

3.5.Sass / SCSS

一、Sass / SCSS

Sass (Syntactically Awesome StyleSheets) 是一種 CSS 的擴充，是為 CSS 的超集合 (透過編譯會 compiled 成傳統 CSS，讓瀏覽器可以閱讀)。使用 Sass 的出現是為了解決在大型專案時傳統 CSS 會遇到的重複、可維護性差等問題 (新增了 nested rules, variables, mixins, selector inheritance 等特性)。讓開發者可以撰寫簡潔、富語意 (expressive)、重複性佳 (reusable)、可維護性佳和可延展性佳的 CSS 程式碼。

Sass 的語法分為新的 SCSS (Sassy CSS、Sass 3，檔案名稱為 *.scss) 和舊的 SASS (向 HamI 取經，具備不使用大括弧格式、使用縮排，不能直接使用 CSS 語法、學習曲線較高等特性，檔案名稱為 *.sass)。由於新的 SCSS 語法是 CSS3 的超集合，所以傳統的 CSS3 檔案就算直接複製過來也不會出錯，學習曲線相對較緩，因此以下將使用 SCSS 語法介紹。

(一) 一般 CSS 語法

```
.header {  
    height: 80px;  
}  
.header .logo {  
    position: relative;  
}  
.header .logo a {  
    display: block;  
}  
.header .logo a:hover {  
    color: red;  
}  
.header .menu {  
    width: 500px;  
}
```

(二) Sass 語法

```
.header  
  height: 80px  
  .logo  
    position: relative  
    a  
      display: block  
      &:hover  
        color: red  
  .menu  
    width: 500px
```

(三) SCSS 語法

```

.header {
  height: 80px;
  .logo{
    position: relative;
    a{
      display: block;
      &:hover{
        color: red;
      }
    }
  }
  .menu{
    width: 500px;
  }
}

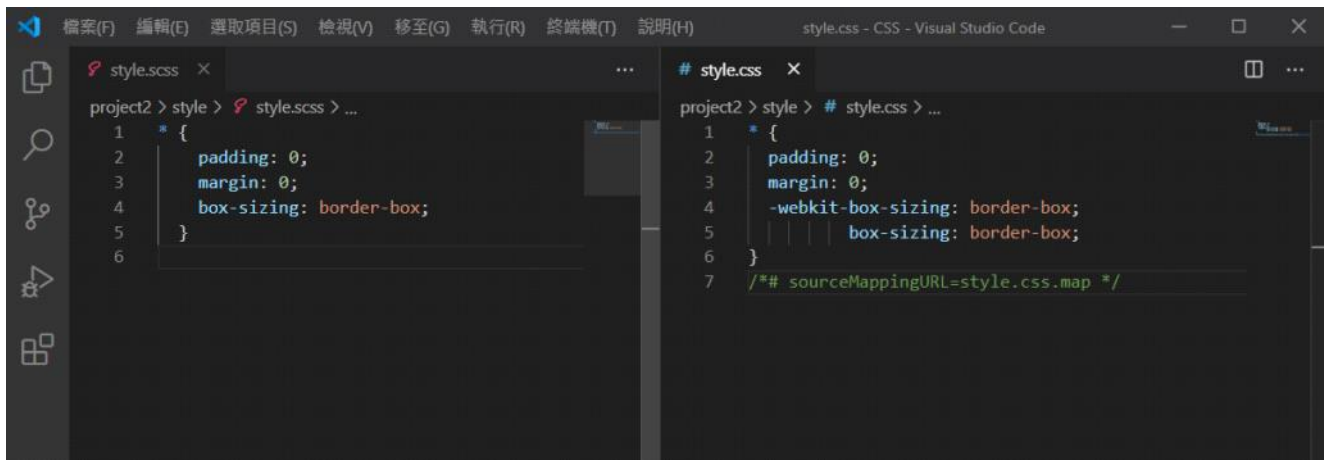
```

