



基礎html5-css3

v.24

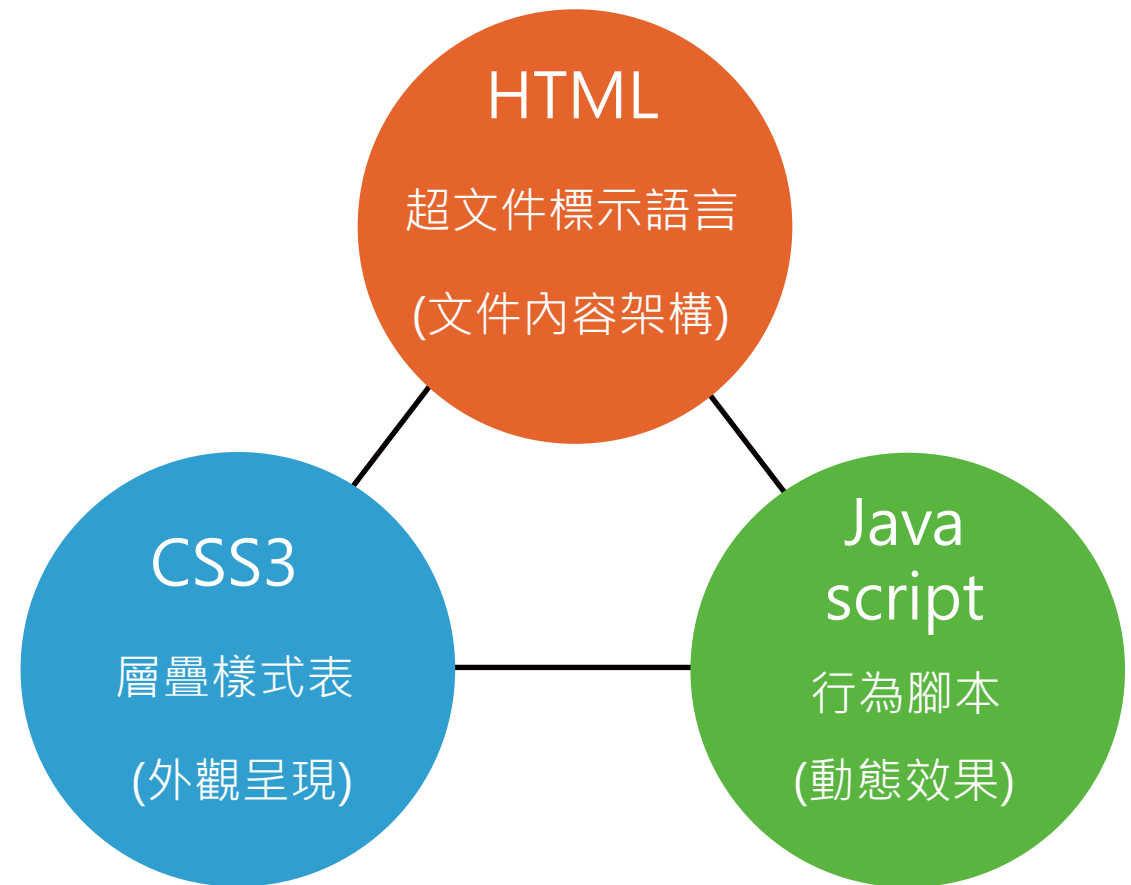
HTML





HTML5是包含了：

HTML、CSS和JavaScript三個部分。
不單單只是HTML部分增加新標籤，
CSS3和JavaScript也有許多的創新，
讓整個網頁程式功能更加繽紛。





- 是最新版本的HTML語言
- HTML5標準，於2014年才正式完成
- 大多數瀏覽器早已開始支援HTML5標準
- 結合html+css3+javascript三種語言
- 設定JavaScript與CSS為預設技術(可不另外宣告)
- 統一許多標準，讓跨瀏覽器的支援度提升
- 精簡舊版標籤，已取消支援許多舊版元素
- 新的語義，並加強元件控制
- 加入新的媒體元件與繪圖功能，建立網站架構有更多元件可選擇
- 許多新功能需依靠JavaScript才能使用
- 可建立網路應用程式API



語意化

語意標籤可以讓我們自然而然的將網頁內容分割成內容、外觀和行為。

重新認識HTML標籤-結構與語意

先思考內容，再決定標籤

撰寫HTML時，只需專注於內容與結構，不用考慮外觀顯示的問題

撰寫CSS時，給語意標籤附加樣式，並再次驗證是否還有可套用之語意標籤



HTML4&HTML5內容結構

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```



```
<!DOCTYPE html>
```

```
<link rel="stylesheet" type="text/css" href="">
```



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

```
<script type="text/javascript"> </script>
```



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

```
<meta http-equiv="Content-Type" content="text/html" charset="UTF-8">
```



```
<meta charset="UTF-8">
```



HTML4&HTML5內容結構

```
<div class=" header" >
```

```
<div class=" nav" >
```

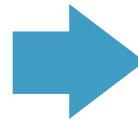
```
<div class=" article" >
```

```
<div class=" section" >
```

```
<div class=" section" >
```

```
<div class=" aside" >
```

```
<div class=" footer" >
```



```
<header>
```

```
<nav>
```

```
<article>
```

```
<section>
```

```
<section>
```

```
<aside>
```

```
<footer>
```



