

#2019.7.10 開始一更;參考 <https://www.codecademy.com/learn/learn-css>

#2019.7.12 結束一更, 剩下字體、Grid 等進階 topic 待之後有空補

#2019.7.19 開始大略二更(完成), 更新衆多選擇器 參考 <https://code.tutsplus.com/zh-hant/tutorials/the-30-css-selectors-you-must-memorize--net-16048>

前言:What is CSS

CSS 是串接樣式表(Cascade style sheet), 顧名思義就是用串接表的方式美化(style)網頁

實用心得:

1.也可以自己改網頁去廣告 (加入 style 屬性 display:none), 這也告訴我們為甚麼 CSS 是用: 而不是 = 因為它算是一個屬性而已 (多個 CSS 需加 ;)

```
<div class="monti-player-wrapper _2Rxd7Q0xtLb-TEbSmF91-u--breakpoint-mini-player" style="position: absolute; height: 0px; display:none;">
```

目錄

A.使用 CSS

B.CSS 語法

(1).選擇器

(2).CSS code 語法

C.常用 CSS style

(1)文字相關

(2)色彩相關

(3)背景相關

D.Boxing model

width,height & padding & border & margin

E.position

A.使用 CSS 的方式

Recall:CSS 引入的三種方式, 參考 CSS 概念筆記

(1).inline style (style 屬性)

任何 HTML tag 都可以使用屬性 style 來進行 inline style

例如

```
<p style="color:red; font-size:20px;"> this paragraph is red </p>
```

(2).style tag

\*HTML 有 style element 供一次選擇某些種的元素

\*Style tag 放置於 head 中  
\*注意與上述方法格式上的差異  
如

```
<head>
  <style>
    p {
      color:red;
      front-size:20px;
    }
  </style>
</head>
```

(3) link the .css file (分檔)

\*正式的大網頁會分開 HTML 與 CSS 檔 (管理、易讀)

\*使用 link element

\*放置於 head

\*重要:link 介紹

(a)<link> 沒有 closing tag

(b)<link>有三個重要屬性

href 放入連結位置 \*/myStyle.css, 也可以用相對路徑

./myStyle.css

type 輸入檔案類型, CSS 為 text/css

rel 檔案間的關係, CSS 為 stylesheet

如

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

## B.CSS 語法

(1)選擇器(selector)的概念, 有多種與法可以組合

(一)簡單選擇器

1 選擇 tag

```
elementName{
```

```
} 則所有此 element 都會被選擇, 然後套用{ }內的 CSS style
```

例如

```
p{} 就是選擇所有<p>
```

2 選擇 class(屬性)

Class 名前要加 .

eg

```
.randomClass{
```

```
}  
選擇了所有有 class="randomClass"的標籤  
(可彌補 element 一次選擇所有的缺點)  
(class 屬性可給多個值，中間用空白隔開即可)
```

### 3 選擇 id(屬性)

id 名前要加#

eg

```
#idName{
```

```
}
```

id 是唯一的

### \*class vs id

class 通常會用來 reuse (比如顏色)，因 class 可多值能達到互相搭配的作用(就像調色)

id 是處理特例時使用，所以對元素是唯一的，可以覆蓋其他選擇的 style 效果

(property 加上 !important > id > 組合選擇 > class > element)

\*在 property rule 中可加入 !important 阻止被覆蓋

例如:

```
p{
```

```
    color:red !important;
```

```
}
```

### 4 子孫 selector

選擇器中間用空白隔開，就成為選前者的子孫 element

\*例如

h1 .article (選 h1 中 class 為 article 的)

(當然也可以 h1 #id，但 id 是唯一的所以直接 #id 就可)

\*也可以

.article h1 (選 class 為 article 中 child 是 h1 的)

### 5 子選擇器

選擇器中間用 > 隔開，就成為選前者的直屬 child 的 element

### 6 多重 selector

可以一次選多個元素，用逗號，隔開 (省重複 code)

eg: p, .myClassA, #id {color:blue;}

### 7 \*選擇

選擇所有元素，通常用於快速將 margin、padding 歸零

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

8 屬性選擇器

elem[attri]  
如 a[href]

(二)偽類別與偽元素

9 pseudo class(reference:<https://footmark.info/web-design/css/css-pseudo-class-selectors/>)

用來表示元素的狀況或狀態，如位置、是否被點擊....

語法 elem:pseudo-class

\*常用:

a.表示 elem 的第 n 個 sibling，同時也要是 elem 型態

```
.nth-child-number p:nth-child(1),
.nth-child-number p:nth-child(2),
.nth-child-number p:nth-child(4) {
    background-color: skyblue;
}
```

b.連結顏色

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

10 pseudo element (<https://footmark.info/web-design/css/css-pseudo-element-selectors/>)

用來表示實際存在網頁中卻又不是元素，包含第一行，第一個字，前後....

