrolling 屬性設定是否顯示捲軸；屬性值有 auto, yes, no  
 noresize 屬性是設定是否鎖定分割視窗的大小；無屬性值  
 marginwidth 屬性是設定分割視窗的左右邊界；屬性值是任意整數  
 marginheight 屬性是設定分割視窗的上下邊界；屬性值是任意整數

試試看建立以下例子：

```
<HEAD>
  <TITLE>分割視窗</TITLE>
  <FRAMESET cols=*,3*> //分割為左右二個框架，比例為 1:3
    <FRAME src=test2.htm name=left> //命名為 left，顯示網頁為 test2.htm
    <FRAME src=test3.htm name=right> //命名為 right，顯示網頁為 test3.htm
  </FRAMESET>
</HEAD>
```

## 第 2 節、FRAME 的進階應用

### 第 1 項、巢狀結構

視窗分割之後，還可以再分割，只要將原本使用的 <FRAME> 取代為 <FRAMESET>... </FRAMESET> 標籤，就可以再使用 <FRAME> 標籤建立框架。

看看底下的例子：

```
<HTML>
<HEAD>
  <TITLE>分割視窗</TITLE>
  <FRAMESET cols=*,3*> //第一個<FRAMESET>
  |
  | <FRAME src=test2.htm name=left>
  | | <FRAMESET rows=25%, *> //第二個<FRAMESET>
  | | | <FRAME src=test3.htm name=right>
  | | | <FRAME src=test6.htm name=main>
  | | </FRAMESET>
  | </FRAMESET>
</HEAD>
</HTML>
```

### 第 2 項、target 屬性 設定目的視窗

這個屬性必須設定於超連結的標籤之中。使用方式如下：

```
<A href="URL" target="框架名稱">...</A>
```

這樣子就可以將超連結的內容顯示於指定的框架之中了。

### 第 3 項、target 屬性的再應用

target=\_blank

會開啟一個新的瀏覽器，將超連結顯示在新瀏覽器中。

target=\_self

將超連結顯示在目前的框架中。

target=\_top

將所有的框架取消，將超連結顯示於目前的瀏覽器中。

target=\_parent

回到上一層的<FRAMESET>...</FRAMESET>框架之中。這個功能多使用於某一個框架中，又有一個框架頁設定了該框架中的分割框架。

### 第 4 項、預設目標框架

```
<BASE target="框架名稱"> 重設該網頁的目的視窗
```

這是一個典型的應用方式。

網頁中若有多個超連結都要連結至某一框架中，則必須在每個超連結中加入 target 屬性，非常麻煩，因此，我們可以使用此標籤將所有的連結指向某一個框架。

```
<BODY>  
  <BASE target=MAIN>  
  <P><A href=test.htm>靜夜思</A>  
  <P><A href=http://www.kingbig.idv.tw>山賊老巢</A>  
</BODY>
```

## 第 10 章、雙向互動的表單

單純的顯示網頁，真的不夠看，能夠留下訪客的資料、做個問卷統計，才能讓網站真正的和使用者之間產生互動。

### 第 1 節、<FORM>…</FORM> 表單

使用表單，有二點需要特別注意：

1. 所有的表單內容都是放在這個標籤之中，包括說明文字、輸入欄位等等。
2. 網頁中可以有很多個表單，但不允許巢狀表單（不允許表單中包含表單）。

method 屬性是設定資料傳送的方式；屬性值有 get, post

action 屬性是設定表單資料的處理方式；屬性值通常是 cgi 的 URL

get、post 是二種不同的傳送方式，在以後的 PHP（或其它 cgi 程式）課程中會比較詳細介紹它的差別的，目前我們都以 post 方式為例。

一般而言，表單的資料都是交由 CGI 程式處理，或者也可以直接利用 EMAIL 的方式寄到信箱中。

```
<FORM method=post action="URL" >  
<FORM method=post action=webmaster@kingbig.idv.tw enctype="Text/Plain">
```

如果使用電子郵件的方式傳送表單資料，記得要加上『enctype="Text/Plain"』屬性，才能正確解譯其中的文字，否則，我們收到的資料內容，會以&號隔開，而且，輸入的中文字全部都會變成亂碼！

### 第 2 節、<INPUT> 表單中的輸入欄位

在表單中的欄位，幾乎都是由<INPUT>標籤所建立的，其共通格式為：

```
<INPUT type="輸入欄位的種類" name="欄位名稱" value="初始值">
```

輸入欄位的種類有以下幾種：

## 第 1 項、TEXT 單行文字框

另外可以加上 SIZE=20 設定顯示長度為 20 字元長，而 MAXLENGTH=10 來設定最大輸入字元數為 10 個字元。

```
<INPUT type=TEXT name=NAME size=20 maxlength=10>
```

## 第 2 項、PASSWORD 密碼欄

其中的字元會以星號取代，傳回後還是原本的內容。

```
<INPUT type=PASSWORD name=PASSWORD maxlength=8>
```

## 第 3 項、SUBMIT 送出鈕

按下此按鈕後，將資料經由 method 指定的方式送到 action 指定的 cgi 程式或電子郵件。

```
<INPUT type=SUBMIT value=送出>
```

## 第 4 項、RESET 重設鈕

按下後會清除表單中的所有資料。

```
<INPUT type=RESET value=清除內容>
```

看看這個比較完整的例子：

```
<HTML>
<HEAD>
  <TITLE>表單</TITLE>
</HEAD>
<BODY>
  <FORM method=post action=mailto:master@kingbig.idv.tw enctype="Text/Plain">
    <P>姓名:<INPUT type=TEXT name=NAME> //一般文字欄位
    <P>密碼:<INPUT type=PASSWORD name=PWD maxlength=8> //密碼欄
    <P><INPUT type=SUBMIT value=送出資料> //送出資料
    <INPUT type=RESET value=清除資料> //清除資料
  </FORM>
</BODY>
</HTML>
```

## 第 3 節、更多的輸入欄位

### 第 1 項、RADIO 單選鈕

將多個 RADIO 的 NAME 設為相同即可

```
<INPUT type=RADIO name=SEX value=MALE>男  
<INPUT type=RADIO name=SEX value=FEMALE>女
```

### 第 2 項、CHECKBOX 多選欄

將多個 CHECKBOX 設為同一個 NAME 即為一組

```
<INPUT type=CHECKBOX name=LE value=MOVIE>電影  
<INPUT type=CHECKBOX name=LE value=TV>電視  
<INPUT type=CHECKBOX name=LE value=BOOK>書
```

### 第 3 項、select 列表 下拉清單

以下的方式將會直接傳回該項目的內容

```
<SELECT name=language>  
  <OPTION>ASSEMBLY  
  <OPTION>BASIC  
  <OPTION>C/C++  
</SELECT>
```

另外，還可以設定為：

<OPTION value=1 SELECTED>則傳回的是 VALUE 值 1，且該項目為預設值

<SELECT name=language size=3>則可以設定為三列高的方框

<SELECT name=language MULTIPLE>就可以多選

### 第 4 項、TEXTAREA 多列文字欄

```
<TEXTAREA name=MTEXT rows=5 cols=30>  
</TEXTAREA>
```

再來看看一個更完整的例子：

```
<BODY>  
<FORM method=post action=mailto:master@kingbig.idv.tw enctype="Text/Plain">  
  <P>姓名:<INPUT type=TEXT name=NAME>
```

```
<P>密碼:<INPUT type=PASSWORD name=PWD maxlength=8>
<P>性別:
  <BR><INPUT type=RADIO name=SEX value=MALE>男    //單選
  <BR><INPUT type=RADIO name=SEX value=FEMALE>女
<P>你的休閒活動有:
  <BR><INPUT type=CHECKBOX name=LE value=MOVIE>電影    //多選
  <BR><INPUT type=CHECKBOX name=LE value=TV>電視
  <BR><INPUT type=CHECKBOX name=LE value=BOOK>書
<P>曾經學過的程式語言有:
<BR>
<SELECT name=language>    //下拉清單
  <OPTION>ASSEMBLY
  <OPTION>BASIC
  <OPTION>C/C++
</SELECT>
<P>備註:
<BR>
<TEXTAREA name=MTEXT rows=5 cols=30>    //文字方塊
</TEXTAREA>
<P>
<INPUT type=SUBMIT value=送出資料>
<INPUT type=RESET value=清除資料>
</FORM>
</BODY>
```

# 第 11 章、樣式檔的連結

## 第 1 節、樣式檔的建立及連結

我們可以將所有網頁所需要用到的樣式建立成獨立的樣式檔，然後在網頁中使用連結的方式將樣式檔加入，那麼將來只需要修改樣式檔，則所有的網頁呈現出來的樣式都會改變，非常的方便。

使用記事本將樣式建立後，儲存時記得將副檔名改為『.CSS』即可。例如底下是樣式檔案的內容，不需要加上<STYLE>...</STYLE>的宣告標籤：

```
H1 {
    font-size:36; font-family:華康新儷粗黑,華康儷粗黑,標楷體;
    color:#8888ff; font-weight:normal;
}
H2 {
    font-size:26; font-family:華康新儷粗黑,華康儷粗黑,標楷體;
    color:#8888ff; font-weight:normal;
}
P {
    font-size:13; color=#000000;
}
.type1 {
    color:red; font-size:24
}
```

接著在<HEAD>...</HEAD>中加入連結樣式檔的宣告：

```
<HEAD>
    <Link Rel=StyleSheet Href="樣式檔案的位址及名稱">
</HEAD>
```

## 第 2 節、局部宣告

我們設定的樣式是作用在整個標籤的，如果希望能夠設定該標籤中的局部樣式，可以將樣式宣告如下：

```
<style>
    H1 {color:red; font-size:24 }
    H1 em {color:blue }
</style>
```

那麼在實際的運用時，可以使用如下的方式：

```
<H1>原本<em>的</em>樣式</H1>
```

我個人認為，<SPAN>的標籤加上自訂樣式，更好用、更多樣！

## 第 3 節、宣告樣式注意事項

1. 整頁宣告的<STYLE>...</STYLE>標籤雖然也可以放在<BODY>中，但置於<HEAD>會比較保險
2. 由於 Netscape 有另一套樣式表 JSS(JavaScript Style Sheet)，因此最好將<STYLE>寫成<Style type="Text/css">來指定使用 CSS 的用法
3. 在樣式表中設定樣式可能會加上『-』分隔符號

## 第 4 節、物件的命名

只要是物件，就有個名稱，例如以下的程式，我們在標籤中加入了『ID=TEXT1』的敘述，就可以在<HEAD>...</HEAD>建立 VBScript，將內文做些改變了。