檔頭標籤 <header>

● 檔頭標籤 <header>

<header>元素表示用於介紹內容或一組導航鏈接的容器。一個<header>元素通常包含：

一個或多個標題元素(<h1> - <h6>)標誌、圖標、作者信息。

*<header>標籤不能放置在<footer>、<address>或另一個<header>元素中。

```
<header>
  <h1>田老大工作室</h1>
</header>
<section>
  <header>
    <h2>關於田老大工作室</h2>
  </header>
</section>
```

```
<header>
  <h1>田老大工作室</h1>
  <nav>
</header>
```

```
<article>
  <section>
    <header>
      關於田老大工作室
    </header>
  </section>
```

```
<section>
```

```
<footer>
```




導覽標籤 <nav>

● 導覽標籤 <nav>

建立網頁的導覽區段，連結其他網頁的超連結集合，通常只有主要導覽區段會建立在 <nav> 中。是以 Navigation 縮寫定義。

```
<header>
  <h1>田老大工作室</h1>
  <nav>
    <ul>
      <li><a href="/news/" >最新消息</a></li>
      <li><a href="/services/" >關於我們</a></li>
      <li><a href="/about/" >服務項目</a></li>
    </ul>
  </nav>
</header>
```

```
<header>
```

```
<nav>
```

```
<ul>
```

```
<li><a href="/news/" >最新消息</a></li>
```

```
<li><a href="/services/" >公司服務</a></li>
```

```
<li><a href="/about/" >關於網站</a></li>
```

```
</ul>
```

```
</nav>
```

```
<article>
```

```
<section>
```

```
<section>
```

```
<nav>
```

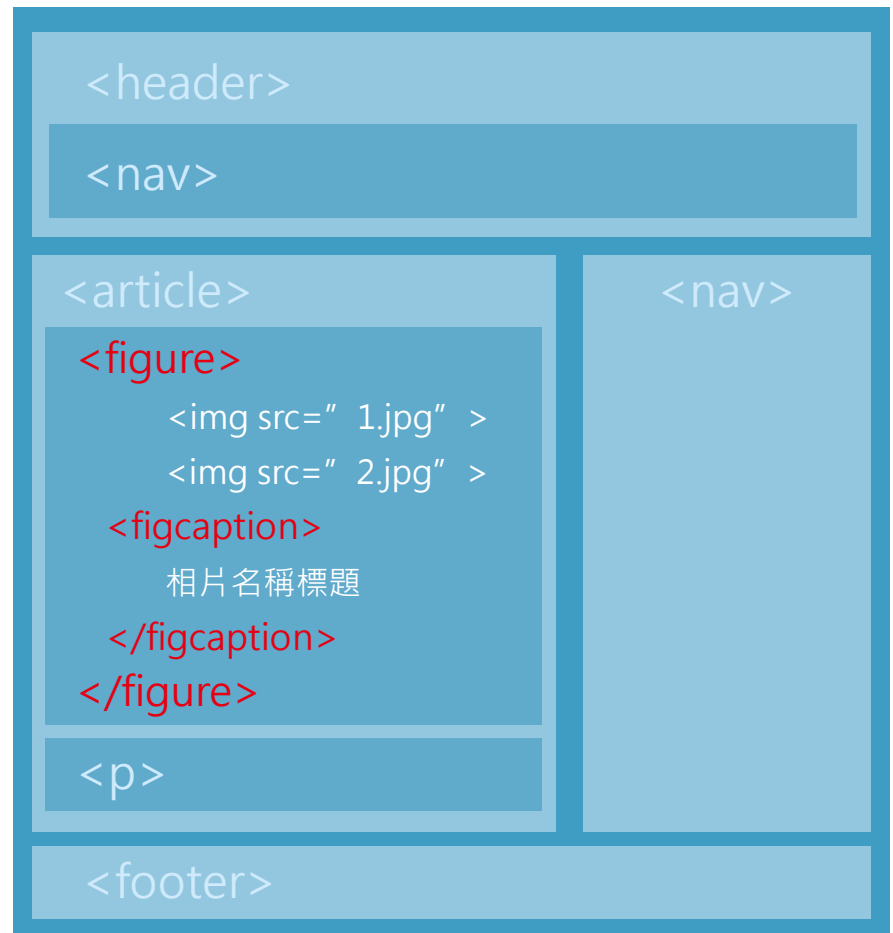
```
<footer>
```



影像群組標籤 <figure>

- 影像群組標籤 <figure>
用來將多個影像封裝成一個群組。
- 相片標題標籤 <figcaption>
用來定義影像名稱。
必須置於 <figure> 中，不可獨立使用

```
<article>
  <figure>
    
    <figcaption>相片名稱標題</figcaption>
  </figure>
  <p>林書豪 ( 英語：Jeremy Shu-How Lin，1988年8月23日 - )，生於美國加州托倫斯，美國職業籃球隊球員，現屬NBA夏洛特黃蜂，場上位置為得分後衛與控球後衛，擅長擋切戰術以及跑轟戰術。</p>
</article>
```





區塊元素說明

【main】

文檔的主要內容

【section】

一段相關主題性的內容區段

【article】

這段內容可以脫離上下文，作為完整的獨立存在的一段內容

【div】

無任何語義，僅僅用作樣式化或者腳本化的容器

原則上來說，能使用article的時候，也是可以使用section的，假如使用article更合適，那麼就不要使用section。
section包含的內容也能算作獨立的一塊，但是它只能算是組成整體的一部分，article才是一個完整的整體。



