



網頁程式設計

CSS樣式設計

Instructor: 馬豪尚

CSS 進階選擇器

HTML父元素和子元素

Example

```
<div class="parent">  
  <div class="child1"> </div>  
    <p> </p>  
  <div class="child2"> </div>  
  <div class="child3"> </div>  
</div>
```

CSS 父子選擇器

- › 可以同時指定選擇在某個父元素底下的某種子元素
 - `div div { color:blue; }`
 - `div div p { color:green; }`
 - `.parent .child1 p {color:green;}`

- › 如果只想指定選擇父元素底下某一個子元素
 - CSS3新增虛擬子元素選擇器
 - `div div:nth-child(n) { color:blue; }`
 - `div div:nth-child(1) p { color:green; }`
 - `div div:nth-child(odd) p { color:green; }`
 - `div div:nth-child(even) p { color:green; }`

CSS 多重類別選擇器

› HTML多重類別宣告

- `<div class="one two"> </div>`

› CSS多重類別選擇器

- `.one.two{` /*兩個 class 中沒有空格*/

- › 某個區塊必須同時具有 one 和 two 的 class 時，才能被 CSS 所選擇到

- `.one .two{` /*兩個 class 中有空格*/

- › 必須要是在 one 裡面的 two，才會被選擇到

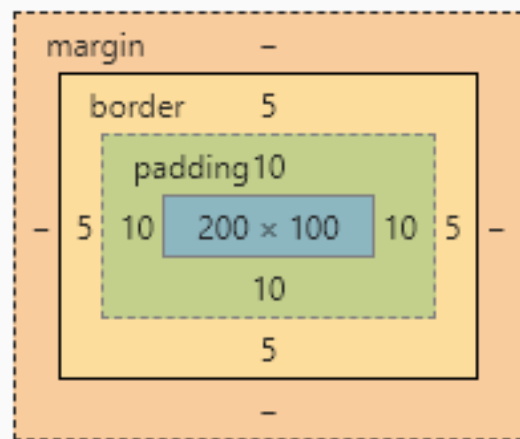
- `.one, .two{` /*兩個 class 中出現逗號*/

- › class 中有 one 或 two，都會被編輯器所選擇到

CSS Box Model

CSS BOX Model

- › Box model 是將每個html元素看成一個方形的盒子
- › Box structure
 - Content: 資料內容本身
 - Padding: 留白，環繞在內容四周的部分
 - Border: 框線，加在留白區域外圍的線條
 - Margin: 邊界，控制html元素彼此之間的距離



Margin邊界屬性

- › margin: 以下屬性值預設都為0
 - margin-top 長度 | 百分比 | auto → p{margin-top: 10px}
 - margin-bottom
 - margin-left
 - margin-right
- › margin: value1 value2 value3 value4
 - 只有一個值: 套用到全部邊界
 - 兩個值: 第一個值套用到上下邊界，第二個值套用到左右邊界
 - 三個值: 上邊界 左右邊界 下邊界
 - 四個值: 上邊界 右邊界 下邊界 左邊界

Padding 留白屬性

- › Padding: 以下屬性值預設都為0
 - padding-top: 長度 | 百分比 → p {padding-top: 10%;}
 - padding-bottom
 - padding-left
 - padding-right
- › padding: value1 value2 value3 value4
 - 只有一個值: 套用到全部留白
 - 兩個值: 第一個值套用到上下留白，第二個值套用到左右留白
 - 三個值: 上留白 左右留白 下留白
 - 四個值: 上留白 右留白 下留白 左留白

Border框線屬性

› border-style框線樣式

- border-top-style: 樣式
- border-bottom-style: 樣式
- border-left-style: 樣式
- border-right-style: 樣式

› border-style

- 可用樣式有none、hidden(隱藏)、dotted(點線)、dashed(虛線)、solid(實線)、double(雙線)、groove(3D立體內凹)、ridge(3D立體外凸)、insert(內凹)、outset(外凸)

