

網頁設計入門

資展國際股份有限公司

王緯宸

2024/02/05

eddiewang@ispan.com.tw

課程目錄

Module 1. 網頁運作基本觀念

Module 2. 網頁文件架構

Module 3. 網頁撰寫基礎

Module 4. CSS基礎知識

Module 5. CSS基礎知識

Module 6. CSS基礎知識

Module 7. CSS基礎應用

Module 8. 方塊模型

Module 9. 表格元素與樣式

Module 10. 表單標籤與樣式

Module 11. 表單標籤與樣式

Module 12. input類型

Module 13. HTML元素

Module 14. CSS樣式屬性

補充教材

Module 1 – 網頁運作基本觀念

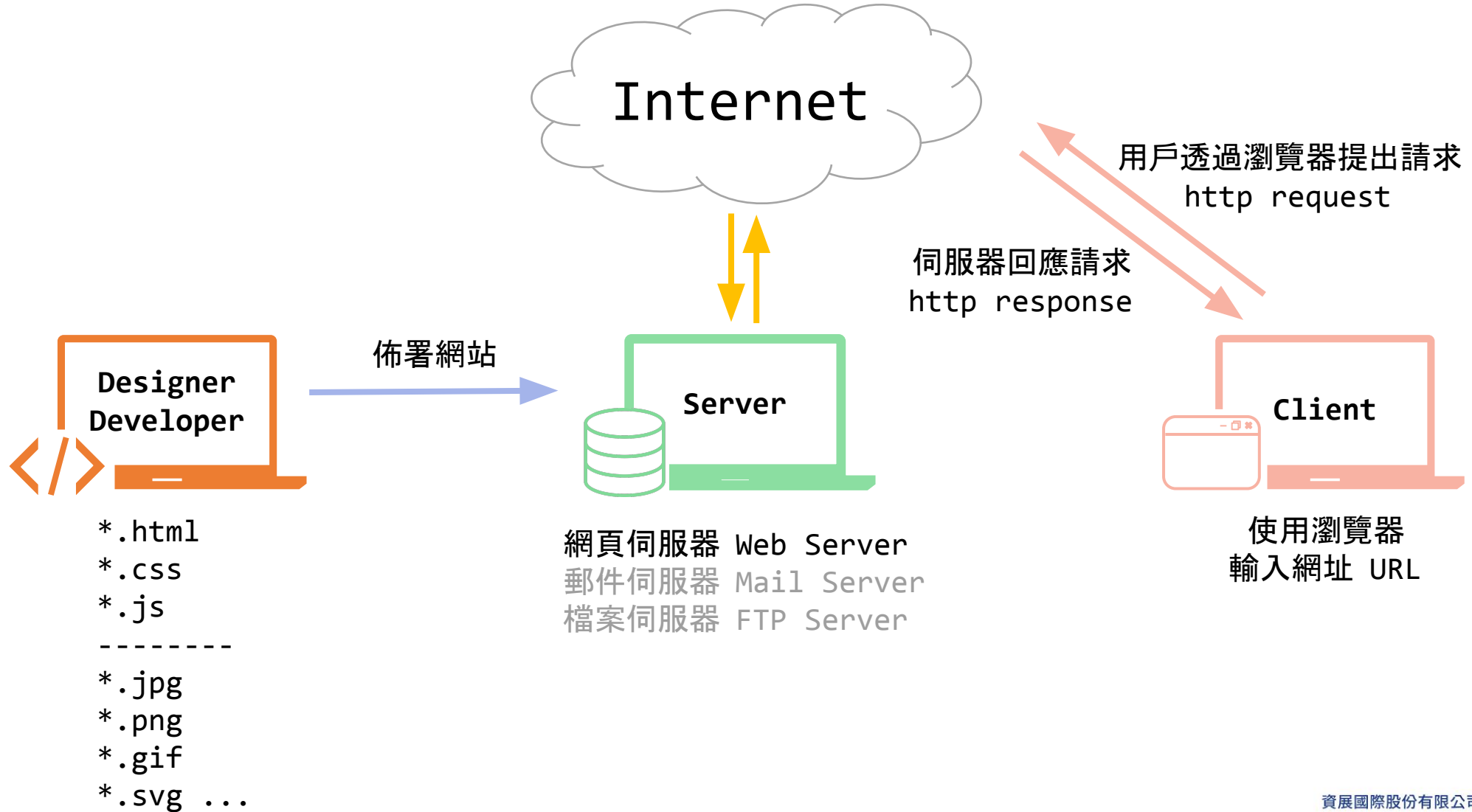
1-1. 網站運作原理及網頁三兄弟

1-2. HTML5演進與標準

1-3. 安裝與熟悉 VS Code

1-4. HTML標籤語法與結構

網站運作原理



網頁三兄弟



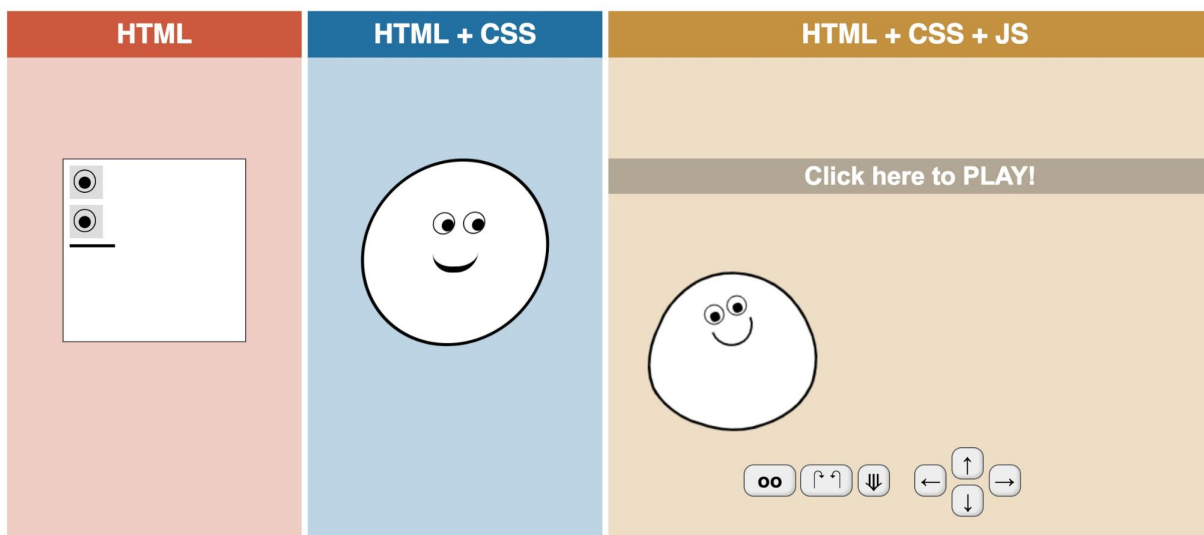
負責網頁的內容



負責美化網頁外觀



根據使用者行為
負責控制網頁內容以及樣式



圖片引用(陳奕帆, Andy Chen)

HTML的歷史

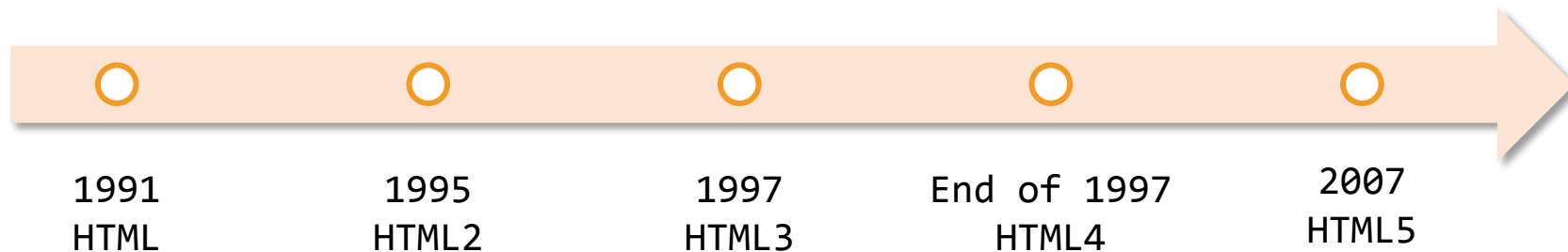
Hypertext Markup Language 超文字標記(標籤)語言

由Berners-Lee於1991年所發明，最初設計的目的是要將電腦中的文字與圖片用一種**固定的格式**來呈現，並透過**超連結**的方式，將這些資料整合在一起，分享給其他使用者。

- W3C (World Wide Web Consortium) 全球資訊網協會制定推薦使用的標準
- W3C官網 | <https://www.w3.org>
- WHATWG (Web Hypertext Application Technology Working Group) 網頁超文字應用技術工作小組
- WHATWG官網 | <https://whatwg.org>

HTML的演進

- HTML的演進



- HTML5

- HTML5在2014年成為標準的最新版本；
- HTML5已經不單純是網頁文件，而是網頁應用程式平台，技術包含HTML、CSS、JS APIs。

如何知道瀏覽器對HTML5支援狀態？

1. <https://html5test.co/>

- 檢查各個瀏覽器支援 HTML5 的程度
- 從不同瀏覽器執行 Chrome / Edge / Safari / Firefox ...

2. <https://caniuse.com/>

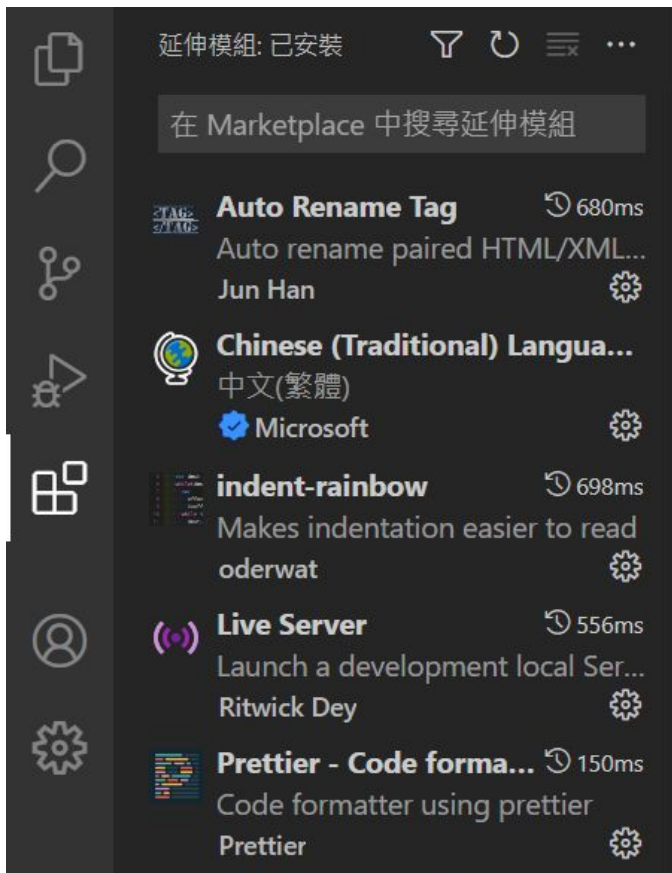
- 查詢標籤、屬性、語法等在各個瀏覽器是否有支援
- 例如: border-radius、zoom-in
- W3C推薦標準的四階段
 1. 工作草案(WD)
 2. 候選推薦標準(CR)
 3. 提議推薦標準(PR)
 4. 推薦標準(REC)

Visual Studio Code

VS Code只是一個開發工具，HTML則是一種語言。

- 為什麼使用 Visual Studio Code([VS Code](#))?
 1. 符合不同作業系統。
 2. 安裝延伸模組可以加速開發。
 3. 支援Emmet語法。
- Emmet
 1. Emmet 使用類似 CSS Selectors 的語法幫助開發人員「快速」的使用HTML標籤，也可以透過縮寫的方式加速開發。
 2. <https://docs.emmet.io/cheat-sheet/>

安裝延伸模組



搜尋並安裝

1. Auto Rename Tag - 同步修改結束標籤
2. Chinese (Traditional) - 繁體中文介面
3. indent-rainbow - 巢狀結構顏色區分
4. Live Server - 瀏覽網頁工具
5. Prettier - 自動排版

設定本機站台

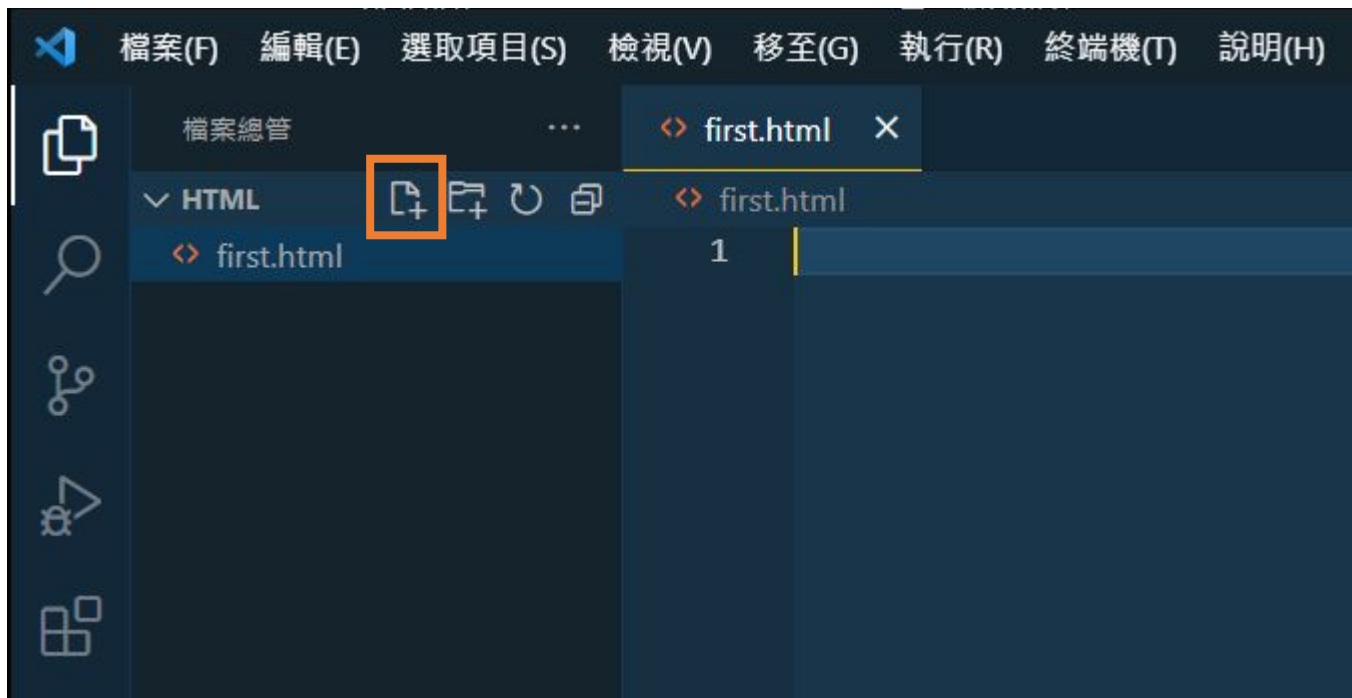
在桌面建立「HTML」資料夾。

點選上方工具列 -> 檔案(F) -> 開啟資料夾 -> 選擇資料夾



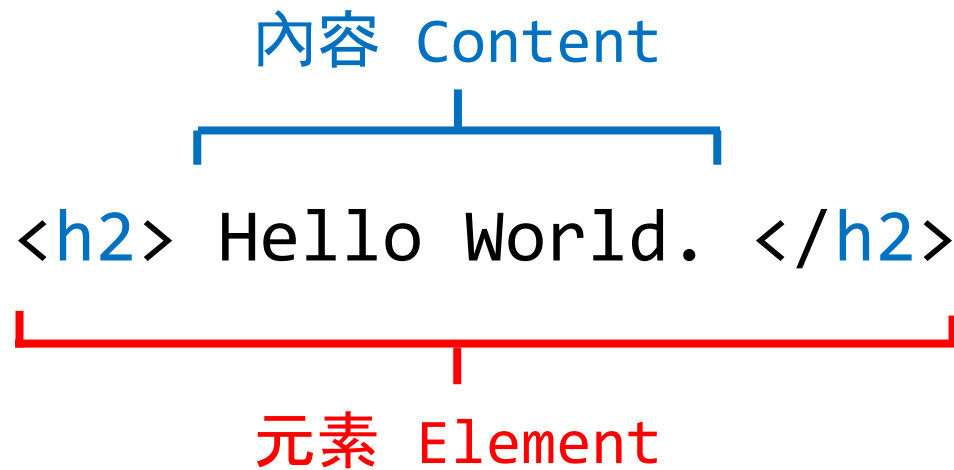
新增空白檔案

點選資料夾(HTML)右方第一個圖示 -> first.html



HTML元素的組成

- HTML元素包含「 開始標籤 Opening Tag 」、「 內容 Content 」及「 結束標籤 Closeing Tag 」三個重要的部分。



空元素(單一標籤)

- 有些標籤是沒有結束標籤, 例如:、
、<hr />、<input /> 等等。
- 可以在結尾的「>」符號前加上「/」符號, 表示結束。

標籤 Tag

元素 Element

HTML屬性(屬性名稱、屬性值)

- HTML標籤內的屬性是用來表示該元素的性質、功能或是狀態等資訊。
- 屬性的語法是將屬性放在「 開始標籤 」內, 並用「 半形空格 」與標籤名稱分開。
- 屬性名稱放在等號「 = 」左邊, 屬性值放在右邊, 並用雙引號「 " 」包住。

屬性名稱 屬性值

`<h2 title="搜尋引擎">Google</h2>`

HTML屬性 HTML Attribute

The diagram illustrates the syntax of an HTML attribute. It shows the code snippet `<h2 title="搜尋引擎">Google</h2>`. Above the code, a blue bracket labeled '屬性名稱' (Attribute Name) spans the word 'title', and an orange bracket labeled '屬性值' (Attribute Value) spans the entire 'title="搜尋引擎"' portion. Below the code, a red bracket labeled 'HTML屬性' (HTML Attribute) spans the entire 'title="搜尋引擎"' portion, with the text 'HTML Attribute' written below it.