二、建構 Visual Studio Code Sass / SCSS 環境

安裝 Visual Studio Code Sass / SCSS 套件 Live Sass Compiler

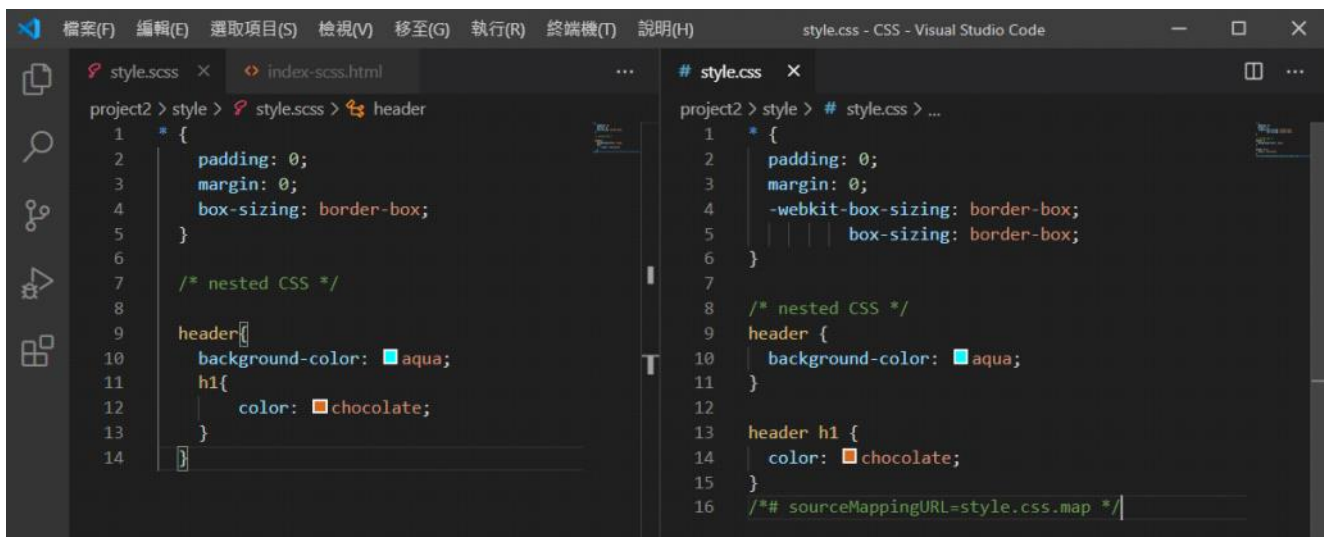
三、Sass / SCSS 常用語法介紹

以之前章節台灣經濟歷史發展網頁來練習，在網站資料夾內建立 style 資料夾與 style.scss 檔案，HTML 檔不能直接讀取 SCSS 檔，所以輸入程式碼後要 Compiler 產生新的 style.css 取代原先的 style.css，如下圖。

