**Html 元素分類**

**block 區塊元素:**

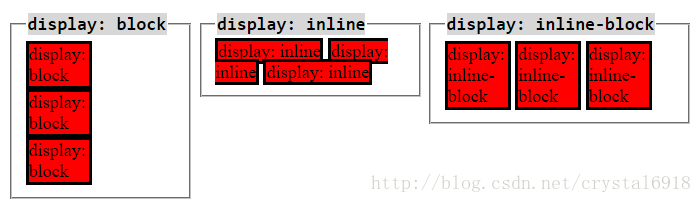
元素寬度預設會撐到最大，使其占滿整個容器可以設定長寬margin、padding，但仍會占滿一整行

**inline 行內元素(內聯元素):**

元素可在同一行內呈現，圖片或文字均不換行，也不會影響其版面配置，不可設定長寬，元素的寬高由它的內容撐開，幾乎不能設定margin/padding

**inline-block行內的區塊:**

對內表現成block對外表現成inline，以inline的方式呈現，但同時擁有block的屬性．可設定元素的寬高/margin/padding．可水平排列



**Html 元素嵌套(https://www.programmersought.com/article/10661886530/)**

1. 區塊元素可以包含行內元素
2. 區塊元素不一定能包含區塊元素
3. 行內元素一般不能包含區塊元素 (a>div)

**Html 元素默認樣式**

若不想使用默認樣式，可以上網找CSS reset的樣式來重置樣式

快速重置法

<style>

    \*{

      margin: 0;

      padding: 0;

    }

</style>

**真題**

1. Doctype的意義是什麼
   1. 讓瀏覽器以標準模式渲染
   2. 讓瀏覽器知道元素合法性
2. HTML5 有什麼變化
   1. 新的元素
   2. 表單增強
   3. 新的api (圖形、影音、實時通訊、本地儲存、設備能力(手機應用))
   4. 分類及嵌套規則變更
3. 語意化的意義
   1. 開發者容易理解
   2. 機器容易理解(SEO)
   3. 可以快速抓到網頁架構和每個區塊的位置
4. 那些元素在H5可以可以自閉合
   1. Input
   2. Img
   3. br hr
   4. meta link
5. HTML和DOM的關係
   1. HTML是藍圖
   2. DOM是解析HTML後的產物
   3. JS可以使用DOM
6. property和attribute的區別
   1. attribute是藍圖
   2. property是DOM解析之後，是實際的值

**CSS選擇器**

元素選擇器 a{}

偽元素選擇器 ::before{} ::after{}(基於原有的DOM，創建另一個不同css的DOM)

類別選擇器 .class{}

屬性選擇器 [type=radio]{}

偽類別選擇器 :hover{} (用於某狀態下)

ID選擇器 #id{}

組合選擇器 [type=checkbox] + label{}

否定選擇器 :not(.link){}

通用選擇器 \*{}

範例

偽類別選擇器 a:link{ color: red;} =>tag a 沒被點擊過時

組合選擇器 ul + p { color: red; } => 選擇ul下相鄰的第一個p

**選擇器權重**

最高類 !important => 加上important的元素不管誰都不可以覆蓋

第一類 ID選擇器 +100

第二類 類別 屬性 偽類別 +10

第三類 元素 偽元素 +1

第四類 其他 +0

若選擇器要修改同一個DOM就必須算權重來決定誰疊加在上層，同樣權重下後寫的生效

例如:

#id .link a[herf] => #id+100 .link+10 a+1 [href]+0 = 111

#id .link.active => #id+100 .link+10 .active+10 =120

有上可計算出第2列的權重較高，所以第2列會優先套用

特別規定 : 每一類的權重是不進位的所以第一類永遠大於第二類以此類推

類如 11個第二類選擇器權重為 0 11 0 不會因為大於10而進位

**非布局樣式**

字體、字粗、顏色、大小、行高

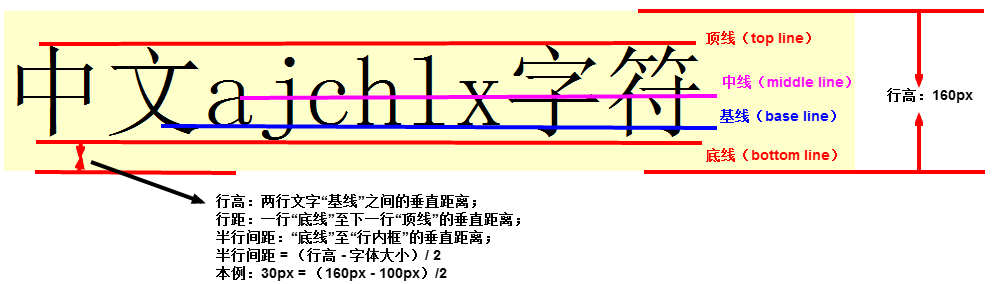
背景、邊框

滾動、換行

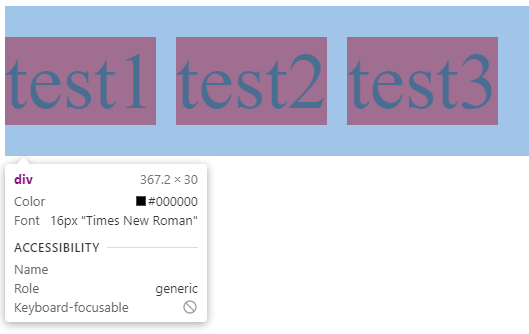
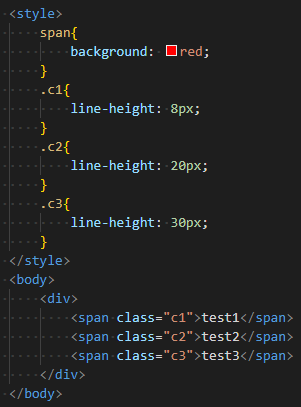
粗體、斜體、下劃線

其他

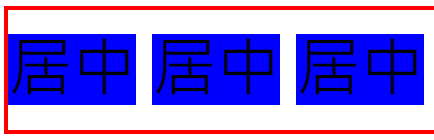
行高(line-height)



line-height 會把外層元素撐大，但不會影響本身內層顯示，內層顯示大小必須用font-size來調整



若line-height 大於font-size，可以實現另類垂直居中的效果



文字對齊正常是以基線來排

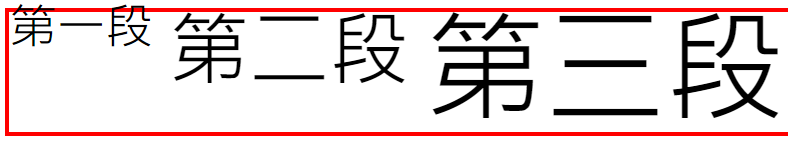


若要設置對齊方式需用vertical-align(top,middle,bottom,?px)

居中vertical-align:middle



頂部vertical-align:top，以頂線對齊



頂部vertical-align:bottom，以底線對齊



圖片因為屬於inline結構所以會以基線對齊，所以會導致底部出現的空白



所以必須設置vertical-align:bottom，使其以底線對齊



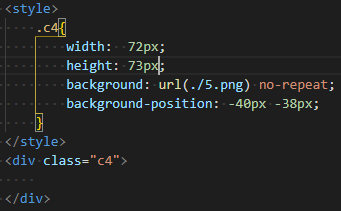
也可以直接使用display:block將inline屬性改變，但這樣會導致換行



背景(background)