程式範例

```
<HEAD>
<script language="vbs">
  sub chbig()
    text1.style.fontSize=36
    text1.style.color="red"
  end sub
  sub chsmall()
    text1.style.fontSize=12
    text1.style.color="blue"
  end sub
</script>
</HEAD>
<BODY>
  <P id="text1" onmouseover="chbig()" onmouseout="chsmall()">
    會改變大小的文字
  </P>
</BODY>
```



## 第 12 章、JavaScript

### 第 1 節、Java 與 JavaScript

#### 第 1 項、Java

是一種新一代的程式語言，類似於 C++ 的物件導向語言，和開發傳統程式一樣，程式撰寫之後都需要先經過編譯器（Compiler）的處理。

將 Java 使用於網頁中，只是它的一部份功能，多是為了追求網頁『炫』的效果。

#### 第 2 項、JavaScript

描述語言，最大的特色是可以讓一般的使用者很快的上手使用，它是一般的文字檔案，不需要經過編譯的過程，原始碼是直接置入網頁中的，只需要瀏覽器就可以執行。

執行效率較 Java 差些，且僅使用物件的基礎觀念。

### 第 2 節、JavaScript 語言結構

#### 第 1 項、先來看看簡單的 JavaScript 範例

只用一行來產生 JavaScript 事件：

```
<A href="http://www.kingbig.idv.tw" onclick="alert('這將連結到山賊老巢');">加上 JavaScript 的超連結</a>
```

使用<JavaScript>…</JavaScript>標籤來產生一個完整的程式：

```
<SCRIPT LANGUAGE="JavaScript">
document.write("大家好");
</SCRIPT>
```

#### 第 2 項、結構

以剛才程式的『document.write("大家好");』而言，我們稱為敘述。

其中的『document』是一個物件，『write』是物件中的方法或當成函數，『(“大家好”)』是方法或函數的參數，『;』是起源於C語言的習慣。

在<JavaScript>...</JavaScript>標籤之中，有可能會出現敘述、函數、變數、運算式等等。

### 第 3 項、函數

剛才使用的 write 就是一種內建函數。

自訂函數的方式為：

```
function add(a,b) {  
    var result=a+b;  
    return result;  
}
```

### 第 4 項、變數

剛才的 add(a,b)之中的 a,b 就是變數，主要是要存放某些資料。

我們可以使用 var a 來定義變數，或是直接使用，a 就成為變數了。

### 第 5 項、陣列

我們可以使用 ab=new Array(10)的方式來定義 ab 陣列，請注意大小寫。

建立 js31.htm 文件，並輸入以下：

```
<SCRIPT LANGUAGE="JavaScript">  
ab=new Array(10);  
ab[0]=3+4;  
ab[2]=4+5;  
document.write("<br>",ab[0],"<br>");  
document.write(ab[2]);  
</SCRIPT>
```

### 第 6 項、運算式

result=a+b 就是運算式的一種。

## 第 7 項、物件、屬性、方法

物件本身可以分為二種：存放資料的屬性、處理資料的方法

## 第 8 項、註解

//為單行註解

/\*.....\*/為多行註解，在這之中的文字、敘述都不會處理

## 第 9 項、特殊字元

在 JavaScript 中要使用一些特殊字元，要用以下的方式：

字元	顯示
\	“
\\	\
%	餘數；7 % 4
++	遞增；a ++
--	遞減；b --
==	比較前後是否相等，a==b 時為 true

## 第 3 節、JavaScript 與表格的實作練習

當 JavaScript 使用的數量較多之後，要注意一些事項，避免發生錯誤而使得網頁顯示不正確。

1. 凡是 JavaScript 定義宣告相關的部份（例如函數），撰寫於<HEAD>區，使這部份的內容優先執行，才不會呼叫一個未定義的函數
2. 變數及函數的大小寫請注意，因為大小寫在這裡是不相同的

請建立 JS32.HTM 檔案，並輸入以下內容：

```
<HTML>
<HEAD>
<TITLE>JavaScript 加表格</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function printrow(name,age,birthday){
```

```

document.write("<TR><TD>",name,"</TD><TD>",age,"</TD><TD>",birthday,"</TD></T
R><BR>");
}
</SCRIPT>
</HEAD>
<BODY>
<TABLE border=2>
<SCRIPT LANGUAGE="JavaScript">
    printrow("張三",27,"7 月 17 日")
    printrow("李四",28,"5 月 12 日")
    printrow("王五",29,"2 月 23 日")
</SCRIPT>
</TABLE>
</BODY>
</HTML>

```

## 第 4 節、條件式與迴圈

### 第 1 項、條件式

條件式有三種不同的表示法：

#### 1. 一行敘述

```
if (a==1) document.write("目前 A=1");
```

#### 2. 多行敘述

```
if (a==1) {
document.write("目前 A=1");
a=0;
}
```

#### 3. 若條件式不成立時

```
if (a==1) {
    document.write("目前 A=1");
    a=0;
} else {
    document.write("目前 A<>1");
    a=0;
}
```

#### 4. 更簡便的寫法

```
(a==1) ? "目前 A=1":"目前 A<>1";
```

## 第 2 項、迴圈

for 迴圈

```
for(初始值; 條件式; 步進值){}
```

用一個簡單的例子（JS35.HTM）來介紹迴圈：

```
for (l=1; l<10; l++) {
    document.write("這是第 ", l, " 行<BR>");
}
```

while 迴圈

while(條件){}，當條件成立時就繼續執行。

用一個例子（JS36.HTM）來介紹：

```
N=0;
while(n<10) {
    document.write("這是第 ", n, " 行<BR>");
    n=n+1
}
```

## 第 3 項、break 中斷

我們在執行迴圈時，要小心造成無窮迴圈！我們可以使用 break 敘述來中斷迴圈的執行，繼續執行其它的敘述。

## 第 4 項、事件

亦即發生了某些動作，例如：按下按鈕、游標移到某個元件上…都是一個事件。

事件	說明
onAbort	放棄影像載入時
onBlur	物件失去焦點時
onChange	當文字欄位被使用者改變時
onClick	按下項目時
onError	當文件或影像不能正確載入時
onFocus	當項目取得焦點時
onLoad	當文件或影像完成載入時
onMouseOver	滑鼠游標移入項目時
onMouseOut	滑鼠游標由項目上移開時
onSelect	在文字欄位上選取文字時

onSubmit	遞送表單時
onUnload	離開文件時

之前我們就實際運用 onClick 的事件了。

## 第 5 節、date 物件

使用『today=new Date()』可以設定變數 today 中儲存目前的日期、時間。而 date 物件的方法如下表：

方法	說明
getDate()	取得『日』
getDay()	取得『星期』
getHours()	取得『時』
getMinutes()	取得『分』
getMonth()	取得『月』
getSeconds()	取得『秒』
getTime()	以毫秒 ( $10^{-3}$ ) 為單位，自 1970 年 0 秒算起
getYear()	取得『年』

以下我們來看看實際的例子：

範例 JS43.HTM：

```
<SCRIPT LANGUAGE="JavaScript">
  today=new Date();
  document.write("今天的日期是 ", today.getYear(), " 年 ", today.getMonth()," 月 ",
today.getDate(),"日 ");

  if (today.getDay()==0) document.write("星期日");
  if (today.getDay()==1) document.write("星期一");
  if (today.getDay()==2) document.write("星期二");
  if (today.getDay()==3) document.write("星期三");
  if (today.getDay()==4) document.write("星期四");
  if (today.getDay()==5) document.write("星期五");
  if (today.getDay()==6) document.write("星期六");

  document.write("<br>")
  document.write("現在的 시간은 ", today.getHours(), " 時 ", today.getMinutes()," 分
", today.getSeconds(),"秒<br>");
</SCRIPT>
```

## 第 6 節、navigator 物件

主要是獲得瀏覽器版本的資訊。

屬性	說明
appCodeName	瀏覽器代號，通常是 Mozilla
appName	瀏覽器名稱
appVersion	瀏覽器版本
userAgent	使用者介面名稱

底下有個簡單的例子 (JS44.HTM)：

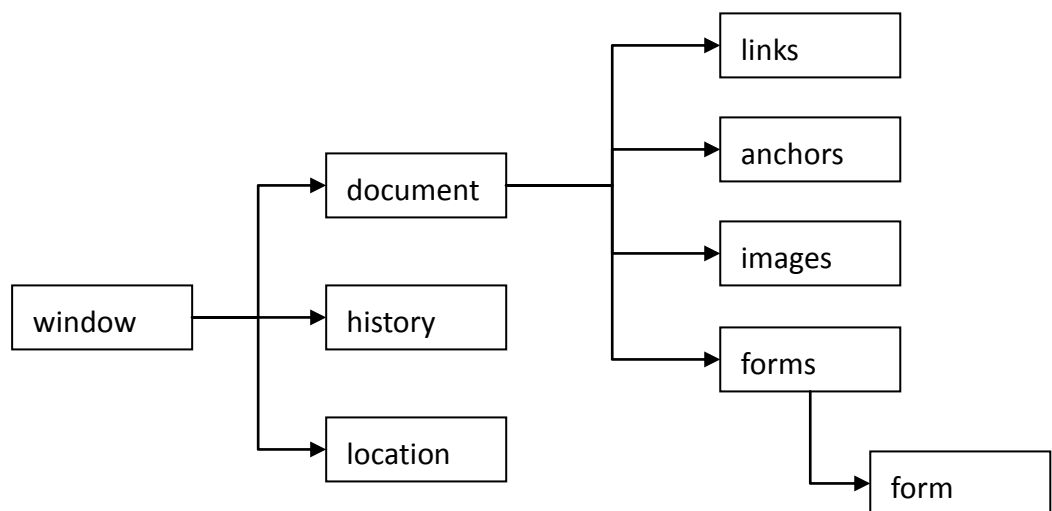
```
<SCRIPT LANGUAGE="JavaScript">
  document.write("瀏覽器代號：",navigator.appCodeName, "<br>");
  document.write("瀏覽器名稱：",navigator.appName, "<br>");
  document.write("瀏覽器版本：",navigator.appVersion, "<br>");
  document.write("使用者介面：",navigator.userAgent, "<br>");
</SCRIPT>
```