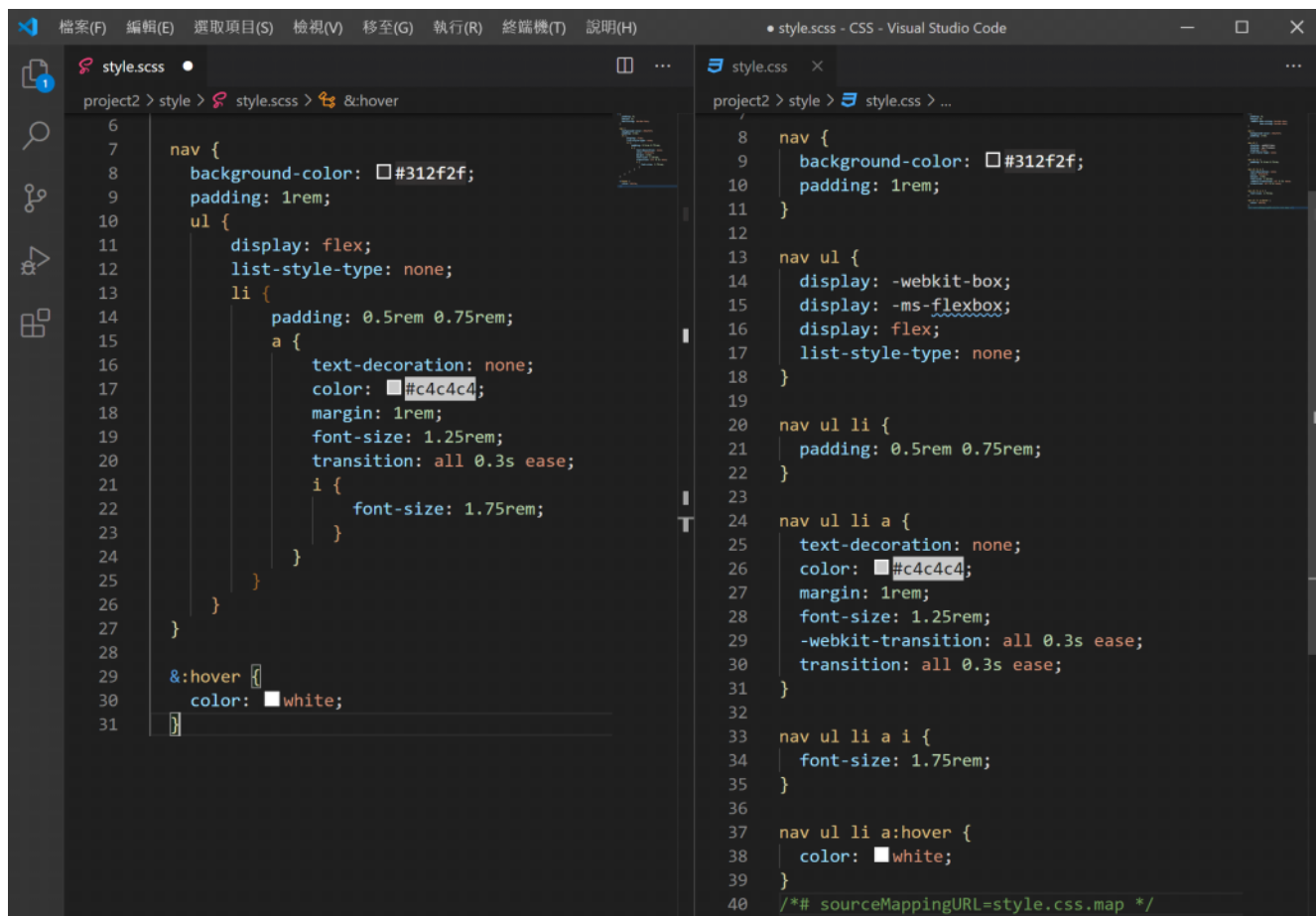


(一) Nested (嵌套)