語法 elem::pe

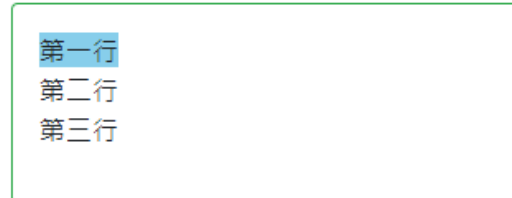
CSS

```
1 | .first-line::first-line {
2 |     background-color: skyblue;
3 | }
```

HTML

```
1 | <p class="first-line">
2 |     第一行<br>
3 |     第二行<br>
4 |     第三行
5 | </p>
```

<code>E::first-line</code>	偽元素	E 元素的第一行
<code>E::first-letter</code>	偽元素	E 元素的第一個字
<code>E::before</code>	偽元素	E 元素之前 新增文字內容
<code>E::after</code>	偽元素	E 元素之後 新增文字內容



(2)CSS code syntax

CSS code 的語法寫於 CSS 選擇器中  
格式為

property:value;

- \*注意中間是用：
- \*注意打完一行要加；
- \*這也是 CSS 是一種 sheet 的原因(表格語言)

property 可以是 size、color 等想要設計的 style  
value 則為 18px、blue 等設定值

### C.常用 CSS style

#### (1)文字相關:

##### 1.font-family (字形)

- \*browser 預設字形為 Times New Roman
- \*Web 是以使用者為目的，所以字形須選使用者有的
- \*字形同一 website 控制在 2-3 種會比較好 (載入效率、整齊)
- \*如果字形名稱中間有空格，應使用" "框好

##### 2.font-size(字體大小)

- \*單位是 px (pixel)
- \*也可以使用 em，代表是父元素的倍數  
如 0.7em，表示是父元素字體大小的 0.7 倍

##### 3.font-weight:bold (字體加粗)

- \*也有 normal 的值
- \*需要 font 有此字形才能使用

##### 4.text-align(文字開始的位置)

- \*就是靠左、靠中或靠右(left、center、right)
- \*預設是靠左

#### (2)色彩相關

##### 1.color & background-color (前景(元素本身)與背景色) 如題

##### 2.opacity (不透明度)

- \*連同背景前景一起，整個元素的不透明度
- \*從 0(完全透明)到 1(完全不透明)

補充:設定色彩的規則:

前言:共有三種方式設定顏色

1.name (string) (只有 147 種)

2.RGB (number) (理論上可有  $256^3$  種)

### 3.HSL (number)

\*RGB 有兩種方式

1.要用#開頭

例如:#8FBC8F 或#0FA

記住是在 116 進位下的 R、G、B 數值

2.rgb(x,y,z) 其中 xyz 從 0-255

\*HSL:

hsl(色相(參考色相環)、色度(%)、亮度(%))

(還有加入透明度的方法, 不過這裡不深入討論)

(3)背景相關:

1.background-image(背景圖)

\*值要填入 url("myurl") 可以是外部 or 內部

(4)區塊相關:

參考 D.box model

(5)位置相關:

參考 E.position

### D. box model

前言

\*Boxing model 目的是從視覺上呈現 HTML 的區塊性, 讓設計更方便美觀

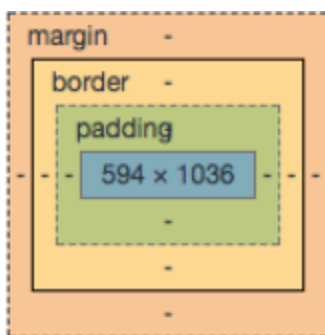
\*Boxing model 包含了諸多 property

元素長、寬(藍色部分) (內容大小)

padding(內距) (內容與邊框的距離)

border(邊框)

margin(外距) (邊框與其他元素的距離)



(此圖可由 chrome F11 > element > style >box modeling 找到)

(1)height & width(長度與寬度)

預設為 raw content 大小

## (2)border 邊界

### (a)basic

就像畫框一樣框住元素

\*可被設定 width、style、color

\*設定方式為

```
border: 3px solid coral;
```

\*預設為

medium none element`s color

\*width 除了數值 px 外也有 thin medium thick

參考 <https://developer.mozilla.org/en-US/docs/Web/CSS/border-style>

\*color 參考 [https://developer.mozilla.org/en-US/docs/Web/CSS/color\\_value](https://developer.mozilla.org/en-US/docs/Web/CSS/color_value)

### (b)border-radius

\*類似從四個角開始的曲率

\*越大越接近圓(px)