## 第 7 節、window 物件

使用物件階層架構的方式來處理網頁是 JavaScript 的重點。

在 JavaScript 物件階層架構中的最上層，是瀏覽器本身，稱為 window 物件，而 window 物件的屬性，也可以是物件，還可以再有下層的屬性。

可以參考下圖，第二層的 document 本身是 window 的屬性，也可以是物件，還包含自己的屬性：



window 還有個 status 屬性，最常被拿來運用，我們來看看 JS45.HTM 的例子：

```
<SCRIPT LANGUAGE="JavaScript">
    window.status="你使用的瀏覽器為" + navigator.appName + navigator.appVersion;
</SCRIPT>
```

這樣就可以將狀態列上的文字換成你自己的訊息了！

window 的方法

方法	說明
alert()	警示框
prompt()	顯示文字輸入框
confirm()	顯示確認框
setTimeout()	設定計時器
clearTimeout()	清除 Timeout 計時器
close()	關閉視窗
open()	開啟新視窗
blur()	將視窗移至背景後

我們來看看 JS46.HTM 中使用了文字輸入框、警示框、確認框：

```
<BODY>
<SCRIPT LANGUAGE="JavaScript">
    document.write("歡迎 ",window.prompt('請輸入你的大名:', '你的大名'), " 的光顧！");
<P>");
</SCRIPT>

<A href="http://www.kingbig.idv.tw" onclick="window.alert('這將連結到山賊老巢');">警示框</a><P>
<A href="http://www.kingbig.idv.tw" onClick="window.confirm('這將連結到山賊老巢');">確認框</a><P>
</BODY>
```



## 第 13 章、HTML5 新增的標籤

### 第 1 節、基本介紹

從早期的 html，其實就應該要注意標籤本身所具備的意義。例如 <p> 與 <h3>，本來就具有不同的意義。

網頁結構相關：

標籤	說明
<section>	文件裡的分項，差不多就等於書籍或論文的小節，網頁裡可能也會區分為好幾個分項
<nav>	只有主要導覽區塊才適用 nav 元素
<article>	通常是一篇文章，例如一篇新聞、一篇部落格的文章
<aside>	可能是和網頁周圍不同的分項，例如廣告，可以用 <aside>
<hgroup>	將一組 h1~h6 包含為群組的標籤
<header>	一般重要的 logo、搜尋表單都應該放在這個標籤裡
<footer>	大多數都放在網頁最底端的網頁註解

語義相關的：

標籤	說明
<time>	使用 24 小時制的時間
<mark>	強調網頁裡的文字、或是加上標記

網頁表單相關的：

標籤	說明
<input>	新增很多個 subtypes
<datalist>	輸入資料時可以產生一個下拉清單
<meter>	以水平長條的圖形顯示數值
<progress>	以水平長條的方式顯示目前的進度
<command>	建立按鈕(目前所有瀏覽器都不支援)
<output>	專門用來輸出結果的標籤

影音相關的：

標籤	說明
----	----

<audio>	聲音
<video>	影片
<source>	指定前二個標籤的來源位置
<embed>	指定要在網頁裡嵌入的外部檔案
<canvas>	畫布

## 第 2 節、網頁結構

對於網頁上的排版方式，最早期我們是使用 table 方式，後來有了 div(division)，被大量運用在網頁的排版，div 標籤再加上一些 css 的設定，真的很方便，而且可以有很多很多不同的排版結果。不過，當我們看到 div 標籤時，沒有辦法很容易就明白其中所包含的內容是什麼，我們還需要再判斷其中所包含的項目，才知道它的用處。

### 第 1 項、舊的網頁結構

以下的網頁範例，是舊的網頁結構裡很常使用的方式，在<body>...</body>的標籤裡，用一個<div id=main>...</div>做為主要的容器，其中再使用三個（或數個）div 標籤，分別設定為 header、content、footer 區塊，再放入適當的內容。

```
<style>
  #main {
    width: 90%;
    margin: auto;
  }

  div.header {
    background: #aaaaff;
    font-family: '標楷體';
    text-align: center;
  }
  div.header h1 {
    font-size: 36px;
    color: #fff;
  }
  div.header h2 {
    font-size: 20px;
  }

  div.content {
```

```

        min-height: 300px;
    }

    div.footer {
        background: #ccf;
        height: 80px;
        text-align: center;
    }
</style>

<h2>舊的文件架構</h2>

<div id=main>
    <div class="header">
        <h1>黃石國家公園</h1>
        <h2>黃石國家公園 ( 簡稱黃石公園 , 英語 : Yellowstone National Park )
是美國第一個國家公園。</h2>
        <p>from http://zh.wikipedia.org/wiki/黃石國家公園</p>
    </div>
    <div class="content">
        <p>主要位於懷俄明州，部分位於蒙大拿州和愛達荷州，於 1872 年 3 月 1
日美國總統尤利西斯·辛普森·格蘭特簽署國會通過的法案後建立，也是世界上第一個國家公
園。</p>
    </div>
    <div class="footer">
        <p>本頁面最後修訂於 <time>2014-05-13</time>
    </div>
</div>

```

## 第 2 項、HTML5 的基本網頁架構

在新的 HTML5 所提供的標籤裡，我們修改了一些網頁標籤，如下：

1. 將原本的<div id=main>改為<article>標籤。
2. 將原本的<div class="header">改為<header>標籤。
3. 將原本的<div class="footer">改為<footer>標籤，同時移到</article>標籤下方。
4. 相對的 CSS 也做了一些設定。

```

<style>
    article {
        width: 90%;
        margin: auto;
    }

    header {

```

```
        background: #aaaaff;
        font-family: '標楷體';
        text-align: center;
    }
    header h1 {
        font-size: 36px;
        color: #fff;
    }
    header h2 {
        font-size: 20px;
    }

    div.content {
        min-height: 300px;
    }

    footer {
        width: 90%;
        margin: auto;
        background: #ccf;
        height: 80px;
        text-align: center;
    }
</style>

<h2>HTML5 的文件架構</h2>

<article>
    <header>
        <h1>黃石國家公園</h1>
        <h2>黃石國家公園 ( 簡稱黃石公園 , 英語 : Yellowstone National Park )
是美國第一個國家公園。</h2>
        <p>from http://zh.wikipedia.org/wiki/黃石國家公園</p>
    </header>
    <div class="content">
        <p>主要位於懷俄明州，部分位於蒙大拿州和愛達荷州，於 1872 年 3 月 1
日美國總統尤利西斯·辛普森·格蘭特簽署國會通過的法案後建立，也是世界上第一個國家公
園。
    </div>
</article>
<footer>
    <p>本頁面最後修訂於 <time>2014-05-13</time>
</footer>
```

### 第 3 項、加上 hgroup

以<header>...</header>區塊裡為例，有<h1>與<h2>標籤是連續出現的，我們可以用<hgroup>...</hgroup>標籤將他們包含在裡頭。

```
<header>
  <hgroup>
    <h1>黃石國家公園</h1>
    <h2>黃石國家公園 ( 簡稱黃石公園，英語：Yellowstone National
Park ) 是美國第一個國家公園。</h2>
  </hgroup>
  <p>from http://zh.wikipedia.org/wiki/黃石國家公園</p>
</header>
```

### 第 4 項、加上 figure 標籤

在使用 figure 標籤之前，我們再在網頁裡插入一張圖片、加上圖片說明、再將圖片和文字靠左（或靠右）對齊，大多的做法如下：

```
<style>
  div.figure {
    float: left;
    margin: 0px 4px;
    background: #fcc;
    width: 150px;
  }
  div.figure p {
    max-width: 150px;
  }
</style>

<div class="content">
  <div class=figure>
    <img
      src=images/YellowstonefallJUN05.jpg
      width=150px>
    <p>黃石公園內的大峽谷
  </div>
  <p>主要位於懷俄明州，部分位於蒙大拿州和愛達荷州，於 1872 年 3 月 1
日美國總統尤利西斯·辛普森·格蘭特簽署國會通過的法案後建立，也是世界上第一個國家公
園。</p>
  </div>
```

現在將<div class=figure>...</div>換成<figure>...</figure>標籤，再將圖片說明的文字放在<figcaption>...</figcaption>標籤裡，結果可能更好一些。

```

<style>
  figure {
    float: left;
    margin: 0px 4px;
    background: #fcc;
    width: 150px;
  }
</style>

<div class="content">
  <figure>
    <img
      src=images/YellowstonefallJUN05.jpg
      width=150px>
    <figcaption>黃石公園內的大峽谷</figcaption>
  </figure>
  <p>主要位於懷俄明州，部分位於蒙大拿州和愛達荷州，於 1872 年 3 月 1
日美國總統尤利西斯·辛普森·格蘭特簽署國會通過的法案後建立，也是世界上第一個國家公
園。</p>
</div>

```

## 第 5 項、使用 aside 標籤

當我們想要在文章的旁邊加上廣告、或是某些重要訊息，可以使用 `<aside>...</aside>` 標籤，參考如下：

```

<style>
  aside {
    float: right;
    border: 1px #77f solid;
    font-size: 13px;
    padding: 4px 10px;
    margin: 4px;
  }
</style>

<div class="content">
  <figure>
    <img
      src=images/YellowstonefallJUN05.jpg
      width=150px>
    <figcaption>黃石公園內的大峽谷</figcaption>
  </figure>
  <aside>
    <p>美國懷俄明州帕克郡、提頓郡
    <br>蒙大拿州加拉廷郡、帕克郡
  </aside>
</div>

```

```

<br>愛達荷州弗里蒙特郡
<br>坐標 44°36'N 110°30'W
<br>面積 2,219,790.71 英畝 ( 8,983 平方公里 )
<br>建立時間 1872 年 3 月 1 日
<br>主管團體 美國國家公園管理局
</aside>
<p>主要位於懷俄明州，部分位於蒙大拿州和愛達荷州，於 1872 年 3 月 1
日美國總統尤利西斯·辛普森·格蘭特簽署國會通過的法案後建立，也是世界上第一個國家公
園。</p>
</div>

```

## 第 6 項、nav 標籤

一些超連結的集合，常常會放在<nav>...</nav>標籤裡，它可能是橫向的放在 banner 下方、或是網頁右上方，也可能會以縱向區塊方式放在網頁左邊或右邊，若設定縱向區塊的方式，就有些像 aside 標籤的作用。