主要是解決 CSS 編寫過程中，父元素重複編寫問題，可以縮短開發流程、方便後續維護、樣式模組化。



(二) & : 引用父選擇器。

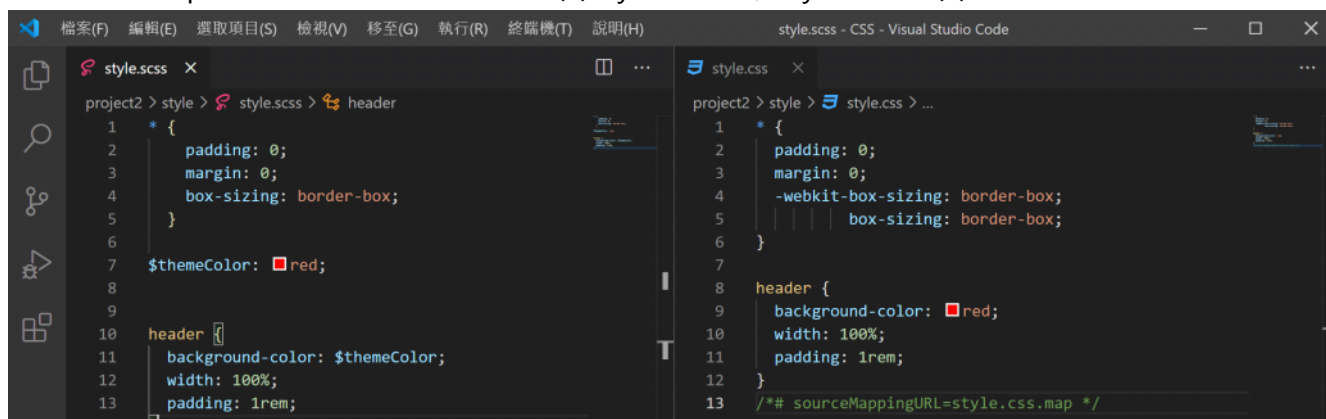


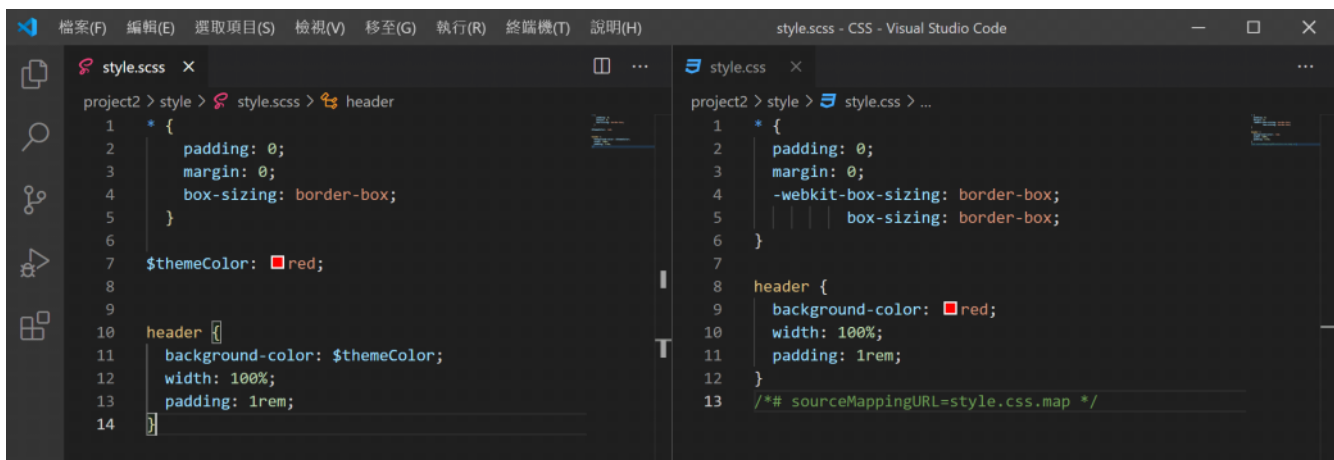
(三) 變數

在實際編寫 CSS 的過程中，常會遇到多次相同的顏色、字型、字體大小等重複出現，如遇到全部需要更改時，將是項大工程，所以 SASS 可以協助解決這問題，就如同 JS 變數邏輯一樣，只需改最源頭的變數值即可，其他所有引用該變數值的地方，值都會自動更改。

SASS 中的變數可使用以下 7 種主要類型：

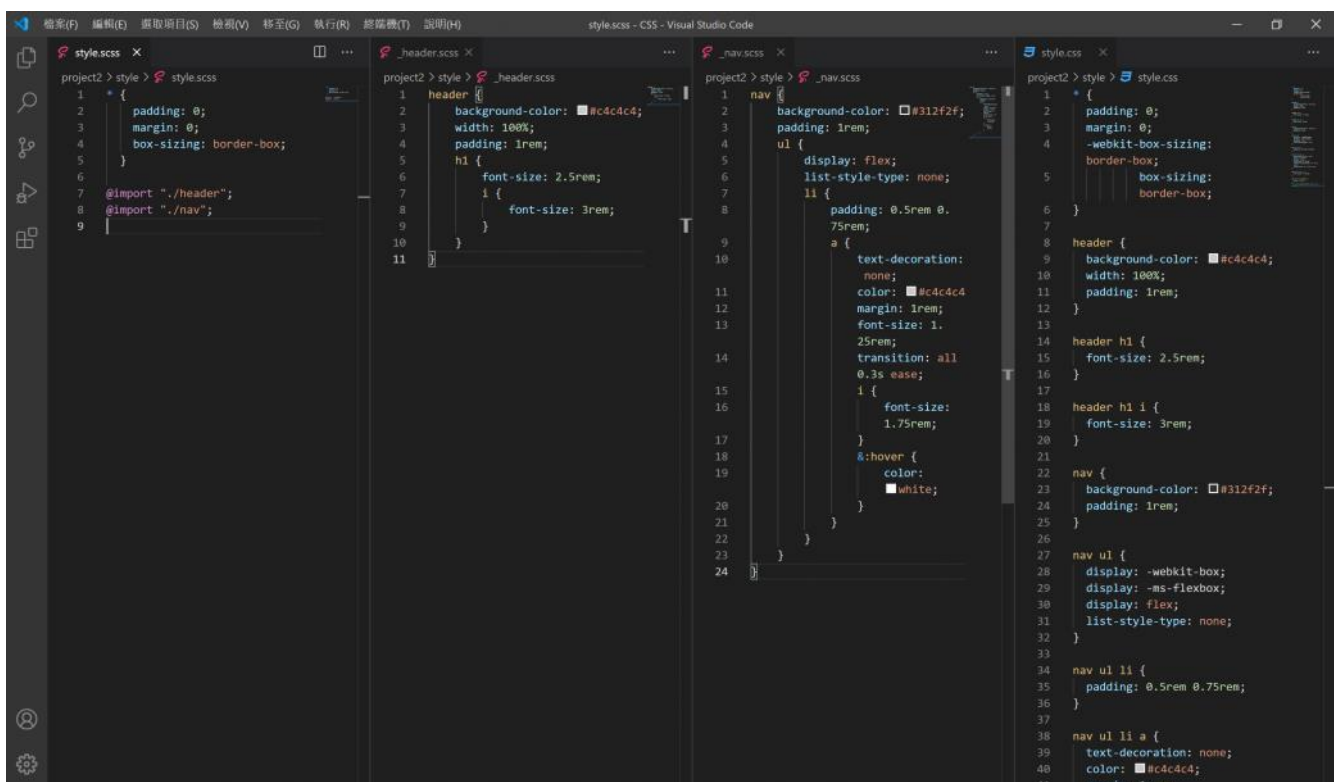
- 數字 (1, 1.5, 20px)
- 字符串，帶引號和不帶引號 ("foo", 'foo', foo)
- 顏色 (red, #28b0b4, rgba(0, 0, 0, 0.5))
- 布林值 (true, false)
- 空值 (null)
- 列表 (list)，用空格或逗號分開 (2em 1em 0 1em、Helvetica, Arial)
- maps：從一個值映射到另一個 ((key1: value1, key2: value2))