主要標籤 <main>

● 主要標籤 <main>

文檔的主要內容。元素中的內容應該是唯一的文檔。
它不應該包含跨檔案的文件，如側邊欄，導航列，版權宣告，網站標誌(logo)和搜索表單的內容。
一個文件中只能有一個<main>元素，不能在<article>，<aside>，<footer>，<header>，<nav>裡面。

```
<main>
  <section>
    <h1>公司歷史</h1>
    <p>公司歷史內容</p>
  </section>
  <section>
    <h1>經營理念</h1>
    <p>經營理念內容</p>
  </section>
</main>
```

<header>

<nav>

<main>

```
<article>
  <h1>關於我們</h1>
  <p>關於我們內容</p>
</article>
```

```
<article>
  <h1>產品介紹</h1>
  <p>產品介紹內容</p>
</article>
```

</main>

<footer>



文章標籤 <article>

● 文章標籤 <article>

包含自我完整內容成份的區域，能獨立發送重覆使用的資料。它比section具有更明確的語義,它代表一段脫離了整體還能作為一個完整的、獨立的相關內容。

```
<article>
  <section>
    <h1>公司歷史</h1>
    <p>公司歷史內容</p>
  </section>
  <section>
    <h1>經營理念</h1>
    <p>經營理念內容</p>
  </section>
</article>
```

```
<header>
<nav>

<article>
  <section>
    <h1>公司歷史</h1>
    <p>公司歷史內容</p>
  </section>

  <section>
    <h1>經營理念</h1>
    <p>經營理念內容</p>
  </section>
</article>

<footer>
```



區段標籤 <section>

● 區段標籤 <section>

用來群組一般用途的文件或應用程式，區段**相關**的網頁內容，絕不可包含不相關的內容，常用於文章的章節、標籤對話框中的標籤頁、或者論文中有編號的部分。例如：一個網站的首頁可以分成簡介、最新消息和聯絡方式等多個區段。

```
<article>
  <section>
    <h1>公司簡介</h1>
    <p>公司簡介內容</p>
  </section>
  <section>
    <h1>經營理念</h1>
    <p>經營理念內容</p>
  </section>
</article>
```

```
<header>
```

```
<nav>
```

```
<article>
```

```
  <section>
```

```
    <h1>公司歷史</h1>
```

```
    <p>公司歷史內容</p>
```

```
  </section>
```

```
  <section>
```

```
    <h1>經營理念</h1>
```

```
    <p>經營理念內容</p>
```

```
  </section>
```

```
</article>
```

```
<footer>
```



額外資訊標籤 <aside>

● 額外資訊標籤 <aside>

與article相關的內容片斷，可提供額外資訊進一步補強主要內容，非指向網站其他內頁之超連結，如：說明、提示、引用、附加註解等。

```
<article>
  <section>
    <h1>林書豪簡介</h1>
    <p>林書豪（英語：Jeremy Shu-How Lin，1988年8月23日
- ），生於美國加州托倫斯，美國職業籃球隊球員，現屬NBA夏
洛特黃蜂，場上位置為得分後衛與控球後衛，擅長擋切戰術以及跑
轟戰術。</p>
  </section>
  <aside>
    ^ 2.0 2.1 kevin Ding. Lin is the NBA's Asian-American
inspiration [在NBA成為美籍亞裔榜樣的林書豪].
  </aside>
</article>
```

<header>

<nav>

<article>

<section>

<section>

<aside>

<footer>



頁腳標籤 <footer>

● 頁腳標籤 <footer>

作為區段內容的結束，同一網頁中可以有多個。不可用於<header>及<footer>本身。

<footer>

<small>

老田設計 版權所有 © 2015 . All Rights Reserved.

</small>

</footer>

<small> 標籤：在HTML5中為輔助說明之意。

<header>

<nav>

<section>

<aside>

<article>

<article>

<footer>

<small>

老田設計 版權所有 © 2015 . All Rights Reserved.

</small>

</footer>



多媒體

<audio>

音樂的原生播放支援

HTML5定義了Audio標籤，可以用來撥放音樂，這表示以後的瀏覽器便不需要外掛一些軟體就可以達到這樣的目地，Audio標籤可以和CSS與JavaScript整合，讓我們更容易進行操控音樂的撥放。

```
<!DOCTYPE html>
<html>
<body>
<audio controls="controls">
  <source src="horse.ogg" type="audio/ogg">
  <!--部分版本的Firefox,Opera，不支援mp3或mp4檔案格式的播放，
    所以可設定改用ogg檔-->
  <source src="horse.mp3" type="audio/mpeg">
  您的瀏覽器不支援audio，請升級您的瀏覽器
</audio>
</body>
</html>
```

src (必填)：影片來源url

controls：是否顯示播放影片控制面板

loop：是否自動循環

autoplay：自動播放

muted：是否靜音

「muted」、「''」、不輸入

preload：是否預先載入

「none」、「auto」、「metadata」



多媒體

<video>

影片的原生播放支援

HTML5定義了<video>標籤，可以用來撥放影片，瀏覽器不需要外掛軟體就可以撥放，<video>標籤可以和CSS與JavaScript整合，讓我們更容易進行操控影片或音樂的撥放。

