

HTML & CSS3

HTML是什麼？

HTML

Hyper Text Markup Language

超文字標記語言

標籤組成

```
<p>內容</p>
```

```

```

文件類型

```
<!doctype html>
```

網頁結構

所有的網頁，都是由下列三個主要標籤所組成。

- html

開始使用HTML，原則上所有的HTML都要在此標籤內。

- head

網頁中的相關資訊，但不會顯示在網頁內容。例如作者、敘述、編碼...等

- body

網頁的所有圖文內容，都要放在此標籤之間

常用標籤

標題

```
<h1>標題一</h1>
```

```
<h2>標題二</h2>
```

```
<h3>標題三</h3>
```

```
<h4>標題四</h4>
```

```
<h5>標題五</h5>
```

```
<h6>標題六</h6>
```

段落

```
<p>  
  lorem...  
</p>
```

超連結

```
<a href="...">位置</a>
```

連結到外部網站

```
<a href="https://www.pcschool.com.tw/">巨匠電腦</a>
```

連結到同一網站其他分頁

```
<a href="about.html">關於</a>  
<a href="about/index.html">關於</a>  
<a href="../about/index.html">關於</a>
```

連結到同一網頁其他區塊

```
<a href="#about">關於</a>  
  
<div id="about">  
  lorem...  
</div>
```

連結用分頁開啟

```
<a href="https://www.pcschool.com.tw/" target="_blank">巨匠電腦</a>
```

連結說明

```
<a href="https://www.pcschool.com.tw/" title="前往巨匠電腦">巨匠電腦</a>
```

使用Mail與電話

```
<a href="mailto:xxx@xxx.com">聯絡我</a>
```

```
<a href="tel:+886912345678">聯絡我</a>
```

相對路徑與絕對路徑

- 相對路徑

當要連結到同一網站內的其他網頁時，可使用相對路徑來存取。如

```
<!-- 同一資料夾 -->
<a href="about.html">關於</a>

<!-- 子資料夾about -->
<a href="about/index.html">關於</a>

<!-- 上一層（父層）的about資料夾 -->
<a href="../about/index.html">關於</a>
```

簡單來說，相對路徑是以自己所在的頁面為基準而產生的路徑。由於是以所在頁面為基準，當在不同頁面時所引用的路徑也有可能會不同。

- 絕對路徑

以網站根目錄為基礎的路徑。由於是以根目錄為基礎，所以不管哪個頁面，所引用的路徑皆是相同的。

```
<a href="/about.html">關於</a>

<a href="/about/index.html">關於</a>
```

連結到其他外部網站時，也是一種絕對路徑，如：

```
<a href="https://www.pcschool.com.tw/">巨匠電腦</a>

<a href="https://www.google.com/">Google</a>
```

圖片

插入圖片

```

```

- src

圖片來源位置，可使用相對或絕對路徑

- alt

圖片替代文字，若圖片無法展示，則會以替代文字呈現

- title

圖片說明，將滑鼠停留放置一小段時間，文字會出現在滑鼠游標旁

寬度與高度

```

```

- width

圖片寬度，單位為像素。

- height

圖片高度，單位為像素

在沒有寬度高度的狀態下，圖片會以原始解析度置放在頁面中，可以透過寬度高度調整。但要特別注意寬度高度比例，若比例錯誤圖片比例也會錯誤。

網站推薦

[unsplash](#)

清單

項目清單

```
<ul>
  <li>項目清單</li>
  <li>項目清單</li>
  <li>項目清單</li>
  <li>項目清單</li>
</ul>
```

編號清單

```
<ol>
  <li>編號清單</li>
  <li>編號清單</li>
  <li>編號清單</li>
  <li>編號清單</li>
</ol>
```

定義清單

```
<dl>
  <dt>清單標題</dt>
  <dd>清單敘述</dd>
  <dt>清單標題</dt>
  <dd>清單敘述</dd>
  <dt>清單標題</dt>
  <dd>清單敘述</dd>
  <dt>清單標題</dt>
  <dd>清單敘述</dd>
</dl>
```