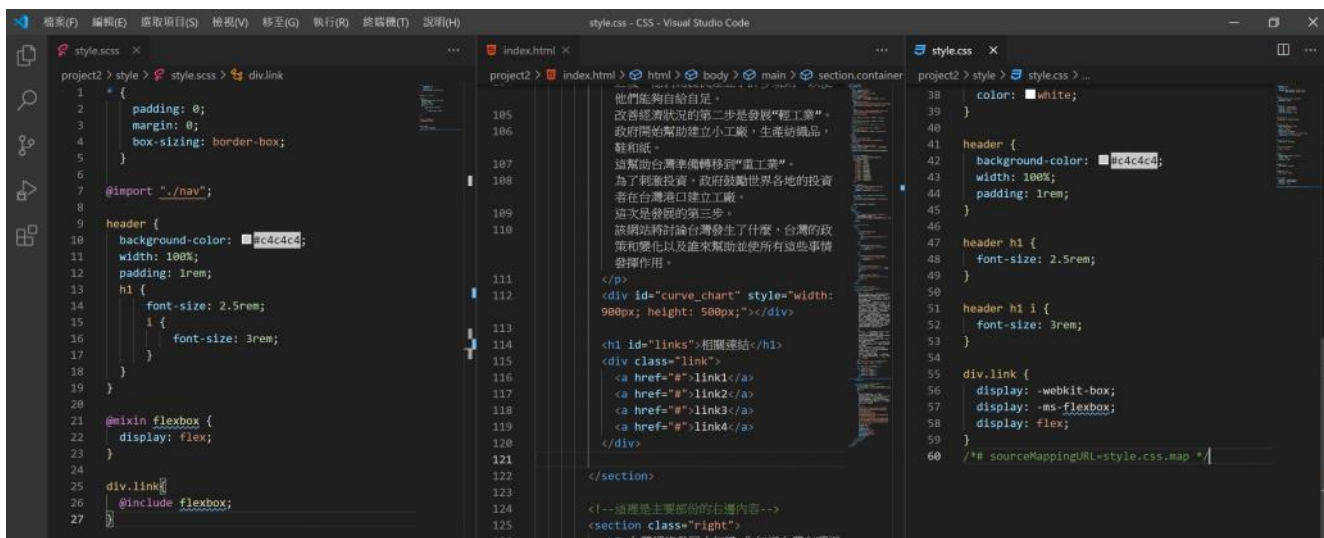
(四) 導入

SASS 強化了 CSS 的 @import 功能，允許在文件內導入其他的 SASS 或 SCSS 子樣式。在樣式進行編譯後，被導入的 SASS 或 SCSS 子文件，會全部合併到同一個 CSS 文件呈現。