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<video width="320" height="240" controls="controls"
src="movie.mp4" >
```

您的瀏覽器不支援video，請升級您的瀏覽器

```
</video>
```

```
</body>
```

```
</html>
```

src (必填)：影片來源url

width：影片寬度

height：影片高度

controls：是否顯示播放影片控制面板

loop：是否自動循環

autoplay：自動播放

muted：是否靜音

/muted/、/" " /、/不輸入/

poster：指定影片影像(預設影像)

preload：是否預先載入

/none/、/auto/、/metadata/

playsinline：內聯播放

(手機版瀏覽器對於影片自動播放時，會自動展開滿版的燈箱效果來播放影片，設定playsinline可讓影片inline內聯，以避免這個問題)

// playsinline 必須和 autoplay 属性一起使用



多媒體

<picture>

偵測設備尺寸來提供圖像版本

<picture> 元素通過包含零或多個 <source> 元素和一個 元素來為不同的顯示設備，提供圖像版本。瀏覽器會選擇最匹配的 <source> 子元素，如果沒有匹配的，就選擇 元素的 src 屬性中的 URL。

所選圖像呈現於 元素所設定的空間中。

該 <picture> 元素包含兩個不同的標籤：一個或多個 <source> 標籤和一個 標籤。

```
<!DOCTYPE html>
<html>
<body>
<picture>
  <source media="(min-width: 650px)" srcset="photo_l.jpg">
  <source media="(min-width: 465px)" srcset="photo_m.jpg">
  
</picture>
</body>
</html>
```

srcset (必填)：定義要顯示的圖像的URL

media：接受在CSS中可定義的任何有效媒體查詢



多媒體

<canvas>

2D的繪圖功能支援

畫布

<canvas>

<canvas>元素能建立一個直接從瀏覽器中執行繪圖平台，類似一個畫布。本身沒有樣式，必須使用腳本語言來繪製圖形（通常是JavaScript）。預設的畫布大小是 300px * 150px (寬 * 高)。

<!DOCTYPE html>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" >

您的瀏覽器不支援canvas，請升級您的瀏覽器

</canvas>

</body>

</html>



<style>

canvas { border:1px solid #000000; }

</style>



多媒體

<svg> 向量圖支援

SVG是什麼？

可伸縮的向量圖形SVG代表。
SVG是用來定義用於Web的向量圖形。
SVG圖形利用XML格式定義。
SVG圖形縮放或調整，不會失真。
在SVG文件中，每一個元素或屬性都可以是動畫。
SVG是W3C推薦使用。
SVG圖像自動延展父層尺寸。

SVG的特色

可以用任何文本編輯器創建和編輯SVG圖像。
SVG圖像可以被搜索，索引，腳本，和壓縮。
SVG圖像具有可擴展性。
SVG圖像可以在任何平台上提供高品質的印刷。
SVG圖像可縮放。（圖像可以進行放大，縮小，而不會降低）



```
<svg xmlns="http://www.w3.org/2000/svg" width="50" height="50" viewBox="0 10 25 25" fill="#30745E">  
  <path d="M0 16.67l2.829 2.83 9.175-9.339 9.167 9.339 2.829-2.83-11.996-12.17z"/>  
</svg>
```

*viewBox="0 10 25 25" | (min x, min y, width, height)

CSS





區塊

【圓角border-radius】

The border-radius

CSS3 **border-radius**屬性可以被用來創建圓角。
 在CSS2中要建立圓角是很困難的。需要建立背景圖片、切片或Javascript，才能做到。但在CSS3中輕易就可以做到。

數值說明：

border-radius: 四角;

border-radius: 左上 右上 左下 右下;

border-radius: 左上 右上 右下 左下;

各種組合數值：

border-top-right-radius : 右上

border-bottom-right-radius : 右下

border-top-left-radius : 左上

border-bottom-left-radius : 左下

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
border:2px solid #a1a1a1;
padding:10px 40px;
background:#dddddd;
width:300px;
border-radius:25px;
-moz-border-radius:25px; /* Firefox 3.6 and earlier */
}
</style>
</head>
<body>

<div>The border-radius</div>

</body>
</html>
```



區塊

【陰影box-shadow】



CSS3 **box-shadow** 屬性可以被用來製造陰影效果：

box-shadow : x y 模糊 顏色;

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
width:300px;
height:100px;
background-color:yellow;
-moz-box-shadow: 10px 10px 5px #888888;
/* Firefox 3.6 and earlier */
box-shadow: 10px 10px 5px #888888;
}
</style>
</head>
<body>

<div> </div>

</body>
</html>
```




區塊

【透明度opacity】

CSS3 **opacity**屬性可以被用來製造出
半透明效果：

值：0~1;

1 = 100%

0.1 = 10%

套用於父層時，子元素全部都會套用
半透明。

```
<html>
<head>
<style>
div.transbox {
    width:400px;
    height:180px;
    margin:30px 50px;
    background-color:#ffffff;
    border:1px solid black;
    opacity:0.6;
    filter:alpha(opacity=60); /* For IE8 and earlier */
}
</style>
</head>

