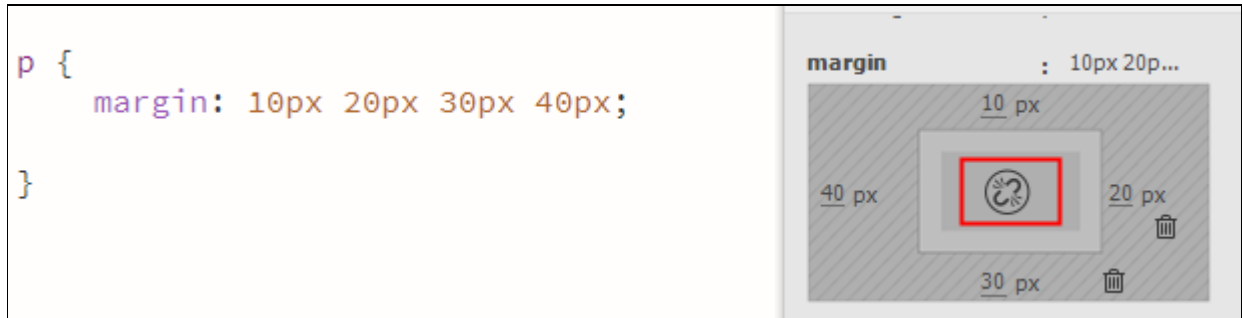


## Margin、Padding 單行寫法

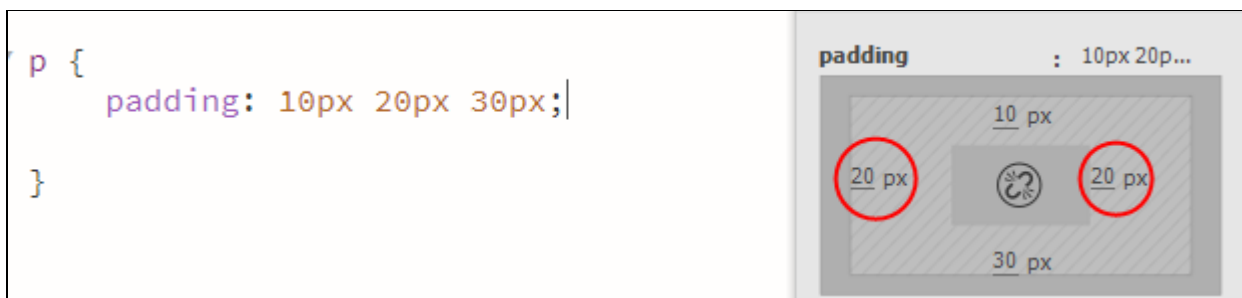
Margin(或 padding): 上 右 下 左 ; (四個值不同)

EX / Margin : 10px 20px 30px 40 px;



Margin(或 padding): 上 左右 下 ; (左右值相同、上下值不同)

EX / Padding : 10px 20px 30px;



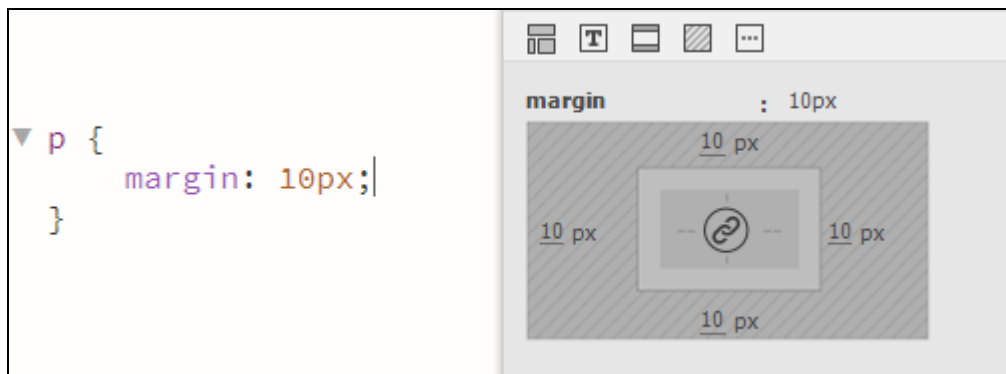
Margin(或 padding): 上下 左右 ; (上下值相同、左右值相同)

EX / Padding : 10px 20px;



Margin(或 padding): 上下左右 ; (四個值均相同)

EX / Margin : 10px;



## CSS 的大於(>)、加號(+)、取代符號(~)

有時我們會在 CSS 裡看到「>」、「+」、「~」這些符號的使用，它們到底有甚麼功用呢？

### 大於符號(>)

大於(>)就是選取底下**直接**的子元素。

以大於符號(.box > p)和空格(.box p)來做比較，先看以下例子：

```
<div class="box">
  <p>001</p>
  <div>
    <p>002</p>
    <p>003</p>
  </div>
  <p>004</p>
</div>
```

