

資料庫專題

資應菁英班四甲

第 5 週 CSS 排版

許雯綾
chiaohsu@nutc.edu.tw

大 綱

- 5-1 CSS 選擇器
- 5-2 CSS 放置位置
- 5-3 外框與邊距
- 5-4 位置屬性
- 5-5 區塊與行內
- 5-6 浮動與清除

5-1 CSS 選擇器

- CSS (Cascading Style Sheets) 串接樣式表，是網頁的美容師，其用途是美化網頁與編排版型。
- CSS 選擇器用於指定「被 CSS 設定的對象」。
- 對象包含網頁中的內容，如文字、圖片、及其他元素。

5-1-1 通用選擇器

- 對象為網頁中的所有元素。
- 表示方式使用 *
- 例如：將所有元素的 margin(外距)設為 0px
`*{margin:0px;}`

5-1-2 類型選擇器

- 對象為 HTML 元素。
- 例如：將<p>元素的文字大小設為 24px

```
<head>
  <style>
    p{ font-size:24px; }
  </style>
</head>
.....
<body>
  <p>一行文字</p>
</body>
```

設定樣式

套用樣式

5-1-3 ID 選擇器

- 對象為 HTML 指定 id 的元素 (id 為獨一無二)。
- 表示方式使用 #
- 例如：將 id 名稱為 wrapper 的<div>元素內的文字大小設為 24px

```
<head>
  <style>
    #wrapper{ font-size:24px; }
  </style>
</head>
.....
<body>
  <div id="wrapper">一行文字</div>
</body>
```

設定樣式

套用樣式

5-1-4 類別選擇器

- 設定 CSS 類別的內容，HTML 元素可指定套用該類別（可多次套用）。
- 表示方式使用。
- 例如：將<p>元素及<h1>元素內的文字對齊方式設為置中

```
<head>
  <style>
    .center{ text-align: center; }
  </style>
</head>
.....
<body>
  <p class="center">一行文字置中對齊</p>
  <h1 class="center">一行文字置中對齊</h1>
</body>
```

設定樣式

套用樣式

5-1-5 群組選擇器

- 同時設定多個 HTML 元素為相同的樣式。
- 使用，為元素做區隔
- 例如：將<p>元素及<h1>元素內的文字顏色均設為藍色

```
<head>
  <style>
    p, h1{ color: blue; }
  </style>
</head>
.....
<body>
  <p>一行藍字的文字</p>
  <h1>一行藍字的文字</h1>
</body>
```

設定樣式

套用樣式

5-1-6 後代選擇器

- 設定某元素之下，所有特定 HTML 子孫元素的樣式。
- 使用「空白」為祖孫元素做區隔
- 例如：將<h1>元素內所有的元素文字顏色皆設為紅色

```
<head>
  <style>
    h1 span{ color: red; }
  </style>
</head>
.....
<body>
  <h1>Hello <span>這裏會變紅色字</span></h1>
  <h1>Hello <em><span>這裏會變紅色字</span></em></h1>
</body>
```

設定樣式

套用樣式

5-1-7 子選擇器

- 設定某元素之下，特定 HTML 子元素的樣式。
- 使用 > 為子孫元素做區隔
- 例如：將<h1>元素內子元素為的文字顏色設為紅色

```
<head>
  <style>
    h1>span{ color: red; }
  </style>
</head>
.....
<body>
  <h1>Hello <span>這裏會變紅色字</span></h1>
  <h1>Hello <em><span>這裏不會變紅色字</span></em></h1>
</body>
```

設定樣式

套用樣式

5-1-8 虛擬選擇器

- 依據不同的情況變更為不同的樣式。
- 例如 <a>元素有四種狀態：a:link(未點擊連結前)、a:visited(點擊連結後)、a:hover(滑鼠移至連結時)、a:active(滑鼠按下時)

```
<head>
  <style type = "text/css/">
    a:link {color: #000000; }
    a:visited {color: #FF0000; }
    a:hover {color: #00FF00; }
    a:active {color: #0000FF; }
  </style>
</head>
```