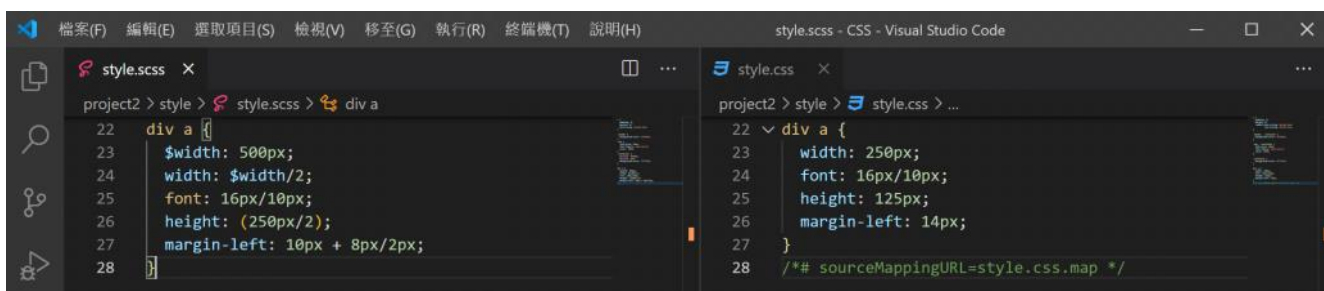
(五) 混合

若是有重複的代碼會不斷使用到，就可用混合指令 (Mixin Directives)，把樣式封裝成一個類別名稱，就可以重複使用。使用 @mixin 定義一個「混合」，使用 @include 引用一個「混合」，可同時引用多個「混合」。使用「混合」功能，可設置參數 (可設定默認值)，讓整體功能變更彈性，有點像函數的傳入參數。