HTML元素多個屬性

- HTML標籤內可以放入多個屬性，兩個屬性間用「 半形空格 」隔開。
- HTML常見屬性：
 - title: 滑鼠移動到元素上，呈現的補充文字。
 - style: 元素的行內樣式，使用CSS的語法。

```
<h2 title="搜尋引擎" style="color:red;">Google</h2>
```

title屬性

style屬性

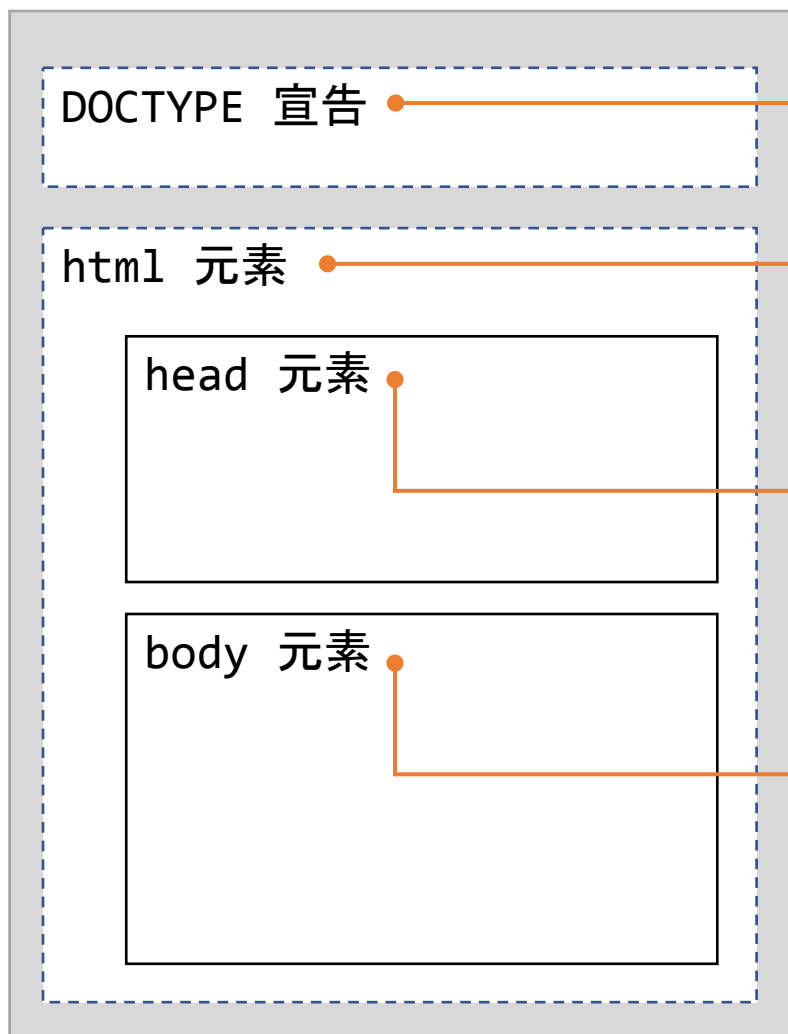
Module 2 – 網頁文件架構

2-1. HTML文件架構與宣告

2-2. HTML元素及lang屬性

2-3. head元素及子元素

HTML 文件架構



DOCTYPE 宣告

用來宣告使用的HTML版本，目前最新版本為 HTML5。

html 元素

是HTML文件最上層的元素。(根元素)

head 元素 - 網頁檔頭

網頁相關資訊。

包含頁面標記、文字編碼、內容概要、關連檔案...，此處內容不會在瀏覽器顯示出來。

body 元素 - 網頁內容

網頁的內容。

此元素中的內容都會在瀏覽器中顯示出來。

DOCTYPE 宣告

- 什麼是DOCTYPE宣告?

- DOCTYPE要寫在網頁的**第一行**告知瀏覽器，此網頁內所使用的HTML標籤是哪一個版本，瀏覽器就會使用宣告的版本來解析程式碼。

- HTML4 DOCTYPE 三種寫法

- 嚴格標準(Strict) - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
- 過渡標準(Transitional) - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
- 框架標準(Frameset) - `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">`

DOCTYPE 宣告

- HTML5 DOCTYPE 寫法

<!DOCTYPE html> - DOCTYPE習慣上用大寫表示。

```
<!DOCTYPE html>
<html id="atomic" lang="zh-Hant-TW" class="atomic my3columns ua-ed ua-win ua-10.0 ua-
<head>
  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <title>Yahoo奇摩</title><meta http-equiv="x-dns-prefetch-control" content="on"><1:

  <meta name="description" content="Yahoo奇摩提供即時新聞、氣象、購物、信箱、搜尋、政治
  <meta name="keywords" content="yahoo, yahoo奇摩, yahoo奇摩首頁, yahoo首頁, yahoo搜
  <meta property="og:title" content="Yahoo奇摩" />
  <meta property="og:type" content='website' />
  <meta property="og:url" content="http://tw.yahoo.com" />
  <meta property="og:description" content="Yahoo奇摩提供即時新聞、氣象、購物、信箱、搜
  <meta property="og:image" content="https://s.yimg.com/cv/apiv2/social/images/yaho
  <meta property="og:site_name" content="Yahoo奇摩" />
  <meta property="fb:app_id" content="399384933466174" />
  <meta name="format-detection" content="telephone=no" />
```

html元素

- 網頁的**根元素**，表示網頁文件開始的根結點。
- 子元素包含**<head>**及**<body>**。
- 常用屬性：
 - lang屬性：支援語言設定(Language)，幫助瀏覽器和搜尋引擎辨識網頁的語言。
- 網頁要呈現多種語言與lang屬性無關，要設定文字編碼。

```
<html lang="zh-hant-TW"> 繁體中文(台灣)
```

lang="語言-字體-地區"

語言 -> zh中文、en英文、ja日文、ko韓文

字體 -> hant繁體、hans簡體

地區 -> TW台灣、CN大陸、HK香港

head元素及子元素

- head元素

- 存放描述網頁的資料(metadata), 所有內容都不會顯示在該網頁(title、icon除外)。

- head元素的子元素

- meta元素: 設定HTML文件的相關資訊
- title元素: 網頁標題
- style元素: 樣式表宣告
- link元素: 外部檔案連結

```
<head>  
  <meta charset="UTF-8">  
  <title>網頁標題</title>  
  <style> /* CSS語法 */ </style>  
  <link rel="stylesheet" href="style.css">  
</head>
```

meta元素

- meta元素中常用屬性

- charset屬性:

宣告網頁文件編碼, 放在<head>元素中第一行。(HTML5 新增編碼屬性)

- http-equiv屬性:

關於網頁的內容屬性資訊, 包含「自動重整/跳轉頁面 refresh」、

「宣告編碼 content-type」.....等等。

```
<meta charset="UTF-8" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<!-- 跳轉頁面 content='秒數' / content='秒數; url=網址' -->
<meta http-equiv="refresh" content="5;" />
<meta http-equiv="refresh" content="5; url=https://www.yahoo.com.tw" />
```

meta元素

- meta元素中常用屬性

- name屬性:

表示文件中相關資訊名稱, 包含「作者 author」、「編輯器 generator」、「描述 description」、「關鍵字 keywords」、「響應式網頁設計(RWD) viewport」.....等等。

- content屬性:

表示文件中資訊內容, 對應到name屬性。

```
<meta name="author" content="Eddie Wang" />
<meta name="generator" content="VS code" />
<meta name="description" content="iSpan養成班 - HTML課程" />
<meta name="keywords" content="HTML,CSS" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```


meta元素實際案例

pazzo官網

```
<!DOCTYPE html>
<html lang="zh-Hant-TW">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <title>PAZZO 生活好感衣著</title>
  <meta name="referrer" content="no-referrer-when-downgrade">
  <meta name="description" property="og:description" itemprop="description"
content="PAZZO 生活好感衣著，將回歸衣著最簡單純粹的質地，講求最基本的質感，由挑選好布料
開始傳遞幸福感的新生活哲學，與大家一起共享對衣著的用心堅持與自在有感生活，立刻線上挑選女
裝・男裝・飾品・配件，享受快速配送服務。IT IS THE ACCESSORY OF LIFE." />
  <meta property="fb:app_id" content="2381134424120320" />
  <meta property="fb:pages" content="102230776487685" />
  <meta property="og:locale" content="zh_TW" />
```

<https://www.pazzo.com.tw>

PAZZO 生活好感衣著

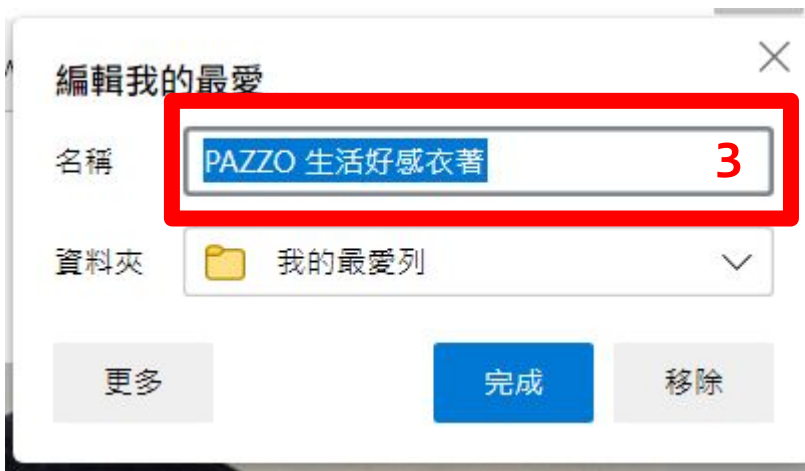
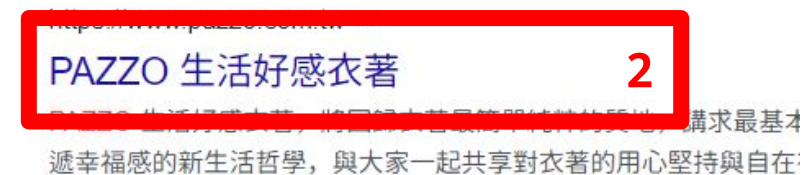
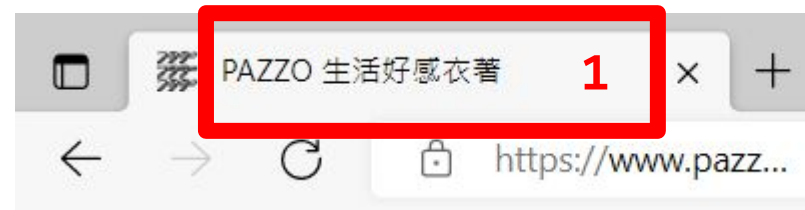
PAZZO 生活好感衣著，將回歸衣著最簡單純粹的質地，講求最基本的質感，由挑選好布料開始傳遞幸福感的新生活哲學，與大家一起共享對衣著的用心堅持與自在有感生活， ...

title元素

- title元素

- 設定網頁名稱，一個<head>元素中只能有一個<title>元素。
- 網頁名稱會顯示在以下幾個區域：
 1. 瀏覽器的頁籤
 2. 搜尋引擎的搜尋結果
 3. 書籤(我的最愛)

```
<!DOCTYPE html>
<html lang="zh-Hant-TW">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>PAZZO 生活好感衣著</title>
  <meta name="description" property="og:d
```



link元素

- icon設定

- 設定網頁縮圖，通常會顯示在瀏覽器的頁籤、書籤(我的最愛)。
 1. 製作ico圖檔([連結](#))並下載檔案，並將檔名修改為「favicon.ico」
 2. 將檔案放在「本機站台(專案資料夾)」的根目錄中



Module 3 – 網頁撰寫基礎

- 3-1. body元素及六大常用元素
- 3-2. 文字元素、區塊/行內元素及常用特殊字元
- 3-3. 圖片元素(image)及內容元素(figure/figcaption)
- 3-4. 清單元素
- 3-5. 超連結元素

body元素

- body元素
 - 主要用來呈現網頁內容，與網頁相關的內容都放在這裡面。
- body元素的子元素：
 - 文字元素。
 - 圖片元素。
 - 清單元素。
 - 表格元素。
 - 超連結元素。
 - 表單元素。
 -

常見六大網頁基礎元素

文字

標題: <h1> ~ <h6>

段落: <p>

換行:

分隔線: <hr>

行內容器:

區塊容器: <div>

圖片

清單

項目清單:

數字清單:

定義清單: <dl>

<dt></dt>

<dd></dd>

</dl>

超連結

<a>

表格

<table>

表單

<form>

<input>

<textarea>

文字元素 - 標題

- 標題 (headings) <h1> ~ <h6>
 - 預設樣式為粗體字且大小皆不同, <h1>最大 ~ <h6>最小。
 - h1元素最為重要, 一個網頁中只能有一個。
 - 原則上不可有階層亂跳或上下關係互換。
 - 標題元素為區塊元素。



文字元素 - 段落

- 段落 (paragraph) <p>
 - 用來作為文章段落區塊。
 - 字型大小與字體不會改變，但<p>元素包起來的地方會有預設邊距。
 - 段落元素為區塊元素，但子元素無法為區塊元素



▲德國權威研究團隊合成出最純淨的「LK-99」單結晶，證實LK-99不是超導體，消息一出讓韓國相關概念股股價雪崩。（圖／Pascal Puphal）

[NOWnews今日新聞] 今年七月底韓國團隊宣稱找到常溫超導體「LK-99」，但知名科學期刊《自然》（Nature）昨（16）日刊登了德國頂尖研究機構馬克斯—普朗克固體研究所（Max Planck Institute for Solid State Research）的論文，直言「LK-99不是超導體」，消息一出也讓韓國相關概念股應聲全倒。

區塊元素 / 行內元素

- 按照元素佔據位置方式分成

- 區塊元素(block)

特性:佔一整列, 前後有其他元素時會換行顯示, 與內容長短無關。

範例:<h1>~<h6>、<p>、<div>、<hr>、、.....。

內層:1. 可以包含區塊元素、行內元素;

2. 例外:段落元素(p)只能包含行內元素。

- 行內元素(inline)

特性:佔據位置就是內容本身, 內容越多空間越大, 超過單行的寬度才會換行。

範例:、
、、<a>.....。

內層:1. 行內元素只能包含行內元素。

2. 例外:超連結(a)可以包含任何元素, 但不能包自己。

文字元素 - 容器及換行

- 容器元素：

- 行內容器 ``
- 區塊容器 `<div>`

- 換行/分隔線元素：

- 換行 `
`
- 分隔線 `<hr />`

因此這個系列文正是希望將這些 React 的學習門檻、核心觀念、設計思維進行一個盤點並深入解析，希望能對於正在學習 React 或是已經有 React 的經驗但是苦惱於難以進一步掌握的朋友們有所幫助。

誰適合閱讀此系列文

這個系列文並不是一個 React 的新手教學或手把手的課程，因此不會把所有的 React API 都一一介紹一遍有什麼參數要怎麼調用，也不會有大量的專案範例帶著大家實作練習怎麼寫，而是針對其核心思想與最容易被誤解的觀念做透徹的解析，希望能夠幫助大家一次打破 React 的學習門檻與觀念誤解：

- 適合已經正在學習 React 中的人

文字元素 - 常用特殊字元

● 常用特殊字元

| 特殊字元 | 描述 | HTML原始碼 |
|------|------|---------|
| < | 小於 | < |
| > | 大於 | > |
| | 空白 | |
| & | 和 | & |
| " | 引號 | " |
| ® | 註冊商標 | ® |
| © | 版權 | © |
| TM | 商標 | ™ |

如果使用鍵盤上的「 < 」、「 > 」符號, 會被程式當作html語法解析, 所以要使用特殊自元表示。

HTML文件中「 空格 」需要使用此特殊字元, 空白鍵只會接收一次。

更多的字符 <https://tool.chinaz.com/tools/htmlchar.aspx>

文字元素 - 常用特殊字元 範例

- 在HTML中寫入一段HTML程式碼。