設定樣式

5-2 CSS 放置位置

- CSS 可以放置在「網頁內部」或「網頁外部」。

5-2-1 網頁內部

(1)HTML 標籤內

- 寫在 HTML 標籤內的 style 屬性裏。

```
<body>
  <h1 style="text-align:center;">Hello World</h1>
  <p style="font-size:20px;">Hello World</p>
</body>
```

(2)head 的 style 元素內

- 將樣式集中寫在 style 元素中。

```
<head>
  <style type="text/css/">
    h1 { text-align:center; }
    p { font-size:20px; }
  </style>
</head>
```

5-2-2 網頁外部

- 將樣式獨立撰寫為一個 CSS 文件儲存。
- 網頁中透過 link 或 import 的方式連結

(1)link

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

CSS 檔案相對路徑

告訴瀏覽器，要導入
一個 CSS 檔案

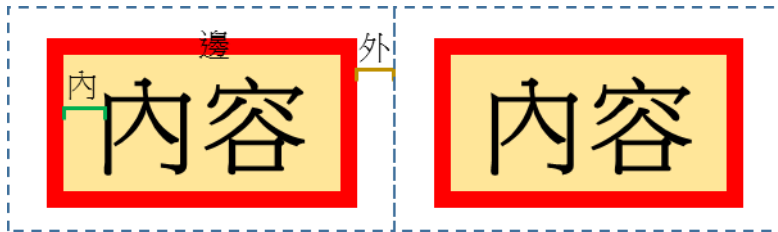
(2)import

```
<head>
  <style type="text/css/">@import url("css/style.css");</style>
</head>
```

- 網頁元素依不同的選擇器設定樣式時，具有「由外至內層」層層套用的效果，使用時應特別留意最後影響的效果為何。

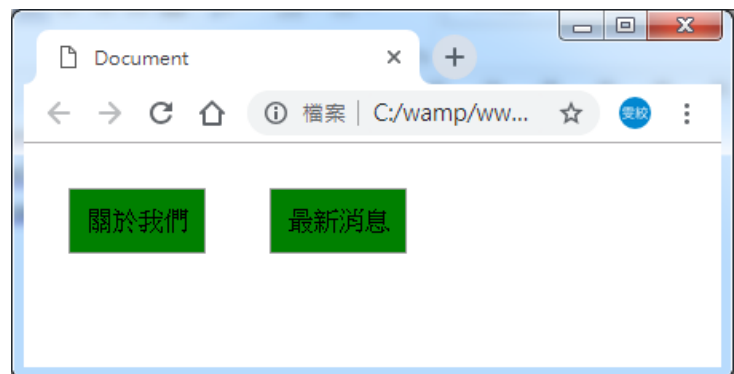
5-3 外框與邊距

- 元素真正佔用視覺空間 = content + padding + border + margin
- 內邊距(padding)：可視範圍
- 邊框(border)：可視範圍
- 外邊距(margin)：不可視範圍，只有空間(距離感)，沒有底色



■ 範例

```
<head>
  <style>
    div div{
      padding:10px;
      border: 1px solid #999;
      margin: 20px;
      background-color: green;
      float: left;
    }
  </style>
</head>
<body>
  <div>
    <div>關於我們</div>
    <div>最新消息</div>
  </div>
</body>
```



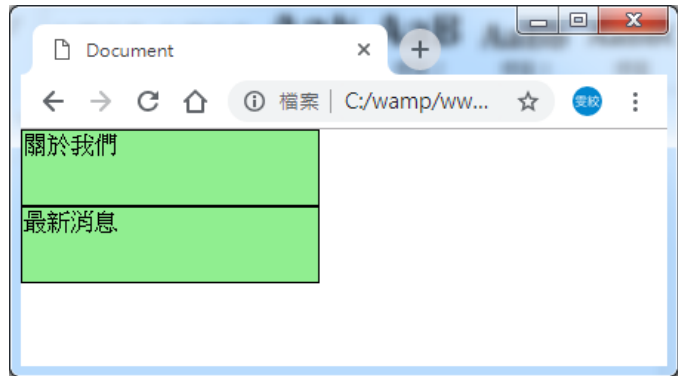
- padding 和 margin 均分別可設定 top、left、right、bottom 的值，例如：
padding-top: 10px;