雪碧圖



靠著長寬及圖片的偏移就可以產生一圖多用的效果

**CSS Hack**

在某些瀏覽器下才會生效的CSS，為不合法但生效的寫法

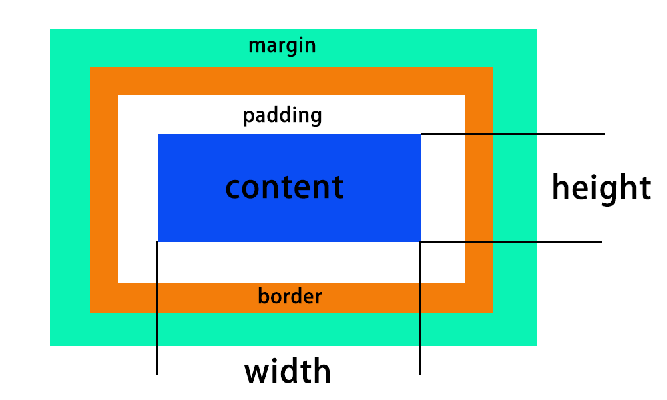
主要用於區分瀏覽器，但難維護、易失效

替代方案:特性檢測、針對性增加class

**真題**

**CSS 布局**

盒模型



display/position

display => 確定元素顯示類型 block/inline/inline-block/flex

position => 確定元素位置 static/relative/absolute/fixed

position:static => 依正常位置排列

position:relative => 相對元素本身正常位置來偏移搭配left top等偏移元素使用，不影響正常  
 布局的範圍及計算但在正常布局中也佔有位置，不會因為他而改變其他正  
 常布局

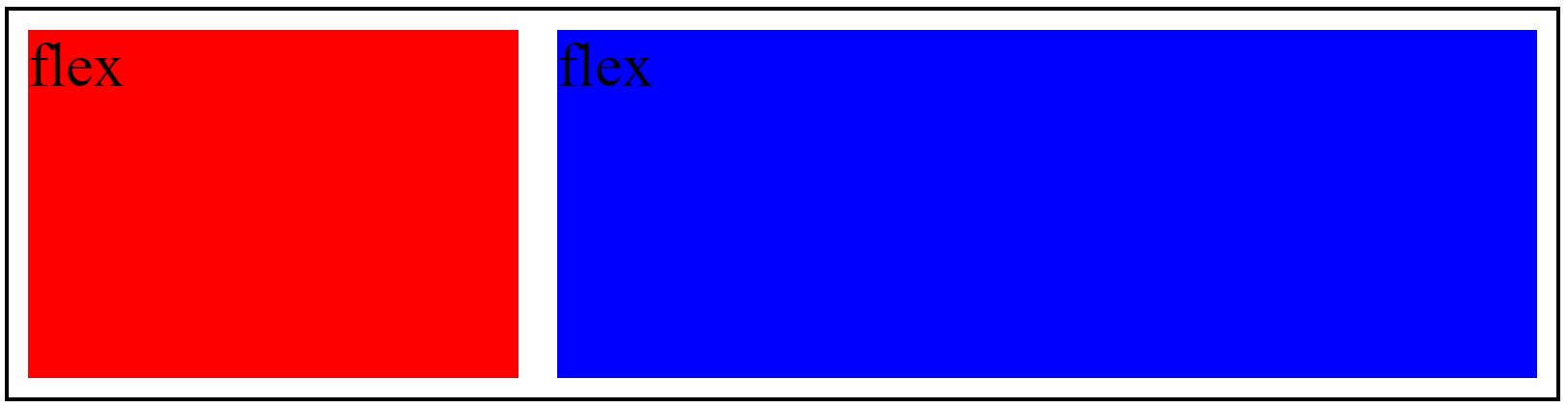
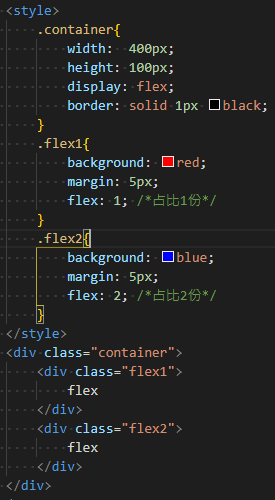
position: absolute => 絕對定位，相對於最近的父級，脫離正常布局計算完全獨立出來，不會  
 影響其他正常布局，搭配left top等偏移元素使用