```
<code>
```

```
&lt;h2&nbsp;title=&quot;標題元素&quot;&gt;這是標題&lt;h2&gt;
```

```
</code>
```

```
<h2 title="標題元素">這是標題<h2>
```

- 在網頁最下方宣告版權。

```
<small>Copyright&copy;2022 iSpan-Eddie</small>
```

```
Copyright©2022 iSpan-Eddie
```

圖片元素

- 圖片元素

```

```

可以搭配以下五個屬性，設定圖片相關資訊：

- **src屬性** - 圖片的來源路徑，必需填寫。
- **alt屬性** - 1.當圖片失效時，可以顯示出指定文字。
2.幫助視障者理解網頁內容。
- **title屬性** - 滑鼠移到圖片上時出現說明文字。
- **width屬性/height屬性** - 圖片寬度與高度，單位為像素(px)。

```

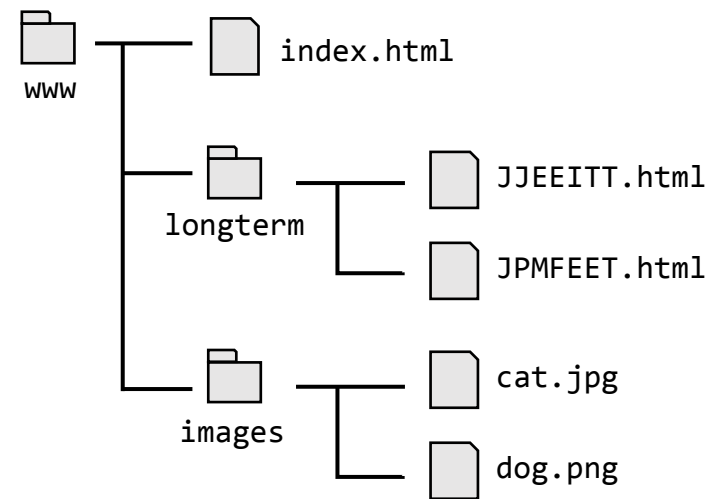
```

絕對路徑 & 相對路徑

插入圖片或是超連結時必須指定「路徑」，分為以下兩種：

- **絕對路徑** - 檔案位置是以「**http**開始的完整網址(URL)」，位置的最上層稱為根目錄。
 - 範例：以右方網站為例，dog.png 的絕對路徑為「`https://www.ispan.com.tw/images/dog.png`」
- **相對路徑** - 「當前檔案」與「目標檔案」之間的相對位置關係。
 - **同一階層路徑**：以「./目標檔案」表示，通常會省略「./」。
 - **下一階層路徑**：以「資料夾名稱/目標檔案」。
 - **上一階層路徑**：以「../目標檔案」。

`https://www.ispan.com.tw`



<figure> / <figcaption>

- figure 元素

- 表示一段獨立的內容。
- 經常放在其中的是圖片、相片、影片、引用文字、表格、程式碼...

- figcaption 元素

- <figcaption>元素是標題/說明，必須寫在<figure>元素內。
- 一個<figure>元素內只需要放一個<figcaption>元素即可。

<body>

 <figure>

 <figcaption>2022新北市國際紀錄片月主視覺。圖:新北市新聞局提供</figcaption>

 </figure>

</body>

清單元素

- 清單元素

- 內容是有上下排列、條列式的資料。
- 分為三種：
 1. 項目清單 (Unordered Lists)
 2. 序號清單 (Ordered Lists)
 3. 定義清單 (Definition Description)

清單元素 - 項目清單

- 項目清單 - 用來製作與順序無關的條列式資料，預設項目符號為黑點。

- 項目清單(Unordered Lists) : ` `

- 單一項目內容(List Item) : ` `

```
<body>
```

```
  <h2>iSpan課程</h2>
```

```
  <ul>
```

```
    <li>就業養成班</li>
```

```
    <li>在職進修班</li>
```

```
    <li>企業包班</li>
```

```
  </ul>
```

```
</body>
```

清單元素 - 序號清單

- 序號清單 - 用來製作有先後順序的條列式資料，預設序號符號為數字。

- 序號清單(Ordered Lists) : ` `

- 單一項目內容(List Item) : ` `

```
<body>
```

```
  <h2>程式語言熱門程度排名(StackOverflow,2022)</h2>
```

```
  <ol>
```

```
    <li>JavaScript</li>
```

```
    <li>HTML/CSS</li>
```

```
    <li>SQL</li>
```

```
  </ol>
```

```
</body>
```

清單元素 - 巢狀清單

- 在清單元素中再新增一個清單元素，要寫在單一項目內容元素中，語法如下：

iSpan課程

- 就業養成班
 - 1. 前端工程師
 - 2. 全端工程師
 - 3. 3D遊戲美術設計
- 在職進修班
- 企業包班

```
<h2>iSpan課程</h2>
<ul>
  <li>就業養成班
    <ol>
      <li>前端工程師</li>
      <li>全端工程師</li>
      <li>3D遊戲美術設計</li>
    </ol>
  </li>
  <li>在職進修班</li>
  <li>企業包班</li>
</ul>
```

清單元素 - 定義清單

- **定義清單** - 用來表示一系列的特殊名詞定義。
 - 定義清單 (Definition Lists) : `<dl> </dl>`
 - 項目內容 (Definition Term) : `<dt> </dt>`
 - 項目內容說明 (Definition Description): `<dd> </dd>`

```
<body>
  <h2>櫻花鉤吻鮭</h2>
  <dl>                                一個<dt>至少要搭配一個<dd>。
    <dt>學名</dt>
    <dd>Oncorhynchus masou formosanus</dd>
    <dt>俗名</dt>
    <dd>台灣鮭魚、台灣鱒、次高鱒、台灣山女</dd>
  </dl>
</body>
```

超連結元素 - a元素(anchor)

1. 外部連結 - 連結到其他網頁

`連結名稱`

常用屬性：

■ href 屬性

- 設定要連結的網頁路徑。
- 使用絕對路徑, 以「 http:// 」或是「 https:// 」開頭的完整網址。

■ target 屬性

- 設定開啟連結的方式
 - 1. `_self` => 直接跳轉畫面 (預設)
 - 2. `_blank` => 開啟新視窗

`<body>`

`Google`

`</body>`

超連結元素 - a元素(anchor)

2. 網站內部連結 - 連結到網站內指定網頁

`頁面名稱`

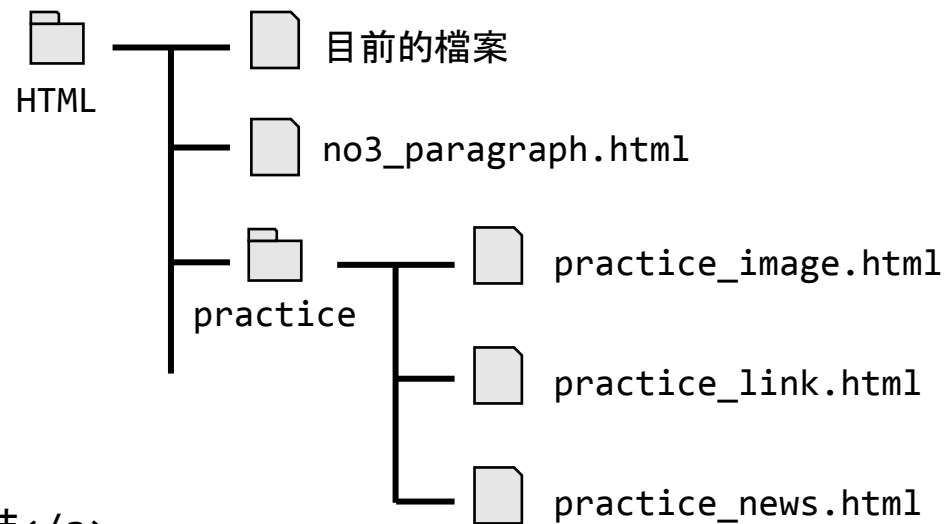
使用方式：

1. 確認專案資料夾結構。
2. 找到目前檔案位置。
3. 使用**相對路徑**找到目標檔案：
 - 同一階層：「./」，可以省略。
 - 上一階層：「../」。

`<body>`

`網站內部連結`

`</body>`



超連結元素 - a元素(anchor)

3. 網頁內部連結(錨點) - 連結到網頁內指定位置

`連結名稱`

使用方式：

1. 在指定的元素加上「id屬性」。
2. 將href的路徑填入「#id名稱」。
3. 點擊連結後就會移動到指定元素位置。

`<body>`

`網頁內部連結`

`<p>前段內容...</p>`

`<p style="background-color: pink;" id="dot1">目標區域</p>`

`<p>後段內容...</p>`

`</body>`

超連結元素 - a元素(anchor)

4. 電子郵件 - 傳送電子郵件到指定信箱

`連結名稱`

使用方式：

1. 在 href 屬性中以「mailto:收件人電子信箱」的語法即可。
2. 此方法可能會被利用來寄垃圾郵件或是惡意郵件，需要審慎評估。
3. 可以預先輸入**主旨 subject / 內容 body**

`連結名稱`

- subject=主旨
- body=內容
- 換行可以使用 %0a

`<body>`

`聯絡我們`

`請假`

`</body>`

超連結元素 - a元素(anchor)

5. 檔案下載 - 下載指定檔案(PDF/IMAGE/DOCX...)

```
<a href="檔案路徑" download>下載檔案</a>
```

使用方式：

1. 在<a>的開始標籤中增加download屬性。

```
<body>
```

```
  <a href="./PDF_example.pdf" download>下載檔案</a>
```

```
</body>
```

Module 4 – CSS3基礎知識

- 4-1. CSS的3種寫法(1) 行內樣式inline-style
- 4-2. CSS的基本語法
- 4-3. CSS的常用屬性-文字(顏色、大小、粗細)

CSS基礎介紹

- 什麼是CSS?
 - CSS(Cascading Style Sheets)階層樣式表, 是一種美化網頁的語言, 用來改變HTML元素預設的樣式。
- 在HTML中使用CSS語法的3種方式
 - 行內樣式 (Inline Style)
 - 內部樣式 (Internal Style Sheet)
 - 外掛樣式 (External Style Sheet)

行內樣式(Inline Style)

一、行內樣式(Inline Style)

- 位置寫在<開始標籤>內的style屬性。
- 行內樣式CSS語法：

<開始標籤 style="CSS屬性名稱:CSS屬性值;">內容</結束標籤>

- 可以同時給予多組CSS屬性, 只需用半形分號「 ; 」隔開。

style="CSS屬性名稱:CSS屬性值; CSS屬性名稱:CSS屬性值;"

<body>

<h2 style="color:red;">Google</h2>

</body>

CSS常用屬性

- 文字 - color, font-size, text-decoration...
- 清單 - list-style
- 邊框 - border
- 背景 - background, background-image...
- 邊界 - margin, padding
- 並排 - float, display
- 定位 - position
- 溢位 - overflow

文字色彩 color屬性

1. 文字色彩

color: 顏色名 || RGB值 || HEX碼 || ... ;

- 文字顏色預設為黑色。
- 常用表達方式(擇一):
 - 顏色名:brown
 - RGB值:rgb(165,42,42)
 - HEX碼:#a52a2a
- 參考網站:[色碼對應表](#)

| | | | |
|----|-------|------------------|---------|
| 褐色 | Brown | rgb(165, 42, 42) | #a52a2a |
|----|-------|------------------|---------|

顏色轉換

● HEX碼轉換成RGB

- HEX碼是16進位的方式表示顏色。
- 「 # 」之後的前兩碼代表紅色、中間兩碼代表綠色、最後兩碼代表藍色。
- 16進位：數字由0-9不變，而10-15則以a-f代替。
- 以褐色的HEX碼為例：紅色的數值為「 a5 』， $a*16 + 5 = 165$ 。

| | | | |
|----|-------|------------------|---------|
| 褐色 | Brown | rgb(165, 42, 42) | #a52a2a |
|----|-------|------------------|---------|

- 線上輔助工具：[色碼轉換器](#)

文字色彩 color屬性

- 範例

```
<body>  
  <h3>我是文章</h3>  
  <p style="color:red;">我是文章</p>  
  <p style="color:#a43363;">我是文章</p>  
  <p style="color:rgb(11, 25, 96)">我是文章</p>  
</body>
```

- 滑鼠移動到color屬性值的時候會出現調色盤。



1. 更換顏色的表達方式
2. 調整顏色的明度(垂直向)、彩度(水平向)
3. 調整透明度(1為不透明、0為透明)
4. 調整顏色的色相。

文字大小 font-size屬性

2. 文字大小

font-size: 絕對長度 || 相對長度 || 絕對尺寸 || 相對尺寸;

- 文字預設大小為16px。(不包含標題元素)
- 常用表達方式(擇一):

| | |
|------|--|
| 絕對長度 | 像素 px, 英吋 in, 公分 cm, 公釐 mm, 點 pt, Pica pc ... |
| 相對長度 | %, em, rem, vw, vh |
| 絕對尺寸 | xx-small, x-small, small, medium, large, x-large, xx-large |
| 相對尺寸 | larger, smaller |

補充: [一次搞懂 CSS 字體單位](#)

文字大小 font-size屬性

- 範例

```
<body>  
  <h3>文字大小</h3>  
  <span style="font-size: 20px">apple</span>  
  <span style="font-size: 70%; color: blue">apple</span>  
  <span style="font-size: 2em; color: navy">apple</span>  
  <span style="font-size: 16px; color: red">apple</span>  
</body>
```

- 比例單位中「 % 」、「 em 」與父層的文字大小有關。

文字粗細 font-weight屬性

3. 文字粗細

font-weight: 數值 || 絕對粗細 || 相對粗細

- 數值: 100(細)~900(粗)
- 絕對粗細: normal(相當於數值400)、bold(相當於數值700)
- 相對粗細: bolder、lighter(以目前文字粗細當作基準)

● 範例

```
<body>
  <h3>文字粗細</h3>
  <p style="font-weight:lighter;">資展國際</p>
  <p style="font-weight:200;">資展國際</p>
  <p style="font-weight:normal;">資展國際</p>
  <p style="font-weight:bolder;">資展國際</p>
</body>
```

Module 5 – CSS3基礎知識

- 5-1. CSS的3種寫法(2) 內部樣式表 Internal Style Sheet
- 5-2. 標籤選擇器 - Type Selector
- 5-3. 類別選擇器 - Class Selector
- 5-4. CSS的常用屬性-文字(字型、樣式、大小寫轉換)
- 5-5. CSS的常用屬性-文字(文字裝飾)

內部樣式表(Internal Style Sheet)

二、內部樣式表(Internal Style Sheet)

- 撰寫位置在<head>元素中的<style>元素內。
- 將HTML及CSS兩種語言分開寫, 更好管理也更易維護。
- CSS語法:

```
<head>
  <style>
    /* 這是註解 */
    選擇器 {
      CSS屬性名稱:CSS屬性值;
    }
  </style>
</head>
```

內部樣式表的寫法

- 內部樣式表的CSS語法