\*如果輸入 100% 代表半徑為 height、weight 的橢圓

## (3)padding (內距)

\*padding 可以使元素不那麼狹窄，以及拓展背景顏色

\*可以指定上下左右的順序給 px，如果只給一個就是統一大小，兩個就相當於高與寬

\*也可以用 padding-top..方式設定

## (4)margin(外距)

\*指定方式類似於 padding

\*auto (通常用於 margin width)，margin: 0 auto，能使 margin 居中平分 Content 沒有用完的 width

\*垂直 margin 有 collapse(邊界重疊)的現象，即兩元素的 margin 會取最大的那個

(水平、padding 都是取總和)

\*padding 可視為拓展，margin 可視為限制

## (5)其他輔助 property

\*min-width、max-width...:

用來控制縮放時的大小限制，例如 min-width:100px，縮的再小也要有 100px 的寬

\*overflow:

當元素太大時如何呈現，有 hidden、scroll、visible 等值

\*user-agent:UA

可參考網路筆記，簡而言之就是 browser 告訴網頁的訊息(名稱、版本、渲染引擎)

也會規定瀏覽器 CSS 的 default, 因此我們可先 reset

eg:

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

//0 就不需要單位了

\*visibility:hidden && display:none

前者會留下空的空間, 後者則完全隱藏, 但都能在 source code 看到

### (6)改良簡化的 boxing

前言:複雜繁瑣的元素區塊(box)大小計算麻煩, 因此有些語法糖幫我們簡化  
width+padding+border=元素寬度

\*box-sizing:border-box;

可以使 content 自動調整使 padding+content 大小=weight 大小

### E.position:

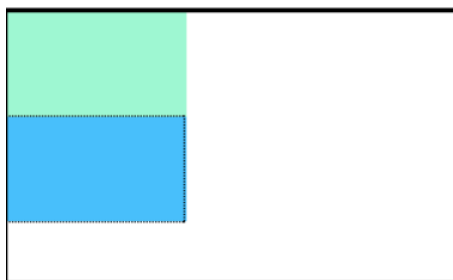
前言:正常而言 browser 會將 element 由執行順序排版, 但我們可藉由 CSS 控制它們的位置(Flow)

property 如下

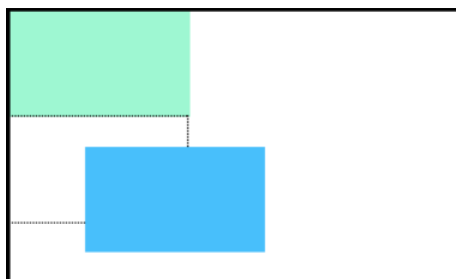
#### (1)position:

有四種值:static(預設)、relative、absolute、fixed

\*static:預設順序



\*relative:相對於預設順序調整(要設定 top、bottom、right、left, 變成位移量), 不影響其他元素



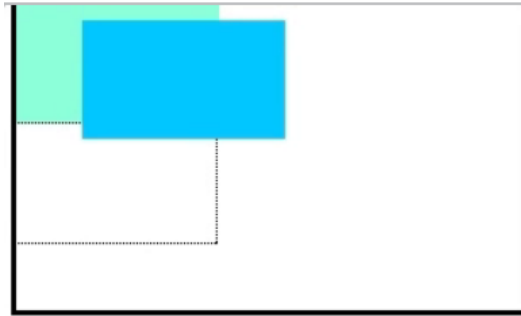
\*absolute:絕對位置

\*預設為 parent 的位置

\*以 parent 位置 t、b、r、l 為位移量



- \*其他元素會當沒這元素存在(可以隨意蓋在其他元素上)
- \*就像絕對黏於"網頁"上，不受滾動影響



- \*fixed: 固定位置
  - \*類似於絕對位置
  - \*但就像固定於"視窗"上，就算滾動也永遠保持在該位置(例如廣告)

### (2)z-index:

- \*如字面意思代表深度，即重疊時要顯示哪個元素(前後關係)
- \*index 越大表示越前面(權重的概念)
- \*position:static 對 z-index 是沒有效果的(預設靜態不會管深淺)

### (3)display:有三種值

- \*inline: 限定元素大小的空間 (一定在一排解決)

QUESTION 1 QUESTION 2 QUESTION 3 QUESTION 4 QUESTION 5 QUESTION 6

- \*block:使該元素佔滿它的水平線 (仍可設定寬度)  
常預設(em strong ; 大多數 element))

(兩者為通)

QUESTION 1  
QUESTION 2  
QUESTION 3  
QUESTION 4  
QUESTION 5  
QUESTION 6

(行行分明)

\***Inline-Block**:結合兩者特性，既能盡量縮在一排，又能限定寬高



QUESTION 1 QUESTION 2 QUESTION 3 QUESTION 4 QUESTION 5  
QUESTION 6

(4)**float&clear**:

- \***float** 有兩個值，左與右 (**right & left**)

- 能盡可能把元素推到最左或最右

- \***clear** 有四個值，是用來控制 **float** 的

- \*左右、**both**、**none** 表示 **float** 不可碰觸的元素 (**both** 是最常用的)