

# 開發工具

安裝Visual Studio Code : <https://code.visualstudio.com/download>

在VS Code安裝擴充套件(建議) :

- Angular Language Service
- Prettier - Code formatter (幫助排版)
- TSLint (偵錯用)

# 開發環境

Angular CLI : 可以快速建立Angular測試及開發環境

環境要求 :

- 安裝Node.JS(版本10.x以上) : <https://nodejs.org/en/>
- 安裝的原因是因為要NPM : 可以用來管理安裝的套件, 已包含在Node.JS裡
- 檢查是否安裝成功 : 開cmd(用系統管理員開) → 輸入**node -v** → 看到安裝版本代表成功
- 安裝Angular CLI : cmd輸入**npm install --global @angular/cli**
- 檢查是否安裝成功 : cmd輸入**ng version**

```
C:\Windows\system32>node -v
v10.13.0

C:\Windows\system32>npm -v
6.4.1

C:\Windows\system32>npm install --global @angular/cli
C:\Users\shengsen\AppData\Roaming\npm\ng -> C:\Users\shengsen
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported pla
"} (current: {"os":"win32","arch":"x64"})

+ @angular/cli@7.0.5
added 268 packages from 205 contributors in 41.32s
```

CA 系統管理員: 命令提示字元

```
C:\Windows\system32>ng version

Angular CLI
Angular CLI: 7.0.5
Node: 10.13.0
```

- 輸入**--help**可以看CLI能做什麼

# 使用Angular CLI

- 建立專案：先建立一個資料夾，到資料夾下再建立，(建議不要在C槽做)。

- **ng new <Project Name>**

- 會有兩個問題，看自己需要選擇，這裡選擇如圖。
- 第一個是看我們有沒有需要用到routing
- 第二個是看選我們習慣用的style(樣式)

```
C:\>ng new todomvc
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? CSS
```

- 開起專案：

- 先到該專案所在資料夾下
- 開啟專案：**code** .

```
C:\>cd todomvc
C:\todomvc>code .
```

- 啟動專案

- **ng serve**

- 建置專案

- **ng build**
- **ng build --prod**：會比較久，但會優化檔案

- 基本語法：()裡的是縮寫，[]是可打可不打

- **ng generate(可打縮寫g就好) [type] [name]**

- 建立component：

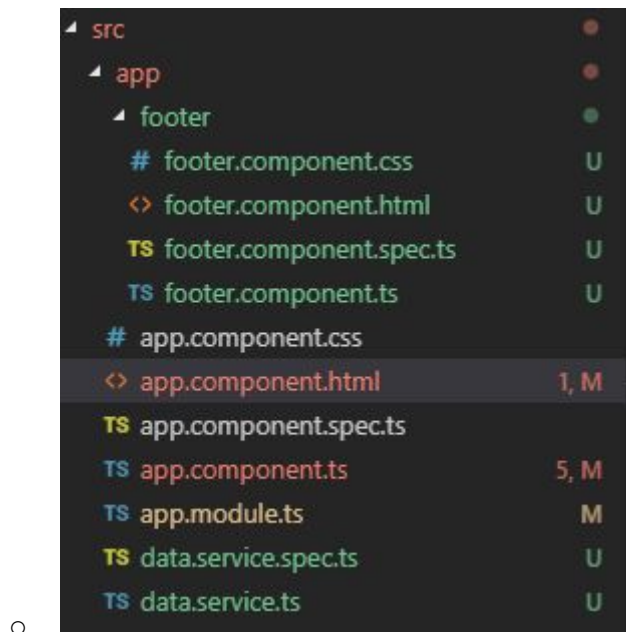
- **ng g c component名稱**

```
C:\todomvc>ng g c footer
CREATE src/app/footer/footer.component.html (25 bytes)
CREATE src/app/footer/footer.component.spec.ts (628 bytes)
CREATE src/app/footer/footer.component.ts (269 bytes)
CREATE src/app/footer/footer.component.css (0 bytes)
UPDATE src/app/app.module.ts (458 bytes)
```

- 建立Service：

- **ng g s service名稱**

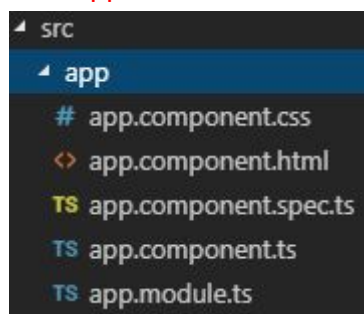
```
C:\todomvc>ng g s data
CREATE src/app/data.service.spec.ts (323 bytes)
CREATE src/app/data.service.ts (133 bytes)
```