```
<style>  
    選擇器 { CSS屬性名稱:CSS屬性值; }  
</style>
```

- 透過選擇器(Selector)找到目標的元素。
- CSS語法寫在大括號「 { } 」中。
- 每組屬性名稱及屬性值中間使用半形冒號「 : 」隔開。
- 每一組屬性設定之間必須用半形分號「 ; 」隔開。
- 註解的表示方式為「 /* 這裡面是已經被註解的內容 */ 」。

CSS選擇器

- 什麼是CSS選擇器？

- CSS是要修改HTML元素預設樣式，要透過選擇器找到目標元素。
- 常用的選擇器如下：
 1. 標籤選擇器 - Type Selector
 2. 類別選擇器 - Class Selector
 3. ID選擇器 - ID Selector
 4. 通用選擇器 - Universal Selector

- 更多選擇器

- 可以參考w3官方文件：<https://www.w3.org/TR/selectors/>

選擇器 { CSS屬性名稱:CSS屬性值; }

標籤選擇器

- 標籤選擇器 – Type Selector

- 使用標籤來控制HTML中所有相同標籤名稱的元素。

`標籤 { CSS屬性名稱: CSS屬性值; }`

- 範例

```
<style>
```

```
    h1 { color: red; }
```

```
    p { color: orange; font-size: 24px; }
```

```
    h2 { color: red; }
```

```
</style>
```

```
<body>
```

```
    <h1>apple</h1>
```

```
    <p>orange</p>
```

```
    <p>mango</p>
```

```
    <h2>banana</h2>
```

```
</body>
```


CSS語法的整理

- 如果有多個選取器使用相同CSS屬性設定，可以使用半形逗號「 , 」隔開。

```
<style>
  h1 { color: red; }
  p  { color: orange; font-size: 24px; }
  h2 { color: red; }
</style>
```

可以寫成：

```
<style>
  h1,h2 { color: red; }
  p  { color: orange; font-size: 24px; }
</style>
```

類別選擇器

- 類別選擇器 – Class Selector
 - 在HTML開始標籤中加入Class屬性。
 - 使用時要在選擇器前面加上點號「 . 」。

`.class選擇器 { CSS屬性名稱: CSS屬性值; }`

`<tag class="class選擇器">內容</tag>`

- 範例

```
<style>
    .fruit1 { color: orange; font-size: 24px; }
    .fruit2 { color: yellow; font-size: 30px; }
</style>
```

```
<body>
    <p class="fruit1">orange</p>
    <p class="fruit2">mango</p>
</body>
```

類別選擇器 - 指定用法

- 只讓特定標籤套用

- 在類別選擇器前面加上標籤名稱。
- 標籤名稱及class選擇器中間不能有空格。

標籤名稱.class選擇器 { CSS屬性名稱: CSS屬性值; }

<tag class="class選擇器">內容</tag>

- 範例 範例中CSS選擇器指定只有是<p>元素才能使用名為fruit2的樣式;所以<h3>標籤不會有效果。

```
<style>
  .fruit1 { color: orange; font-size: 24px; }
  p.fruit2 { color: yellow; font-size: 30px; }
</style>
```

```
<body>
  <p class="fruit1">orange</p>
  <p class="fruit2">mango</p>
  <h3 class="fruit2">mango</h3>
</body>
```

類別選擇器 - 綁定多個class選擇器

- 一個元素可以綁定多個class選擇器
 - 每個class選擇器名稱要以半形空格分開。

```
<tag class="class選擇器">內容</tag>
```

- 範例

```
<style>
```

```
.red { color: red; }
```

```
.bigword { font-size: 50px; }
```

```
.bold { font-weight: 900; }
```

```
</style>
```

```
<body>
```

```
<h4 class="red bold bigword">紅色大寫粗體字</h4>
```

```
</body>
```

文字字型 font-family屬性

4. 文字字型

```
font-family: '新細明體', '標楷體', '細明體' ...
```

- 僅限於電腦中已安裝的字體。
- 瀏覽器預設為正黑體。
- 若使用者裝置中沒有前者字體, 就會使用下一個字體。

● 範例

```
<style>
```

```
    .font1 {
```

```
        font-family: '標楷體';
```

```
    }
```

```
</style>
```

```
<body>
```

```
    <h3>文字字型</h3>
```

```
    <p>我是瀏覽器預設字體(正黑體)</p>
```

```
    <p class="font1">我是標楷體</p>
```

```
</body>
```

文字字型進階設定(一) @font-face

- @font-face — 提供伺服器的字型供網頁下載使用, 語法如下:

```
@font-face {  
    src: url('fonts/Cursive_Script.ttf');  
    font-family: 'Cursive_Script';  
}
```

- src: 指定字型路徑。
- font-family: 命名字型名稱。

- 指定標籤套用:

```
.font2 { font-family: 'Cursive_Script'; }
```

```
<p class="font2">內部字型</p>
```

- font-family: 與@font-face設定中名稱相同即可使用。

文字字型進階設定(二) 外部字體

- 使用外部字體：讓使用者連結至提供字型的網站

- Google Fonts <https://fonts.google.com>

- 引入方法(一):

- 使用<link>元素, 就像載入CSS的做法一樣。
- 放在<style>元素之前。

```
<link rel="stylesheet" href="外部字體路徑">
```

- 引入方法(二):

- @import寫在style元素裡面**第一行**開始的位置。

```
<style>  
    @import url("外部字體路徑");  
</style>
```

文字樣式 font-style屬性

5. 文字樣式

```
font-style: normal || italic || oblique;
```

- normal:預設樣式, 正常字體。
- italic:斜體。
- oblique:透過數學演算的方式將正常字體傾斜一個角度。

● 範例

```
<style>
```

```
.font4 { font-style: normal; }
```

```
.font5 { font-style: italic; }
```

```
.font6 { font-style: oblique; }
```

```
</style>
```

```
<body>
```

```
<h3>文字樣式</h3>
```

```
<p class="font4">正常字體</p>
```

```
<p class="font5">斜體</p>
```

```
<p class="font6">斜角體</p>
```

```
</body>
```


文字大小寫轉換 text-transform屬性

6. 大小寫轉換(英文字)

```
text-transform: none || capitalize || uppercase || lowercase;
```

- none:預設值, 無轉換。
- capitalize:單字第一個字母大寫
- uppercase:全部大寫
- lowercase:全部小寫

● 範例

```
<style>
  .font7{ text-transform: capitalize; }
  .font8{ text-transform: uppercase; }
  .font9{ text-transform: lowercase; }
</style>
```

```
<body>
  <h3>文字大小寫轉換</h3>
  <p>applE samsunG</p>
  <p class="font7">applE samsunG</p>
  <p class="font8">applE samsunG</p>
  <p class="font9">applE samsunG</p>
</body>
```

文字裝飾 text-decoration-line 屬性

7. 文字裝飾(線條)

```
text-decoration-line: none || underline || overline || line-through;
```

- 設定文字裝飾線條, 包含無線條、底線、頂線、刪除線。

● 範例

```
<style>
```

```
.underline { text-decoration-line: underline; }
```

```
.overline { text-decoration-line: overline; }
```

```
.linethrough { text-decoration-line: line-through; }
```

```
</style>
```

```
<body>
```

```
<h3>文字裝飾(線條)</h3>
```

```
<h2 class="underline">Java班</h2>
```

```
<h2 class="overline">Java班</h2>
```

```
<h2 class="linethrough">Java班</h2>
```

```
</body>
```

文字裝飾 text-decoration-style 屬性

8. 文字裝飾(線條樣式)

```
text-decoration-style: solid || double || dotted || dashed || wavy;
```

- 設定該線條的樣式, 包含實線、雙線、點線、虛線、波浪線。

● 範例

```
<style>
```

```
  .solid { text-decoration-style: normal; }
```

```
  .double { text-decoration-style: double; }
```

```
  .wavy { text-decoration-style: wavy; }
```

```
</style>
```

```
<body>
```

```
  <h3>文字裝飾(線條)</h3>
```

```
  <h2 class="underline wavy">Java班</h2>
```

```
  <h2 class="overline double">Java班</h2>
```

```
  <h2 class="linethrough solid">Java班</h2>
```

```
</body>
```

文字裝飾 text-decoration-color 屬性

9. 文字裝飾(線條顏色)

text-decoration-color: 色彩;

- 設定文字裝飾線條之顏色。

● 範例

```
<style>
    .red { text-decoration-color: red; }
    .blue { text-decoration-color: blue; }
    .green { text-decoration-color: green; }
</style>
```

```
<body>
    <h3>文字裝飾(線條)</h3>
    <h2 class="underline wavy red">Java班</h2>
    <h2 class="overline double blue">Java班</h2>
    <h2 class="linethrough solid green">Java班</h2>
</body>
```

文字裝飾(速記) text-decoration屬性

10. 文字裝飾速記

text-decoration: 顏色 + 線條 + 樣式;

- 屬性值沒有順序之分，中間以半行空白分隔即可。
- 不顯示文字線條，可以直接使用「 text-decoration: none; 」。

● 範例

```
<style>
    .all { text-decoration: red underline wavy; }
    .none { text-decoration: none; color: pink; }
</style>

<body>
    <h2 class="all">Java班</h2>
    <a href="https://www.google.com" class="none">Google</a>
</body>
```

文字行距 line-height屬性

11. 文字行高

line-height: 長度單位 || 比例單位 ... ;

- 增加行高會使文字更易閱讀。
- 建議使用「 em 」為單位。

● 範例

```
<style>
  p { line-height: 2em; }
</style>

<body>
  <p>lorem</p>
</body>
```

lorem

8px

16px

8px

文字水平對齊 text-align屬性

12. 文字水平對齊

```
text-align: left || center || right || justify ... ;
```

- left: 文字靠左。
- center: 文字水平置中。
- right: 文字靠右。
- justify: 文字左右對齊。(最後一行不會對齊)
 - 如需最後一行也左右對齊, 可以使用text-align-last屬性。

● **範例**

```
<style>  
  p { text-align: center; }  
</style>
```

文字垂直對齊 vertical-align屬性

13. 文字垂直對齊

```
vertical-align: top || middle || bottom ... ;
```

- 只對表格儲存格<td>有效, <div>沒有作用。
- top:文字對齊上方。
- middle:文字垂直置中。
- bottom:文字對齊下方。

文字垂直對齊

- 讓文字在<div>中垂直對齊

方式一：使用line-height

只適用於單行文字且沒有使用其他元素。

```
<style>
.div1 {
    border: 1px solid red;
    line-height: 100px;
    height: 100px;
    text-align: center;
}
</style>
<div class="div1">文字垂直置中</div>
```

方式二：使用padding

可以是多行文字且使用其他元素。

```
<style>
.div2 {
    border: 1px solid red;
    padding: 50px 0;
    text-align: center;
}
</style>
<div class="div2">文字垂直置中</div>
```

Module 6 – CSS3基礎知識

6-1. ID選擇器 - ID Selector

6-2. 通用選擇器 - Universal Selectors

6-3. CSS的3種寫法(3) 外掛樣式表 External Style Sheet

ID選擇器

- ID Selector – ID選擇器
 - 在HTML開始標籤中加入ID屬性, ID必須是唯一值(unique)。
 - 使用時要在選擇器前面加上井號「 # 」。

#id屬性值 { CSS屬性名稱: CSS屬性值; }

```
<head>                                <body>
    <style>                            <h1 title="搜尋引擎" id="theH1">Google</h1>
        #theH1 {                      <h2>Yahoo</h2>
            color: red;                </body>
        }
    </style>
</head>
```

ID選擇器

- ID只能是唯一值的原因？

- JavaScript使用id修改樣式時，例如修改文字顏色：

```
<body>
  <h1 title="搜尋引擎" id="theH1">Google</h1>
  <h2 id="theH1">Yahoo</h2>
  <script>
    document.getElementById("theH1").style.color = "blue";
  </script>
</body>
```

- JavaScript使用id控制元素，只會抓到第一個，無法改變其他元素。
- 如需同時修改多個元素的樣式，建議使用class選擇器。

通用(全域)選擇器

- Universal Selectors – 通用(全域)選擇器
 - 套用到HTML中所有的元素上。

```
* { CSS屬性名稱: CSS屬性值; }
```

```
<head>  
  <style>  
    * {  
      font-size: 20px;  
    }  
  </style>  
</head>
```

CSS權重

- 以下四個元素的顏色會是什麼？

```
<style>
  * { color: yellow; }
  h1 { color: red; }
  .font1 { color: blue; }
  #font2 { color: green; }
</style>

<body>
  <h1 class='font1'>CSS選擇器權重</h1>
  <h1 class='font1' id="font2">CSS選擇器權重</h1>
  <h1 id="font2" style="color: pink;">CSS選擇器權重</h1>
  <h1>CSS選擇器權重</h1>
</body>
```

CSS權重

CSS權重

所謂的權重就是指 css 的優先權，例如：

1. 相同權重但是後寫的css屬性可以覆蓋先寫的css屬性。
2. 當兩個選擇器同時作用在一個元素，權重高的選擇器優先生效。

Inline Style > Id > Class > Element > *

建議使用class選擇器修改樣式，id選擇器保留給JavaScript使用。

外掛樣式表(External Style Sheet)

三、外掛樣式表(External Style Sheet)

- 優點: 多個網頁可以共用同一個css, 方便修改及管理。

```
<head>  
  <!-- 方法一 link元素 -->  
  <link href="CSS檔案的路徑" rel="stylesheet">  
</head>
```

- 將內部樣式表剪下, 建立外部css檔案並貼上css語法。

```
<head>  
  <link rel="stylesheet" href="style/first.css">  
</head>
```


外掛樣式表(External Style Sheet)

三、外掛樣式表(External Style Sheet)

- 優點: 多個網頁可以共用同一個CSS, 方便修改及管理。

```
<head>
  <!-- 方法二 style元素 -->
  <style>
    @import url(CSS檔案的路徑);
  </style>
</head>
```

```
<head>
  <style>
    @import url(style/first.css);
  </style>
</head>
```

CSS 總結

● CSS三種寫法：

行內樣式 inline Style	內部樣式表 Internal Style Sheet	外掛樣式表 External Style Sheet
-------------------	----------------------------	----------------------------

- 建議使用外掛樣式表，不同網頁可以重複使用且更方便修改。
- 如有需要單獨修改，才會使用內部樣式表。
- 最不建議使用行內樣式，權重太高不易修改。

● CSS選擇器、權重：

選擇器	全域	標籤	class	id	行內樣式
使用方式	* { }	tag { }	.class值 { }	#id值 { }	<tag style=" ">
權重	<div> <div>最低</div> <div>←————→</div> <div>最高</div> </div>				

- 建議使用class選擇器。（ id選擇器不建議使用，留給JavaScript控制 ）
- 相同權重的選擇器之下，後寫的樣式會覆蓋前者。

Module 7 – CSS3基礎應用

7-1. 與超連結有關的選擇器

7-2. 清單樣式的屬性

超連結相關選擇器

- 超連結樣式透過下面四個選擇器修改。

- `a:link` 未瀏覽的超連結
- `a:visited` 已瀏覽的超連結
- `a:hover` 滑鼠移至超連結上
- `a:active` 滑鼠點選的超連結

```
<style>
```

```
  a:link { text-decoration:none; }
```

```
  a:visited { color:red; }
```

```
  a:hover { text-decoration: underline; color:green; }
```

```
  a:active { color:yellow; }
```

```
</style>
```

清單符號 list-style-type屬性

● 清單符號

`list-style-type: none || disc || circle ...;`

- 改變清單的預設符號，常用清單符號屬性：

•	實心圓	disc
○	空心圓	circle
■	實心方形	square
1、2、3.....	阿拉伯數字	decimal
01、02、03.....	有0開頭的10進位數字	decimal-leading-zero
a、b、c.....	小寫英文字母	lower-alpha
A、B、C.....	大寫英文字母	upper-alpha
i、ii、iii.....	小寫羅馬字母	lower-roman
I、II、III.....	大寫羅馬字母	upper-roman

清單符號(圖片) list-style-image屬性

● 清單符號(圖片)

```
list-style-image: url(...);
```

- 清單符號可以使用圖片代替。
- 此屬性會覆蓋清單符號。

```
<head>
  <style>
    ul {
      list-style-type:none;
      list-style-image:url(xxx.jpg);
    }
  </style>
</head>
```

```
<body>
  <ul>
    <li>就業養成班</li>
    <li>在職進修班</li>
    <li>企業包班</li>
  </ul>
</body>
```

清單符號位置 list-style-position 屬性

● 清單符號位置

list-style-position: inside || outside ;

- 變更清單符號的起始位置，預設為 outside 。
- 修改整個清單位置時可以使用此屬性。

