

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>
```