<body>
<div class="transbox">
    <p>一些內容</p>
</div>
</body>
</html>
```



區塊

【多欄：column-count】

【欄距：column-gap】

CSS3

column-count /*多欄*/

column-gap /*欄距*/

屬性可以被用來製作出多欄排版效果：

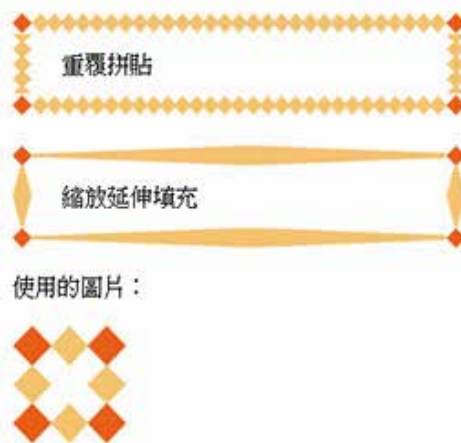
```
<!DOCTYPE html>
<html>
<head>
<style>
.newspaper {
    -moz-column-count:3; /* Firefox */
    -webkit-column-count:3; /* Safari and Chrome */
    column-count:3;

    -moz-column-gap:40px; /* Firefox */
    -webkit-column-gap:40px; /* Safari and Chrome */
    column-gap:40px;
}
</style>
</head>
<body>
<div class="newspaper"> </div>
</body>
</html>
```



邊框

【影像邊框border-image】



CSS3

在邊框圖像屬性採用圖像和切片成九個部分，就像一個井字棋板。然後，它會將邊角處的四角，中間部分被反復拉伸或您指定。

border-image: **source slice;**

屬性可以被用來製造圖片邊框效果：

border-image-width: 設定圖片邊框的寬度;

border-image-outset: 邊框圖片超出邊框的量;

border-image-repeat:

repeat: 用重複方式填滿;

round: 可指定圖片在邊框上以圍繞的方式呈現;

stretch: 可指定圖片在邊框上以伸縮的方式呈現;

```
<style>
div
{
border-width:15px;
width:250px;
padding:10px 20px;
}
#round/*圍繞*/
{
-moz-border-image:url(border.png) 30 30 round; /* Firefox */
-webkit-border-image:url(border.png) 30 30 round; /* Safari and Chrome */
-o-border-image:url(border.png) 30 30 round; /* Opera */
border-image:url(border.png) 30 30 round;
/*border-image:url(border.png) 上右下左 round;*/
/*border-image:url(border.png) 上下左右 round;*/
}
#stretch/*伸縮*/
{
-moz-border-image:url(border.png) 30 30 stretch;
-webkit-border-image:url(border.png) 30 30 stretch;
-o-border-image:url(border.png) 30 30 stretch;
border-image:url(border.png) 30 30 stretch;
}
</style>
```



背景

【伸展背景圖：**background-size**】

CSS3

background-size屬性可以被用來製造背景圖片效果。

參數包含：

auto：自動/默認值。背景圖像以其原始大小顯示。

length：設置背景圖像的寬度和高度。

第一個值設置寬度，第二個值設置高度。

如果僅給出一個值，則將第二個設置為“自動”。

percentage：百分比。以父元素的百分比設置背景圖像的寬度和高度。

第一個值設置寬度，第二個值設置高度。

如果僅給出一個值，則將第二個設置為“自動”。

cover：覆蓋。調整背景圖像的大小以覆蓋整個容器，自動拉伸圖像，較長邊寬邊緣裁切（套用較小邊寬，較長邊寬超過容器範圍裁切掉）。

contain：包含。調整背景圖片的大小，以確保該圖片完全可見。

initial：初始。將此屬性設置為其html默認值。

inherit：繼承。從其父元素繼承此屬性。



背景

【指定背景圖區域：**background-origin**】

background-origin: border-box;



background-origin: content-box;



CSS3

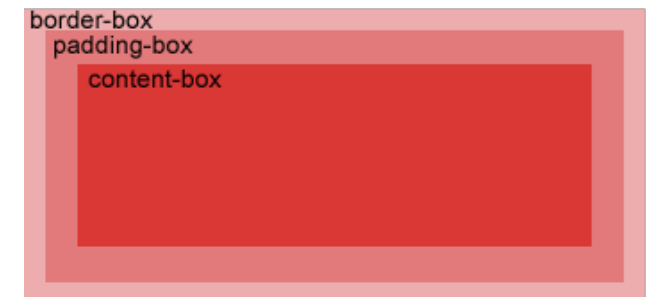
background-origin屬性可以被用來指定背景圖片顯示的區域位置。

border-box可指定圖片在邊框區域。

content-box可指定圖片在內容區域。

。

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
border:1px solid black;
padding:35px;
background-image:url('icon.png');
background-repeat:no-repeat;
background-position:left;
}
#div1
{
background-origin:border-box;
<!--背景來源:邊框區域 -->
}
#div2
{
background-origin:content-box;
<!--背景來源:內容區域 -->
}
</style>
</head>
<body>
<p>background-origin:border-box:</p>
<div id="div1"> </div>
<p>background-origin:content-box:</p>
<div id="div2"> </div>
</body>
</html>
```





背景

背景:線性漸層:



背景:放射性漸層



CSS3

線性漸層：

`background-image: linear-gradient(#fff,#000);`

放射性漸層：

`background-image:radial-gradient(#fff,#000);`

【漸層背景色：`background:gradient`】

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
  border:1px solid black;
  padding:35px;
  background-image:url('icon.png');
  background-repeat:no-repeat;
  background-position:left;
}
#div1
{
  background-image: linear-gradient(#fff,#000);
  <!-- 背景:線性漸層 -->
}
#div2
{
  background-image: radial-gradient(#fff,#000);
  <!-- 背景:放射性漸層 -->
}
</style>
</head>
<body>
  <p>background-origin:border-box:</p>
  <div id="div1"></div>
  <p>background-origin:content-box:</p>
  <div id="div2"></div>
</body>
</html>
```

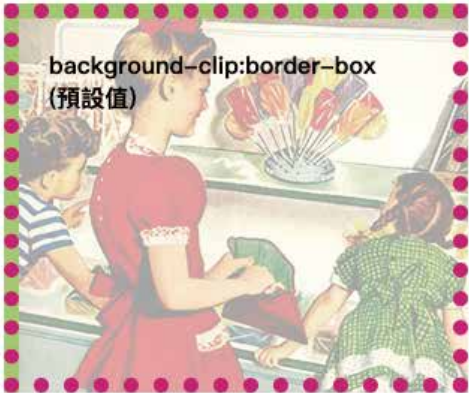


背景

【裁切背景範圍：background-clip】

選取器設定不同的裁切範圍，控制背景圖片顯示區域。

background-clip: border-box | padding-box | content-box



```
width:250px;  
height:200px;  
padding:20px;  
border:10px dotted #E90080;  
background:#A9E969 url(clip_bg.jpg) no-repeat;
```

border-box

左上框線顯示背景色，不會顯示圖片，
右下顯示背景圖，不顯示背景色。



padding-box

內距區及內容區顯示背景圖片。

content-box

內距區不顯示背景圖片，
內容區顯示背景圖片。



文字

【 陰影Text Shadow 】

Text-shadow effect!

CSS3

text-shadow 屬性可以被用來製造文字陰影效果：

text-shadow: x y 模糊 顏色;

```
<!DOCTYPE html>
<html>
<head>
<style>
h1
{
text-shadow: 5px 5px 5px #FF0000;
}
</style>
</head>
<body>

<h1>Text-shadow effect!</h1>

</body>
</html>
```




文字

【字體樣式@font-face】

WITH CSS3, WEBSITES CAN FINALLY
USE FONTS OTHER THAN THE PRE-
SELECTED "WEB-SAFE" FONTS!

CSS3

@font-face屬性可以用來指定字體樣式效果：

有了CSS3，網站終於可以使用預先選定的“網路安全”字體。

注意:Internet Explorer 8之前的版本是不支援新的 @font-face規則

```
<!DOCTYPE html>
<html>
<head>
<style>
@font-face {
    font-family: myFirstFont;
    src: url('http://randsco.com/fonts/lexograph.eot')
        ,url('http://randsco.com/fonts/lexograph.ttf'); /* IE9+ */
}
div {
    font-family:myFirstFont;
    font-size:2em;
}
</style>
</head>
<body>
<div>
    有了CSS3，網站終於可以使用預先選定的“網路安全”字體。
</div>
</body>
</html>
```



文字

【書寫模式 writing-mode】

CSS3

writing-mode: horizontal-tb | vertical-rl | Vertical-lr;

horizontal-tb :

讓內容從左到右水平流動，從上到下垂直流動

Vertical-rl :

讓內容從上到下垂直流動，從右到左水平流動

Vertical-lr :

讓內容從上到下垂直流動，從左到右水平流動

horizontal-tb 讓內容
從左到右水平流動，從
上到下垂直流動。

Vertical-rl 讓內容從上到下垂直流
動，從右到左水平流動。
Vertical-lr 讓內容從上到下垂直流
動，從左到右水平流動。

```
<!DOCTYPE html>
<html>
<head>
<style>
.test1 {
  writing-mode: horizontal-tb;
}
.test2 {
  writing-mode: vertical-rl;
}
.test3 {
  writing-mode: vertical-lr;
}
</style>
</head>
<body>
<p class="test1">
  horizontal-tb 讓內容從左到右水平流動，
  從上到下垂直流動。
</p>
<p class="test2">
  Vertical-rl 讓內容從上到下垂直流動，
  從右到左水平流動。
</p>
<p class="test3">
  Vertical-lr 讓內容從上到下垂直流動，
  從左到右水平流動。
</p>
</body>
</html>
```



轉換

【transform】

https://www.w3schools.com/cssref/css3_pr_transform.asp

CSS3

transform: transform-function;

利用變形函數控制元素變化

2D變形函數:

rotate(θ)

指定元素以參考點為中心軸 2D 旋轉 θ 度。

skewX(θ)

指定元素以參考點為中心軸沿著橫向傾斜 θ 度。

skewY(θ)

指定元素以參考點為中心軸沿著縱向傾斜 θ 度。

skew(θ_x, θ_y)

指定元素以參考點為中心軸沿著橫向傾斜 θ_x 度、縱向傾斜 θ_y 度。參數如果只指定 1 個，省略的第 2 個參數，會視為 0，也就是只有沿橫向傾斜。September, 2012 W3C 草書又復原此項

scaleX(m)

指定元素由參考點橫向縮放 m 倍。

scaleY(m)

指定元素由參考點縱向縮放 m 倍。

scale(m_x, m_y)

指定元素由參考點 2D 橫向縮放 m_x 倍、縱向縮放 m_y 倍，等於是結合 scaleX(m_x), scaleY(m_y)。參數如果只指定 1 個，省略的第 2 個參數，會等於第 1 個，也就是橫向、縱向以相同比例縮放。

translateX(o)

指定元素由參考點橫向移動 o 距離。

translateY(o)

指定元素由參考點縱向移動 o 距離。

translate(o_x, o_y)

指定元素由參考點 2D 橫向移動 o_x 距離、縱向移動 o_y 距離，等於是結合 translateX(o_x), translateY(o_y)。參數如果只指定 1 個，省略的第 2 個參數，會視為 0，也就是只有橫向移動。

matrix(a, b, c, d, e, f)

指定元素由參考點依據數學變形矩陣 (transformation matrix) 的 6 個參數值產生 2D 變形。矩陣的目的主要在運算出四邊形四角的座標，屬於數學問題，這裡就不作太多討論了。事實上，前述的各項變形函數都是簡化這個矩陣而拆解出來的，稍後範例會看到對照。有一些線上工具可以比較容易幫我們產生這些參數值。



旋轉

【transform】

【rotate()】

CSS3

transform:rotate (角度deg)屬性可以被用來**轉換**製造出**旋轉**效果：