› border-style: value1 value2 value3 value4

- 套用規則同邊界和留白

框線樣式demo

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

Border框線屬性

- › Border-color 框線色彩
 - border-top-color: 色彩 | transparent(透明)
 - border-bottom-color
 - border-left-color
 - border-right-color
- › border-color: value1 value2 value3 value4
 - 套用規則同邊界和留白

Border框線屬性

- › border-width框線寬度
 - border-top-width: thin | medium | thick | 長度
 - border-bottom-width
 - border-left-width
 - border-right-width
- › border-width : value1 value2 value3 value4
 - 套用規則同邊界和留白

Border框線速記

- › 可以直接設定框線的樣式、顏色、寬度等屬性
- › Border: <border-style> || <border-color> || <border-width>
- › 若設定的屬性不只一個，屬性值沒有順序之分，也可以不寫使用預設

Example:

```
<h1 style="border: solid 10px hotpink">內容</h1>
```

```
<h1 style="border: 10px hotpink">內容</h1>
```

```
<h1 style="border-top: 10px hotpink">內容</h1>
```

Border框線屬性

› Border-radius框線圓角

- border-top-left-radius: 長度 | 百分比 [長度 | 百分比]
- border-top-right-radius
- border-bottom-right-radius
- border-bottom-left-radius

› border-radius: value1 value2 value3 value4

- 一個值: 就套用在全部角落
- 兩個值: 第一個值套用到左上和右下，第二個值套用到左下和右上
- 三個值: 左上角 右上角和左下角 右下角
- 四個值: 左上角 右上角 右下角 左下角

Box的大小套用類型

- › `box-sizing: content-box | border-box`
- › `box-sizing: content-box`
 - 這個是預設屬性，就是我們一般作用得模式
- › `box-sizing: border-box`
 - 這個屬性的話，就會把 `padding` 等地考慮進來，而自動做內縮調整

Box的內容寬度和高度

- › width: 長度 | 百分比 | auto
- › height: 長度 | 百分比 | auto
- › min-width: 長度 | 百分比 | auto
- › max-width: 長度 | 百分比 | auto
- › min-height: 長度 | 百分比 | auto
- › max-height: 長度 | 百分比 | auto
- › overflow: visible | hidden | scroll | auto

Box的長度和寬度計算

› content-box

- 總寬度 = 內容寬度 + 左留白寬度 + 右留白寬度 + 左框線寬度 + 右框線寬度 + 左邊界寬度 + 右邊界寬度
- 總高度 = 內容高度 + 上留白高度 + 下留白高度 + 上框線高度 + 下框線高度 + 上邊界高度 + 下邊界高度

› border-box

- 總寬度 = 內容寬度
- 總高度 = 內容高度

```
div{  
    width: 320px;  
    padding: 10px;  
    border: 5px solid gray;  
    margin: 0;  
}
```



BOX的顯示層級

HTML元素層級

› Block level(區塊層級元素)

- `<div>`、`<h1>~<h6>`、`<p>`、``、``、``、`<table>`、`<form>`、`<section>`、`<article>`
- 元素的內容在瀏覽器中會另起一行

› Inline level(行內層級)

- ``、`<i>`、``、``、`<a>`、`<sub>`等
- 元素在瀏覽器中不會另起一行

BOX 元素的顯示層級

- › HTML已經有預設的顯示層級，但是我們還是可以用CSS來加以變更
- › display: 改變元素的顯示層級或特性
 - block: 將元素設定為區塊層級，我們可以設定其寬度、高度、留白與邊界
 - Inline-box: 將元素設定為行內層級，無法設定其寬度、高度，元素的寬高由它的內容撐開，可設定留白和邊界，但實際上會影響版面的只有水平方向
 - Inline-block: 令元素設定為像行內層級一樣不換行，但是我們可以設定其寬度、高度、留白與邊界
 - none: 不顯示
 - flex: 令元素依照Flexbox Model編排內容
 - grid: 令元素依照grid model編排內容