表格

有一些具有序列或是規則性的資料，可以使用表格來呈現，像是旅遊行程、交通時刻表等。

基本表格

```
<table>
  <tr>
    <td>表格內容</td>
    <td>表格內容</td>
    <td>表格內容</td>
  </tr>
  <tr>
    <td>表格內容</td>
    <td>表格內容</td>
    <td>表格內容</td>
  </tr>
  <tr>
    <td>表格內容</td>
    <td>表格內容</td>
    <td>表格內容</td>
  </tr>
</table>
```

表格標題

通常表格都上方都會需要標題，這時可以將第一列的td修改成th

```
<table>
  <tr>
    <th>表格標題</th>
    <th>表格標題</th>
    <th>表格標題</th>
  </tr>
  <tr>
    <td>表格內容</td>
    <td>表格內容</td>
    <td>表格內容</td>
  </tr>
  <tr>
    <td>表格內容</td>
    <td>表格內容</td>
    <td>表格內容</td>
  </tr>
</table>
```

合併表格

列合併

```
<table>
  <tr>
    <th>表格標題</th>
    <th>表格標題</th>
    <th>表格標題</th>
  </tr>
  <tr>
    <td>表格內容</td>
    <td>表格內容</td>
    <td rowspan="2">列合併</td>
  </tr>
  <tr>
    <td>表格內容</td>
    <td>表格內容</td>
    <!-- <td>列合併</td> -->
  </tr>
</table>
```

欄合併

```
<table>
  <tr>
    <th>表格標題</th>
```

```
<th>表格標題</th>
<th>表格標題</th>
</tr>
<tr>
  <td>表格內容</td>
  <td>表格內容</td>
  <td>表格內容</td>
</tr>
<tr>
  <td colspan="3">欄合併</td>
  <!-- <td>欄合併</td> -->
  <!-- <td>欄合併</td> -->
</tr>
</table>
```

長表格

```
<table>
  <thead>
    <tr>
      <th>表格標題</th>
      <th>表格標題</th>
      <th>表格標題</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>表格內容</td>
      <td>表格內容</td>
      <td>表格內容</td>
    </tr>
    <tr>
      <td>表格內容</td>
      <td>表格內容</td>
      <td>表格內容</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>表格註解</td>
      <td>表格註解</td>
      <td>表格註解</td>
    </tr>
  </tfoot>
</table>
```

表單

表單是在網頁應用程式中非常重要的元素，不管是搜尋、會員登入或是線上金流等，表單都是不可或缺的。

表單運作流程

表單結構

```
<form action="" method="">
  <!-- 略 -->
</form>
```

- action

接收表單資訊的路徑

- method

表單傳送的方法，有get與post兩種

常用文字輸入

```
<form action="" method="">
  <!-- 單行文字 -->
  <input type="text" name="user">

  <!-- 密碼 -->
  <input type="password" name="pw">

  <!-- email -->
  <input type="email" name="email">

  <!-- 日期 -->
  <input type="date" name="birth">

  <!-- 數值 -->
  <input type="number" name="num">
</form>
```

- type

控制input標籤的類型

- name

後端程式取值時所需要的屬性。

單選核取按鈕

```
<form action="" method="">
  <input type="radio" name="gender" value="male">男
```



```
<input type="radio" name="gender" value="female">女
</form>
```

- value

單選核取按鈕的值。若是自行輸入的input，像是單行文字、密碼等，輸入的文字就是值。但如果是選取的，像是單選核取按鈕、多選核取按鈕與下拉式選單，則都要在後方設定value。

多選核取按鈕

```
<form action="" method="">
  <input type="checkbox" name="hobby" value="吃">吃
  <input type="checkbox" name="hobby" value="喝">喝
  <input type="checkbox" name="hobby" value="玩">玩
  <input type="checkbox" name="hobby" value="樂">樂
</form>
```

下拉式選單

```
<form action="" method="">
  <select name="edu">
    <option value="國小">國小</option>
    <option value="國中">國中</option>
    <option value="高中職">高中職</option>
    <option value="大專院校">大專院校</option>
  </select>
</form>
```