## 第 7 項、section 標籤

我們可以用<section>...</section>標籤將一個標題標籤、以及幾個段落文字包起來，變成一個 section，例如：

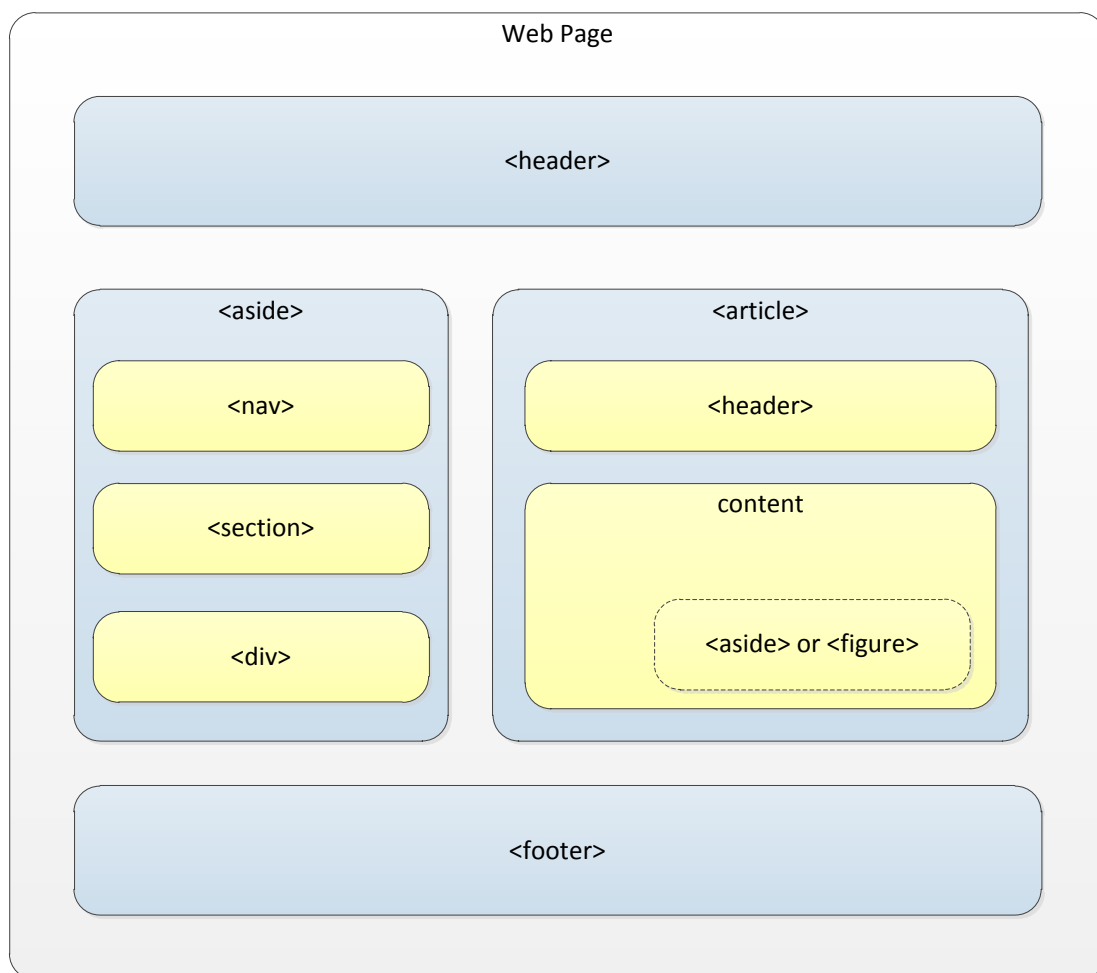
```

<section>
  <h3>標題三</h3>
  <p>第一段文字
  <p>第二段文字
</section>

```

## 第 8 項、HTML5 網頁架構小結

以下的網頁架構是很粗略的介紹，不同的藍色或黃色區塊裡，其實還可以包含其他的標籤。



參考 Oreilly.HTML5: The Missing Manual, P66 內容繪製

## 第 3 節、語義相關的新標籤

### 第 1 項、time 標籤

我們可以將日期時間，加在 time 標籤裡，參考如下：

```
<br>建立時間 <time datetime="1872-03-01">1872 年 3 月 1 日</time>
```

### 第 2 項、mark 標籤

被 mark 標籤框住的文字，預設會被標示為黃底的背景色，就像是將它畫上重點一樣。

```
<br>主管團體 <mark>美國國家公園管理局</mark>
```



## 第 4 節、表單相關的標籤

### 第 1 項、input 的 placeholder 屬性

```
<p>姓名 : <input type=text placeholder="請輸入您的大名" name=u01>
```

### 第 2 項、input 的 focus 狀態

設定 input:focus 的樣式，如下：

```
<style>
  input:focus {
    background: #ffc;
  }
</style>
```

### 第 3 項、Input 的 autofocus 屬性

```
<p>姓名 : <input type=text placeholder="請輸入您的大名" name=u01 autofocus>
```

### 第 4 項、Input 的 required 屬性

```
<form>
  <p>姓名 :
    <input type=text
      placeholder="請輸入您的大名"
      name=u01
      autofocus
      required
    >
  <p><input type=submit value=send>
</form>
```

## 第 5 項、使用正規運算式驗證資料

```
<p>身份證字號 :
  <input type=text name=u02
    placeholder="請輸入您的身份證字號"
    pattern="[A-Za-z][0-9]{9}"
  >
```

## 第 6 項、Output 標籤

如果我們使用 javascript 將一些結果輸出，通常會使用 div、span 之類的標籤定義一個 id，然後使用 id 來存取或修改它的值，而在 HTML5，專門就用 output 標籤負責輸出的動作，參考以下例子：

```
<form oninput="result.value=parseInt(a01.value)+parseInt(a02.value)">
  <input type="text" id="a01" value="50" style="width: 80px; ">
  +
  <input type="text" id="a02" value="50" style="width: 80px; ">
  =
  <output name="result" for="a01 a02"></output>
</form>
```

## 第 7 項、Input 的 number 型態

```
<form oninput="result.value=parseInt(a01.value)+parseInt(a02.value)">
  <input type="number" id="a01" value="50" style="width: 80px; ">
  +
  <input type="number" id="a02" value="50" style="width: 80px; ">
  =
  <output name="result" for="a01 a02"></output>
</form>
```

## 第 8 項、Input 的 range 型態

```
<form oninput="result.value=parseInt(a01.value)+parseInt(a02.value)">
  <input type="range" id="a01" value="0" min="0" max="10">
  +
  <input type="range" id="a02" value="0" min="0" max="10">
```

```
=  
<output name="result" for="a01 a02"></output>  
</form>
```

## 第 9 項、Input 的 email 型態

當我們設定『type=email』時，這個欄位會自動判斷是否符合 email 的格式，具備前述驗證資料的功能，同時，在行動裝置上，它的小鍵盤還有顯示有『@』符號的鍵盤配置。

```
<p>email : <input type=email placeholder="請輸入您的 email" name=u03>
```

## 第 10 項、Input 的 url 型態

```
<p>web : <input type=url placeholder="請輸入您的 web" name=u04>
```

## 第 11 項、Input 的 search 型態

在行動裝置上使用這個欄位，最右方會顯示一個『x』符號，方便我們清除已經輸入的關鍵字。

```
<p>站內搜尋 : <input type=search placeholder="請輸入關鍵字" name=u05>
```

## 第 12 項、Input 的 tel 型態

在行動裝置上，它的小鍵盤會顯示為九宮格的數字鍵盤（依不同裝置而有所差異）。

```
<p>聯絡電話 : <input type=tel placeholder="請輸入您的聯絡電話" name=u06>
```

## 第 13 項、Input 的 date 型態

對於支援這個功能的瀏覽器而言，它會自動顯示為下拉的日期清單，方便使用者可以選擇日期。

```
<p>生日 : <input type=date placeholder="請輸入您的生日" name=u07>
```

除了 date 型態之外，還有 time、datetimelocal、datetime、month、week 這幾個型態可以使用。

## 第 14 項、Input 的 color 型態

瀏覽器會自動顯示為調色盤，方便使用者選擇色彩。

```
<p>色彩： <input type=color placeholder="請挑選您最喜歡的顏色" name=u08>
```

## 第 15 項、Progress 標籤

主要用來顯示任務的進度，預設是由 0 到 1 之間的數值，有 max 屬性和 value 屬性可以設定。

```
<p>下載進度： <progress value=0.4></progress>
```

## 第 16 項、Meter 標籤

Meter 標籤適合用來標示已知範圍的數值，例如以下範例是顯示某科的分數，除了指定 min 與 max 的屬性，還可以再指定 low 與 high 的屬性表示過低或非常好的數值，在 low 與 high 之間的數值、low 以下或者是 high 以上的數值，它們的顏色是不同的。

```
<p>分數： <meter value=80 min=0 max=100 low=40 high=90></meter>
```

# 第 5 節、影音

## 第 1 項、HTML5 的音樂

Audio 標籤是 HTML5 新增的標籤，專門用來播放音樂的，它有 src、controls、preload、autoplay、loop 幾個屬性可以設定。

Src 屬性，通常會省略，而使用<source... />標籤來指定來源，使用 source 標籤可以支援多種不同的音訊格式，瀏覽器會由第 1 項開始檢查是否支援，若有支援的話就停止往下檢查。

Controls 屬性則可以顯示播放的控制介面。

Preload 屬性，則是指定載入網頁時，同時就先載入音訊檔案。

若指定 autoplay 屬性，載入網頁時即自動播放。同時，也可以忽略 preload 屬性。

Loop 屬性則可以自動重新播放。

完整網頁請參閱 t04.htm。

```
<h2>html5 的音訊播放</h2>
<audio controls="controls" autoplay=autoplay>
  <source src="Vladimir_Sterzer_-_For_You.ogg" type="audio/ogg" />
  <source src="Vladimir_Sterzer_-_For_You.mp3" type="audio/mpeg" />
  您的瀏覽器不支援 html5 的 audio 標籤。
</audio>

<p>音樂來源： http://www.jamendo.com/en/track/592183/for-you
<p>詳細的 audio 標籤屬性，請參考以下連結：
<br><a href="http://www.w3schools.com/html5/html5_audio.asp target=_blank">英文
</a>
<br><a href="http://www.w3school.com.cn/html5/html_5_audio.asp target=_blank">
簡體中文</a>
```

## 第 2 項、直接使用 html5 的 video 播放影片

Video 的標籤和 audio 標籤所擁有的屬性和作用相同。

完整網頁請參閱 t02.htm。

```
<h2>html5 的影音播放</h2>

<video controls="controls">
  <source src="myvideo.ogv" type="video/ogg">
  <source src="myvideo.mp4" type="video/mp4">
  您的瀏覽器不支援 html5 的標籤。
</video>

<p>在 html5 可以直接使用標籤播放影片，而且，可以指定多種影片格式，瀏覽器會
由上往下，播放第一個支援的影片格式。
<p>詳細的 video 標籤屬性，請參考以下連結：
<br><a href="http://www.w3school.com.cn/html5/html_5_video.asp target=_blank">
簡體中文</a>
```