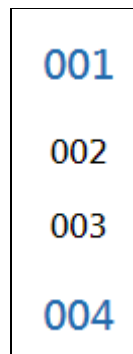
### 使用空格的情況

.box p{ font-size:20px; color:#216AAD;}

001  
002  
003  
004

## 使用大於符號的情況下

`.box > p{font-size:20px; color: #216AAD;}`



由上可知`.box p`會影響到 `box` 這個 `div` 底下所有的 `p` 元素，而因為「`>`」只會影響到直接的子元素，所以例子中 `box` 底下直接接觸到的 `p` 元素只有 `001` 跟 `004`，所以只有這兩者會改變。

## 加號符號(+)則是會影響到後方同層級的第一個元素

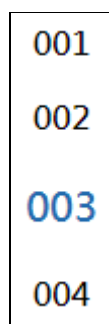
範例：

html

```
<div>
  <div>001</div>
  <div><p>002</p></div>
  <p>003</p>
  <p>004</p>
</div>
```

CSS

`div + p{ font-size:20px; color:#216AAD;}`



由於跟 div 同層級又直接位於 div 後方的只有 003，所以只有 003 改變。

而取代符號(~)是影響到後方同層級的**全部**元素，因此除了 003 以外 004 也會改變。

```
div ~ p{ font-size:20px; color:#216AAD;}
```

001
002
003
004

## 網頁文字單位 - pt、px、%、em、rem

網頁的文字單位大致上分成: pt、px、%、em、rem

pt、px 為最常見的，屬於絕對單位；%、em、rem 則為相對單位。

%單位的算法：

預設網頁上的文字大小為 16px=12pt，

假如設定 font-size: 75%，意思就是將預設文字大小改成：16 X 0.75=12px

假如設定 font-size: 0.75em，意思就是將預設文字大小改成：16 X 0.75=12px

那麼 em、rem 看起來很像的兩者，又有什麼差異呢？

**em** 是以父元素為基準的文字單位，適合使用在限制區塊的內容，例如：ul li、p、div 等等，em 可以快速地从父元素改變內容文字的大小。

```
.em-demo p {  
  
    font-family:微軟正黑體; font-size: 2em;  
  
    /*文字大小：16px * 2 = 32px，因為其父元素是 <body> 標籤預設值 16px。*/  
  
}  
  
.em-demo p span{  
  
    font-size: 0.5em;  
  
    /*文字大小：32px * 0.5 = 16px，因為其父元素是 <p> 標籤。*/  
  
}
```

**rem** 的文字則是以根元素為基準，而根元素就是指標籤。適合用在整體網頁的尺寸切換，可以依據不同的螢幕尺寸，統一改變網頁全部的文字大小。

```
.rem-demo p {  
  
    font-family:微軟正黑體; font-size: 2rem;  
  
    /*文字大小：16px * 2 = 32px，因為其父元素是 <body> 標籤預設值 16px。*/  
  
}  
  
.rem-demo p span{  
  
    font-size: 0.5rem;  
  
    /*文字大小：16px * 0.5 = 8px，因為其父元素是 <p> 標籤預設值 16px。*/  
  
}
```

## CSS 選擇器的優先權順序

CSS 提供許多選擇器類型，當使用不同的選擇器時，套用樣式的優先權也會不同。

### A.元素內的樣式>頁面內的樣式>外部載入

#### 元素內的樣式

例子：

```
<div class="text" style="width:200px;height:50px;color:#fff;"></div>
```

#### 頁面內的樣式(嵌入樣式)

例子：

```
<style type="text/css">
.box{
width:400px;
height:100px;
background:#000;
width:250px;
height:50px;
background:#fff;
}
</style>
```

#### 外部載入

例子：

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

### B.後設定>前設定

最後設定的樣式將蓋過之前設定的樣式

例子：

```
<style type="text/css">
.box{
width:400px;
height:100px;
background:#000;
width:250px;
height:50px;
background:#fff;
}
</style>
```

最後會套用的屬性是 width:250px; height:50px; background:#fff;

## C. 名稱 Id > class

由於 id 具有不可重複性，而 class 則是可重複的，所以 id 在層級上大於 class，如果一個元素同時有 id 跟 class 兩種屬性，且互有所衝突時，id 優先權大於 class。

```
<!-- Service Grid Section -->
<section class="service" id="service">
  <div class="container py-3">
```

## D.使用 !important 為最高優先

!important 的作用為強制提高優先權，照理說越後面設定的越優先，但若使用了 !important，會使得最前面的設定成了最優先，使用!important 的 CSS 樣式幾乎等於無敵。

```
.bg-primary {
  background-color: #007bff !important;
}
```

當出現二個!important 時，優先權就是依先後順序，與其它的選擇器的原理相同，因此非必要的話，最好盡量少用!important。