```
<head>
  <style>
    .inside { list-style-position: inside; }
    .outside { list-style-position: outside; }
    .orange { background-color: orange; }
    .yellow { background-color: yellow; }
  </style>
</head>
```

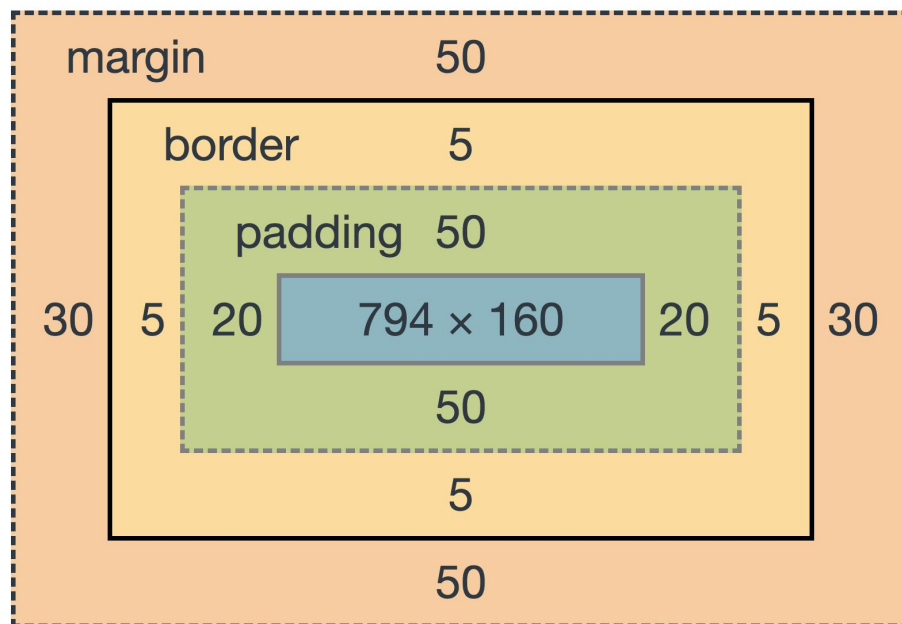
```
<body>
  <ul class="inside yellow">
    <li class="orange">就業養成班</li>
    <li>在職進修班</li>
    <li>企業包班</li>
  </ul>
  <ul class="outside yellow">
    <li class="orange">就業養成班</li>
    <li>在職進修班</li>
    <li>企業包班</li>
  </ul>
</body>
```

Module 8 – 方塊模型

- 8-1. 認識方塊模型(Box Model)
- 8-2. float、clear樣式
- 8-3. 彈性盒子 flex box

方塊模型 Box Model

- 什麼是方塊模型？
 - 方塊模型主要由四個部分主成，由內而外分別是「內容 Content」、「內邊距 Padding」、「邊框 Border」和「外邊距 Margin」。



內容寬度 width屬性

● 內容

width: 長度單位 || 百分比 ...;

- 內容為元素內的文字、圖片、影片.....等等。
- 可以藉由css屬性設定內容的寬度及高度。
- 寬度屬性:width。
- 高度屬性:height。

● 範例

```
img { width: 300px; border: 2px solid blue; }
```

```

```

```

```

```

```

```

```

邊框樣式 border屬性

● 邊框 border

border: 寬度 + 樣式 + 顏色 ;

- 設定元素的邊框, 由三個邊框相關屬性集合而成, 分別是「 寬度border-width屬性」、「 樣式border-style屬性」、「 顏色border-color屬性」。
- 寬度及樣式必須給予數值才會呈現。
- 包含以下屬性及屬性值:
 1. border-width: thick(厚) | medium(普通) | thin(薄) | 長度單位 ;
 2. border-style: none(無) | solid(實線) | double(雙線) | dotted(點線)...;
 3. border-color: 顏色名 | RGB值 | HEX碼 | ...;

```
<head>
  <style>
    .border1 { border: 2px solid orange; }
  </style>
</head>
```

邊框圓角 border-radius屬性

- 邊框圓角 border-radius

border-radius: 長度單位 || 百分比 ...;

- 設定元素的邊框圓角, 可以提供四邊圓角不同的數值:

1. 單一數值(四角相同)

```
border-radius: 10px;
```

2. 兩個數值(左上&右下 + 右上&左下)

```
border-radius: 10px 50px;
```

3. 四個數值(左上 + 右上 + 右下 + 左下)

```
border-radius: 10px 30px 60px 120px;
```

內邊距 padding屬性

● 內邊距

padding: 長度單位 || 百分比 ...;

- 內邊距為元素內容及邊框之間的距離。
- 背景顏色會依據內容而定。
- 四邊的內邊距可以給予相同/不同的數值。

● 範例

```
img { width: 300px; border: 2px solid blue; background-color: pink; }  
.image1 { padding: 20px; } /* 單一數值 = 同時設定四邊; */  
.image2 { padding: 20px 30px; } /* 兩個數值 = 上下邊 左右邊; */  
.image3 { padding: 20px 30px 10px; } /* 三個數值 = 上邊 左右邊 下邊; */  
.image4 { padding: 20px 30px 10px 40px; } /* 四個數值 = 上邊 右邊 下邊 左邊; */
```

外邊距 margin屬性

● 外邊距

margin: 長度單位 || 百分比 ...;

- 外邊距就是元素與其他元素的距離。
- 背景顏色為透明。
- 四邊的外邊距可以給予相同/不同的數值。

● 範例

/* 單一數值 = 同時設定四邊; */

```
.image1 { padding: 20px; margin: 20px; }
```

/* 兩個數值 = 上下邊 左右邊; */

```
.image2 { padding: 20px 30px; margin: 20px 30px; }
```

/* 三個數值 = 上邊 左右邊 下邊; */

```
.image3 { padding: 30px 20px 10px; margin: 30px 20px 10px; }
```

/* 四個數值 = 上邊 右邊 下邊 左邊; */

```
.image4 { padding: 30px 20px 10px 40px; padding: 30px 20px 10px 40px; }
```

更改方塊模型寬度計算方式 box-sizing屬性

- 更改方塊模型寬度計算方式

```
box-sizing: content-box || border-box ;
```

- content-box計算方式是將元素的內容寬度固定，並外加padding及border。
- border-box計算方式是將元素的寬度扣掉padding及border即為內容的寬度。

- 範例

```
img { width: 300px; border: 30px solid blue; padding: 10px; margin:20px; }  
.image3 { box-sizing: content-box; }  
.image4 { box-sizing: border-box; }  
  

```

瀏覽器預設外邊界參數

- 瀏覽器預設外邊界參數

- 各家瀏覽器四邊皆有預設外邊界參數。
- 例如: edge、chrome皆為8px。
- 可以使用通用選擇器修改, 內容即可貼齊邊線:

```
* { margin: 0; padding:0; }
```


置中對齊

- 將元素置中對齊

```
margin: auto;
```

- 所有元素預設由上而下、左而右排列；
- 要將元素置中對齊的方式：
 1. 該元素要設定寬度。
 2. 使用margin屬性平均分配左右寬度。

- 將元素中的文字置中對齊

```
text-align: left || center || right || justify ...;
```

- 使用text-align屬性設定文置對齊方式。
- 分別為「從左邊開始 left」、「置中對齊 center」、「從右邊開始 right」、「兩側對齊 justify」.....。

行內元素間的空白

- 行內元素間空白

- 行內元素並行時，中間會有空白，可以使用以下兩種方式解決。

1. 將元素寫在同一行，如有換行會產生空白字元。

```
<img ... /><img... />
```

2. 將父元素的文字大小設定為0。

```
父元素 { font-size: 0; }
```

- 範例

```
div { font-size: 0; }  
<div>  
      
      
      
      
</div>
```

浮動樣式 float 屬性

● 浮動樣式

```
float: none || left || right ;
```

- 用來將元素設為浮動樣式，形成文繞圖的效果。
- none: 預設，沒有浮動。
- left: 浮動到左邊。
- right: 浮動到右邊。

● 範例

```
.image3 { float: left; }  
.image4 { float: right; }  
<div>  
      
      
    <p>lorem*10</p>  
</div>
```

清除浮動元素 clear屬性

● 清除浮動元素

```
clear: none || left || right || both ;
```

- 用來解除浮動的元素，下方的元素不會跟著向上浮動。
- none:預設，後方元素跟著一起浮動。
- left:禁止左邊有浮動元素。
- right:禁止右邊有浮動元素。
- both:禁止左右兩邊有浮動元素。

● 範例

```
p { clear: both }  
<div>  
      
      
    <p>lorem*10</p>  
</div>
```

彈性盒子 flex-box

- 什麼是彈性盒子 flex-box?
 - 讓排版變得更容易的方式，具有主軸(main axis)、切軸(cross axis)、內容排序及對齊方式等等。
- 什麼是主軸 main axis?
 - 彈性盒子中的內容出發方向，有起點及終點。
- 什麼是切軸 cross axis?
 - 彈性盒子中單行內容超過容器大小，需要換行的方向，有起點及終點。



彈性盒子 flex-box display屬性

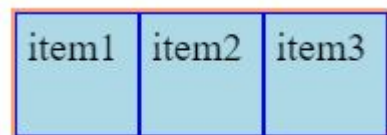
● 彈性盒子

```
display: flex || inline-flex ;
```

- 要創造一個彈性盒子，需要使用display屬性。
- flex: 將盒子內的元素並排呈現。



- inline-flex: 將盒子內容並排呈現，並且盒子的大小縮減成內容大小。



彈性盒子 flex-box display屬性 範例

/* 外層容器 */

```
.box {  
    border: 2px solid lightsalmon;  
}
```

/* 內層元素(內容) */

```
.item {  
    width: 50px;  
    height: 50px;  
    border: 1px solid blue;  
    background-color: lightblue;  
    font-size: 20px;  
    padding: 5px;  
}
```

/* 設定彈性盒子 flex-box */

```
.flex { display: flex; }  
.inline-flex { display: inline-flex; }
```

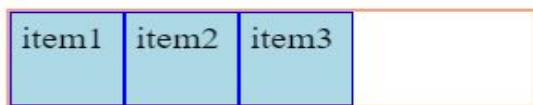
```
<div class="box flex">  
    <div class="item">item1</div>  
    <div class="item">item2</div>  
    <div class="item">item3</div>  
</div>  
  
<div class="box inline-flex">  
    <div class="item">item1</div>  
    <div class="item">item2</div>  
    <div class="item">item3</div>  
</div>
```

內容的出發方向 flex-direction屬性

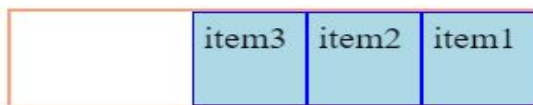
● 出發方向

`flex-direction: row || row-reverse || column || column-reverse ;`

- 設定flex-box中內容的出發方向。
- row: 並排呈現, 由左而右。(預設值)
- row-reverse: 並排呈現, 由右而左。
- column: 垂直呈現, 由上而下。
- column-reverse: 垂直呈現, 由下而上。



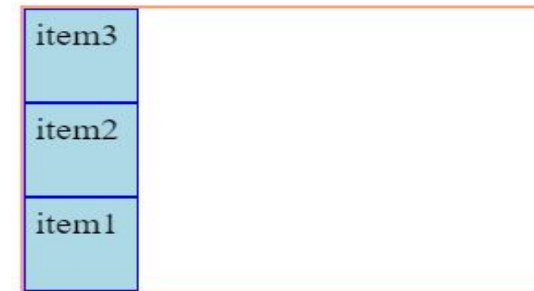
row



row-reverse



column



column-reverse

內容的出發方向 flex-direction屬性 範例

/ 外層容器(彈性盒子) */*

```
.box {  
    border: 2px solid lightsalmon;  
    display: flex;  
}
```

/ 內層元素(內容) */*

```
.item {  
    ...與上個範例相同 ...  
}
```

/ 內容的出發方向 */*

```
.row { flex-direction: row; }  
.row-reverse { flex-direction: row-reverse; }  
.column { flex-direction: column; }  
.column-reverse { flex-direction: column-reverse; }
```

```
<div class="box row">  
    <div class="item">item1</div>  
    <div class="item">item2</div>  
    <div class="item">item3</div>  
</div>
```


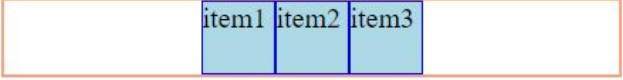

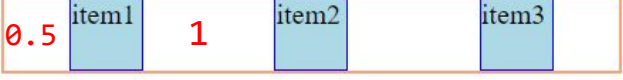
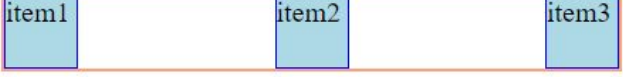
```
<div class="box ...">*4
```

對齊方式(主軸) justify-content屬性

● 對齊方式

justify-content: flex-start || center || flex-end ... ;

- 設定flex-box中內容的主軸對齊方式。

屬性值	功用	示意圖
flex-start	靠左	
center	置中	
flex-end	靠右	
space-around	左右留白、元素間平均分配	
space-between	左右不留白、元素間平均分配	

對齊方式(主軸) justify-content屬性 範例

/* 外層容器(彈性盒子) */

```
.box {
    border: 2px solid lightsalmon;
    display: flex;
    flex-direction: row;
}
```

/* 內層元素(內容) */

```
.item {
    ...與上個範例相同 ...
}
```

/* 主軸對齊方式 */

```
.flex-start { justify-content: flex-start; }
.flex-end { justify-content: flex-end; }
.center { justify-content: center; }
.space-around { justify-content: space-around; }
.space-between { justify-content: space-between; }
```

```
<div class="box flex-start">
    <div
        class="item">item1</div>
    <div
        class="item">item2</div>
    <div
        class="item">item3</div>
</div>
```

```
<div class="box ...">*5
```

Module 9 – 表格元素與樣式

9-1. 表格的基本架構與標籤

9-2. 表格範例

9-3. CSS 表格樣式

表格元素

● 表格元素

- 透過格子的格式呈現資訊。
- 表格是放在<table>標籤內，表格中包含以下元素：
 1. <caption>: 表格標題。
 2. <thead>: 欄位標題存放位置。
 3. <tbody>: 表格內容存放位置。
 4. <tfoot>: 註腳、總和...等等。
 5. <tr>: 表格列(table row)，裡面有一個或多個儲存格/標題。
 6. <th>: 標題儲存格(table heading)，用來呈現表格標題。
 7. <td>: 一般儲存格(table data)，用來存放表格資料。
- 如果沒有使用<tbody>，瀏覽器會自動產生。

<table>	表格標題
<caption></caption>	
<thead>	欄位標題
<tr>	
<th></th>	
</tr>	
</thead>	
<tbody>	表格內容
<tr>	
<td></td>	
</tr>	
</tbody>	
<tfoot>	表格註腳
<tr>	
<td></td>	
</tr>	
</tfoot>	
</table>	

表格元素 範例

- 表格樣式

- 預設沒有任何樣式，只有呈現資訊。

- 範例

```
/* 設定表格外框 */  
.table1 {  
    border: 3px solid #0072E3;  
    width: 400px;  
}  
/* 設定儲存格內框線 */  
.table1 td, .table1 th {  
    border: 1px solid gray;  
}
```

```
<table class="table1">  
    <thead>  
        <tr>  
            <th>標題1</th>*4  
        </tr>  
    </thead>  
    <tbody>  
        <tr>  
            <td>儲存格1</td>*4  
        </tr>  
        <tr>  
            <td>儲存格5</td>*4  
        </tr>  
    </tbody>  
</table>
```

表格<table> 常用樣式

- 邊框共用

```
border-collapse: separate || collapse ;
```

- 設定表格及儲存格是否要共用邊框。
- separate: 不共用邊框, 邊框之間會有格線。(預設值)
- collapse: 共用邊框, 邊框之間不會有空格線。

- 範例

```
.table1 {  
    border: 3px solid #0072E3;  
    width: 400px;  
    border-collapse: collapse;  
}
```

合併儲存格

- 表格欄位及標題數目要相同，但合併儲存格後要做刪減，可以在<th>或是<td>開始標籤加上以下屬性：

- colspan: 水平合併儲存格，屬性值填寫合併的格數(含自身)。

