[**版本紀錄**](#_heading=h.30j0zll) **2**

[**命名原則\_Naming Rules**](#_heading=h.3znysh7) **3**

[Sass / SCSS 檔名命名規則](#_heading=h.2et92p0) 3

[Sass / SCSS 變數命名規則](#_heading=h.tyjcwt) 3

[Sass / SCSS 樣式命名規則](#_heading=h.3dy6vkm) 3

[**格式\_Formatting**](#_heading=h.1t3h5sf) **4**

[透過 WebStorm 設定 CSS / Scss 自動排版](#_heading=h.4d34og8) 4

[**程式碼結構\_Source File Structure**](#_heading=h.2s8eyo1) **6**

[樣式撰寫原則](#_heading=h.17dp8vu) 6

[編輯器文件撰寫規範](#_heading=h.3rdcrjn) 7

[**註解**](#_heading=h.26in1rg) **8**

[時機及規則](#_heading=h.lnxbz9) 8

[**其他規範**](#_heading=h.1ksv4uv) **9**

[保持一致性與整潔 (HTML / SCSS)](#_heading=h.44sinio) 9

[使用 Class 下去撰寫樣式，盡量不使用 inline-style (行內樣式)](#_heading=h.2jxsxqh) 9

[同一組 HTML 的樣式，多使用嵌套/巢狀的寫法](#_heading=h.z337ya) 9

[注重 Production (正式生產環境)程式碼品質](#_heading=h.3j2qqm3) 9

[**自動檢查工具**](#_heading=h.4i7ojhp) **10**

[安裝 Stylelint plugin](#_heading=h.2xcytpi) 10

[引用 .stylelintrc.json](#_heading=h.1ci93xb) 10

[Package.json](#_heading=h.3whwml4) 10

[Npm run lint](#_heading=h.qsh70q) 11

[**參考資料：**](#_heading=h.1pxezwc) **12**

[談一下 CSS Specificity](#_heading=h.49x2ik5) 12

[SCSS 物件導向模組設計 (1)](#_heading=h.2p2csry) 12

# 

# 版本紀錄

|  |  |  |
| --- | --- | --- |
| 日期 | 作者 | 修改項目 |
| 20200702 | Polly\_Lee | 初始版本 |
|  |  |  |

# 

# 命名原則\_Naming Rules

### Sass / SCSS 檔名命名規則

* 1. 不需要被編譯出來的 SCSS 檔，請在檔名前方加上 **\_ (底線)**
  2. 檔名有多個單字時，請用 - 隔開 (烤肉串寫法)
  3. 請依照頁面 or 共用元件去命名
     + AboutUs.vue ⇒ \_about-us.scss
     + Model Component ⇒ \_model.scss

### Sass / SCSS 變數命名規則

* 1. 錢字號 **$ + 小寫英文名稱**
  2. 變數名稱含有多個單字時，請用 **\_ (底線)** 隔開
     + EX：$color\_primary
  3. 命名名稱不使用縮寫，請完整寫出單字，以好辦認為主
     + EX：$font\_size (O)、$ft\_size (X)
  4. Mixin、Extend 命名含有多個單字時，請用 - 隔開 (烤肉串寫法)
     + EX：@mixin bg-setting() {}、%title-setting {}