```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width:100px;
    height:75px;
    background-color:red;
    border:1px solid black;
}
div#div2 {
    -ms-transform:rotate(30deg); /* IE 9 */
    -moz-transform:rotate(30deg); /* Firefox */
    -webkit-transform:rotate(30deg); /* Safari and Chrome */
    -o-transform:rotate(30deg); /* Opera */
    transform:rotate(30deg);
}
</style>
</head>
<body>

<div>Hello. This is a DIV element.</div>

<div id="div2">Hello. This is a DIV element.</div>

</body>
</html>
```



鏡射

【transform】
【3D rotateX】
【3D rotateY】

CSS3

transform:rotateX(角度deg)

transform:rotateY(角度deg)

屬性可以被用來轉換製造出鏡射效果：

Hello. This is a
DIV element.

Hello. This is a
DIV element.

Hello. This is a
DIV element.

Hello. This is a
DIV element.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width:100px;
    height:75px;
    background-color:red;
    border:1px solid black;
}
div#div2 {
    transform:rotateX(120deg);
}
</style>
</head>
<body>
<div>Hello. This is a DIV element.</div>
<div id="div2">Hello. This is a DIV element.</div>
</body>
</html>
```



傾斜

【transform】
【skew()】

CSS3

`transform:skew(x,y)`屬性可以被用來
轉換製造出傾斜效果：



```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width:100px;
    height:75px;
    background-color:red;
    border:1px solid black;
}
div#div2 {
    transform:skew(30deg,20deg);
}
</style>
</head>
<body>

<div>Hello. This is a DIV element.</div>

<div id="div2">Hello. This is a DIV element.</div>

</body>
</html>
```



轉場

【transition】

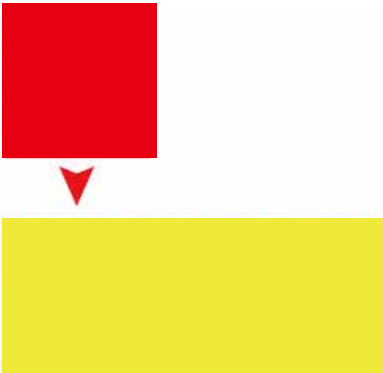
CSS3 transition可以添加轉場動畫效果，無需使用Flash動畫或JavaScript。
CSS3轉換效果，讓元素逐漸改變。

你必須指定事件：

- **transition-property:** (轉場屬性)
none | all | property | initial | inherit ;
- **transition-duration:** (轉場持續時間)
time(2s) | initial | inherit ;
- **transition-timing-function:** (轉場計時功能)
linear | ease | ease-in | ease-out | ease-in-out |
step-start | step-end | steps (int,start|end) |
cubic-bezier(n,n,n,n) | initial|inherit ;
https://www.w3schools.com/cssref/css3_pr_transition-timing-function.asp
- **transition-delay:** (轉場延遲)
time(2s) | initial | inherit ;

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 115px;
    height: 115px;
    background: red;

    transition: width 2s, background 2s;
}
div:hover /*滑入*/ {
    width: 300px;
    background: yellow;
}
</style>
</head>
<body>
<div>可將滑鼠移到上面的div元素，瀏覽轉換效果。
IE瀏覽器不支援這個功能。 </div>
</body>
</html>
```





動畫

【@keyframes Rule】

CSS3

@keyframes

要建立CSS3動畫，你必須了解

@keyframes (@關鍵框架規則)。

@keyframes 指定CSS樣式和動畫逐步從目前的樣式改變為新的樣式。

當創建在動畫@keyframes時，一定要指定一個選項，否則動畫不會有任何效果。

你必須指定兩件事情：

- 1,指定CSS屬性效果。
- 2,指定持續時間效果。

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width:100px;
    height:100px;
    background:red;

    -moz-animation:myfirst 5s;
    -webkit-animation:myfirst 5s;
    -o-animation:myfirst 5s;
    animation:myAmt 5s;
}
@-moz-keyframes myAmt {
    from { background:red; }
    to { background:yellow; }
}
@-webkit-keyframes myAmt {
    from { background:red; }
    to { background:yellow; }
}
@-o-keyframes myAmt {
    from { background:red; }
    to { background:yellow; }
}
@keyframes myAmt {
    from { background:red; }
    to { background:yellow; }
}
</style>
</head>
<body>
<div></div>
</body>
</html>
```




動畫

【Animations】

可簡寫：

animation: myAmt 5s 2s infinite;

| 自訂名稱 | 動畫時間 | 動畫延遲 | 播放次數 | 等...