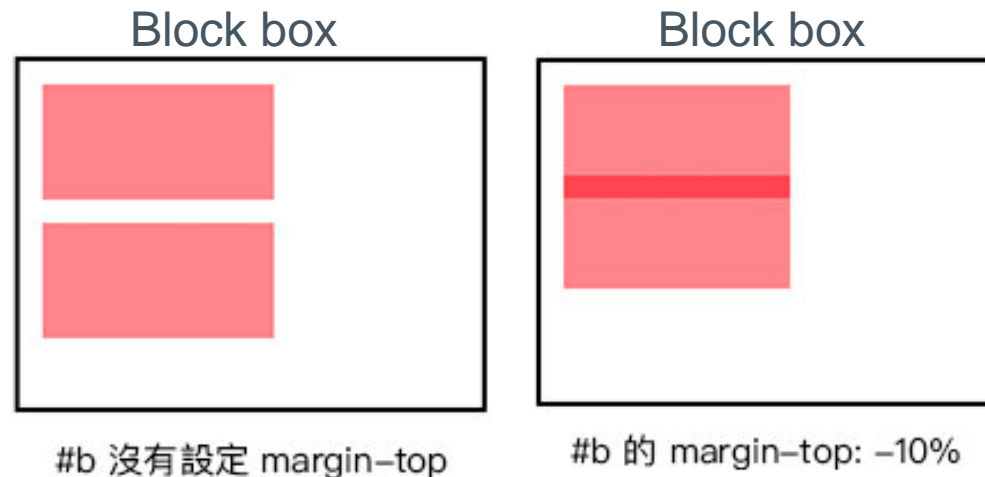
BOX的定位方式

BOX定位方式

- › position: static | relative | absolute | fixed
- › static: 正常順序
- › relative: 相對定位，也就是相對於正常順序來定位
- › absolute: 絕對定位
- › fixed: 固定定位，屬於絕對定位方式的另一種形式，差別在於box會顯示在固定的位置，不會隨著內容捲動

Box定位方式 - 正常順序

- › Block box的位置會取決於它在HTML程式碼中出現的順序，並且根據垂直順序一一顯示，而block box彼此之間的距離是以其上下邊界來界定



Box定位方式 - 正常順序

› 父元素和子元素皆為block box的定位

– 元素寬度預設占滿父元素寬度100%

– 子元素會顯示在父元素內

- › 根據指定的寬度和高度來顯示
- › 沒有指定就是預設佔滿父元素的寬度



```
<div class="parent">  
  <div class="child"> </div>  
</div>
```

```
.parent {  
  height:200px;  
  width:200px;  
  background-color: #A3AADE;  
}  
.child {  
  height:70px;  
  background-color: pink;  
}
```

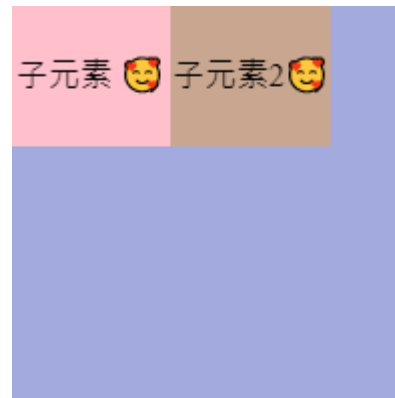
Box定位方式 - 正常順序

- › Inline box的位置會依照水平方向排列，而inline box彼此之間的距離是以左右留白、左右框線和左右邊界來界定
 - 圖中是有設定padding和margin所以整個inline box會被上下撐開，但不會影響排版

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Temporibus perspiciatis incidunt Necessitatibus, voluptatem dolorum rem animi commodi labore ullam. Mollitia, dolorem! Inventore, omnis iusto dolor aspernatur provident deserunt fuga neque labore atque eos doleniti eulae, edit illum exaniet accusantium quem, quos sint ead blanditiis! Dolores et ratione Obcaecati animi saepe sint laudantium? Earum voluptatum iusto ducimus quis tempora provident sit veritatis ullam rem Maxime vel nam ipsam explicabo voluptas quos optio nostrum fuga aspernatur aliquid, eligendi, rem necessitatibus molestiae ipsum ipsa! Necessitatibus magni doloribus cumque ipsum.