5-4 位置屬性

- position 屬性用於設定元素的定位，包含：static(預設值)、absolute、relative、fixed，例如：
position: static;
- 另外搭配 top、bottom、left、right 表示位移的距離

5-4-1 position: static

```
<head>
  <style>
    *{
      margin:0px;
      padding:0px;
    }
    div {
      border: 1px solid #000;
      width:200px;
      height:50px;
      background-color: lightgreen;
    }
  </style>
</head>
<body>
  <div>關於我們</div>
  <div>最新消息</div>
</body>
```



- 沒有設定 position 屬性，預設為 static
- 使用通用選擇器 * 設定 padding 和 margin，解決不同瀏覽預設不同問題。

5-4-2 position: relative

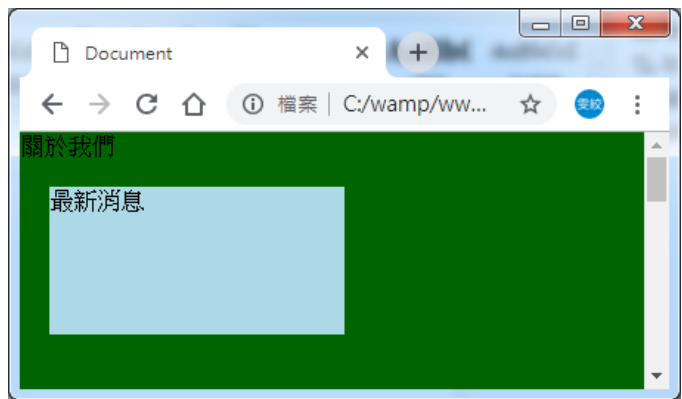
```
<head>
  <style>
    *{
      margin:0px;
      padding:0px;
    }
    div {
      border: 1px solid #000;
      width:200px;
      height:50px;
      background-color: lightgreen;
    }
    .relative{
      background: lightblue;
      position: relative;
      top:20px;
      left:50px;
    }
  </style>
</head>
<body>
  <div>關於我們</div>
  <div class="relative">最新消息</div>
</body>
```



5-4-3 position: fixed

- 設定為 fixed 時，當網頁往下捲時，元素的位置會一直固定在瀏覽器的某個位置。

```
<head>
  <style>
    *{
      margin:0px;
      padding:0px;
    }
    #wrapper {
      height:800px;
      background-color: darkgreen;
    }
    .fixed{
      background: lightblue;
      position: fixed;
      left: 20px;
      width:200px;
      height: 100px;
    }
  </style>
</head>
<body>
  <div id = "wrapper">
    <div>關於我們</div>
    <br />
    <div class="fixed">最新消息</div>
  </div>
</body>
```



5-4-4 position: absolute

- 設定為 absolute 時，會依據「可定位的父元素」進行定位。
- 可定位的父元素的屬性必須為 relative、absolute、fixed。
- 若父元素不可定位，則以 body 最左上角為定位點。

```
<head>
  <style>
    *{
      margin:0px;
      padding:0px;
    }
    #wrapper {
      height:800px;
      background-color: darkgreen;
    }
    .fixed{
      background: lightblue;
      position: fixed;
    }
  </style>
</head>
<body>
  <div id = "wrapper">
    <div>關於我們</div>
    <br />
    <div class="fixed">最新消息</div>
  </div>
</body>
```



```

        left: 20px;
        width: 400px;
        height: 200px;
    }
    .absolute {
        width: 100px;
        height: 50px;
        background-color: red;
        position: relative;
        top: 20px;
        left: 20px;
    }
</style>
</head>
<body>
    <div id="wrapper">
        <div>關於我們</div>
        <br />
        <div class="fixed">最新消息(父元素)
            <div class="absolute">子元素</div>
        </div>
    </div>
</body>