```
<th colspan="數值">內容</th>
```

- rowspan: 垂直合併儲存格，屬性值填寫合併的格數(含自身)。

```
<td rowspan="數值">內容</td>
```

標題	標題	標題	標題
儲存格1	子標題	子標題	
	儲存格2	儲存格3	儲存格4
	儲存格6	儲存格7	儲存格8

rowspan

colspan

合併儲存格 範例

- 垂直合併、水平合併、方塊合併：

```
<table class="table1">
  <thead>
    <tr>
      <th>標題</th><th>標題</th><th>標題</th><th>標題</th>
    </tr>
    <tr>
      <th colspan="2">子標題</th><!-- <th>標題</th> --><th colspan="2">子標題</th><!-- <th>標題</th> -->
    </tr>
  </thead>
  <body>
    <tr>
      <td rowspan="2">儲存格1</td><td>儲存格2</td><td rowspan="2" colspan="2">儲存格3</td><!-- <td>儲存格4</td> -->
    </tr>
    <tr>
      <!-- <td>儲存格5</td> --><td>儲存格6</td><!-- <td>儲存格7</td> --><!-- <td>儲存格8</td> -->
    </tr>
  </body>
</table>
```

欄位標題<thead>、表格內容<tbody> 常用樣式

- 改變欄位標題、表格內容的背景色

```
background-color: 顏色名 || RGB值 || HEX碼 || ... ;
```

- 將背景色下在<thead>、<tbody>元素上，可以直接改變其中表格列<tr>的顏色。

- 偶數列、奇數列背景色不同

- 使用CSS偽類選擇器，語法「 :nth-child() 」，括弧內可以填寫列數。
- 奇數列選擇器

```
tbody > tr:nth-child( 2n+1 | odd ) { CSS屬性; }
```

- 偶數列選擇器

```
tbody > tr:nth-child( 2n | even ) { CSS屬性; }
```

欄位標題<thead>、表格內容<tbody> 常用樣式

- 滑鼠移到<tbody>內的表格列<tr>或是儲存格<td>時會變顏色

```
tbody > tr:hover { CSS屬性; }
```

- 欄位標題<thead>中也有表格列<tr>, 所以要限制只有表格內容<tbody>中的表格列<tr>才會變換顏色。

- 範例

```
/* 設定欄位標題之背景色 */
thead { background-color: black; color: white; }
/* 設定表格內容之背景色 */
tbody { background-color: rgb(230, 230, 230); }
/* 設定奇數列之背景色 */
tbody > tr:nth-child(odd) { background-color: lightsalmon; }
/* 設定偶數列之背景色 */
tbody > tr:nth-child(even) { background-color: khaki; }
/* 設定滑鼠移至該表格列之背景色 */
tbody > tr:hover { background-color: deepskyblue; }
```

表格標題<caption>

- 表格標題放在<caption>元素中

<caption>表格標題</caption>

- <caption>中可以放入標題元素。

- 範例

```
<table>
  <caption>
    <h2>表格練習</h2>
  </caption>
  <thead> ..... </thead>
  <tbody> ..... </tbody>
</table>
```

Module 10 – 表單元素與樣式

10-1. 表單結構

10-2. 文字輸入控制項

10-3. 按鈕控制項

表單元素 - form

- 表單 (form)

`<form action="..." method="..." enctype="..."></form>`

- 所有表單控制項, 都要放在<form>元素當中。

- 重要屬性:

1. action屬性: 伺服器上等待接收資料的URL

2. method屬性: 傳送表單資料的方式

- get(預設): 透過網址列傳送資料, 沒有任何加密效果。
- post

3. enctype屬性: 資料傳送時的編碼設定

- application/x-www-form-urlencoded(預設, 使用 百分比編碼URL encoding方式處理資料)
- multipart/form-data(表單中有上傳檔案時需使用)
- text/plain(傳送純文字)

`<form action="#" method="get" enctype="multipart/form-data"></form>`

文字輸入控制項 - input

- 單行文字輸入

姓名：

```
<label for="...">標籤</label>
```

```
<input type="text" id="..." name="..." value="..." />
```

- 使用<input>元素, type屬性是指<input>的類型, 屬性值為text。
- <input>常搭配<label>元素使用, for屬性值要對應到<input>的id屬性值。
- 常用屬性
 - name: 讓伺服器得知目前的控制項代表什麼。
 - id: 對應到<label>的for屬性, 可以直接點擊標籤就focus到<input>。
 - value: 使用者輸入的資料, 如有填寫value屬性值會顯示在輸入框中。

文字輸入控制項 - input、textarea

● 密碼輸入

密碼：

```
<label for="...">標籤</label>
```

```
<input type="password" id="..." name="..." maxlength="..." />
```

- 使用<input>元素, type屬性是指<input>的類型, 屬性值為password。

● 多行文字輸入

```
<label for="...">標籤</label>
```

```
<textarea id="..." name="..." cols="..." rows="..." ></textarea>
```

- 使用<textarea>元素。
- 不支援value屬性。
- 常用屬性
 - cols: 元素的寬度, 必須為正數。
 - rows: 元素的高度, 必須為正數。

備註：

文字輸入控制項 - input、textarea 範例

```
<form action="#" method="get" enctype="multipart/form-data">
  <!-- 單行文字輸入 -->
  <label for="name">姓名:</label>
  <input type="text" id="name" name="myname" value="" placeholder="請輸入姓名">
  <!-- 密碼輸入 -->
  <label for="password">密碼:</label>
  <input type="password" id="password" name="mypassword" value="" maxlength="8">
  <!-- 多行文字輸入 -->
  <label for="textarea">備註:</label>
  <textarea name="memo" id="textarea" cols="30" rows="10"></textarea>
</form>
```

按鈕控制項 - input

- 送出按鈕

```
<input type="submit" value="..." />
```

- 表單預設送出按鈕，將type屬性設定為submit。
- 按鈕預設文字為「提交」，可以使用value屬性修改顯示文字。
- 點擊送出按鈕之後，會將資料傳至後端伺服器。(不是直接寫入資料庫)

- 清除按鈕

```
<input type="reset" value="..." />
```

- 表單預設清除按鈕，將type屬性設定為reset。
- 按鈕預設文字為「重設」，可以使用value屬性修改顯示文字。

按鈕控制項 - input

- 一般按鈕

`<input type="button" id="..." value="按鈕名稱" />`

`<button type="button" id="...">按鈕名稱</button>`

- 使用<input>元素, type屬性設定為button。
- 這兩種寫法沒有預設功能, 用來讓開發人員透過JavaScript去自訂功能。
- 常用屬性
 - disabled: 把控制項的功能關閉。(預設為enable)

按鈕控制項 - input 範例

```
<form action="#" method="get" enctype="multipart/form-data">
```

```
    . . . 文字輸入控制項程式碼 . . .
```

```
<!-- 送出按鈕 -->
```

```
<input type="submit" value="送出">
```

```
<!-- 清除按鈕 -->
```

```
<input type="reset" value="清除" >
```

```
<!-- 一般按鈕 -->
```

```
<input type="button" value="一般按鈕">
```

```
</form>
```

調整表單結構 範例

使用<div>元素包住每一組控制項。

```
<form action="#" method="get" enctype="multipart/form-data">
  <div class="q1">
    <label class="t1" for="name">姓名:</label>
    <input type="text" id="name" name="myname" value="" placeholder="請輸入姓名">
  </div>
  <div class="q1">
    <label class="t1" for="password">密碼:</label>
    <input type="password" id="password" name="mypassword" value="" maxlength="8">
  </div>
  <div class="q1">
    <label class="t1" for="textarea">備註:</label>
    <textarea name="memo" id="textarea" cols="30" rows="10"></textarea>
  </div>
  <div class="btn1">
    <input type="submit" value="送出">
    <input type="reset" value="清除" >
  </div>
</form>
```

```
/* 控制項盒子 */
.q1 {
  width: 450px;
  border-bottom: 2px dashed gray;
  margin: 20px;
  padding-bottom: 20px;
}

/* 按鈕對齊 */
.btn1 {
  width: 450px;
  margin: 20px;
  text-align: center;
}

/* 項目標題文字對齊 */
.t1 {
  float: left;
  width: 100px;
  text-align: right;
}
```

Module 11 – 表單元素與樣式

11-1. 選擇控制項

11-2. 檔案上傳

11-3. 表單群組化

選擇控制項 - radio、checkbox

- 單選控制項

性別： ☐ 男 ☐ 女

```
<input type="radio" name="..." value="..." />
```

- 單選控制項，將type屬性設定為radio。

- 複選控制項

興趣： ☐ 游泳 ☐ 旅遊 ☐ 畫畫 ☐ 追劇

```
<input type="checkbox" name="..." value="..." />
```

- 多選控制項，將type屬性設定為checkbox。

- 常用屬性

- name: 給予相同值才會被視為是同一個題目的選項。
- value: 目的是判斷使用者選擇什麼選項。
- checked: 網頁載入時，預先被選取的選項。

選擇控制項 - radio、checkbox 範例

1. 使用取代<label>的項目標題文字。
2. 將<input>及選項名稱包在<label>中也可以達成點擊選項名稱連動選項。

```
<form action="#" method="get" enctype="multipart/form-data">
  <div class="q1">
    <!-- 單選控制項, name屬性值相同才會被視為是同一題目。 -->
    <span class="t1">性別:</span>
    <label><input type="radio" name="gender" value="male">男</label>
    <label><input type="radio" name="gender" id="female" value="female">女</label>
  </div>
  <div class="q1">
    <!-- 多選控制項 -->
    <span class="t1">興趣:</span>
    <label><input type="checkbox" name="interest" value="swimming">游泳</label>
    <label><input type="checkbox" name="interest" value="traveling">旅遊</label>
    <label><input type="checkbox" name="interest" value="drawing">畫畫</label>
    <label><input type="checkbox" name="interest" value="netflix">追劇</label>
  </div>
</form>
```


選擇控制項 - select、option

- 下拉式選單

縣市：

```
<select id="..." name="...">
  <option value="..." >選項1</option>
  <option value="..." >選項2</option>
  <option value="..." >選項3</option> ...
</select>
```

- 下拉式選單使用<select>元素包住選項<option>元素。

- <select>常用屬性

- multiple: 選單如需複選功能，可以加上此屬性。

- <option>常用屬性

- selected: 網頁載入時，預先被選擇的項目。

選擇控制項 - select、option 範例

```
<form action="#" method="get" enctype="multipart/form-data">
  <div class="q1">
    <span class="t1">縣市:</span>
    <!-- 如需多選, 在<select>加入multiple屬性。【滑鼠+ctrl】點擊選項即可多選 -->
    <select name="city" id="">
      <option value="taipei">台北市</option>
      <option value="newtaipet">新北市</option>
      <option value="taoyuan">桃園市</option>
    </select>
  </div>
  <div class="q1">
    <label class="t1" for="address">地址:</label>
    <input type="text" name="address" id="address"/>
  </div>
</form>
```

檔案上傳控制項 - input

- 上傳檔案

上傳檔案： 沒有選擇檔案

```
<label for="...">標籤</label>  
<input type="file" name="..." />
```

- 範例

```
<form action="#" method="get" enctype="multipart/form-data">  
  <div class="q1">  
    <label for="" class="t1">上傳檔案:</label>  
    <input type="file" name="file1" />  
  </div>  
</form>
```

表單控制項群組 - fieldset、legend

- 表單控制項群組

<fieldset>

 <legend>群組名稱</legend>

 ... 表單內容 ...

</fieldset>

- 使用<fieldset>元素將表單控制項群組化。
- <legend>元素是用來設定群組名稱, 要放在<fieldset>元素內起始位置。

基本資料

姓名：

密碼：

表單控制項群組 - fieldset、legend 範例

```
<form action="#" method="get" enctype="multipart/form-data">
  <!-- 表單控制項群組化 -->
  <fieldset>
    <!-- 表單控制項群組標題 -->
    <legend>基本資料</legend>
    ... 姓名、密碼 ...
  </fieldset>
  <fieldset>
    <legend>地址</legend>
    ... 地址、完整地址 ...
  </fieldset>
  <fieldset>
    <legend>其他資料</legend>
    ... 性別、興趣、備註、上傳檔案 ...
  </fieldset>
</form>
```

```
/* fieldset 群組 */
fieldset {
    width: 500px;
    margin: 15px;
    border: 2px solid gray;
    border-radius: 10px;
}

/* legend 標題 */
legend {
    padding: 5px 15px;
    margin-left: 10px;
}
```

Module 12 – HTML5 新增input類型

- 12-1. 日期控制項
- 12-2. 文字輸入控制項
- 12-3. 其他控制項

日期控制項 - input

● 日期控制項

`<label for="...">標籤</label>`

`<input type="..." id="..." name="..." value="..." />`

■ 新增type屬性值如下：

- date:顯示日歷, 格式為YYYY-MM-DD。
- datetime-local:顯示日曆、時間, 格式為YYYY-MM-DDThh:mm。
- month:顯示年份、月份, 格式為YYYY-MM。
- time:顯示時間, 格式為hh:mm。
- week:顯示年份、第幾週, 格式為YYYY-W00。

■ 常用屬性：

- min/max:限制使用者可選擇範圍。
- value:設定預設顯示日期/時間。

date

datetime-local

month

time

week

日期控制項 - input 範例

```
<form action="#" method="get" enctype="multipart/form-data">
  <div class="box">
    <label for="date1" class="t1">日曆:</label>
    <input type="date" id="date1" name="date1" />
  </div>
  <div class="box">
    <label for="date2" class="t1">日曆及時間:</label>
    <input type="datetime-local" id="date2" name="date2" />
  </div>
  <div class="box">
    <label for="time" class="t1">時間:</label>
    <input type="time" id="time" name="time" />
  </div>
  <div class="box">
    <label for="month" class="t1">年/月份:</label>
    <input type="month" id="month" name="month" />
  </div>
  <div class="box">
    <label for="week" class="t1">年份/週:</label>
    <input type="week" id="week" name="week" />
  </div>
</form>
```

```
.box {
  width: 600px;
  margin-bottom: 10px;
}
.t1 {
  width: 100px;
  float: left;
  text-align: right;
}
```


文字輸入控制項 - input

- 文字輸入控制項

<label for="...">標籤</label>

<input type="..." id="..." name="..." />

- 新增type屬性值如下：

- email:郵件格式, 送出時會簡單驗證是否為郵件格式。
- url:網址格式, 送出時會驗證是否為http://開頭的網址。
- tel:手機模式會自動轉換數字鍵盤。
- search:部分瀏覽器在輸入框內最右側會出現刪除按鈕。

文字輸入控制項(手機模式) - input

type="text"

A mobile form with four input fields: Name, Phone, Email, and URL. The Name field is active, showing a text keyboard. The keyboard has a 'go' button. The status bar at the top shows the time 11:12 and battery level 63%.

type="tel"

A mobile form with four input fields: Name, Phone, Email, and URL. The Phone field is active, showing a numeric keyboard. The keyboard has a 'go' button. The status bar at the top shows the time 11:12 and battery level 63%.

type="email"

A mobile form with four input fields: Name, Phone, Email, and URL. The Email field is active, showing an email keyboard. The keyboard has a 'go' button. The status bar at the top shows the time 11:12 and battery level 63%.

type="url"

A mobile form with four input fields: Name, Phone, Email, and URL. The URL field is active, showing a URL keyboard. The keyboard has a 'go' button. The status bar at the top shows the time 11:12 and battery level 64%.

文字輸入控制項 - input 範例

```
<div class="box">
  <label for="email" class="t1">郵件:</label>
  <input type="email" id="email" name="email" />
</div>
<div class="box">
  <label for="tel" class="t1">手機:</label>
  <input type="tel" id="tel" name="tel" />
</div>
<div class="box">
  <label for="url" class="t1">網址:</label>
  <input type="url" id="url" name="url" />
</div>
<div class="box">
  <label for="search" class="t1">搜尋:</label>
  <input type="search" id="search" name="search" />
</div>
<div class="box">
  <input type="submit" value="送出" />
</div>
```

其他控制項 - input

- 其他控制項

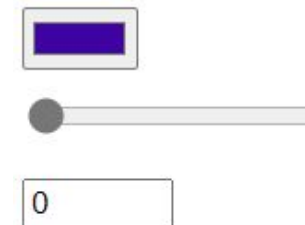
`<label for="...">標籤</label>`

`<input type="..." id="..." name="..." value="..." />`

- 新增type屬性值如下：

- color: 顯示顏色調色盤讓使用者選擇，格式為HEX碼。
- range: 可輸入數值，有控制鈕可以拖曳。
- number: 可以輸入數值。

- range及number會搭配範圍驗證屬性。



其他控制項 - input

- 範圍驗證屬性
 - max: 上限值。
 - min: 下限值。
 - step: 每次增加/減少多少數值。
 - value: 預設數值。
- 一般驗證屬性
 - maxlength: 最大值。
 - minlength: 最小值。
 - required: 必填項目。
 - pattern: 驗證使用者輸入資料的方式。

其他控制項 - input 範例