- 動畫自訂名稱

animation-name: myAmt;

- 動畫時間

animation-duration: 5s;

- 動畫速度曲線

animation-timing-function: linear | ease | ease-in | ease-out |
ease-in-out |

step-start | step-end | steps(int,start|end) |

cubic-bezier(n,n,n,n) | initial | inherit;

animation-timing-function: ease;

- 動畫延遲

animation-delay: 2s;

animation-delay: -2s; /*直接從2秒後開始播*/

- 播放動畫次數

animation-iteration-count: number | infinite | initial | inherit;

animation-iteration-count: 2;

animation-iteration-count: infinite; /*無限次播放*/

- 動畫方向

animation-direction: normal | reverse | alternate | alternate-reverse | initial | inherit;

/* 執行結束後會回到元素設定樣式 */

animation-direction: normal; /*正常播放(默認值)*/

animation-direction: reverse; /*反方向播放*/

animation-direction: alternate; /*正方向來回播放*/

animation-direction: alternate-reverse; /*反方向來回播放*/

- 動畫填充模式

animation-fill-mode: none | forwards | backwards | both | initial |
inherit;

animation-fill-mode: none; /*元素執行後，不會改變原樣式*/

animation-fill-mode: forwards;

/* forwards會抓取指定之元素設定的屬性，並停留在最後一格 */

animation-fill-mode: backwards;

/*執行from,to樣式設定樣式，執行結束後會回到元素設定樣式 */

animation-fill-mode: both;

/*直接執行from,to樣式，元素不需設定動畫樣式。

both不會抓取元素設定的屬性，並停留在最後一格 */

- 動畫播放狀態

animation-play-state: paused | running | initial | inherit;

animation-play-state: paused; /*動畫暫停*/

animation-play-state: running; /*動畫運行(默認值)*/



效果

濾鏡

【filter】

https://www.w3schools.com/cssref/css3_pr_filter.asp

CSS3濾鏡屬性：**filter**

- 模糊 /blur(px)/blur(5px);
- 亮度 /brightness(0%~)/brightness(200%);
- 對比 /contrast(0%~)/contrast(200%);
- 飽和度 /saturate(0%~);/saturate(8%);
- 色相 /hue-rotate(0deg~360deg)/hue-rotate(90deg) ;
- 陰影 /drop-shadow(x軸 y軸 模糊 展開(選填) 顏色)/drop-shadow(8px 8px 10px gray);
- 灰度 /grayscale(0%~100%)/grayscale(100%);
- 反轉 /invert(0%~100%)/invert(100%);
- 不透明度 /opacity(0%~100%)/opacity(30%);
- 棕褐色調 /sepia(0%~100%);/sepia(100%);
- 混用 /contrast(200%) brightness(150%);
- 不要濾鏡 /none;



效果

混合模式

【mix-blend-mode】

https://www.w3schools.com/cssref/pr_mix-blend-mode.asp

CSS3 混合模式屬性：**mix-blend-mode**

- 正常/normal {mix-blend-mode: normal;}
- 色彩增值/multiply {mix-blend-mode: multiply;}
- 濾色/screen {mix-blend-mode: screen;}
- 覆蓋/overlay {mix-blend-mode: overlay;}
- 變暗/darken {mix-blend-mode: darken;}
- 變亮/lighten {mix-blend-mode: lighten;}
- 加亮顏色/color-dodge {mix-blend-mode: color-dodge;}
- 加深顏色/color-burn {mix-blend-mode: color-burn;}
- 差異化/difference {mix-blend-mode: difference;}
- 排除/exclusion {mix-blend-mode: exclusion;}
- 色相/hue {mix-blend-mode: hue;}
- 飽和度/saturation {mix-blend-mode: saturation;}
- 顏色/color {mix-blend-mode: color;}
- 明度/luminosity {mix-blend-mode: luminosity;}



效果

背景濾鏡

【backdrop-filter】

<https://developer.mozilla.org/zh-CN/docs/Web/CSS/backdrop-filter>

/* 關鍵詞值 */

backdrop-filter: none;

/* 指向 SVG 濾鏡的 URL */

backdrop-filter: url(commonfilters.svg#filter);

/* <filter-function> 濾鏡函數值 */

backdrop-filter: blur(2px); //模糊

backdrop-filter: brightness(60%); //亮度

backdrop-filter: contrast(40%); //對比度

backdrop-filter: drop-shadow(4px 4px 10px blue); //陰影

backdrop-filter: grayscale(30%); //灰度

backdrop-filter: hue-rotate(120deg); //色相旋轉

backdrop-filter: invert(70%); //反轉

backdrop-filter: opacity(20%); //不透明度

backdrop-filter: sepia(90%); //棕褐色

backdrop-filter: saturate(80%); //飽和度

/* 多重濾鏡 */

backdrop-filter: url(filters.svg#filter) blur(4px) saturate(150%);

/* 全局值 */

backdrop-filter: inherit;

backdrop-filter: initial;

backdrop-filter: unset;

END.