```



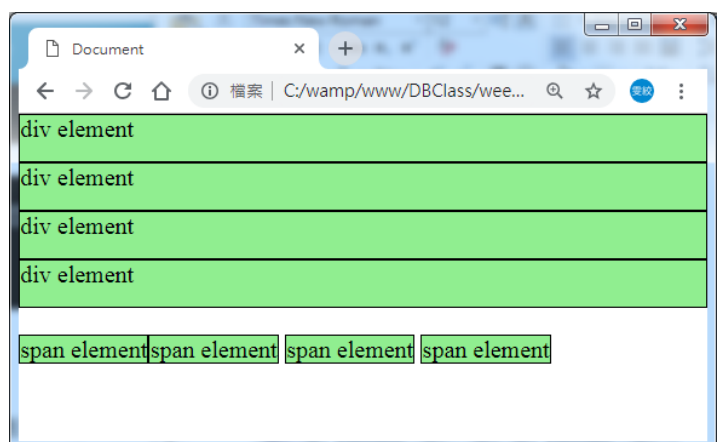
5-5 區塊與行內

- 在網頁排版裡，可以粗略的分成兩大系統：區塊(Block)、行列(Inline)元素，用來設定「內容的排列環境(formatting context)」。
- **div** 和 **span** 這兩個元素，就是支援這兩種元素觀念的 HTML 表現形式。

```

<head>
    <style>
        div, span {
            border: 1px solid #000;
            width: 50px;
            height: 30px;
            background-color: lightgreen;
        }
    </style>
</head>
<body>
    <div>div element</div><div>div
element</div>
    <div>div element</div>
    <div>div element</div>
    <br />
    <span>span element</span><span>span element</span>
    <span>span element</span>
    <span>span element</span>
</body>

```



- 原始碼無論是橫排還是直排，是不會影響最後呈現的結果。
- **div** 的內容由上到下的垂直排列
- **span** 的內容由左至右的水平排列

■ <div> 的預設行為

- **block** 類型元素的代表
- 自然排列是直排(強迫換行)
 - **div** 元素寬與它的容器大小一樣(**width: 100%;**)。
 - **div** 元素可以控制高度。
- **div** 本身的作用
 - 界定區塊尺寸與留白範圍
 - 賦予執行 **block** 元素的 CSS 特性

■

- **inline** 類型元素的代表
- 自然排列是橫排(不強迫換行)
 - **span** 元素寬與它的內容撐開，所以可以「水平的一個接著一個」。
 - **span** 元素高由它的內容撐開。
- **span** 本身的作用
 - 界定內容選取範圍
 - 賦予執行 **inline** 元素的 CSS 特性

- **display** 屬性中，有兩個值 **inline** 和 **block**。

- 因此 **div** 也可以透過設定 **display: inline** 將 **div** 變成 **inline** 元素。
(但不建議這麼做，因為每個 HTML 標籤都有預設 CSS 行為，不要任意改變)

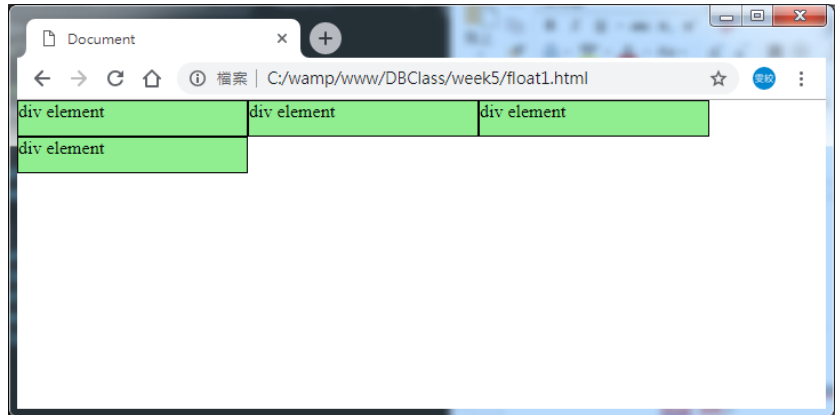
5-6 浮動與清除

- 網站常見將文字繞著一個圖案顯現。這是用 **float** (浮動) 這個屬性來達成的。
- **float** 屬性有三個可能的值：**left**、**right**、和 **none**。
- **div** 標籤是「區塊」的功能，預設是由上到下垂直排列。透過設定「**float**」可以改變排列的方式。

```

<head>
  <style>
    div {
      border: 1px solid #000;
      width: 200px;
      height: 30px;
      background-color: lightgreen;
      float: left;
    }
  </style>
</head>
<body>
  <div>div element</div>
  <div>div element</div>
  <div>div element</div>
  <div>div element</div>
</body>