### 第 3 項、加上 JavaScript 控制 video 物件

若使用 javascript 控制 video 物件，除了可以用 play()與 pause()方法控制影片播放與暫停，還有一些可能常用到的，例如，stop()方法讓影片停止、currentTime 屬性指定秒數 (v01.currentTime=0)、playbackRate 屬性指定播放影片的速率 (v01.playbackRate=2、v01.playbackRate=0.5)。

完整網頁請參閱 t03.htm。

```

<h2>video 的控制</h2>

<div style="text-align:center; ">
  <button onclick="playAndPause()">Play/Pause</button>
  <button onclick="big()">Big</button>
  <button onclick="normal()">Normal</button>
  <button onclick="small()">Small</button>

  <br>
  <video id=v01 controls="controls">
    <source src="myvideo.ogv" type="video/ogg">
    <source src="myvideo.mp4" type="video/mp4">
    您的瀏覽器不支援 html5 的標籤。
  </video>
</div>

<script type="text/javascript">
  //由網頁裡取得 id=v01 的物件。這裡會對應到測試影片
  var v01=document.getElementById("v01");

  //播放或暫停的函式
  function playAndPause() {
    if (v01.paused) { //如果 v01 的狀態目前是暫停的，就進行影片播放
      v01.play();
    } else { //暫停影片
      v01.pause();
    }
  }

  function big() {
    v01.width=480;
  }

  function normal() {
    v01.width=320;
  }

  function small() {

```

```

        v01.width=160;
    }

</script>

<p>在 html5 要控制影片的播放更簡單了，直接操作物件的播放與暫停就可以達成。
<p>此網頁功能參考來源如下：
<br><a href=http://www.w3schools.com/html5/html5_video_dom.asp
target=_blank>w3schools - Take Control Using the DOM</a>

```

## 第 6 節、Canvas

完整網頁請參閱 t05.htm。

### 第 1 項、繪製矩形

Canvas 本身只是一張空白的紙張，要畫圖，可以由 javascript 控制。

```

<canvas id="myCanvas00" width="200" height="100" style="border:1px solid
#c3c3c3;">
    Your browser does not support the canvas element.
</canvas>

<script type="text/javascript">

    var c00=document.getElementById("myCanvas00");
    var ct00=c00.getContext("2d");
    ct00.fillStyle="#0000ff"; //設定填滿的顏色
    ct00.fillRect(0,0,150,75); //設定填滿矩型的起點(x,y)、寬、高

</script>

```

### 第 2 項、繪製矩形時加上透明度

在 javascript 裡可以使用 rgb 設定顏色，也可以使用 rgba 設定顏色與透明度。

```

<canvas id="myCanvas01" width="300" height="300" style="border:1px solid
#c3c3c3;">

```

```

    Your browser does not support the canvas element.
</canvas>

<script type="text/javascript">

    var c01=document.getElementById("myCanvas01");
    var ct01=c01.getContext("2d");
    ct01.fillStyle="rgb(255,0,0)";
    ct01.fillRect(0,0,200,200);
    ct01.fillStyle="rgba(0,0,255,0.5)"; //使用 rgba 設定了透明度
    ct01.fillRect(100,100,200,200);

</script>

```

### 第 3 項、繪製三角形

```

<canvas id="myCanvas02" width="300" height="300" style="border:1px solid
#c3c3c3;">
    Your browser does not support the canvas element.
</canvas>

<script type="text/javascript">

    var c02=document.getElementById("myCanvas02");
    var ct02=c02.getContext("2d");
    ct02.beginPath(); //宣告開始描繪路徑
    ct02.moveTo(10,10); //指定開始描繪的起點
    ct02.lineTo(10,260); //使用線條畫至定點
    ct02.lineTo(260,260);
    ct02.closePath(); //宣告描繪動作結束，由於有 ct02.fill() 動作，
    這行其實可以省略
    //以下這二行是進行填滿的工作
    ct02.fillStyle="#cc0000"; //指定填滿的顏色，若這行省略，則填入黑色
    ct02.fill(); //進行實際繪製工作。如果沒有這行，前面的動作都
    不會真正的繪製到 canvas
    //設定以下這三行，就可以描繪外框，若沒有以下這三行，則會顯示實心的圖案
    ct02.strokeStyle="#0000cc"; //線條的顏色
    ct02.lineWidth=3; //線條的粗細
    ct02.stroke(); //實際進行外框描繪的工作

</script>

```



## 第 4 項、繪製弧形

繪製弧形的例子，使用 `context.arc(x, y, 半徑, 起始角度, 結束角度, 指定為逆時針旋轉)`

```
<canvas id="myCanvas10" width="300" height="300" style="border:1px solid #c3c3c3;">
  Your browser does not support the canvas element.
</canvas>

<script type="text/javascript">

  var c10=document.getElementById("myCanvas10");
  var ct10=c10.getContext("2d");

  ct10.beginPath();
  var startAngle = 0;           //設定起始角度
  var endAngle = 120 * Math.PI / 180; //設定結束角度
  ct10.arc(150, 150, 50, startAngle, endAngle, false); //繪製弧形
  ct10.strokeStyle="#0000cc";
  ct10.lineWidth=1;
  ct10.stroke();

</script>
```

## 第 5 項、線性漸層色彩

繪製線性漸層色彩，這裡是使用 `CanvasGradient` 物件的 `createLinearGradient` 方法繪製線性漸層。

```
<canvas id="myCanvas20" width="300" height="100" style="border:1px solid #c3c3c3;">
  Your browser does not support the canvas element.
</canvas>

<script type="text/javascript">

  var c20=document.getElementById("myCanvas20");
  var ct20=c20.getContext("2d");

  var grd=ct20.createLinearGradient(0,0,300,100);// 使用 Canvas 的
  createLinearGradient 方法
  grd.addColorStop(0, "#FF0000"); //這裡可以加入好幾個偏移量。
```

```

        grd.addColorStop(0.5, "#00FF00");
        grd.addColorStop(1, "#0000FF");

        ct20.fillStyle=grd; // 將線性漸層的設定值 指定給
fillStyle
        ct20.fillRect(0,0,280,80); // 描繪矩形，這裡矩形的寬高故意縮小一
些，grd 超出的部份就不會顯示出來

</script>

```

## 第 6 項、圓形漸層色彩

繪製圓形漸層色彩，這裡是使用 CanvasGradient 物件的 createRadialGradient 方法繪製圓形漸層，設定方式是 context.createRadialGradient(x1, y1, r1, x2, y2, r2)，這裡可以設定二個各別的圓心與半徑。

```

<canvas id="myCanvas21" width="300" height="300" style="border:1px solid
#c3c3c3;">
    Your browser does not support the canvas element.
</canvas>

<script type="text/javascript">

    var c21=document.getElementById("myCanvas21");
    var ct21=c21.getContext("2d");

    //內部半徑 50 的圓是紅色的，外部半徑 150 的圓是綠色的，中間的區域進行
漸層效果
    var grd=ct21.createRadialGradient(150,150,50,150,150,150);
    grd.addColorStop(0, "#FF0000");
    grd.addColorStop(1, "#00FF00");
    grd.addColorStop(1, "#0000FF");

    ct21.fillStyle=grd;
    ct21.fillRect(0,0,300,300);

</script>

```

## 第 7 項、不同圓心半徑的圓形漸層色彩

繪製圓形漸層色彩，故意指定二個不同的圓心與半徑。

```
<canvas id="myCanvas22" width="300" height="300" style="border:1px solid
#c3c3c3;">
    Your browser does not support the canvas element.
</canvas>

<script type="text/javascript">

    var c22=document.getElementById("myCanvas22");
    var ct22=c22.getContext("2d");

    //內部半徑 50 的圓是紅色的，外部半徑 150 的圓是綠色的，中間的區域進行
    漸層效果
    var grd=ct22.createRadialGradient(30,30,30,170,170,100);
    grd.addColorStop(0.3, "#FF0000");
    grd.addColorStop(0.7, "#00FF00");
    grd.addColorStop(1, "#0000FF");

    ct22.fillStyle=grd;
    ct22.fillRect(20,20,260,260);

</script>
```

## 第 8 項、基本繪圖程式

請參考 h04.htm

```
<!DOCTYPE html>
<html lang="zh-TW">
<head>
    <title>網站標題</title>
    <meta charset="utf-8" />

    <meta name="viewport" content="width=device-width, user-scalable=yes,
initial-scale=1.0, maximum-scale=3.0" />

    <script src=modernizr.custom.24002.js></script>

    <style>
        article, aside, figure, footer, header, hgroup, menu, nav, section { display:
block; }
    </style>
```

```
<style>
  body {
    background: white;
  }

  .Toolbar {
    float: left;
    font-family: 'Trebuchet MS';
    font-size: 14px;
    font-variant: small-caps;
    text-align: center;
    background: #F2F7FE;
    padding: 10px 15px 3px 10px;
    margin-bottom: 1px;
    margin-right: 1px;
    border: 1px solid #7B899B;
  }

  .Toolbar button {
    padding: 6px;
    margin: 7px 2px;
    font-variant: normal;
    font-size: 12px;
  }

  .CanvasContainer {
    clear: both;
  }

  canvas {
    border: 1px solid #7B899B;
  }

  img {
    padding: 2px;
    border: 2px solid #F2F7FE;
  }

  img:hover {
    border: 2px groove #E4F0FE;
    background: white;
  }

  img.Selected {
    border: 2px groove #E4F0FE;
  }

  #savedCopyContainer {
```

```

        display: none;
    }

    #savedCopyContainer img {
        width: 250px;
        height: 150px;
    }
</style>

<script language=javascript>
    var canvas;
    var context;

    window.onload = function() {
        // Get the canvas and the drawing context.
        canvas = document.getElementById("drawingCanvas");
        context = canvas.getContext("2d");

        // Attach the events that you need for drawing.
        canvas.onmousedown = startDrawing;
        canvas.onmouseup = stopDrawing;
        canvas.onmouseout = stopDrawing;
        canvas.onmousemove = draw;
    };

    var isDrawing = false;

    function startDrawing(e) {
        // Start drawing.
        isDrawing = true;

        // Create a new path (with the current stroke color and stroke thickness).
        context.beginPath();

        // Put the pen down where the mouse is positioned.
        context.moveTo(e.pageX - canvas.offsetLeft, e.pageY -
canvas.offsetTop);
    }

    function stopDrawing() {
        isDrawing = false;
    }

    function draw(e) {
        if (isDrawing == true) {
            // Find the new position of the mouse.
            var x = e.pageX - canvas.offsetLeft;
            var y = e.pageY - canvas.offsetTop;

