

Build School 課程 – SASS、RWD

Build School 助教
王維駿

SASS

- <https://sass-lang.com/>
- 以程式思維使CSS更易撰寫與維護
- 更多強大功能
- 已被廣泛使用且持續發展中
- 無法直接載入使用，需先編譯成CSS

SASS語法

- SASS

- 無大括號和分號
- 需非常注意縮排
- 附檔名是.sass

```
p  
  color : #ddd;  
  font-size : 20px;
```

- SCSS

- CSS語法完全相容
- 副檔名是.scss

```
p {  
  color: ■ #ddd;  
  font-size: 20px;  
}
```

開始使用SASS

- <https://marketplace.visualstudio.com/items?itemName=riktwicky.live-sass>
- 更改原有CSS檔案的副檔名".css"為".scss "
- 點擊"Watch SASS"

Nesting

- 節省重複輸入
- 結構清楚一目瞭然
- 最好不超過三層，否則
易因權重增加而造成
難以覆寫的困境

```
div{  
  p {  
    color: ■ #ddd;  
    font-size: 20px;  
  }  
  background-color: ■ black;  
  height: 100px;  
  width: 100px;  
}
```

Variables

- 使用\$符號後面加上名稱來建立
- 可重複使用
- 可存放任意值

```
$main-color : ■ #ff9203 ;  
  
.main{  
    background-color: $main-color;  
}
```

Partials & Import

- **Partials**
- 拆分SASS檔案可使結構更加清楚且便於維護
- 在SASS檔案的名稱前若帶有底線_，該檔案將不會被直接編譯成CSS檔案
- **Import**
- 在SASS檔案中使用@import "檔名 (不用副檔名)" 就可以載入該檔案的內容，若載入的是Partials SASS檔，則檔名的底線_可省略。

檔案_allcolor.scss

```
@import "allcolor";
```



Extend

- 相同的CSS屬性重複使用
- 使用%開頭來撰寫CSS屬性將不會被直接編譯出來
- 使用@extend 名稱 可引入該段規則之內容

Extend

```
%btn {  
  color: red;  
  margin: 10px;  
  height: 30px;  
  width: 20px;  
}  
  
.btn-A {  
  @extend %btn;  
  padding: 5px;  
}  
  
.btn-B {  
  @extend %btn;  
  border: 2px purple;  
}
```

```
.btn-A, .btn-B {  
  color: red;  
  margin: 10px;  
  height: 30px;  
  width: 20px;  
}  
  
.btn-A {  
  padding: 5px;  
}  
  
.btn-B {  
  border: 2px purple;  
}
```

```
.btn {  
  color: red;  
  margin: 10px;  
  height: 30px;  
  width: 20px;  
}  
  
.btn-A {  
  @extend .btn;  
  padding: 5px;  
}  
  
.btn-B {  
  @extend .btn;  
  border: 2px purple;  
}
```

```
.btn, .btn-A, .btn-B {  
  color: red;  
  margin: 10px;  
  height: 30px;  
  width: 20px;  
}  
  
.btn-A {  
  padding: 5px;  
}  
  
.btn-B {  
  border: 2px purple;  
}
```

Mixins

- 當一段CSS屬性需要經常重複使用，但使用時又需依狀況填入不同值，便可用Mixin來處理
- 使用@mixin 名稱(變數) 宣告CSS屬性，再用@include 名稱(變數) 取用

```
@mixin Square($width, $height, $color) {  
  width: $width;  
  height: $height;  
  color: $color;  
  border: 1px solid black;  
}  
  
.box {  
  @include Square(40px, 50px, blue);  
}
```



```
.box {  
  width: 40px;  
  height: 50px;  
  color: blue;  
  border: 1px solid black;  
}
```

Parent References

- 用&符號能表示所處位置的父層selector

```
li {  
  display: flex;  
  flex-basis: auto;  
  margin-bottom: 10px;  
  
  &:last-child {  
    margin-bottom: 0px;  
  }  
  & > a {  
    flex: 0 0 50px;  
    margin-right: 10px;  
  }  
}
```

加減乘除

- 可使用+ - * /等符號直接對宣告的值做計算

```
.main{  
  padding: 0 0 (1080px - 60px)/3;    // 0 0 340px  
}
```

調整顏色

- Darken(色碼, 百分比) 調深顏色
- Lighten(色碼, 百分比) 調亮顏色
- Invert(色碼) 取得互補色
- Mix(色碼1, 色碼2, 色碼1的混和佔比) 兩個色碼的混合色

<https://jsfiddle.net/dg75h94t/5/>

RWD

- Responsive Web Design
- 響應式網頁
- 在不同尺寸的裝置開啟網頁，都能有良好的呈現內容
- 常見的製作策略
 - Desktop first
 - Mobile first

實作RWD

- 變更畫面寬度並決定斷點
- 注意圖片大小對容器的影響
- 調整畫面寬度時要考慮呈現方式
- 可適度增加容器或元素以利排版
- 改變內容的呈現方式符合容器、觀看感覺
- 在不同寬度下依需求顯示/隱藏內容
- 善用jQuery