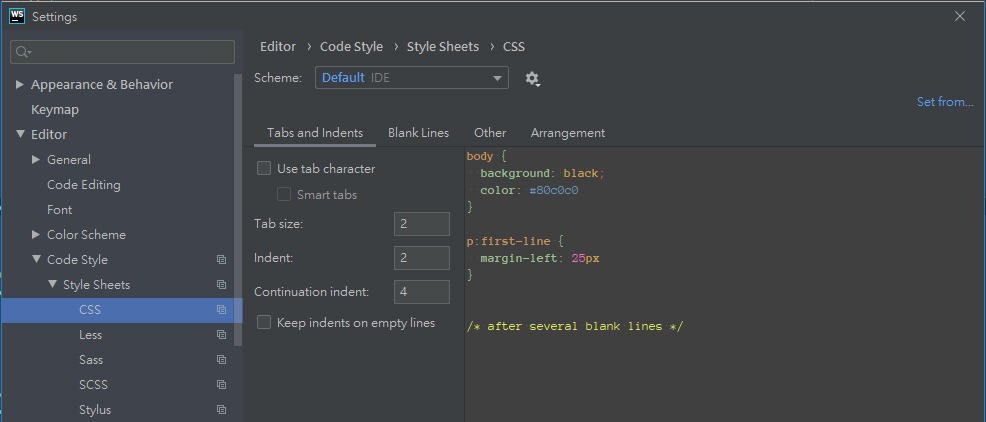
### Sass / SCSS 樣式命名規則

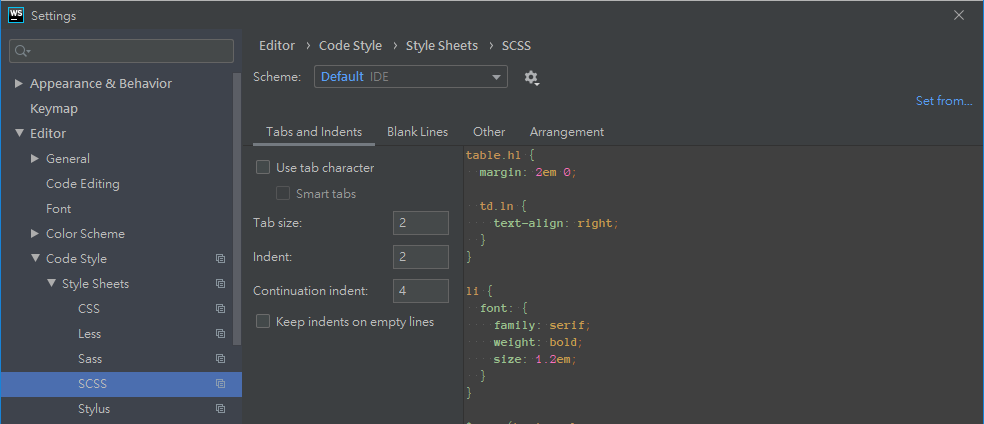
* 1. 區塊、功能性、狀態 + **\_ (底線)** + 小駝峰；*不使用 - hyphen 連字號*
     + EX：.slider\_wrapBlock、.state\_error、.album\_imgBox
  2. 可以容許最常見的名稱縮寫
     + EX：.bg\_colorBox、.btn\_primaryColor、.nav\_items
  3. 名稱需要**有意義 且 不使用縮寫**
     + .state\_error (O)、.st\_er (X)

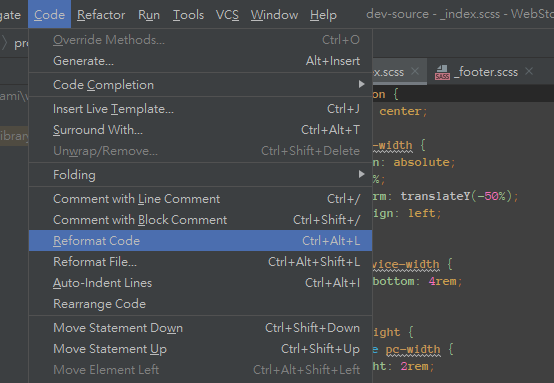
# 格式\_Formatting

### 透過 WebStorm 設定 CSS / Scss 自動排版

* 1. Windows and Linux：
     + File > Settings > Editor > Code Style > Style Sheets > CSS / Scss
  2. macOS：
     + WebStorm > Preferences > Editor > Code Style > Style Sheets > CSS / Scss
  3. 依照下方圖面修改設定