(六) 運算字符

SassScript 支持對數字的運算，如：（加法 +、減法 -、乘法 *、除法 / ...等），簡單來說就是增加了運算功能，有需要也可進行使用。

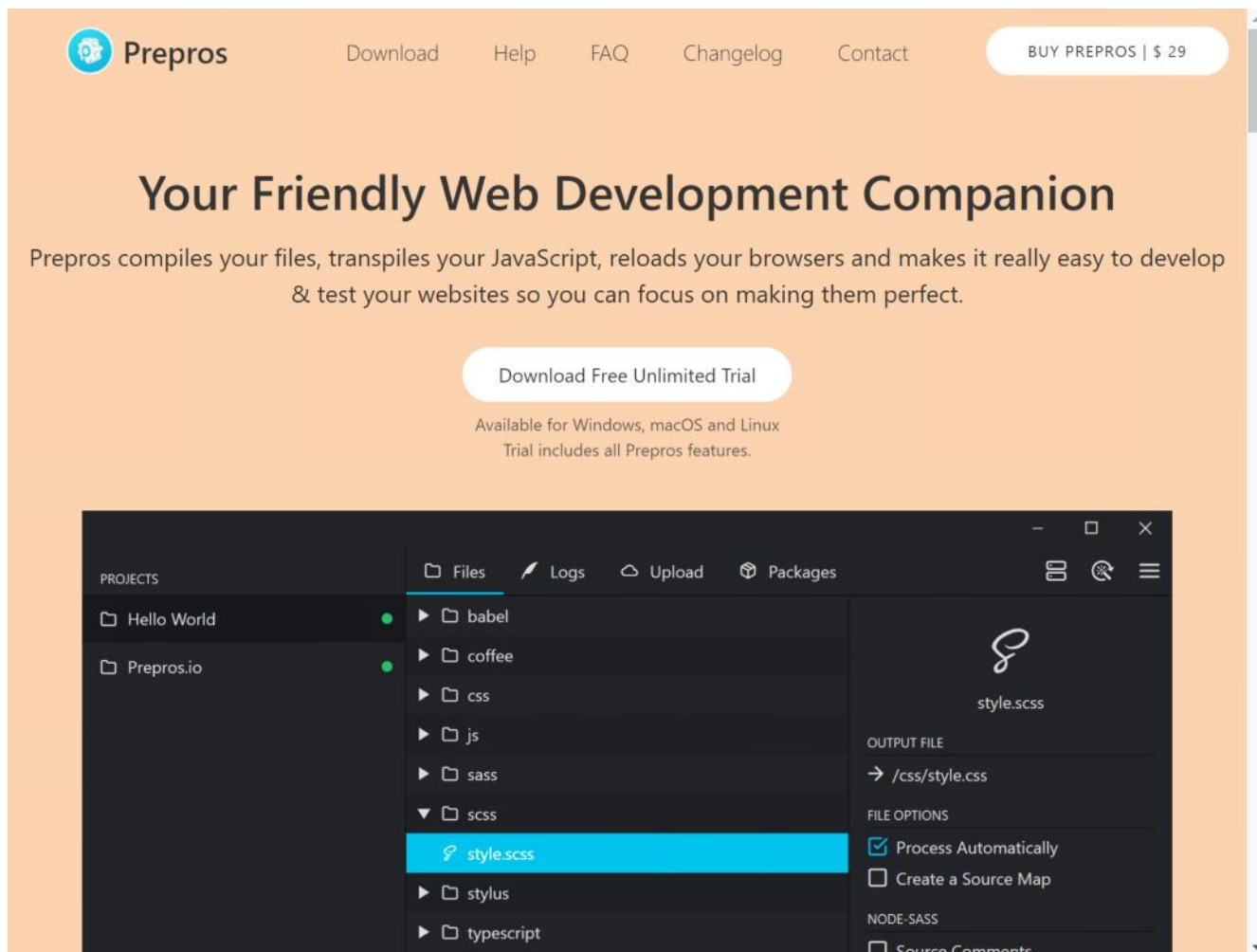


四、Prepros

不是使用 Visual Studio Code 編輯網頁或是使用的編輯器沒有支援 SASS 的套件，可以使用 Prepros 這套軟體來協助。

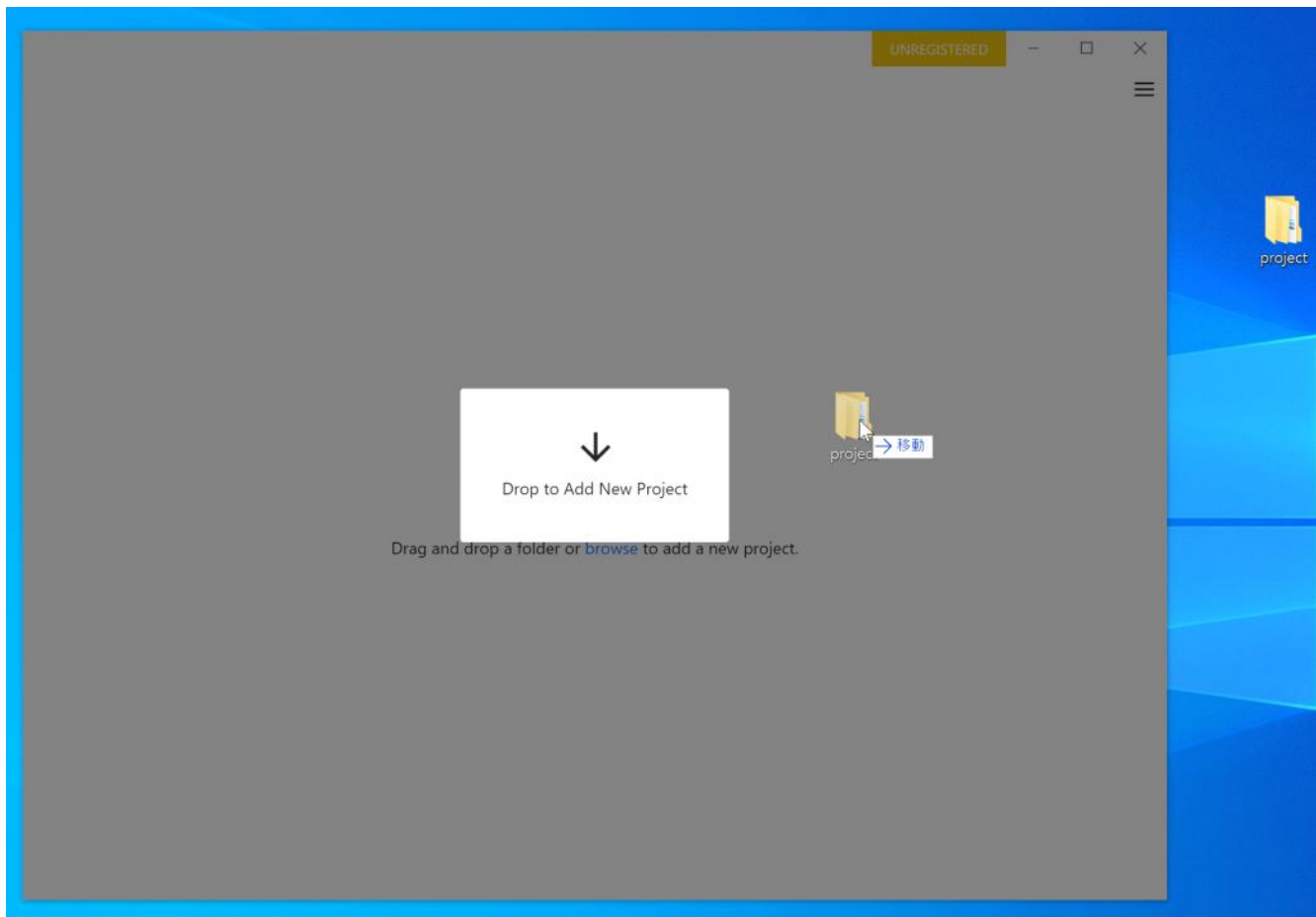
(一) 安裝 prepros