```
<div class="box">
  <label for="color" class="t1">顏色:</label>
  <input type="color" id="color" value="#456789" name="color" />
</div>
<div class="box">
  <label for="range" class="t1">控制條:</label>
  <input type="range" id="range" min="0" max="100" step="5" value="70" name="range" />
</div>
<div class="box">
  <label for="number" class="t1">數量:</label>
  <input type="number" id="number" min="0" max="100" step="5" value="70" name="number" />
</div>
```

加上手機驗證。

```
<div class="box">
  <label for="" class="t1">手機:</label>
  <input type="tel" pattern="^[0][9]\d{8}$" maxlength="10" name="tel" />
</div>
```

其他控制項 - input 範例

即時顯示控制條(range)的數值。

```
<div class="box">
  <label for="range" class="t1">控制條:</label>
  <input type="range" id="range" min="0" max="100" step="5" value="70" name="range" />
  <span id="rangeOutput"></span>
</div>
<script>
  range.addEventListener('input',()=>{
    rangeOutput.innerText = range.value;
  })
</script>
```

表單控制項常用屬性

● 表單控制項常用屬性

- placeholder: 文字輸入框為空時, 會顯示的提示訊息。
- autocomplete:
 - on: 自動記錄過去填寫內容。(預設)
 - off: 關閉。
- autofocus: 網頁載入時, 自動關注到指定輸入欄位。
- size:
 - 下拉式選單可視選項數量。
- multiple:
 - 下拉式選單複選。(滑鼠+ctrl鍵)
 - 檔案上傳。
- readonly: 唯讀, 可以讀取但無法改寫。(適用表單文字輸入控制項)

Module 13 – HTML元素

13-1. 內嵌框架 - iframe

13-2. HTML5 結構語意元素

內嵌框架 - iframe

- 內嵌框架

```
<iframe src="..." width="..." height="..."></iframe>
```

- 將其他網站內容嵌入到自己的網站中，不是每個網站都可以被嵌入。
- 常用屬性
 - src: 要被嵌入網頁的絕對路徑。
 - width/height: 設定內嵌框架的寬/高度。
 - frameborder: 內嵌框架的邊框，0為不顯示框線(預設)，1為顯示框線。
 - allowfullscreen: 允許全螢幕觀看。

內嵌框架 - iframe 範例(一)

<!-- 入口型網站(Google、Yahoo就無法被嵌入) -->

```
<iframe src="https://www.google.com" frameborder="0" width="600"  
height="400"></iframe>
```

```
<iframe src="https://www.ispan.com.tw" frameborder="0" width="600"  
height="400" allowfullscreen></iframe>
```



內嵌框架 - iframe 範例(二)

<!-- 嵌入地圖 -->

```
<iframe  
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3614.7988217416337!2d121.541319  
46537899!3d25.040900494167914!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x3442abd3  
7971c7cb%3A0x40ba641f27b6d4e3!2zMTA25Y-w5YyX5biC5aSn5a6J5Y2A5b6p6IiI5Y2X6Lev5LiA5q61Mzkw6J  
mf!5e0!3m2!1szh-TW!2stw!4v1669540387307!5m2!1szh-TW!2stw" width="600" height="450"  
style="border:0;" allowfullscreen="" loading="lazy"  
referrerpolicy="no-referrer-when-downgrade"></iframe>
```



內嵌框架 - iframe 範例(三)

<!-- youtube影片(有一些影片無法被嵌入) -->

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/5S8jvUaqg-Q"
title="YouTube video player" frameborder="0" allow="accelerometer; autoplay;
clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
```



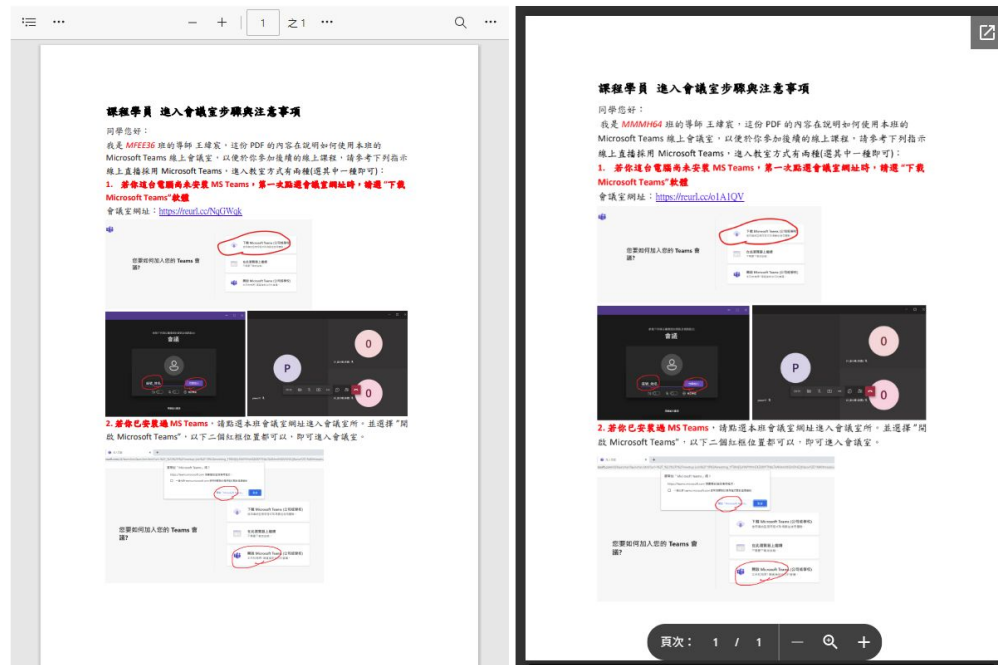
內嵌框架 - iframe 範例(四)

<!-- 嵌入PDF檔案 本地(伺服器) -->

```
<iframe src="./PDF_example.pdf" frameborder="0" width="600" height="800"></iframe>
```

<!-- 嵌入PDF檔案 Google 雲端硬碟 -->

```
<iframe src="https://drive.google.com/file/d/<file_id>/preview" frameborder="0"
width="600" height="800"></iframe>
```



內嵌框架 - iframe 範例(五)

<!-- Google 表單 -->

<iframe

src="https://docs.google.com/forms/d/e/1FAIpQLScs2xeHYzHPVPh3w-X7IuuTb98wEtXfckdus8Um7ic9P

DSi5w/viewform?embedded=true" width="640" height="800" frameborder="0" marginheight="0"

marginwidth="0">載入中...</iframe>

網頁設計入門 (2023)

資展國際股份有限公司 - 王緯宸

chenwei110102@gmail.com 切換帳戶

未共用的項目

作答時間: 30分鐘

繼續 清除表單

請勿利用 Google 表單送出密碼。

Google 表單 Google 並未認可或建立這項內容。

語意結構區塊

- 語意結構區塊 Semantic Layout Section

- 用處是讓網頁文件結構更明確。
- 這些元素沒有預設樣式, 要使用CSS修改樣式。
- 包含以下元素
 - `<section>`
 - `<article>`
 - `<nav>`
 - `<header>`
 - `<footer>`
 - `<main>`
 - `<aside>`

語意結構區塊 - section、article

- `<section>`

- 有明顯含義的區塊。
- 一般來說`<section>`中也會有自己的標題(h1-h6)。
- 如無明顯意義則可以使用`<div>`。

- `<article>`

- 內容本身是獨立且完整，例如部落格中的一篇文章、一則留言，或像論壇中的一個回覆等等。
- 一個頁面可以有多个`<article>`。

語意結構區塊 - nav、header、footer

- **<nav>**
 - 存放此網站或網頁的主要導覽連結，像是網站最上面的主菜單(menu)。
- **<header>**
 - 定義一個標題區塊，通常會包含標題元素、Logo、目錄、搜尋等等。
- **<footer>**
 - 定義一個頁尾或結尾區塊。
 - 通常會包含隱私權政策、版權等等。

語意結構區塊 - main、aside

- **<main>**

- 存放頁面主要資訊的區塊，每個頁面中只能有一個<main>。
- 通常不會放在<nav> 、<article> 、<aside> 、<header> 、<footer>當中。

- **<aside>**

- 此內容是跟主要內容沒直接關係的區塊，通常是作為頁面的額外資訊。
- 顯示額外資訊，廣告、其他連結、推薦文章或商品等等。
- 不一定是側邊位置才能使用。

HTML5 語意標籤主要架構

`<header>` 頁首

`<nav>` 導覽列

`<main>` 主要區塊

`<aside>` 側欄

`<section>` 區塊

`<article>` 文章區塊

`<footer>` 頁尾

Module 14 – CSS樣式屬性

14-1. 背景樣式

14-2. 背景樣式相關設定

14-3. display/visibility/overflow樣式

14-4. 定位 - position

背景樣式

- 背景顏色

`background-color: 顏色名 || RGB值 || HEX碼 || ... ;`

- 文字、超連結、表單、清單、表格等等元素皆可以設定背景顏色。

- 背景圖片

`background-image: none || url(圖片路徑) ;`

- 除了設定顏色之外，也可以透過此屬性設定背景為圖片。
- 預設會從左上角開始填滿指定元素的區塊。

```
<style>
  body {
    background-image: url(images/background1.jpg);
  }
</style>
```

背景重複 background-repeat屬性

● 背景重複

```
background-repeat: repeat || repeat-x || repeat-y || no-repeat ;
```

- 設定背景圖片是否要重複。
- repeat: 重複填滿整個區塊。
- repeat-x: 重複填滿X軸。
- repeat-y: 重複填滿Y軸。
- no-repeat: 不重複, 只顯示一張圖(原始大小)。

```
<style>  
  body {  
    background-repeat: no-repeat;  
  }  
</style>
```

背景尺寸 background-size屬性

● 背景尺寸

background-size: 寬度 + 高度 || cover || contain ;

- 可以設定背景圖片大小。
- 寬度+高度:
 - 可能破壞圖片比例。
 - 如果只有給予一個值，會預設為寬度並按照原始比例調整高度。
- cover: 將圖片整張填滿元素區塊，保持原始寬高比例、必要時會進行裁切。
- contain: 與cover屬性質相似，但差異在於不會裁切，保持完整圖片。

```
<style>
  body {
    background-size: cover;
  }
</style>
```


背景位置 background-position屬性

● 背景位置

background-position: 水平 + 垂直 ;

- 設定背景圖片的位置。
- 水平:left | center | right | 長度單位。
- 垂直:top | center | bottom | 長度單位。
- 如果只有給指一個值, 另一個值會自動設為center。

```
<style>
    .p1 {
        background-image: url(images/space.jpg);
        background-repeat: no-repeat;
        background-size: 300px;
        background-position: center;
    }
</style>
```

顯示設定 display屬性

● 顯示設定

display: none || block || inline || flex ... ;

- 設定元素的顯示方式。
- none: 不顯示, 且此元素不占任何空間。
- block: 顯示, 視為區塊, 前後緊鄰的元素將換行顯示。
- inline: 顯示, 視為行內, 前後緊鄰的元素會並排顯示。
- flex: 顯示, 視為彈性盒子。

```
<style>
  h1 {
    background-color: orange;
    display: none;
  }
</style>
<h1>我是h1標題</h1>
<h2>我是h2標題</h2>
```

顯示設定 visibility屬性

- 顯示設定

```
visibility: hidden || visible || collapse ;
```

- 設定元素的顯示方式。
- hidden: 不顯示, 會佔據空間。
- visible: 顯示。
- collapse: 顯示, 用於表格中。

```
<style>
  h1 {
    background-color: orange;
    visibility: hidden;
  }
</style>
<h1>我是h1標題</h1>
<h2>我是h2標題</h2>
```

顯示設定 overflow屬性

● 顯示設定

```
overflow: auto || scroll || hidden || visible ;
```

- 設定元素的顯示方式。
- auto: 內容有超出元素尺寸的方向會自動出現卷軸。
- scroll: 直接出現水平及垂直卷軸。
- hidden: 內容超出元素尺寸以外的部分會隱藏。
- visible: 全部顯示。(預設值)

```
<style>
    div {
        width: 300px;
        height: 400px;
        border: 1px solid red;
        overflow: auto;
    }
</style>

<div>
    <p>lorem*10</p>
</div>
```

定位方式 position屬性

● 定位方式

```
position: relative || absolute || fixed || static ;
```

- 設定元素於畫面中的定位方式。
- relative:相對位置, 元素的原點在原本位置。
- absolute:絕對位置, 不占空間, 下方元素會往上移動。原點參考父層的原點位置。
- fixed:固定位置, 直接固定在畫面上的指定位置, 卷軸滾動時不會移動。
- static:靜態位置, 依照程式碼原始排序來定位。

● 設定定位

```
top/bottom/left/right: 長度單位 || 比例單位 ;
```

- 設定元素與邊界的距離。

Thank you

補充教材

邊框圓角 border-radius

邊框單一圓角 border-X-X-radius屬性

- 邊框圓角 border-X-X-radius

border-X-X-radius: 長度單位 || 百分比;

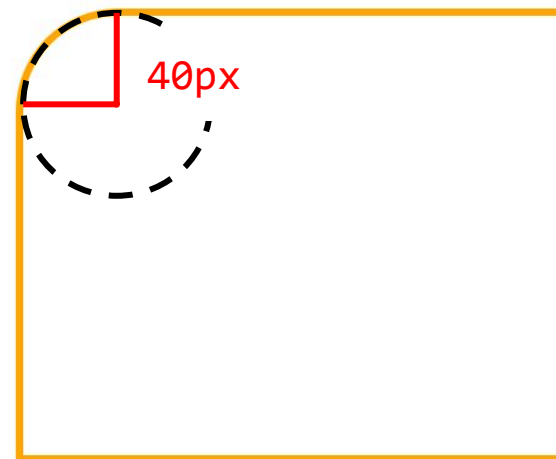
- 設定元素的單一邊框圓角，四角的屬性分別為：

1. 左上角: border-top-left-radius
2. 右上角: border-top-right-radius
3. 左下角: border-bottom-left-radius
4. 右下角: border-bottom-right-radius

- 設定兩個參數即為橢圓形弧線。

- 範例

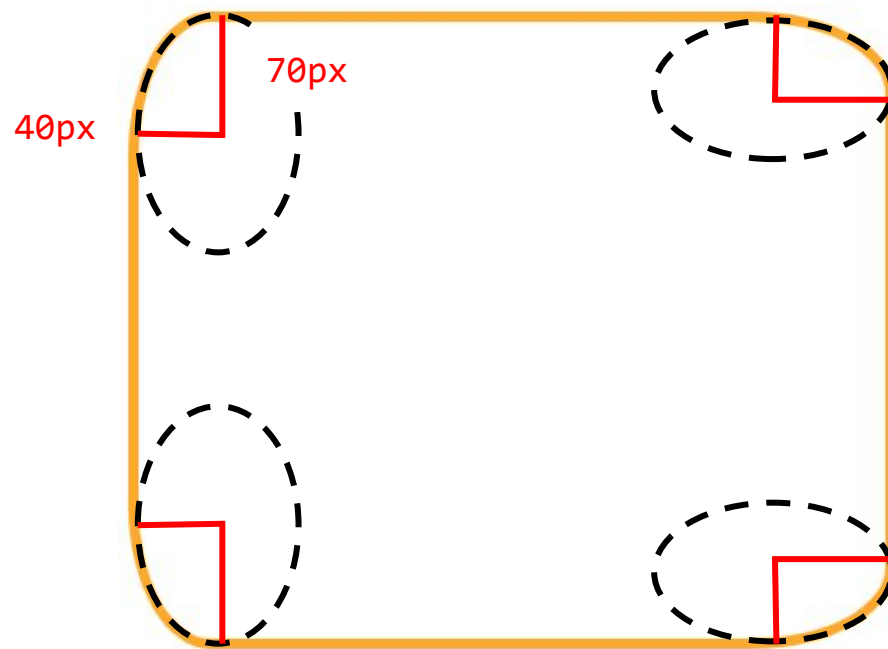
- border-top-left-radius: 40px;



