

新增元件

為什麼要用元件（Component）

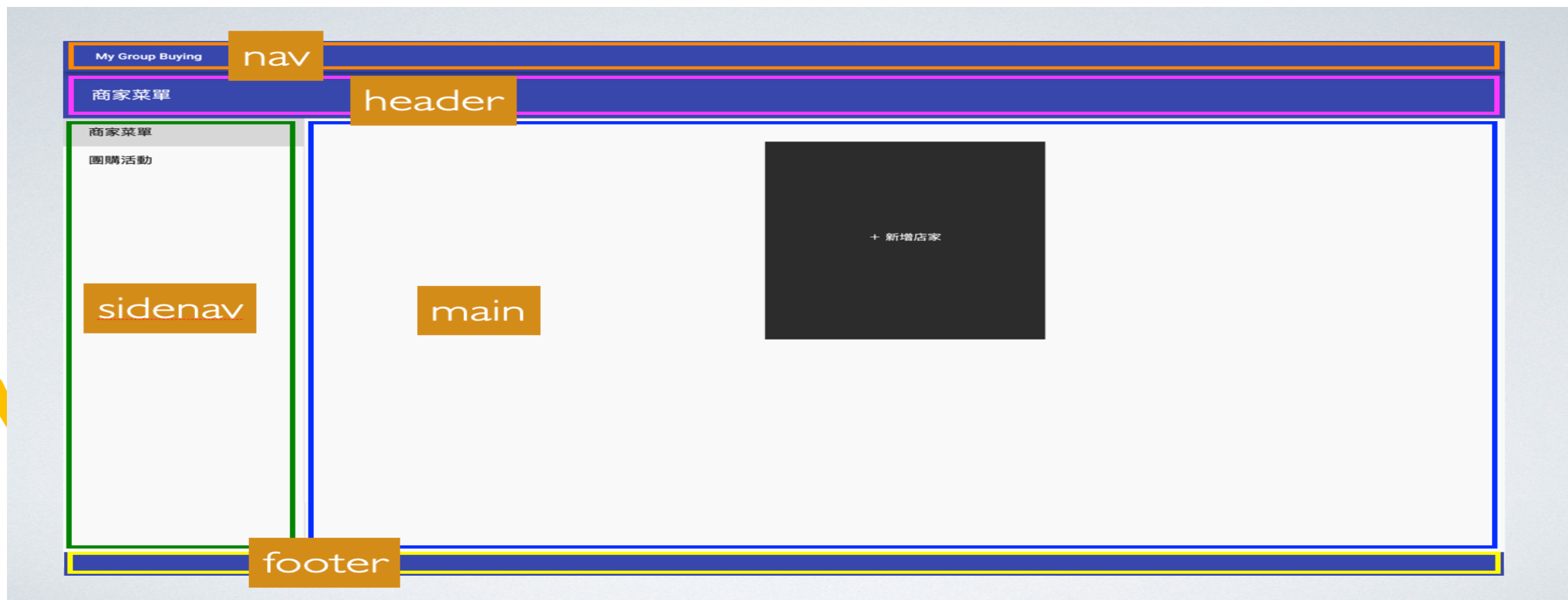
Component是什麼？一個畫面可以由多個Component 共同組合成，而且這些Component 可能有機會一直被重複使用。

傳統的開發很習慣以整個網頁的思考方式寫程式，但後面的開發都建議將一個畫面切換成多個Component。

參考下一頁的後台畫面來分割畫面。

為什麼要用元件（Component）

可大致將畫面拆成這 5 個部分，當然每個部分都還有可能可以拆的更小。



新增元件

打開命令提示字元/終端機，並且切換至專案的根目錄，如果已經開啟專案請先將專案關閉，這邊建議要將檔案新增至components資料夾如果沒有建立資料夾系統會自動幫你建立。

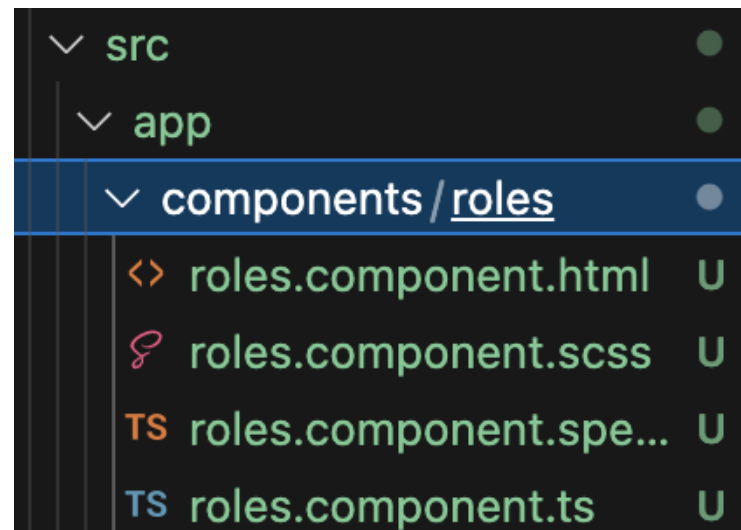
輸入 **ng generate component 資料夾名稱/組件名稱** 系統會自行建立組件所需要的檔案(簡寫 **ng g c 資料夾名稱/組件名稱**)。

```
CREATE src/app/component/roles/roles.component.ts (231 bytes)
lingyilun@lingyilundeMacBook-Pro demo1 % ng g c components/roles
CREATE src/app/components/roles/roles.component.scss (0 bytes)
CREATE src/app/components/roles/roles.component.html (20 bytes)
CREATE src/app/components/roles/roles.component.spec.ts (585 bytes)
CREATE src/app/components/roles/roles.component.ts (231 bytes)
```