Box定位方式 - 正常順序

- › 父元素為block box而子元素為inline box的定位
 - 子元素會顯示在父元素內
 - › 根據指定的寬度水平來顯示
 - › 如果第二個子元素顯示時會超過父元素的寬度就會換行



```
<div class="parent">  
  <div class="child"> </div>  
</div>
```

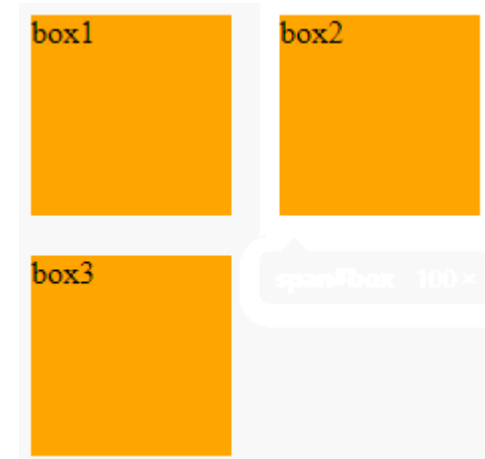
```
.parent {  
  height:200px;  
  width:200px;  
  background-color: #A3AADE;  
}  
.child {  
  width:80px;  
  display:inline-block;  
  background-color: pink;  
}
```

Box定位方式 - 正常順序

› inline-block

- 像 inline 可以併排
- 像 block 可以調整各種屬性
- inline-block 跟 block 最大的差異就在於 inline-block 可以併排

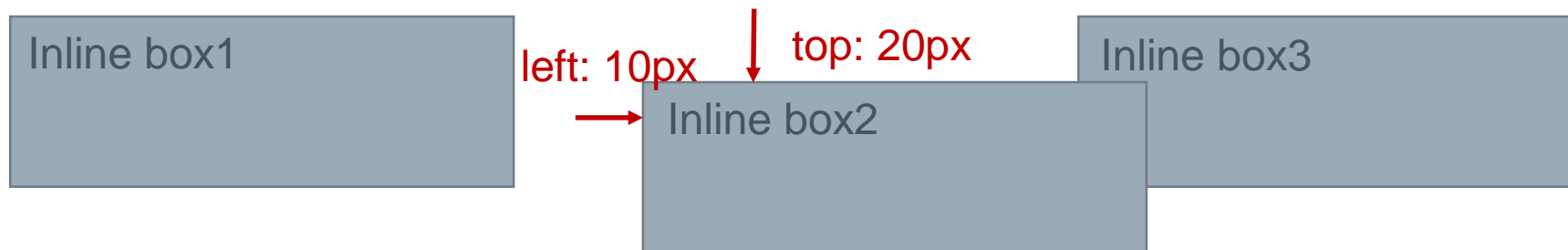
```
#box {  
  width: 100px;  
  height: 100px;  
  background: orange;  
  margin: 10px;  
  display: inline-block;  
}
```



Box定位方式 - 相對定位

- › 可以用以下的屬性來設定box在畫面中上下左右的位移量
- top: 長度 | 百分比 | auto
 - right: 長度 | 百分比 | auto
 - bottom: 長度 | 百分比 | auto
 - left: 長度 | 百分比 | auto

```
#inlineBox2{  
    position : relative;  
    top: 20px;  
    left: 10px;  
}
```



Box定位方式 - 絕對定位

- › position: absolute;
- › 會將html元素的box從正常順序中抽離出來，顯示在指定的位置上，而其它依照正常順序顯示的元素會當絕對定位的元素不存在
- › 可以用top、right、bottom、left屬性來做定位
 - 水平或垂直位移長度
 - top: 長度
 - right: 長度
 - bottom: 長度
 - left: 長度