position: fixed => 相對於可視區域，脫離正常布局計算完全獨立出來，不會影響其他正常布局  
 搭配left top等偏移元素使用

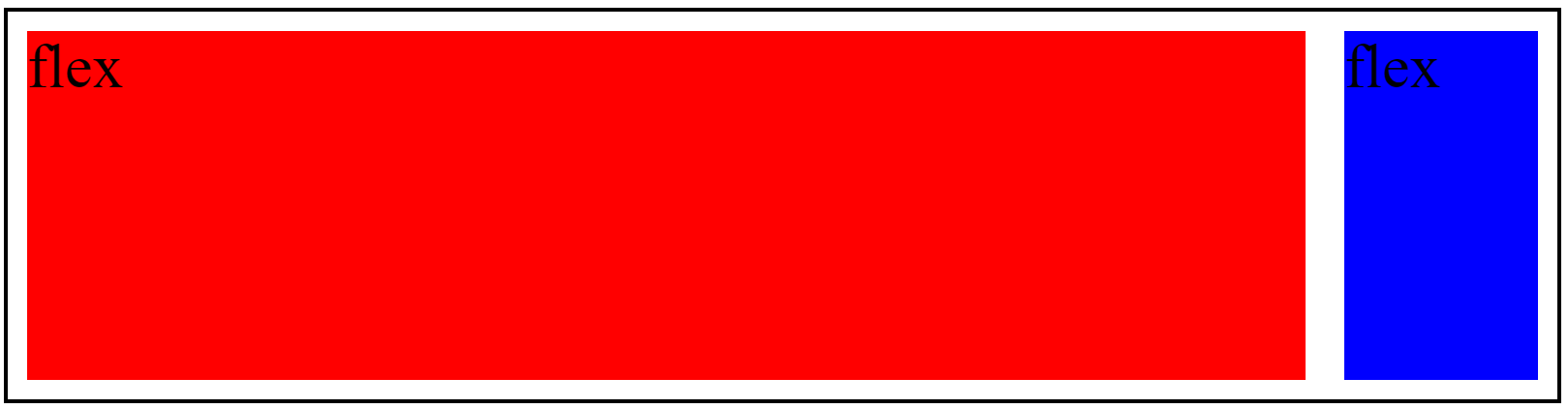
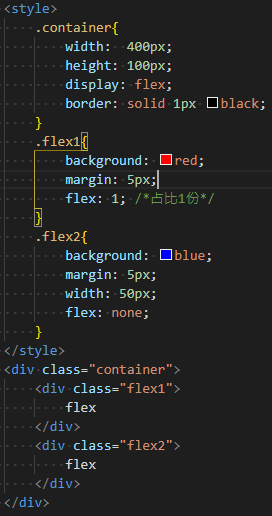
層級依渲染順序或使用z-index來設定層級

display:flex(彈性盒子)(有兼容問題)