邊框單一圓角 - 範例

```
<head>
  <style>
    .border2 {
      border: 5px solid orange;
      border-top-left-radius: 40px 70px;
      border-top-right-radius: 70px 40px;
      border-bottom-left-radius: 40px 70px;
      border-bottom-right-radius: 70px 40px;
      height: 300px;
    }
  </style>
</head>

<body>
  <div class="border2"></div>
</body>
```



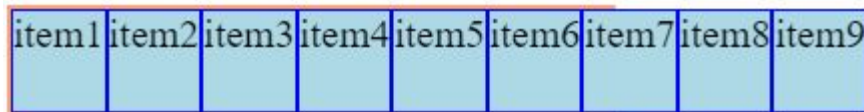
彈性盒子 flex box

換行方式 flex-wrap屬性

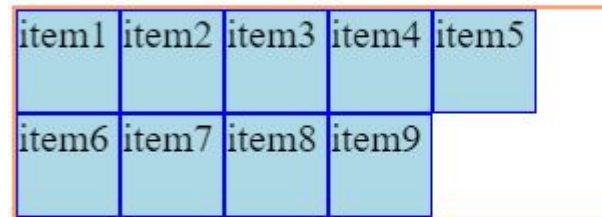
● 換行方式

flex-wrap: nowrap || wrap || wrap-reverse ;

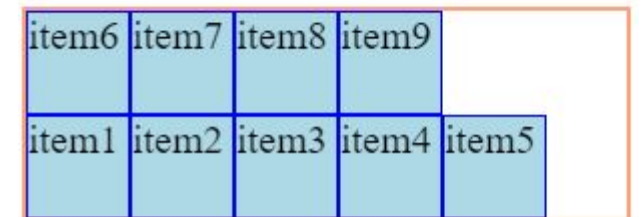
- 設定flex-box中內容的換行方式。
- nowrap:所有內容單行顯示, 不換行。
- wrap:換行, 由切軸起點開始。
- wrap-revrse:換行, 由切軸終點開始。



nowrap



wrap



wrap-reverse

換行方式 flex-wrap屬性 範例

```
/* 外層容器(彈性盒子) */
```

```
.box {  
    border: 2px solid lightsalmon;  
    display: flex;  
    flex-direction: row;  
    justify-content: flex-start;  
    width: 300px;  
}
```

```
/* 內層元素(內容) */
```

```
.item {  
    ...與上個範例相同 ...  
}
```

```
/* 換行方式 */
```

```
.nowrap { flex-wrap: nowrap; }  
.wrap { flex-wrap: wrap; }  
.wrap-reverse { flex-wrap: wrap-reverse; }
```

```
<div class="box nowrap">  
    <div class="item">item1</div>  
    <div class="item">item2</div>  
    <div class="item">item3</div>  
    <div class="item">item4</div>  
    <div class="item">item5</div>  
    <div class="item">item6</div>  
    <div class="item">item7</div>  
    <div class="item">item8</div>  
    <div class="item">item9</div>  
</div>
```

```
<div class="box ...">*3
```

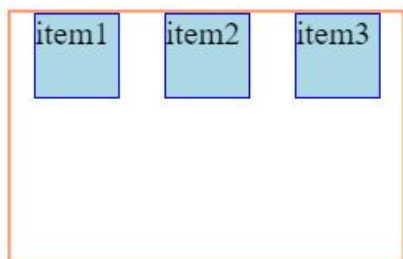
單行對齊方式(切軸) align-items屬性

● 單行對齊方式(切軸)

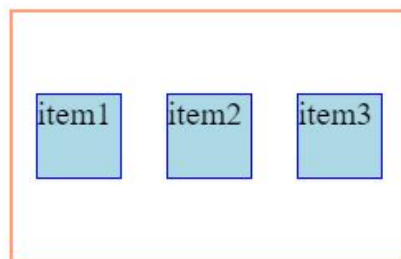
align-items: flex-start || center || flex-end || baseline || stretch ;

- 設定flex-box中內容的切軸對齊方式(單行)。
- flex-start: 對齊切軸起點。
- center: 對齊切軸正中心。
- flex-end: 對齊切軸終點。
- baseline: 對齊文字。
- stretch: 如內容的無設定高度, 會自動占滿全部。(預設值)

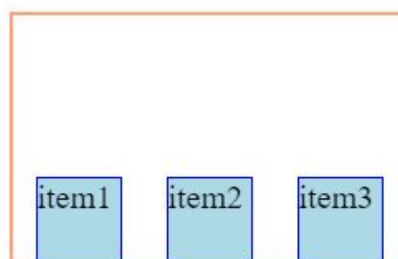
flex-start



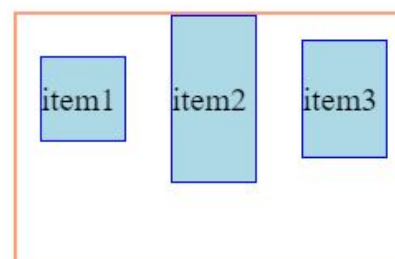
center



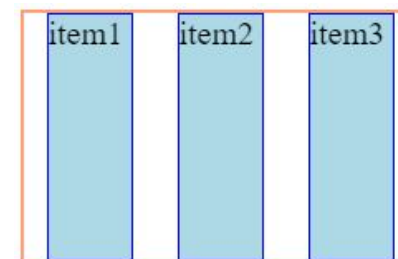
flex-end



baseline



stretch



單行對齊方式(切軸) align-items屬性 範例

/ 外層容器(彈性盒子) */*

```
.box {
  border: 2px solid lightsalmon;
  display: flex;
  width: 300px;
  justify-content: space-around;
  height: 150px;
}
```

/ 內層元素(內容) */*

```
.item { ...與上個範例相同... }
```

/ 展示 baseline 要設定items文字高度 */*

```
.lh50 { height: 50px; line-height: 50px; }
.lh70 { height: 70px; line-height: 70px; }
.lh100 { height: 100px; line-height: 100px; }
```

/ 切軸對齊方式(單行) */*

```
.flex-start { align-items: flex-start; }
.center { align-items: center; }
.flex-end { align-items: flex-end; }
.baseline { align-items: baseline; }
.stretch { align-items: stretch; }
```

```
<div class="box flex-start">
  <div class="item">item1</div>
  <div class="item">item2</div>
  <div class="item">item3</div>
</div>
```

```
<div class="box ...">*5
```

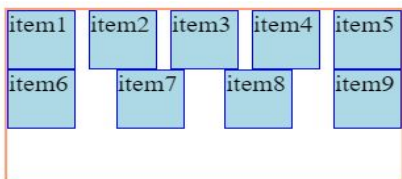
多行對齊方式(切軸) align-content屬性

● 多行對齊方式(切軸)

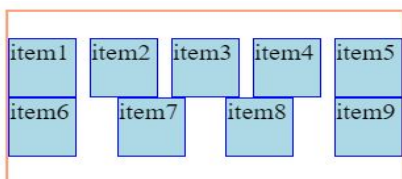
align-content: flex-start || center || flex-end || stretch ... ;

- 設定flex-box中內容的切軸對齊方式(多行)。
- flex-start: 對齊切軸起點。
- center: 對齊切軸正中心。
- flex-end: 對齊切軸終點。
- space-around: 上下留白、平均分配間隔。
- space-between: 上下不留白、平均分配間隔。
- stretch: 如內容的無設定高度, 會自動占滿全部。(預設值)

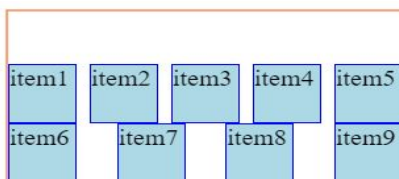
flex-start



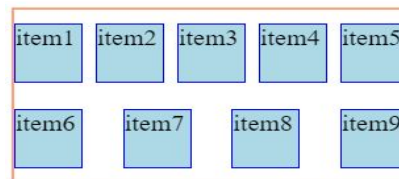
center



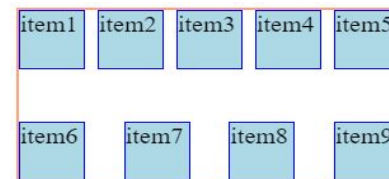
flex-end



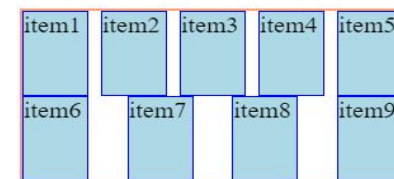
space-around



space-between



stretch



多行對齊方式(切軸) align-content屬性 範例

```
/* 外層容器(彈性盒子) */
```

```
.box {
  border: 2px solid lightsalmon;
  display: flex;
  width: 300px;
  justify-content: space-around;
  height: 150px;
  flex-wrap: wrap;
}
```

```
/* 內層元素(內容) */
```

```
.item { ...與上個範例相同... }
```

```
/* 切軸對齊方式(多行) */
```

```
.flex-start { align-items: flex-start; }
.center { align-items: center; }
.flex-end { align-items: flex-end; }
.space-around { align-content: space-around; }
.space-between { align-content: space-between; }
.stretch { align-items: stretch; }
```

```
<div class="box flex-start">
  <div class="item">item1</div>
  <div class="item">item2</div>
  <div class="item">item3</div>
  <div class="item">item4</div>
  <div class="item">item5</div>
  <div class="item">item6</div>
  <div class="item">item7</div>
  <div class="item">item8</div>
  <div class="item">item9</div>
</div>
```

```
<div class="box ..." *6
```

flex-items的相關設定

● flex-items的伸展

```
flex-grow: 0 || ∞ ;
```

- 當flex-box有剩餘空間時，設定flex-grow屬性可以分配剩餘空間並伸展。
- 不伸展為0(預設值)。

● flex-items的收縮

```
flex-shrink: 0 || ∞ ;
```

- 當flex-box空間不足時，設定flex-shrink屬性可以壓縮其中的items。
- 收縮為1(預設值)，不收縮為0。

● flex-items的寬度

```
flex-basis: 長度單位 ... ;
```

- 設定flex-items的寬度，使用flex-basis屬性。
- 寬度權重: $\text{max-width/min-width} > \text{flex-basis} > \text{width}$ 。

flex-items的相關設定 範例

```
/* 外層容器(彈性盒子) */
```

```
.box {
    border: 2px solid lightsalmon;
    display: flex;
    width: 800px;
    justify-content: space-around;
    height: 150px;
    flex-wrap: wrap;
}
```

```
/* 內層元素(內容) */
```

```
.item { ...與上個範例相同... }
```

```
/* flex-items的伸展 */
```

```
.flex-grow { flex-grow: 1; }
```

```
/* flex-items的收縮 */
```

```
.width500 { width: 500px; }
```

```
.width200 { width: 200px; }
```

```
.flex-shrink { flex-shrink: 0; }
```

```
/* flex-items的寬度設定 */
```

```
/* 寬度權重: max-width/min-width > flex-basis > width */
```

```
.flex-basis { flex-basis: 350px; }
```

```
<div class="box">
    <div class="item flex-grow">item1</div>
    <div class="item">item2</div>
    <div class="item">item3</div>
</div>
<div class="box width500">
    <div class="item width200">item1</div>
    <div class="item width200">item2</div>
    <div class="item width200 flex-shrink">item3</div>
</div>
<div class="box">
    <div class="item">item1</div>
    <div class="item flex-basis">item2</div>
    <div class="item">item3</div>
</div>
```

flex-items的相關設定

- flex-items的個別對齊方式(切軸)

```
align-self: flex-start || center ... ;
```

- 個別設定flex-items的切軸對齊方式, 使用align-self屬性。
- 包含flex-start、center、flex-end、baseline、stretch。

- flex-items的排列順序

```
order: 數值 ;
```

- 個別設定flex-items的排列順序, 使用order屬性。
- 不需要修改html元素順序。
- 任意數值, 可以為負數。

flex-items的相關設定 範例

/* 外層容器(彈性盒子) */

```
.box {
  border: 2px solid lightsalmon;
  display: flex;
  width: 800px;
  justify-content: space-around;
  height: 150px;
  flex-wrap: wrap;
}
```

/* 內層元素(內容) */

```
.item { ...與上個範例相同... }
```

/* flex-items的對齊方式 */

```
.height200 { height: 200px; }
```

```
.align-self { align-self: center; }
```

/* flex-items的排序 */

```
.order-10 { order: -10; }
```

```
.order9 { order: 9; }
```

```
<div class="box height200">
  <div class="item">item1</div>
  <div class="item align-self">item2</div>
  <div class="item">item3</div>
</div>
<div class="box">
  <div class="item order9">item1</div>
  <div class="item order-10">item2</div>
  <div class="item">item3</div>
</div>
```

Media Queries

Media Queries

- 什麼是Media Queries?

- Media Queries 代表網頁會先「詢問 query」「媒體 media」的屬性, 再針對這些屬性定義樣式。
- 媒體類型Media 是大方向, 分為四類:
 - all:所有裝置
 - screen:螢幕裝置
 - print:印刷裝置(包含預覽列印產生的畫面)
 - speech:針對可以「讀出」頁面的設備
- 裝置特性Feature 是小分類, 必須使用()包覆:
 - width/height:螢幕寬度/高度, 可以加上max/min
 - orientation:螢幕旋轉方向, 可以是portrait (直向), landscape (橫向)
 - aspect-ratio:螢幕長寬比, 例如:1/1, 1920/1080
 - color:輸出螢幕的色彩位元數, 若為0則代表黑白裝置
 - resolution:螢幕解析度

Media Queries 語法

- Media Queries語法

- Queries有四種使用方式，分別是or、and、not和only。
- 除了or 使用逗號「,」其他的就直接寫出英文字即可。
- 常用寫法：@media 媒體類型 and (裝置特性) { CSS語法 }。

- 寫法一

```
<style>
  h1 {
    color: blue;
  }
  @media screen and (max-width:600px) {
    h1 {
      color: red;
    }
  }
</style>
```


Media Queries 語法

- 寫法二

```
<head>  
  <link rel="stylesheet" media="screen and (max-width:600px)" href="example.css" />  
</head>
```

- 寫法三

```
<style>  
  @import url(example.css) screen and (max-width:600px);  
</style>
```

陰影
shadow

文字陰影 text-shadow屬性

● 文字陰影

```
text-shadow: h-shadow + v-shadow + blur + color ;
```

- 設定文字的陰影。
- h-shadow: 水平位移的陰影, 負數在左邊、正數在右邊, 數字越大離越遠。
- v-shadow: 垂直位移的陰影, 負數在上方、正數在下方, 數字越大離越遠。
- blur: 陰影模糊程度, 數值越大越模糊。
- color: 顏色。

```
<style>  
  h1 {  
    text-shadow: 5px 5px 5px red;  
  }  
</style>
```

區塊陰影 box-shadow屬性

● 區塊陰影

box-shadow: h-shadow + v-shadow + blur + color ;

- 設定區塊的陰影。
- h-shadow: 水平位移的陰影, 負數在左邊、正數在右邊, 數字越大離越遠。
- v-shadow: 垂直位移的陰影, 負數在上方、正數在下方, 數字越大離越遠。
- blur: 陰影模糊程度, 數值越大越模糊。
- color: 顏色。

```
<style>
  div {
    box-shadow: 5px 5px 5px gray;
  }
</style>
```

文字多欄排版

文字多欄版面

- 文字多欄版面 (指定欄數)

column-count: 數值 ;

- 將一個區塊切割成指定文字欄數。

- 文字多欄版面 (指定欄位寬度)

column-width: 長度單位 || 比例單位 ... ;

- 根據區塊總寬度除以指定欄為寬度。
- 會隨著區塊寬度大小而自動增加/刪減欄位。

```
<style>
```

```
.p1 { column-count: 3; }
```

```
<p class="p1">lorem*10</p>
```

```
.p2 { column-width: 200px; }
```

```
<p class="p2">lorem*10</p>
```

```
</style>
```

文字多欄版面相關設定

- 欄位之間的間距

`column-gap: 長度單位 || 比例單位 ... ;`

- 設定欄位之間的間距寬度。

- 欄位之間的分隔線

`column-rule: 寬度 + 樣式 + 顏色 ;`

- 與border屬性值寫法相同。

```
<style>
    .p1 {
        column-gap: 30px;
        column-rule: 3px dashed blue
    }
</style>
```