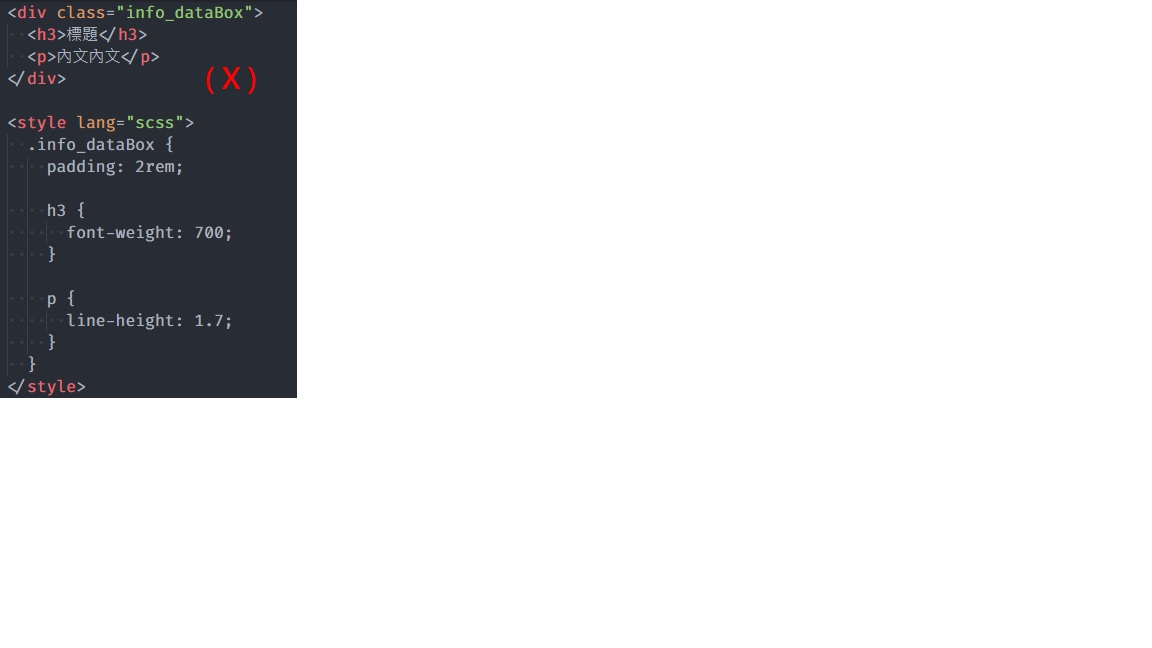
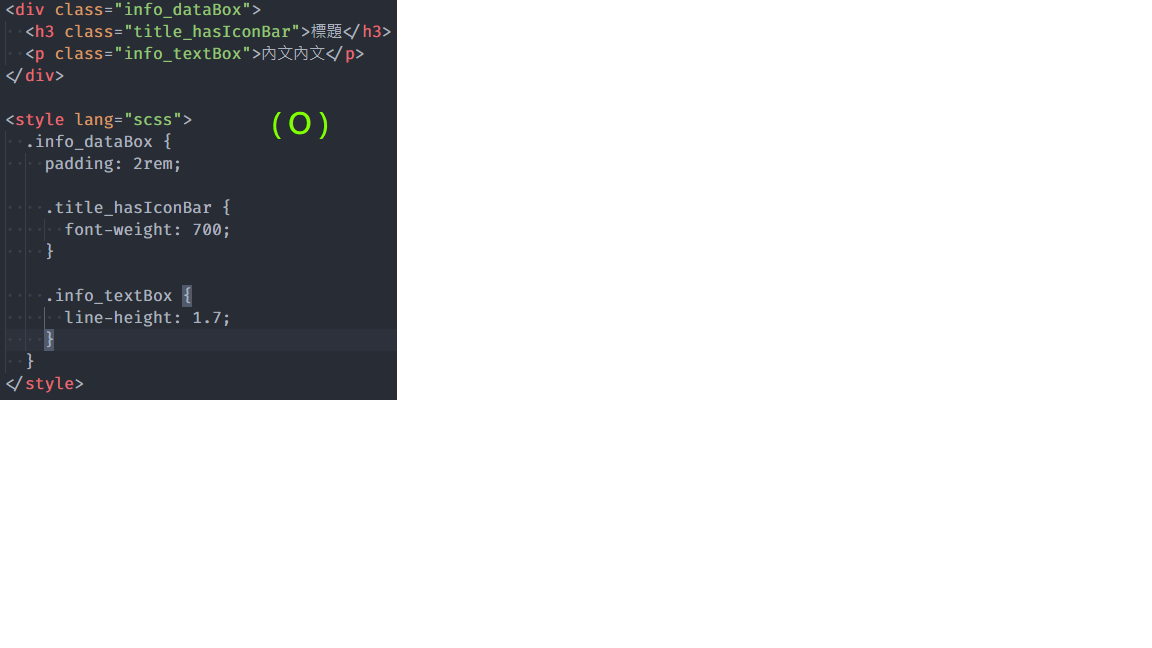