本身為並列，只需指定寬度即可

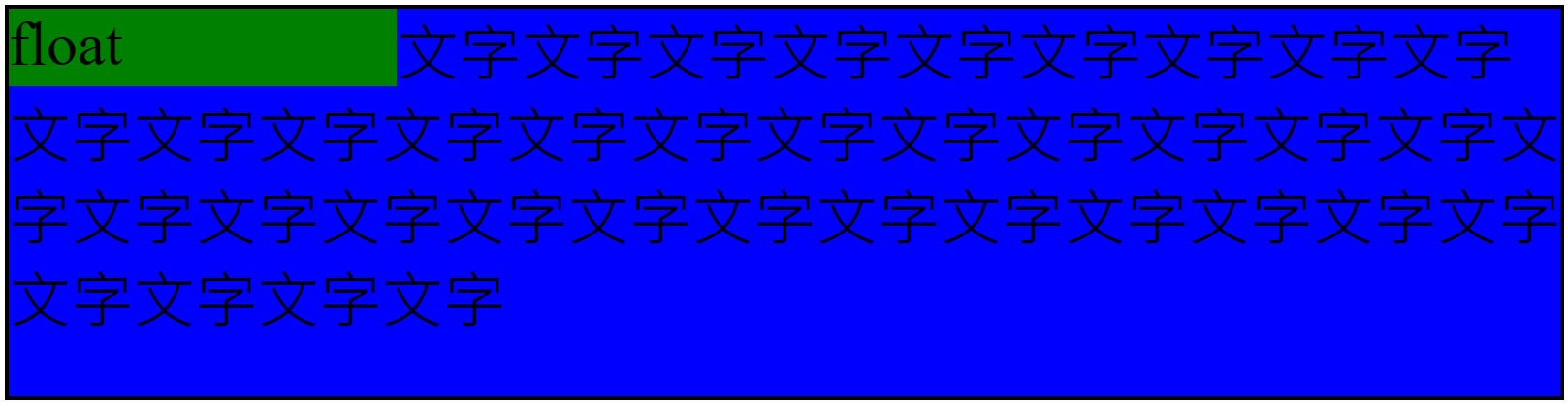
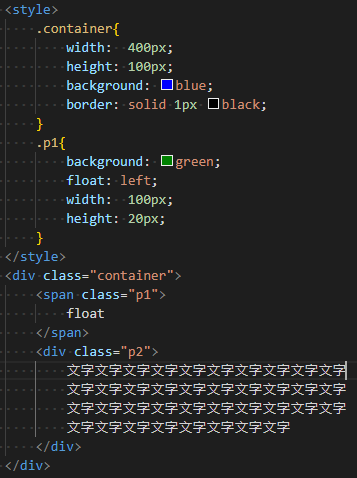
 

若想自寬度需設置flex:none 及width參數

float(浮動)(兼容性高)

讓元素浮動且脫離正常布局計算完全獨立出來，但會影響其他元素文字的布局

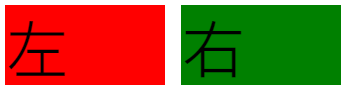
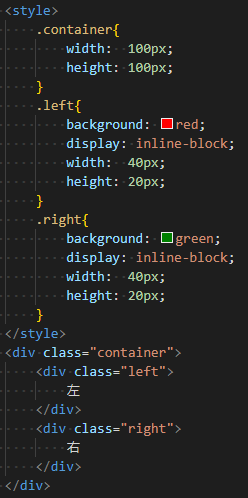
float 會將元素轉成區塊形元素，會依長寬盡量靠上且依浮動方向盡量靠左或右

因為float脫離正常布局所以父級元素會無視而導致父級元素塌陷

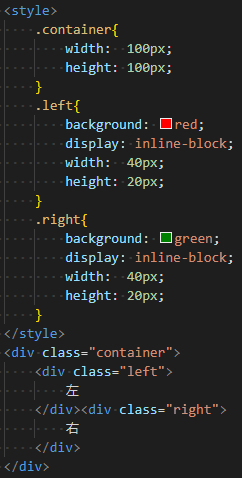
可以將父級元素加上overflow來解決此問題或使用::after及clear:both來新增元素用來撐大塌陷

Inline-block

間隙問題

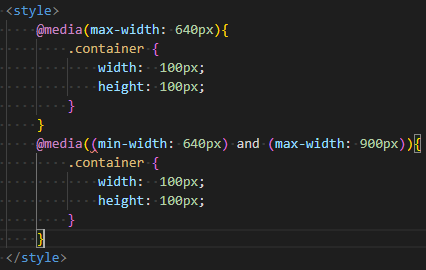
中間的間隙是HTML判斷文字之間有空白，所以解法如下

**響應式設計**



Initial-scale為放大倍數



使用media設定寬度多少必須使用哪個css

**CSS預處理器(less sass)**

基於CSS的另一種語言，通過工具編譯成CSS，添加了許多CSS不具備的屬性

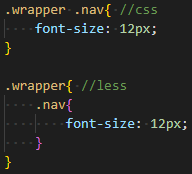
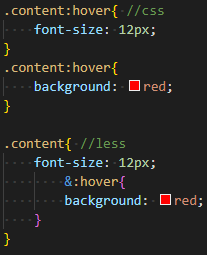
，能提升CSS文件的組織，減少重複代碼