```

```
        // Draw a line to the new position.
        context.lineTo(x, y);
        context.stroke();
    }
}

// Keep track of the previous clicked <img> element for color.
var previousColorElement;

function changeColor(color, imgElement) {
    // Change the current drawing color.
    context.strokeStyle = color;

    // Give the newly clicked <img> element a new style.
    imgElement.className = "Selected";

    // Return the previously clicked <img> element to its normal state.
    if (previousColorElement != null) previousColorElement.className = "";
    previousColorElement = imgElement;
}

// Keep track of the previous clicked <img> element for thickness.
var previousThicknessElement;

function changeThickness(thickness, imgElement) {
    // Change the current drawing thickness.
    context.lineWidth = thickness;

    // Give the newly clicked <img> element a new style.
    imgElement.className = "Selected";

    // Return the previously clicked <img> element to its normal state.
    if (previousThicknessElement != null)
previousThicknessElement.className = "";
    previousThicknessElement = imgElement;
}

function clearCanvas() {
    context.clearRect(0, 0, canvas.width, canvas.height);
}

function saveCanvas() {
    // Find the <img> element.
    var imageCopy = document.getElementById("savedImageCopy");

    // Show the canvas data in the image.
    imageCopy.src = canvas.toDataURL();
}
```

```

        // Unhide the <div> that holds the <img>, so the picture is now visible.
        var                imageContainer                =
document.getElementById("savedCopyContainer");
        imageContainer.style.display = "block";
    }
</script>

</head>
<body>

    <h2>小畫家</h2>
    <p>參考來源 : <a href='http://www.prosetech.com/html5/Chapter 08/Paint.html'
target=_blank>http://www.prosetech.com/html5/Chapter 08/Paint.html</a>

    <div class="Toolbar">
        - Pen Color -<br>
        
        
        
    </div>
    <div class="Toolbar">
        - Pen Thickness -<br>
        
        
        
    </div>
    <div class="CanvasContainer">
        <canvas id="drawingCanvas" width="500" height="300"></canvas>
    </div>
    <div class="Toolbar">
        - Operations -<br>
        <button onclick="saveCanvas()">Save the Canvas</button>
        <button onclick="clearCanvas()">Clear the Canvas</button>
        <div id="savedCopyContainer">
            <img id="savedImageCopy"><br>
            Right-click to save ...
        </div>
    </div>

    <p>

```

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

## 第 9 項、描繪方格紙。

```
<canvas id="myCanvas31" width="300" height="300" style="border:1px solid
#c3c3c3;">
  Your browser does not support the canvas element.
</canvas>

<script type="text/javascript">

  var c31=document.getElementById("myCanvas31");
  var ct31=c31.getContext("2d");

  //描繪出淡淡的漸層底色
  var grd = ct31.createLinearGradient(0, 0, 0, 300);
  grd.addColorStop(0, "#e0e0e0");
  grd.addColorStop(1, "#ffffff");
  ct31.fillStyle = grd;
  ct31.fillRect(0, 0, c31.width, c31.height);

  var gridCols = 10;                //建立欄的數量
  var gridRows = 10;                //建立列的數量
  var cellHeight = c31.height / gridRows; //取得每一列的高度
  var cellWidth = c31.width / gridCols; //取得每一欄的寬度
  ct31.lineWidth = 1;
  ct31.strokeStyle = "#a0a0a0";

  ct31.beginPath();

  //繪製橫線
  for (var col=0; col<=gridCols; col++) {
    var x = col * cellWidth;
    ct31.moveTo(x, 0);
    ct31.lineTo(x, c31.height);
  }

  //繪製縱線
  for (var row=0; row<=gridRows; row++) {
    var y = row * cellHeight;
    ct31.moveTo(0, y);
    ct31.lineTo(c31.width, y);
```



```

    }

    ct31.stroke();

</script>

```

## 第 10 項、在方格紙上繪製資料。

```

<canvas id="myCanvas32" width="300" height="300" style="border:1px solid
#c3c3c3;">
    Your browser does not support the canvas element.
</canvas>

<script type="text/javascript">

    var uriAge = [80, 92, 104, 110, 68, 50, 45, 90, 74, 68, 98, 103];
    drawChart(uriAge);

    function drawChart(data) {

        //繪製基本的方格紙
        var c32=document.getElementById("myCanvas32");
        var ct32=c32.getContext("2d");

        //描繪出淡淡的漸層底色
        var grd = ct32.createLinearGradient(0, 0, 0, 300);
        grd.addColorStop(0, "#e0e0e0");
        grd.addColorStop(1, "#ffffff");
        ct32.fillStyle = grd;
        ct32.fillRect(0, 0, c32.width, c32.height);

        var gridCols = data.length + 1 ;           //建立欄的數量
        var gridRows = 10;                         //建立列的數量
        var cellHeight = c32.height / gridRows; //取得每一列的高度
        var cellWidth = c32.width / gridCols; //取得每一欄的寬度
        ct32.lineWidth = 1;
        ct32.strokeStyle = "#a0a0a0";

        ct32.beginPath();

        //繪製橫線
        for (var col=0; col<=gridCols; col++) {

```

```

        var x = col * cellWidth;
        ct32.moveTo(x, 0);
        ct32.lineTo(x, c32.height);
    }

    //繪製縱線
    for (var row=0; row<=gridRows; row++) {
        var y = row * cellHeight;
        ct32.moveTo(0, y);
        ct32.lineTo(c32.width, y);
    }

    ct32.stroke();

    //求出陣列裡的最大值
    var maxV = 0;
    for (var i = 0; i < data.length; i++) {
        if (data[i] > maxV) { maxV = data[i]; }
    }
    //將最大值調整為 1.1 倍，主要是不要畫到最頂端
    maxV = maxV * 1.1;

    //計算繪製的位置
    var points = [];
    for (var i = 0; i < data.length; i++) {
        var v = data[i];
        //使 x 軸的位置能跳過 y 軸，由第一格開始
        var px = cellWidth * (i+1);
        //由於 y 軸的位置的 0 是在上面，往下增加，因此要反過來計算它
        的值
        var py = c32.height - c32.height * ( v / maxV );
        //使用 key:value 的方式存入陣列，也就是 "x" 與 "y" 是索引的名
        稱，px 與 py 是它對應的值
        points.push({"x":px, "y":py});
    }

    //繪製線條圖表
    ct32.beginPath();
    ct32.moveTo(points[0].x, points[0].y);
    for (var i = 1; i < points.length; i++) {
        ct32.lineTo(points[i].x, points[i].y);
    }
    ct32.lineWidth = 2;
    ct32.strokeStyle = "#ee0000";
    ct32.stroke();

    for(var i in points) {
        var p = points[i];

```

```
        ct32.beginPath();
        ct32.arc(p.x, p.y, 6, 0, 2*Math.PI);
        ct32.fillStyle = "#ee0000";
        ct32.fill();
    }
}

</script>
```

## 第 11 項、皮球彈跳的動畫。

```
<canvas id="myCanvas33" width="300" height="300" style="border:1px solid
#c3c3c3;">
    Your browser does not support the canvas element.
</canvas>

<script type="text/javascript">

    var c33=document.getElementById("myCanvas33");
    var ct33=c33.getContext("2d");

    //皮球的狀態
    var ball = {x:10, y:100, dirx:5, diry:5};
    //設定計時器
    setInterval(drawBall, 100);

    function drawBall() {

        //重新描繪背景，等於擦掉重來
        ct33.fillStyle = "#FFFFFF";
        ct33.fillRect(0, 0, c33.width, c33.height);

        //描繪皮球
        ct33.beginPath();
        ct33.arc(ball.x, ball.y, 5, 0, 2*Math.PI, true);
        ct33.fillStyle = "#0000FF";
        ct33.fill();

        //移動皮球
        ball.x = ball.x + ball.dirx;
        ball.y = ball.y + ball.diry;
```

```
        if (ball.x < 0 || ball.x > c33.width) { ball.dirx = ball.dirx * -1; }  
        if (ball.y < 0 || ball.y > c33.height) { ball.diry = ball.diry * -1; }  
  
    }  
  
</script>
```

## 第 14 章、CSS3 新增功能

### 第 1 節、css3 網頁功能

#### 第 1 項、繪製圓角

以前要繪製圓角的區塊，很麻煩，有人使用影像裁切組合，也有的使用 javascript 處理，現在，簡單很多，直接套用 css3 就可以了。

```
article {  
    border: solid 1px black;  
    padding: 15px;  
    margin: 5px;  
    background: yellow;  
    float: left;  
    width: 330px;  
    border-radius: 25px;  
}
```

還可以給他四個不同的圓角大小：

```
border-radius: 25px 50px 25px 85px;
```

完整網頁請參閱 t07.htm。

```
<style>  
    div {  
        text-align:center;  
        font-size: 32px;  
        padding: 10px;  
        margin: 10px;  
        height: 60px;  
        width: 300px;  
        -webkit-border-radius: 15px;  
        -moz-border-radius: 15px;  
    }  
  
    .box1 {  
        background-color: red;  
    }  
    .box2 {
```

```

        background-image: -moz-linear-gradient(bottom, #841F7A 4%,
#9F3B9F 52%, #BF59BF 76%);
        background-image: -webkit-linear-gradient(bottom, #841F7A 4%,
#9F3B9F 52%, #BF59BF 76%);
    }
    .box3 {
        background-color: blue;
        -webkit-box-shadow: 3px 3px 15px #888;
        -moz-box-shadow: 3px 3px 12px #888;
    }
</style>
<div class=box1>這是圓角</div>
<div class=box2>這是圓角加漸層</div>
<div class=box3>這是圓角加影陰</div>

```

## 第 2 項、設定文字影陰

使用 `text-shadow` 屬性可以設定文字影陰，屬性值包含『影陰色彩 x 位移 y 位移 模糊程度』。

```

article {
    font: 30px '標楷體';
    text-shadow: #fff 4px 4px 3px;
}