按鈕

```
<form action="" method="">
  <input type="submit" value="送出">
  <input type="button" value="自定義">
  <input type="reset">
  <button type="submit">送出</button>
  <button type="button">自定義</button>
  <button type="reset">重設</button>
</form>
```

label標籤

通常會在input的前方加上label標籤當作input的標示，以便讓視障人士也可以透過螢幕閱讀器使用表單。

```
<label for="user">帳號</label>
<input type="text" id="user">

<label for="user">
  帳號
  <input type="text" id="user">
</label>
```

- for

for屬性可以指定該標籤是哪一組的input控制項。

群組表單

```
<form>
  <fieldset>
    <legend>表單標題</legend>
    <!-- 表單內容 -->
  </fieldset>
</form>
```

可使用

語意標籤

```
<nav></nav>
<header></header>
<footer></footer>
<section></section>
<article></article>
<aside></aside>
<main></main>
<figure>
  <figcaption></figcaption>
</figure>
```

文字格式

```
<b></b>
<i></i>
<strong></strong>
<em></em>
<del></del>
<s></s>
<u></u>
```

```
<small></small>  
<hr>
```

區塊元素與行內元素

區塊元素在預設的狀態下，都會新增一行。行內元素則會保持在同一行，直到寬度不夠才會換行。

div與span

div是一種不具任何格式意義的區塊元素，若網頁中的區塊不適用任何語意元素時，就可以使用div；span則是不具任何意義的行內元素，若網頁中的沒有人和元素可以區分文字時，則可使用span元素

CSS基礎

CSS 分類

- 嵌入式樣式
- 內部樣式
- 外部樣式表

常用選取器

- 標籤選取器
- 類別選取器 (class)
- id選取器
- 群組選取器
- 後代選取器
- 萬用選取器
- 虛擬類別

標籤選取器

直接用標籤名稱開頭。

```
h1 {}  
  
/* 選取所有的h1標籤 */
```

類別選取器

英文句點開頭，名稱自訂。

```
<style>  
  .box {}  
  
/* 選取所有class="box"的標籤 */
```

```
p.box {}

/* 只選取p並且class="box"的標籤 */
</style>
<div class="box"></div>
<p class="box"></div>
```

id選取器

#字號開頭，名稱自訂

```
<style>
  #wrapper {}

  /* 選取所有id="wrapper"的標籤 */

  p#box {}

  /* 只選取p並且class="box"的標籤 */
</style>
<div id="wrapper"></div>
<p id="box"></div>
```

id選取器要特別注意，在同一個頁面之下，id選取器只能使用一次。

群組選取器

將選取器用逗號分隔。

```
<style>
  h1,h2,h3 {}
  /* 同時選取h1,h2,h3三個元素 */
</style>
```

後代選取器

將選取器用空格分隔。

```
<style>
  .container h2 {}
  /* 選取container內的h2元素，外層h2不受影響 */
</style>
<div class="container">
  <h2>Hello!</h2>
</div>
```

```
<h2>Hello</h2>
<!-- 此行不受影響 -->
```

萬用選取器

使用*字號

```
<style>
  * {}
  /* 選取所有的元素*/
</style>
```

虛擬類別選取

- link
- visited
- hover
- active

```
<style>
  a:link {}
  /* 連結在一般的狀態時*/
  a:visited {}
  /* 點選過的連結*/
  a:hover {}
  /* 滑過連結*/
  a:active {}
  /* 作用中連結*/
</style>
```

權重

選取器選取的元素愈少，權重愈重；元素愈多，則愈輕

套用順序(權重)

嵌入式 > id > class > tag

若權重相等則考慮套用順序

內部樣式與外部樣式相同的情況下，會以後方寫的套用。

文字相關樣式

文字色彩 - color

在網頁中使用顏色的方式有下列幾種:

- 色彩名稱 red
- rgb色碼 rgb(255,0,0)
- rgba色碼 rgba(255,0,0,0.5)
- 16進位色碼 #ff0000,#f00

文字對齊 - text-align

值：

- left
- center
- right
- justify

文字裝飾 - text-decoration

值：

- none
- underline
- overline
- line-through

字體大小 - font-size

常用單位

值：數值

單位：

- px
- pt
- em
- rem

em 與 rem 的差別

- em 的比例是依循最靠近的父元素；rem 的比例是依循根的比例。假設父元素是 24px，子元素是 1.6em，最後的大小就是 24 x 1.6；若子元素是1.6rem，則是 16 x 1.6。
- 根可以透過 :root 修改。

字體種類 - font-family

值：字體名稱

```
font-family: 'arial','tahoma'
```

通常會在後方設定兩個以上的字體，如果某一個瀏覽器抓不到第一個字體，則會抓第二個字體；若第二個也抓不到則會使用瀏覽器預設字體。

字體粗細 - font-weight

值：

- bold
- bolder
- light
- lighter
- 100-900
- normal

斜體字 - font-style

值：

- italic
- normal

行高 - line-height

值：數值

行高不管設定為多少，文字一定都會行的正中間。

字元間距 - letter-spacing

值：數值

單字間距 - word-spacing

值：數值

文字大小寫 - text-transform

值：

- uppercase
- lowercase
- capitalize

背景

背景色 - background-color

值：色碼

背景圖片 - background-image

值：

- 圖片路徑

```
body {  
    background-image: url("圖片路徑");  
}
```

背景位置 - background-position

值：

- 水平距離與垂直距離
- top
- bottom
- left
- right

背景重複方式 - background-repeat

值：

- repeat
- no-repeat
- repeat-x
- repeat-y

背景固定方式 - background-attachment

值：

- scroll
- fixed

背景大小 - background-size

值：

- 圖片寬度與高度
- cover
- contain

清單

清單樣式 - list-style

值：

- none
- disc

- circle
- square
- decimal
- upper-alpha
- lower-alpha
- upper-roman
- lower-roman
- cjk-ideographic

盒子模型

寬度 - width

值：px,%,vw

高度 - height

值：px,%,vh

%與vh

在預設的狀態下，%需要有父層的高度才能使用，所以製作滿版高度時，要先將html,body設定高度，才能設定100%。

```
<style>
  html,body {
    height: 100%;
  }
  .container {
    height: 100%;
  }
</style>

<div class="container"></div>
```

vh則是指視窗的高度，不管是否有父層都可以製作滿版高度。

```
<style>
  .container {
    height: 100vh;
  }
</style>

<div class="container"></div>
```

邊框 - border

- border-style
- border-width
- border-color
- border-left-style
- border-left-width
- border-left-color
- border-right-style
- border-right-width
- border-right-color
- border-top-style
- border-top-width
- border-top-color
- border-bottom-style
- border-bottom-width
- border-bottom-color
- border
- border-left
- border-right
- border-top
- border-bottom

內距 - padding

- padding
- padding-left
- padding-right
- padding-top
- padding-bottom

寬度與高度計算

總寬度 = width + border-left + border-right + padding-left + padding-right

總高度 = height + border-top + border-bottom + padding-top + padding-bottom

外距 - margin

- margin
- margin-left
- margin-right
- margin-top
- margin-bottom

顯示 - display

- block
- inline
- inline-block
- table
- table-cell
- flex
- none

box-sizing

讓width包含border與padding

- border-box
- content-box

定位-position

靜態 - static

靜態定位是所有元素的預設值。

相對 - relative

以預設位置為基準的指定位置

絕對 - absolute

以最相近父元素為基準的定位。若所有父元素都沒有設定定位（static等同沒有設定），則會以視窗為基準，並隨著頁面滾動而移動。

固定 - fixed

相對於視窗的定位，且不會隨著視窗移動。

黏性 - sticky

依據使用者滾動卷軸的位置定位

表格樣式

表格基礎

- width

- height
- border

border-collapse

- collapse

vertical-align

垂直對齊只有inline, inline-block與table中有效

- top
- middle
- bottom

垂直置中

line-height

行高不管設定多少，文字的內容一定會在行中間。

```
<style>
  .box {
    width: 600px;
    height: 400px;
  }
  .container {
    line-height: 400px;
  }
</style>
<div class="container">
  <div class="box">
    <p>Hello World!</p>
  </div>
</div>
```