less => JS撰寫，編譯速度快

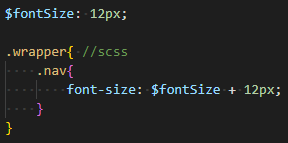
sass(scss)=>ruby撰寫，複雜特性較好撰寫

但大部分撰寫語法一樣

預處理器寫法可以直接嵌套子層

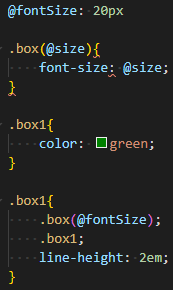
變數



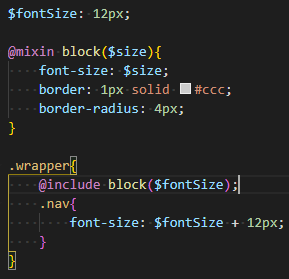
less變數使用@

mixin

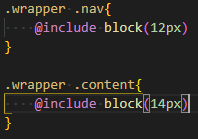
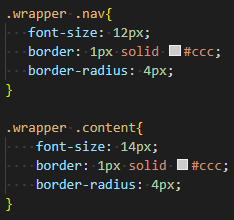
less => 可以把class當mixin用



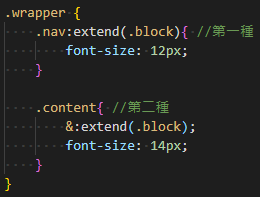
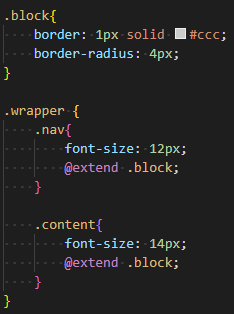
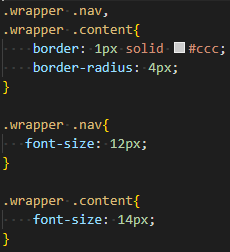
scss => 必須明確定義mixin



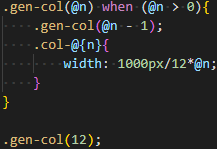
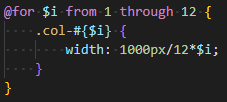
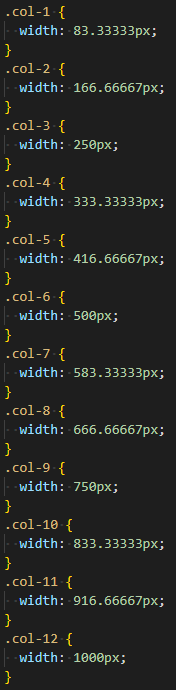
但mixin會導致CSS公用部分重複撰寫

須使用Extend來撰寫

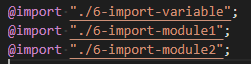
  

loop

Import(模塊化)

less與scss語法一樣 若引用正確變數可以跨文件使用



**Bootstrap客製化**

1. 使用同名類覆蓋 => 會破壞Bootstrap計算好的style，要修改很多地方
2. 修改原始SCSS後重新建構

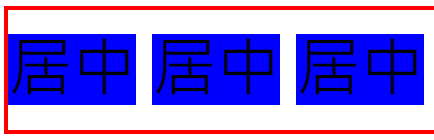
=> 須了解Bootstrap源碼結構，但改動後Bootstrap會自動計算

1. 使用MIXIN方式引用Bootstrap Scss於自定義SCSS文件中使用並修改

=> 須非常了解Bootstrap源碼結構，可以精簡引用的代碼且還能使用Bootstrap自動計算後的CSS

**垂直居中**

1. line-height 大於font-size，可以實現另類垂直居中的效果



**水平居中**

margin: 0 auto;