* 1. 開啟需要排版的 .css / .scss 檔案後，點擊 Code > Reformat Code
  2. 如果套用後無效果，請重啟 WebStorm
  3. 此方式可自動處理部分格式，如縮排、空格...等

# 程式碼結構\_Source File Structure

### 樣式撰寫原則

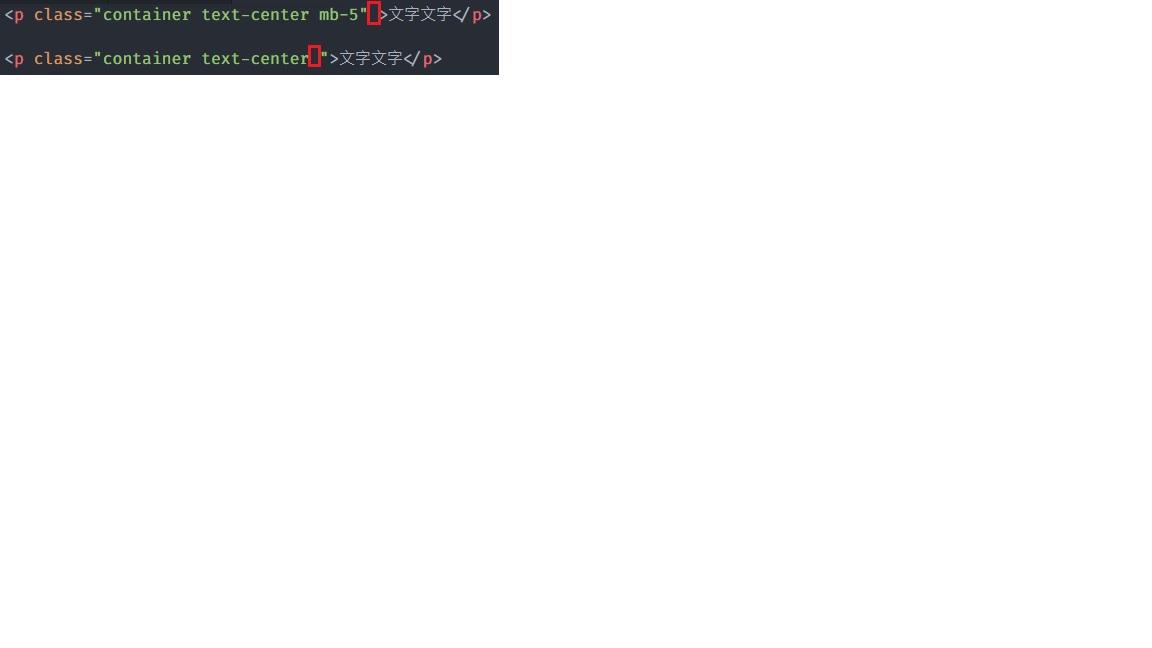
* 1. 前置準備：
     + 請學習、了解 "**CSS權重規則**" **\*Ref (1)**
     + 參考 **極少化 / 介面設計 / 開放封閉 原則** **\*Ref (2)**
  2. 觀念理解與說明：
     + 極少化原則
       - 不過於倚賴 DOM 結構，減少與 DOM 結構的耦合性 (增加複用性)
       - 撰寫樣式時請嘗試將層級攤平，減少多層的巢狀
       - 如需巢狀寫法，請勿超過 5 層
     + 介面設計原則
       - 先設計模組的介面（interface），再依照介面實踐它不同的樣  
         式
       - EX：圓形按鈕  
         先設計按鈕的架構、預設字體大小、padding... [.button]  
         在設計 形狀、固定樣式 [.button\_circle]  
         最後是 顏色、字體大小、字體粗細 [.button\_primary]
     + 開放封閉原則
       - 當有某一單元 或 特殊情況，需要有些許不一樣的樣式變化，  
         請 自訂Class 下去修改
       - EX：首頁的圓形按鈕，需要陰影  
         專為該單元 或 單一特殊情況 定義一個Class [ .button\_userShadow]
  3. 因 RWD 設定，專案皆使用 **rem** 作為尺寸單位，減少使用 px 單位
  4. 使用 Class 的方式進行樣式撰寫，除非必要，撰寫樣式不撰寫在 HTML Tag