line-height置中的缺點

只可以使用單行文字。

table

vertical-align可以在table中實現，所以把區塊轉換成table來實現垂直置中

```
<style>
  .box {
    width: 600px;
    height: 400px;
  }
</style>
```

```
.container {  
  display: table;  
}  
.box {  
  display: table-cell;  
  vertical-align: middle;  
}  
</style>  
<div class="container">  
  <div class="box">  
    <p>Hello World!</p>  
  </div>  
</div>
```

table置中的缺點

至少要包兩層元素

padding

利用上下padding的推擠達到置中效果

```
<style>  
  .box {  
    width: 600px;  
  }  
  .container {  
    padding-top: 100px;  
    padding-bottom: 100px;  
  }  
</style>  
<div class="container">  
  <div class="box">  
    <p>Hello World!</p>  
  </div>  
</div>
```

padding置中的缺點

無法得到確切的高度

排版 - 使用inline-block

inline-block優點

- 可併排
- 有width跟height

inline-block的缺點

- 元素之間有4px不等的空隙
- 無法使用margin:auto置中

inline-block解決方案

1. 父元素的font-size設定為0
2. 子元素的font-size還原
3. 設定垂直對齊

排板 - 使用float

float優點

- 可併排
- 元素之間無空隙

float缺點

- 後方元素會被浮動影響
- 父元素崩塌

float解決方案

- height
- overflow
- clearfix
- flowroot

排板 - 使用flex

- display
- flex-wrap
- flex-direction
- justify-content
- align-items
- align-self
- align-content
- flex-basis
- flex-grow
- flex-shrink
- gap

排版 - 使用grid

grid基本

- display
- grid-template
- grid-template-columns
- grid-template-rows

- grid-auto-flow
- grid-auto-row
- grid-auto-column

track 大小

- grid-column
- grid-column-start
- grid-column-end
- grid-row
- grid-row-start
- grid-row-end

grid區域

- grid-area
- grid-template-areas

grid間距

- gap
- column-gap
- row-gap

grid對齊

- justify-items
- align-items
- place-items
- justify-content
- align-content
- place-content
- justify-self
- align-self
- place-self

單位與屬性

- fr
- minmax
- repeat

RWD 響應式網頁設計

媒體判斷與中斷點

```
@media screen and (min-width:600px){  
    /* ...略 */  
}
```

```
@media screen and (max-width:600px){  
    /* ...略 */  
}  
@media screen and (max-width:1200px) and (min-width:600){  
    /* ...略 */  
}
```

viewport設定

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

圖片處理

img

```
  
  
  
  

```

picture

```
<picture>  
    <source media="(min-width: 750px)"  
           srcset="images/horses-1600_large_2x.jpg 2x,  
                 images/horses-800_large_1x.jpg" />  
    <source media="(min-width: 500px)"  
           srcset="images/horses_medium.jpg" />  
      
</picture>
```

RWD與AWD

一套CSS和多套CSS

RWD(Responsive Web Design)響應式網頁設計，和AWD(Adaptive Web Design)自適應式網頁設計，主要都是針對現今行動裝置所發展出來的CSS方法。簡單來說，RWD是一套CSS通用於各裝置，AWD是針對裝置寫相對應獨立的CSS，可以有多套CSS，例如桌機有桌機的一套CSS，手機有自己的一套CSS。

如何判斷裝置？

RWD的方式會把所有裝置的style寫在一支CSS裡，當使用者打開了網頁，能即時透過網頁的寬度去決定要吃到相對應裝置的style；而AWD的方式必須仰賴程式端(工程師)去寫程式判斷使用者開啟網頁的裝置，再決定匯入相對應的CSS檔案，例如判斷這個裝置是桌機，那就匯入桌機的CSS檔案，若是手機就匯入手機的CSS檔案。