Box定位方式 – 固定定位

- › position: fixed;
- › 屬於絕對定位的一種方式，和絕對定位的差別在於HTML元素的box會顯示在固定的位置，不會隨著內容捲動
- › 可以用top、right、bottom、left屬性來做定位
 - 水平或垂直位移長度
 - top: 長度
 - right: 長度
 - bottom: 長度
 - left: 長度

Z-index重疊順序

- › z-index屬性用來設定HTML元素的重疊順序
- › z-index: auto | 整數
 - auto為預設
 - 數字較大的會顯示在較上面的層面

行內層級元素的對齊

› vertical-align:

- baseline→ 將元素的基準線對齊父元素的基準線(預設)
- sub→ 將元素的基準線對齊父元素的下標基準線
- super→ 將元素的基準線對齊父元素的上標基準線
- text-top→ 將元素的頂端對齊父元素字型的頂端
- text-bottom→ 將元素的底部對齊父元素字型的底部
- middle→ 將元素的中間對齊父元素的中間
- top→ 將元素的頂端對齊整行元素的頂端
- bottom→ 將元素的底部對齊整行元素的底部
- 長度: 將元素往上或下移指定的長度
- 百分比: 將元素往上或下移指定的百分比

Visibility 顯示或隱藏box

- › visibility: 這個屬性可以設定要顯示或隱藏box
- › visibility: visible | hidden | collapse
 - visible→ 顯示
 - hidden→ 隱藏
 - collapse→ 隱藏表格的行列

CSS Display Table

- › Display有提供類似HTML表格元素的顯示方式
 - table：定義一個會產生類似表格的區塊。(table)
 - Inline-table：定義一個會產生行內層級的表格區塊。
 - table-row：表示元素是一列表格基本格。(tr)
 - table-row-group：宣告元素是一個以上的表格列形成的元素群組。(tbody)
 - table-header-group：會顯示於列群組的上方。(thead)
 - table-footer-row：會顯示於列群組的下方。(tfoot)
 - table-column：宣告元素是一行的表格行。(col)
 - table-column-group：宣告元素是一個以上的表格行形成的群組。(colgroup)
 - table-cell：表示一個基本格。(th或td)
 - table-caption：表示為表格的標題。(caption)

HTML Table vs CSS Display Table

› HTML表格

```
<table>
  <tr>
    <td> 標題 </td>
    <td> 內容 </td>
  </tr>
  <tr>
    <td> 文章標題 </td>
    <td> 文章內容 </td>
  </tr>
</table>
```

› CSS display表格

```
<div id="css_table">
  <div class="css_tr">
    <div class="css_td"> 標題 </div>
    <div class="css_td"> 內容 </div>
  </div>
  <div class="css_tr">
    <div class="css_td"> 文章標題 </div>
    <div class="css_td"> 文章內容 </div>
  </div>
</div>
```

```
<style>
  #css_table {
    display:table;
  }
  .css_tr {
    display: table-row;
  }
  .css_td {
    display: table-cell;
  }
</style>
```

float定位方式

- › float: 這個屬性可以將一個正常順序中的元素放在容器的左側或右側
- › float: none → 預設
- › float: left → 將元素放在容器的左側
- › float: right → 將元素放在容器的右側
- › 用這個屬性可以做到類似文繞圖的效果

練習

- › 運用CSS Box Model重新定位你的論文瀏覽網頁架構，和CSS.ppt長的一樣
 - 在你的頁面內顯示一張紙的容器，這個不用設框線，大小可以自行設定(類似文章翻頁的效果)
 - 每一個框都是一個box，每個box的大小和排版的位置盡量要和ppt內的設定接近，邊界留白等可自行調整，每個box都要設定框線，框線粗度自行設定
 - Index Terms做絕對定位(fixed)在最上面
 - 表格用CSS的方式重新製作