```

## 第 3 項、多個背景檔案的設定

之前的 CSS 版本裡，一個區塊元件只能設定一個背景圖片，在 CSS3 改善了，我們可以設定多個背景圖片，同時指定每個背景圖片的位置、是否重覆。

```

article {
    border: solid 1px black;
    padding: 15px;
    margin: 5px;
    background-color: yellow;
    background-image:
        url('images/bg01.jpg'),
        url('images/bg02.jpg'),
        url('images/bg03.jpg');
    background-position:
        left top,
        right bottom,
        right top;
}

```

```
background-repeat:
    no-repeat,
    no-repeat,
    no-repeat;
float: left;
width: 96%;
height: 800px;
}
```

## 第 4 項、Hover 虛擬類別

幾乎所有的樣式，我們都可以加上 hover 虛擬類別進行一些簡單的互動效果。

```
<head>
  <style>
    .slickButton {
      color: white;
      font-weight: bold;
      padding: 10px;
      border: solid 1px black;
      background: lightgreen;
      cursor: pointer;
    }
    .slickButton:hover {
      color: black;
      background: yellow;
    }
  </style>
</head>
<body>

  <button class="slickButton">Hover Here!</button>

</body>
```

## 第 5 項、轉換特效

```
<head>
  <style>
    .slickButton {
      color: white;
      font-weight: bold;
```

```

padding: 10px;
border: solid 1px black;
background: lightgreen;
cursor: pointer;
-webkit-transition: background 0.5s;
-moz-transition: background 0.5s;
-o-transition: background 0.5s;
}
.slickButton:hover {
color: black;
background: yellow;
}
}
</style>

</head>
<body>

<button class="slickButton">Hover Here!</button>

</body>

```

## 第 6 項、轉換特效之 2

先在 body 區域裡，加入以下圖片的區塊：

```

<div class="gallery">
  
  
  
</div>

```

在 style 區塊裡，加入以下的宣告，再將滑鼠移到圖片上看看，是不是很棒，不用寫程式，就完成圖片庫的效果：

```

.gallery {
background: #d8eeff;
margin: 30px;
padding: 10px;
}
.gallery img {
width: 150px;
padding: 10px;
margin: 30px;
border: solid 1px black;
}

```



```
background: white;
-webkit-transition: all 1s;
-moz-transition: all 1s;
-o-transition: all 1s;
}
.gallery img:hover {
-webkit-transform: scale(2.2) rotate(10deg);
-moz-transform: scale(2.2) rotate(10deg);
-o-transform: scale(2.2) rotate(10deg);
}
```

## 第 7 項、轉換特效說明

轉換特效的屬性有以下這幾種：

1. transition-property，指定要進行轉換的 CSS 屬性
2. transition-duration，轉換過程的時間，單位為 s 或 ms
3. transition-delay，延遲多久時間後開始轉換，單位為 s 或 ms
4. transition-timing-function，使用已經定義好的方式進行轉換的效果，包含 linear（線性）、ease（很快的加速後再慢慢減速，預設值）、ease-in（加速，越來越快）、ease-out（減速，越來越慢）、ease-in-out（比 ease 還要平緩的轉換動作）、cubic-bezier（自行定義轉換速度的設定）

如果將以上的屬性縮短填寫，要注意各項屬性值是依前項順序方式填寫的。

Transform 屬性，可以讓 html 的某些元素進行 2D 或 3D 旋轉、傾斜、放大縮小的效果。

參考網站：

<http://codepen.io/>

<http://www.css3maker.com/>

<http://daneden.github.io/animate.css/>

<http://thecssguru.freeiz.com/animate/>

## 第 8 項、多欄版面

css3 很容易就可以做到多欄版面。

完整網頁請參閱 t08.htm。

```
<style>
  .box1 {
    text-align:justify;
    font-size: 16px;
    padding: 20px;
    margin: 20px;
    width: 600px;

    -webkit-border-radius: 15px;
    -moz-border-radius: 15px;

    background-color: #ccaaff;
    -webkit-box-shadow: 3px 3px 5px #888;
    -moz-box-shadow: 3px 3px 5px #888;

    -webkit-column-count: 2;
    -webkit-column-gap: 20px;
    -moz-column-count: 2;
    -moz-column-gap: 20px;

  }
</style>

<div class=box1>
  <p>暴龍 ( 學名 : Tyrannosaurus ) 又名霸王龍...
</div>
```

## 第 2 節、段落或文字功能

### 第 1 項、::first-line 與::first-letter 的 pseudo-elements

::first-line 與::first-letter 這二個虛擬元素，分別可以控制第一行以及第一個字元，設定不同的樣式效果，參考以下範例：

```
<style>
  article {
    border: solid 1px black;
    padding: 15px;
```

```
        margin: 5px;
        background: #ddf;
        float: left;
        width: 40%;
    }
    article p::first-line {
        font-size: 20px;
        color: blue;
    }
    aside {
        border: solid 1px black;
        padding: 15px;
        margin: 5px;
        background: yellow;
        float: left;
        width: 40%;
    }
    aside p::first-letter {
        font-size: 48px;
        color: red;
        float: left;
    }
}

</style>

<article>
  <p>很多字...
  <p>很多字...
</article>

<aside>
  <p>很多字...
  <p>很多字...
</aside>
```

另外，在 CSS3 裡，為了與『:hover』這種 pseudo-class 區別，這裡介紹的這二個 pseudo-element，通常就會表示為『::first-line』與『::first-letter』，不過，使用『::』或『:』目前都可以通用的。

## 第 2 項、:focus

之前介紹的: hover 是游標移到標籤時會發生作用，而: focus 則是輸入焦點移到標籤時會有作用，例如：

```
input:focus { background-color: yellow; }
```

### 第 3 項、::before

這個 pseudo element 可以在符合樣式的內容前面，加入一些內容，例如以下的範例，在 article 裡的 p 段落、使用 hot 自訂樣式的最前面，再加入『HOT!!』的文字、同時也將這個文字設定紅色，其它的樣式設定各位也可以試試看。

```
article p.hot::before {
    content: "HOT!! ";
    color: red;
}

<article>
  <p class=hot>很多字...
  <p>很多字...
</article>
```

### 第 4 項、::after

作用類似::before，它是放在指定標籤的最後面。

### 第 5 項、::selection

這個功能是針對我們在網頁上選取文字時，可以進行樣式的設定，底下的『::-moz-selection {...}』是針對 firefox 做的設定。在『::selection』前方沒有指定標籤時，表示是對所有的標籤都有作用。

```
::selection {
    color: #F00;
    background-color: #000;
}

::-moz-selection {
    color: #F00;
    background-color: #000;
}
```

### 第 6 項、Attribute Selector

我們在設定樣式時，可以在標籤後直接加上[]將特定屬性及屬性值加入，當標籤符合、屬性與屬性值也符合時，才套用樣式的設定。以下的範例，是用 a 標籤，再加上『[href="http://www.google.com"]』的內容，因此，只有一個超連

結會套用這個樣式。另外，要注意的是，一般我們可能會忽略屬性值前後的雙引號，不過，在這裡就不能省略，必須加上雙引號才能發揮作用。

```
<style>
  a[href="http://www.google.com"] {
    color: red;
    text-decoration: underline overline;
  }
</style>

<p><a href="http://www.google.com" target=_blank>Google</a>
<p><a href="http://tw.msn.com" target=_blank>MSN</a>
<p><a href="http://tw.yahoo.com" target=_blank>Yahoo</a>
```

這裡也可以套用正規運算式，例如以下例子的『^』表示是開頭的字串，也就是在『href』字串後面，緊接著開頭的字串必須是『=http://』的字串內容才會符合。簡單說，這個的作用是判斷連到外部網站的連結。

```
a[href^="http://"]
```

剛好反相的例子如下，其中的『\$』表示結尾的字串，也就是指副檔名為『.pdf』的超連結。

```
a[href$=".pdf"]
```

## 第 7 項、child selector

我們可以針對某個標籤的 child 進行樣式設定，以下的例子是選擇第一個與最後一個項目做樣式的設定。

```
<style>
  #ul01 li:first-child {
    color: blue;
  }
  #ul01 li:last-child {
    color: red;
  }
</style>

<ul id=ul01>
  <li>第 01 項
  <li>第 02 項
  <li>第 03 項
  <li>第 04 項
```

```
<li>第 05 項  
<li>第 06 項  
<li>第 07 項  
<li>第 08 項  
</ul>
```

以下的範例，可以針對奇數行(odd)與偶數行(even)做不同的樣式設定，這個功能超好用，早期在使用 PHP 輸出表格的<tr>時，如果希望奇數列與偶數列的顏色不同，還要用程式判斷加上不同的 class，現在不用，直接使用 CSS3 樣式套用就可以，其它的由瀏覽器處理。

```
<style>  
  #ul01 li:nth-child(odd) {  
    color: blue;  
  }  
  #ul01 li:nth-child(even) {  
    color: red;  
  }  
</style>
```

以下的範例，是將第 3 個、第 6 個、第 9 個...的項目套用特定樣式。

```
<style>  
  #ul01 li:nth-child(3n) {  
    color: blue;  
  }  
</style>
```

如果期望由第 1 個、第 4 個、第 7 個...方式套用，可以參考底下的方式，將它加上一個數值，讓它產生位移，就可以得到我們想要的結果。

```
<style>  
  #ul01 li:nth-child(3n+1) {  
    color: blue;  
  }  
</style>
```

其實，還有一些其他的 pseudo class 可以運用，不過，我們暫時先提到這裡。

## 第 3 節、Web Storage

本項目暫時有問題，容後修正！

使用 localStorage 物件，可以自行增加 key 與 value。

早期的 Cookie 可以允許每個網站在 Client 端儲存 4KB 的資料，而且，每次網頁的請求都會連帶傳送 cookie 的值，一來佔用網路頻寬，二來使得安全性降低。

在 HTML5 的規範裡，每個來源可以儲存 5MB 的資料，而且，不會跟著網頁的請求傳送回 server 端。

### 第 1 項、計數器

完整網頁請參考 t09.htm。

```
<p>計數器 1 。
</div id=counterDiv class=box1></div>

<script type="text/javascript">

    //將計數器加 1
    if (localStorage.counter == null) {
        localStorage.counter = 1;
    } else {
        localStorage.counter++;
    }

    //顯示計數器
    var cd01 = document.getElementById("counterDiv");
    cd01.innerHTML = localStorage.counter; //新增資料到 cd01 裡

</script>
```

### 第 2 項、記事本

使用 localStorage 物件，使用 localStorage.setItem("key", value) 的方式儲存資料，使用 localStorage.getItem("key") 的方式取出資料。