元件

來新增元件時我們會發現他新增了4個檔案

- .HTML
我們放html的地方碼。
- .SCSS
我們放style的地方
- SPEC.ts
我們放測試程式碼的地方。
- .ts
我們放ts程式碼的地方。



元件

接下來看到.ts的文件

- 從 @angular/core 這個核心函式庫中匯入 Component 類。如此一來就能使用 @Component 裝飾器來設定元件。
- 定義元件的selector、template和style。
 1. selector：可以把它看作類似標籤的名字，透過這個 selector 就可以在其他 HTML 模板中使用此元件，像 <app-home></app-home> 這樣。
 2. templateUrl：指定這個元件所使用的HTML
 3. styleUrls：指定這個元件所使用的SCSS

```
import { Component } from '@angular/core';  引用路徑

@Component({
  selector: 'app-roles',  標籤名稱
  standalone: true,
  imports: [],  引用哪些文件
  templateUrl: './roles.component.html',
  styleUrls: ['./roles.component.scss']
})
export class RolesComponent {  名稱
}
```

引用元件

當新增完組件後我們就要嘗試將組件引入至頁面中使用，首先我們要先確認元件的名稱請打開剛剛新增的元件的TS，找到元件的名稱將其放入要使用的頁面的TS文件中的import，引用完畢後就可以在HTML頁面使用引用TS文件中的標籤名稱來呈現畫面。

```
import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { RolesComponent } from '../components/roles/roles.component';

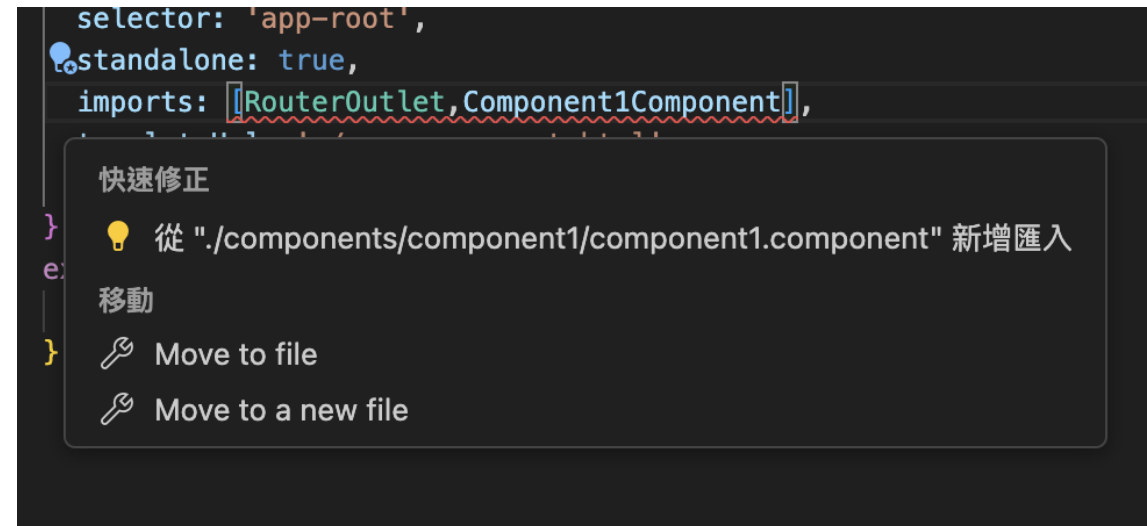
@Component({
  selector: 'app-root',
  standalone: true,
  imports: [RouterOutlet, RolesComponent],
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  title = 'demo2';
}
```

```
<app-roles></app-roles>
```

好用工具

當要新增元件時會覺得自己手動去輸入元件路徑是一件麻煩事，這時候VS Code就有好用的輔助工具可以使用，當你輸入完元件名稱時會發現下面有錯誤的紅線條(系統不知道這個名稱要對應哪個地方)，往你新增的名稱點下去並且注意畫面右邊有的藍色燈泡(如左圖)點擊後，會發現他有個快速修正點下去他會自動將此名稱對應的import新增進你的程式中(如果你的名稱沒有打錯)

```
selector: 'app-root',
standalone: true,
imports: [RouterOutlet, Component1Component],
templateUrl: './app.component.html',
styleUrl: './app.component.scss'
```



練習

新增兩個組件並且放入 app.component.html 中顯示，讓畫面變成以下

Angular引用元件

component1

component2

```
... <body>  
  ... <app-root _ngghost-ng-c3637030675 ng-version="8.2.8"> == $0  
    <h1 _ngcontent-ng-c3637030675 style="text-align: center; color: red;">Angular引用元件</h1>  
    > <app-component1 _ngcontent-ng-c3637030675 _ngghost-ng-c288547766>⋮</app-component1>  
    > <app-component2 _ngcontent-ng-c3637030675 _ngghost-ng-c1142745058>⋮</app-component2>  
  </app-root>
```

结束