到 [Prepros](#) 官網下載軟體並安裝。



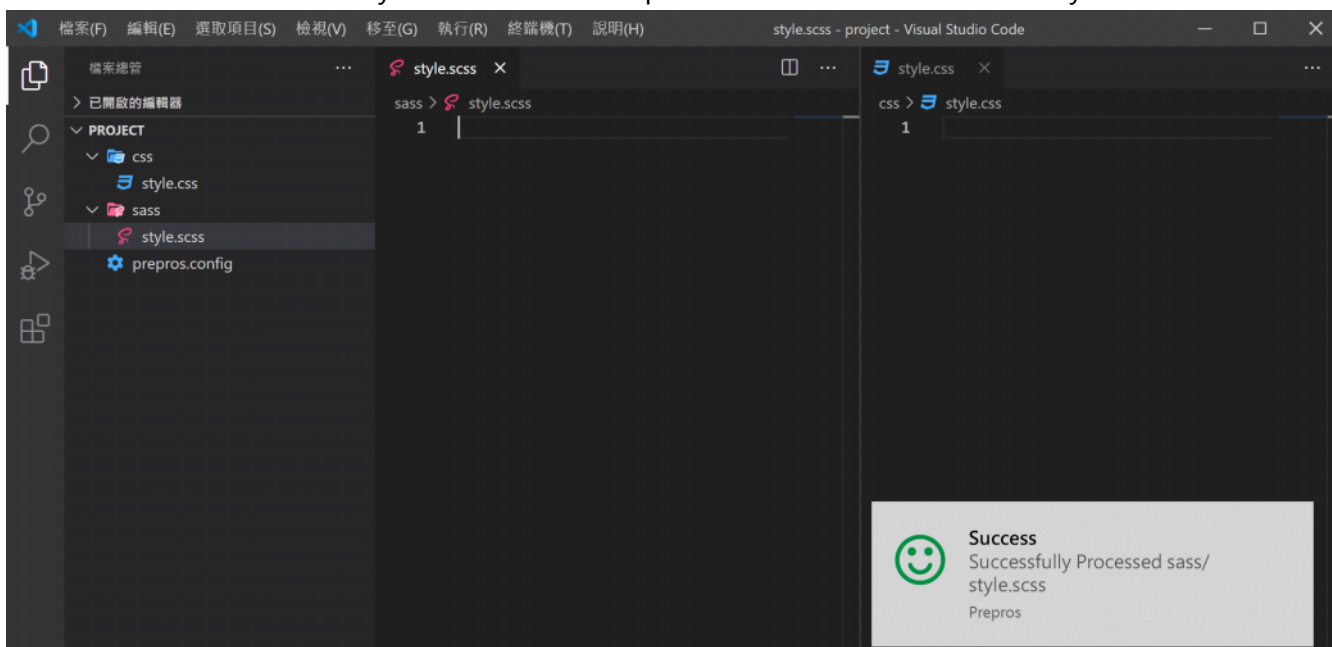
(二) 建立測試資料夾並連結 Prepros

建立 project 資料夾及 sass 子資料夾，完成後將 project 資料夾與 Prepros 連結。



(三) 建立 SASS 檔

在 sass 資料夾內建立 style.scss 並存檔，Prepros 會自動建立 css 資料夾及 style.css 檔。



(四) 使用 Prepros

編輯時存檔就會自動編譯，編輯時不可關閉 Prepros 軟體。