完整網頁請參考 t10.htm。

```
<div id=note1 class=box1>最後存檔時間</div>
```

```

<textarea id=note2 cols=80 rows=20></textarea>

<script type="text/javascript">

    //取得物件
    var div01 = document.getElementById("note1");
    var textarea01 = document.getElementById("note2");

    //視窗載入時的動作
    window.load = function() {
        var b01 = localStorage.getItem("memo01");
        if (b01 != null) {
            textarea01.value = b01;
        }
    }

    //設定變更文章時的事件
    textarea01.onchange = function() {
        localStorage.setItem("memo01", textarea01.value);
        div01.innerHTML = "saved " + new Date().toString();
    }

</script>

```

## 第 4 節、XMLHttpRequest

使用 Server 端的程式語言配合，功能更強大，同時，這裡介紹使用 XMLHttpRequest 方式要求資料，配合一些簡單的程式，不需要重新整理整個頁面，就可以更新網頁資料。

首先，先撰寫一個 PHP 程式，我將它命名為 serverSide.php，內容如下：

```

<?php
    $b01 = $_GET['a01'];
    $b02 = $_GET['a02'];
    $b03 = $b01 + $b02;
    echo $b03;
?>

```

接著，再撰寫一個 html5 的檔案，在 body 區塊裡的內容如下：

```

<h2>XMLHttpRequest</h2>
<div>
    <p>請輸入二個數值:
        <input id="a01" type="number">
        <input id="a02" type="number">

```



```

    </p>
    <p>直接按底下的按鈕就可以得到答案，畫面不用重新整理:
    <button onclick="askServer()">詢問 Server</button>
    </p>
  </div>
  <p id="result"></p>

```

在同一個 html5 的檔案裡，再加入以下的 script 檔案，有一些地方要注意，XMLHttpRequest 的 open() 方法，有三個參數，分別是傳送方式、接收的程式、是否同步。另外，XMLHttpRequest's onReadyStateChange 事件發生時，也要指定某個 function 處理，在這個例子我們是呼叫 handleServerResponse 處理。readyState 的屬性會有四種狀態：1). 初始化要求，2). 傳送，3). 接收到部份資料，4). 接收完成，接下來，就要看看 status 屬性，200). 表示 OK，401). 表示不允許的存取，404). 表示找不到網頁，因此，在這裡就要判斷 readyState==4 而且 status==200 的情況：

```

<script>
  var req = new XMLHttpRequest();

  function askServer() {

    var a01 = document.getElementById("a01").value;
    var a02 = document.getElementById("a02").value;

    var dataToSend = "?a01=" + a01 + "&a02=" + a02;
    req.open("GET", "serverSide.php" + dataToSend, true);    //true 為同步
    req.onreadystatechange = handleServerResponse;
    req.send();

    document.getElementById("result").innerHTML = "訊息已經傳送.";
  }

  function handleServerResponse() {
    if ((req.readyState == 4) && (req.status == 200))
    {
      var result = req.responseText;
      document.getElementById("result").innerHTML = "加總結果是 " +
result;
    }
  }
</script>

```

將這二個都放在 web server 裡，使用瀏覽器瀏覽操作看看。  
 已完成的範例：<http://ntc.kingbig.idv.tw/html5test/h14.htm>

## 第 5 節、地理位置

### 第 1 項、取得地理資訊

完整網頁請參閱 t11.htm。

```
<input type=button value=取得地理資訊 onclick=showPos(>

<div id=div01></div>

<script type="text/javascript">

    //視窗載入時的動作
    function showPos() {

        var d01 = document.getElementById("div01");

        if (navigator.geolocation == undefined) {
            alert("不支援取得位置資訊的功能！");
            return;
        }

        //取得位置的相關資訊
        navigator.geolocation.getCurrentPosition(          successCallback,
errorCallback );

        //成功取得資訊時
        function successCallback(position) {
            var s01="<h2>成功取得</h2>";
            for (var p01 in position.coords) {
                s01 = s01 + p01 + ":" + position.coords[p01] + "<br>\n";
            }
            d01.innerHTML = s01;
        }

        //取得資訊失敗時
        function errorCallback(err) {
            alert("地理資訊取得失敗(" + err.code + ")" + err.message);
        }

    }

</script>
```

## 第 2 項、地理位置—連到 google

完整網頁請參考 t12.htm。

```
<input type=button value=取得地理資訊 onclick=showPos()>

<div id=div02></div>

<script type="text/javascript">

    //視窗載入時的動作
    function showPos() {

        var d02 = document.getElementById("div02");

        if (navigator.geolocation == undefined) {
            alert("不支援取得位置資訊的功能！");
            return;
        }

        //取得位置的相關資訊
        navigator.geolocation.getCurrentPosition(           successCallback,
errorCallback );

        //成功取得資訊時
        function successCallback(position) {

            var lat = position.coords.latitude;
            var lon = position.coords.longitude;
            var uri = "http://maps.google.com.tw/?q=" + lat + "," + lon;
            var s02="<h2>顯示地圖</h2>";
            s02 = s02 + "<a href=" + uri + " target=_blank>連結 google maps
</a>";

            d02.innerHTML = s02;

        }

        //取得資訊失敗時

        function errorCallback(err) {
            alert("地理資訊取得失敗( " + err.code + " )" + err.message);
        }

    }

</script>
```

## 第 3 項、google maps 的 javascript 函式庫

完整網頁請參閱 t13.htm。

```

<!-- 要先匯入 google maps 的 javascript 函式庫 -->

<script type="text/javascript"
        src="http://maps.google.com/maps/api/js?sensor=true">
</script>

<!-- 當網頁載入完成時，就觸發 onload 函式 -->
<script type="text/javascript">
    window.onload = function() {
        var d04 = document.getElementById("div04");

        //傳送 經緯度
        var loc01 = new google.maps.LatLng(22.769374, 121.117607);
        //繪製 google maps
        var gmap = new google.maps.Map(
            d04, { //指定顯示在 d04 的物件
                zoom:15, //設定縮放倍率
                center: loc01, //將 經緯度 的位置顯示地圖於正中央
                mapTypeId: google.maps.MapTypeId.ROADMAP
            }
        );
    }
</script>

<div id=div04 style="width: 640px; height: 480px; "></div>

```

## 第 6 節、串接樣式表

### 第 1 項、螢幕和列印之間

我們可以使用 media 屬性，針對螢幕顯示和列印時，由系統自動套用不同的樣式設定檔案。

```
<link rel="stylesheet" media="screen" href="styles.css">
```

```
<link rel="stylesheet" media="print" href="stylesPrint.css">
```

## 第 2 項、不同解析度設定不同樣式

我們在 style 樣式設定裡，可以針對主要期望的螢幕解析度進行樣式設定，另外，為了符合不同的螢幕解析度，使用 @media{...} 的方式，指定適合的樣式，使得不同解析度的使用者，都可以有良好的瀏覽閱讀的網頁。

```
<style>
  article {
    border: solid 1px black;
    padding: 15px;
    margin: 5px;
    background: yellow;
    float: left;
    width: 330px;
  }
  aside {
    border: solid 1px black;
    padding: 15px;
    margin: 5px;
    background: yellow;
    position: absolute;
    float: left;
    margin-left: 370px;
  }

  @media (max-width: 480px) {
    article {
      float: none;
      width: auto;
    }
    aside {
      position: static;
      float: none;
      background: pink;
      margin-left: 5px;
    }
  }
</style>

<article>
  <p>很多文字...
</article>

<aside>
```

```
<p>很多文字...
</aside>
```

### 第 3 項、更強化的設定

```
/* Normal styles here */

@media (min-width: 600px) and (max-width: 700px) {
  /* Override the styles for 600-700 pixel windows. */
}

@media (min-width: 400px) and (max-width: 599.99px) {
  /* Override the styles for 400-600 pixel windows. */
}

@media (max-width: 399.99px) {
  /* Override the styles for sub-400 pixel windows. */
}
```

以下的設定結果和前一個是完全相同的。

```
/* Normal styles here */

@media (not max-width: 600px) and (max-width: 700px) {
  /* Override the styles for 600-700 pixel windows. */
}

@media (not max-width: 400px) and (max-width: 600px) {
  /* Override the styles for 400-600 pixel windows. */
}

@media (max-width: 400px) {
  /* Override the styles for sub-400 pixel windows. */
}
```

### 第 4 項、不同解析度套用不同的串接樣式檔

```
<link rel="stylesheet" href="standard_styles">
<link rel="stylesheet" media="(max-width: 480px)" href="small_styles.css">
```

這樣的寫法也可以，不過，和上一個的設定方式還是有所不同。

上一個，會先套用預設的樣式設定，如果發現螢幕解析度小於 480px 時，則會再讀取另一個樣式檔，以新的樣式設定覆寫原有的樣式設定。

下方的設定，則是一開始就將二個不同螢幕解析度的樣式分開處理，所以不會有覆寫樣式設定的狀況。

```
<link rel="stylesheet" media="(min-width: 480.01px)" href="standard_styles">
<link rel="stylesheet" media="(max-width: 480px)" href="small_styles.css">
```

## 第 5 項、識別裝置的螢幕解析度

我們如果要識別不同裝置的螢幕解析度，可以使用 `max-device-width` 進行查詢的動作。例如，早期手機解析度的寬度可能只有 480px，我們就可以用以下的方式進行查詢和判斷。

```
<link rel="stylesheet" media="(max-device-width: 480px)" href="mobile_styles.css">
```

對於新的行動裝置而言，有些裝置在小小的螢幕具有很高的解析度，瀏覽器為了讓使用者能有比較好的瀏覽狀態，避免文字和圖案變的很小，因此，有些會主動放大二倍，這部份我們可以用 `pixel ratio` 屬性判斷。

```
<link rel="stylesheet"
      media="(max-device-width: 480px) and (-webkit-min-device-pixel-ratio: 2)"
      href="style2X.css">
```

使用行動裝置瀏覽網頁時，還可以考慮使用者用橫向或縱向的方式，參考以下的方式處理。

```
<link rel="stylesheet"
      media="(max-device-width: 768px) and (orientation: portrait)"
      href="style_portrait.css">

<link rel="stylesheet"
      media="(max-device-width: 768px) and (orientation: landscape)"
      href="style_landscape.css">
```

## 第 6 項、@import 方式匯入樣式檔

以下這種方式也可以匯入樣式檔案，這種方式和前述的結果相同，都可以匯入外部樣式檔案，不過，前述方式比較常使用。

```
<style>  
  @import url(css/styles.css);  
  @import url(css/form.css);  
</style>
```