* 1. 在撰寫或維護其他專案時，除非**已確認不會影響其他頁面**，請勿大量使用 **!important** 去暴力覆蓋樣式
  2. 屬性和值之間需要空格
  3. 不任意或強硬修改 框架 的樣式 (Boostrap 4)
  4. 注意 Class name 命名有無與框架重複

### 編輯器文件撰寫規範

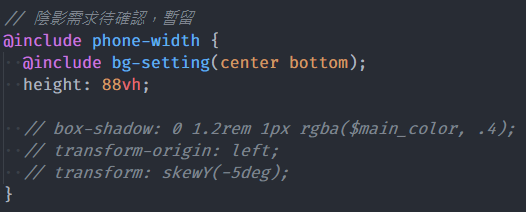
* 1. 文件縮排 (HTML / SCSS)：統一為 **2 格** 且使用 **空白 (Space)**
  2. 行內樣式、Class name 請刪除多餘空白



# 註解

### 時機及規則

* 1. 以說明特殊情況為原則，可透過清楚的 Class name 命名達到快速理解此樣式，一般不需要加註解
  2. Sass / SCSS 註解請使用 **//** 而不是 /\* \*/；HTML 則使用 <!-- -->
  3. 註解請寫在 Class 前方；勿寫在樣式內 or 該樣式後方
  4. 當有被註解樣式 / HTML需要留下時，請補上註解說明



* 1. 盡量不使用大量、重複不必要的註解 (HTML / SCSS)

# 

# 其他規範

### 保持一致性與整潔 (HTML / SCSS)

HTML / SCSS 因在開發時行數多而在維護或 Code review上很仰賴格式排版與縮排，大家一起養成良好習慣。

* 1. 注意 HTML / SCSS 文件內的縮排 (可以利用 IDE 來 Format)
  2. 樣式與樣式之間、Div與Div之間刪除多餘的空白 (最多空一行)
  3. 移除不需要的註解 Code
  4. 移除測試用的樣式
  5. 移除空白 Class (未寫樣式的 Class name)

### 使用 Class 下去撰寫樣式，盡量不使用 inline-style (行內樣式)

* 1. 統一在 SCSS 內管理樣式
  2. 行內樣式權重較高，不當使用會造成後續維護上的問題

### 同一組 HTML 的樣式，多使用嵌套/巢狀的寫法

* 1. 易於管理 (模組化)
  2. 嵌套/巢狀的深度請勿超過 3 層，以免造成共用樣式客製化的困難

### 注重 Production (正式生產環境)程式碼品質

* 1. 移除 Debug 用的 Code
  2. 壓縮圖檔大小
  3. Minify CSS & JS檔案

### 

# 自動檢查工具

### 安裝 Stylelint plugin

* 1. **npm install -D stylelint stylelint-config-sass-guidelines stylelint-config-standard stylelint-order stylelint-scss husky pre-commit**

### 引用 .stylelintrc.json

* 1. [檔案下載](https://drive.google.com/file/d/1QlzhyJhxli13Po5-JHUq1rHEioK0lKrk/view?usp=sharing)
  2. 放入專案資料夾



### Package.json

* 1. 加入 scripts 兩個指令
     + **"lint"**: **"stylelint source/scss/\*\*/\*.scss --syntax scss"**
     + **"lint:fix"**: **"stylelint source/scss/\*\*/\*.scss --syntax scss --fix"**
  2. 加入 pre-commit 的指令

### 

### Npm run lint

* 1. 每次 Commit 之前，先 Run 下方指令，先修復 linter 錯誤之後再 Commit

### 

# 參考資料：

### 談一下 CSS Specificity

<https://askiebaby.github.io/css-specificity/>

### SCSS 物件導向模組設計 (1)

<https://blog.chh.tw/posts/scss-object-oriented/>