```

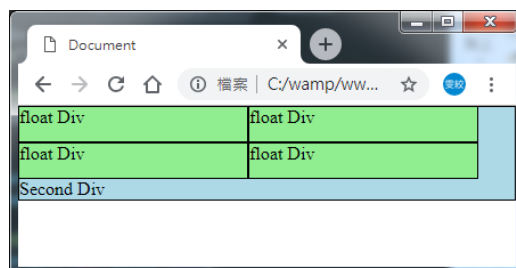
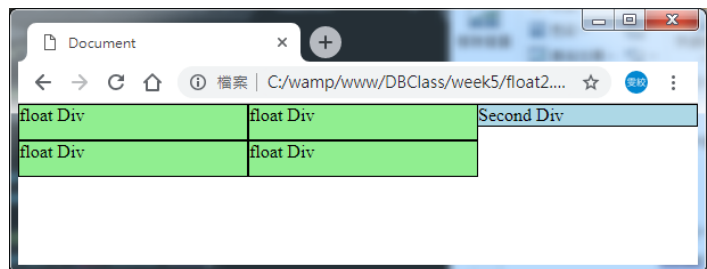


■ float 未清除，即使沒有套用，仍會造成排版的問題

```

<head>
  <style>
    .floatDiv {
      border: 1px solid #000;
      width: 200px;
      height: 30px;
      background-color: lightgreen;
      float: left;
    }
    .afterFloat {
      border: 1px solid #000;
      background-color: lightblue;
    }
  </style>
</head>
<body>
  <div class="floatDiv">float Div</div>
  <div class="floatDiv">float Div</div>
  <div class="floatDiv">float Div</div>
  <div class="floatDiv">float Div</div>
  <div class="afterFloat">Second Div</div>
</body>

```

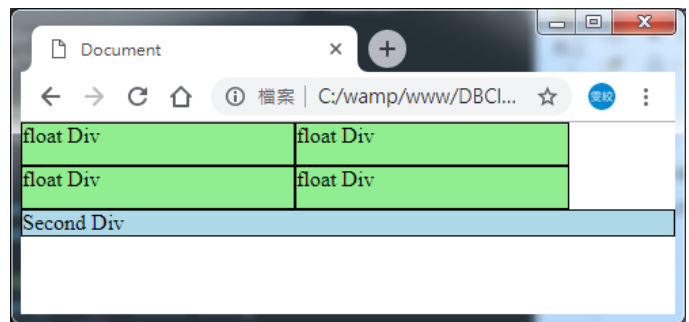


■ 第一種清除的方式：clear

➤ 可能的值有: left, right, both, none。

➤ 上述範例修改 .afterFloat

```
.afterFloat{  
    border: 1px solid #000;  
    background-color: lightblue;  
    clear:left;  
}
```

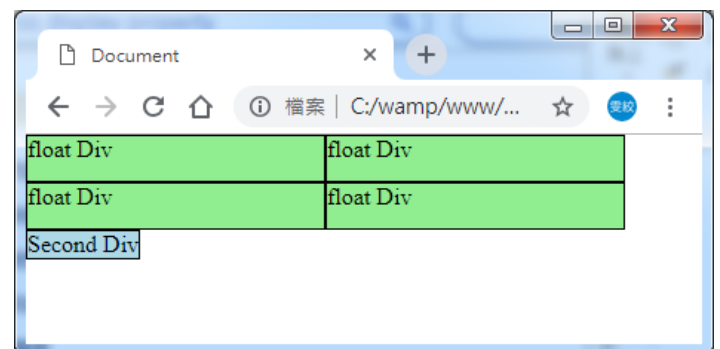


■ 第一種清除的方式：display: inline-block

➤ 可能的值有: left, right, both, none。

➤ 上述範例修改 .afterFloat

```
.afterFloat{  
    border: 1px solid #000;  
    background-color: lightblue;  
    display: inline-block;  
}
```



■ inline-block：切版的重要工具，結合兩者特性，一不會強迫換行，二可以設定寬高尺寸