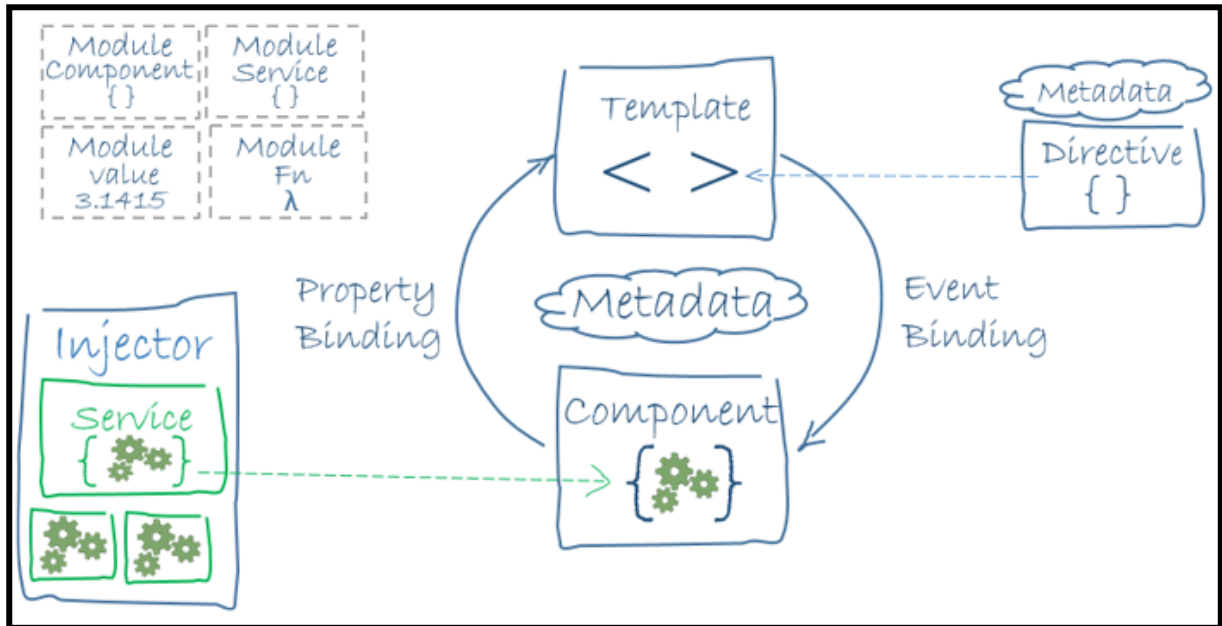
主要資料夾說明：

package.json：存放安裝的套件

src→app：所有主程式放置的位置(重要)



# Angular 架構剖析



首先先看最中間那一塊，是由template、metadata、component所構成的，這三個是一個component必備的元素。

Template：也就是專案裡的`component.html`檔(html的檔案)，可以在裡面用資料綁定與事件綁定與controller裡的物件做繫結。

Component：由HTML、CSS、JS組合而成。

Event Binding(事件繫結，用()符號)、Property Binding用[]：建立Component和Template之間的溝通

Service：只處理邏輯，不處理跟Template有關的事

Directive：擴充功能

Module：

- 定義該模組下可以使用那些元件
- 可設定屬性
- imports：引用其他module
- 定義可用的component/directive/pipe

Component：

- 使用最廣的元件型別
- 負責顯示內容
- 基本架構：@Component

Pipe：資料顯示的轉型，方便我們顯示不同格式的資料，**{{ 變數 | 顯示格式 }}**

RxJS：

- 非常重要，必學(很多，慢慢學)
- 透過Observable, Operator的組合，可以讓非同步與同步的操作，優雅的融合在一起

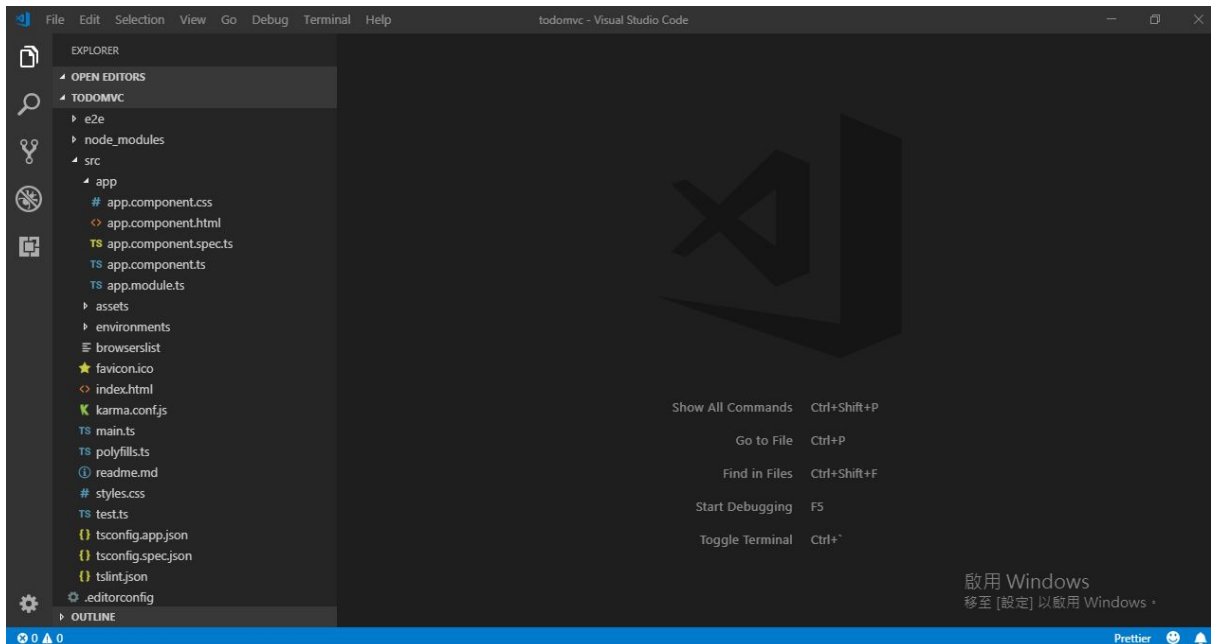
樣板變數：**#名稱**，有點像id，每個元件只能有一個，告訴Angular要取哪個元件的值

自訂的Component：**<名稱></名稱>**

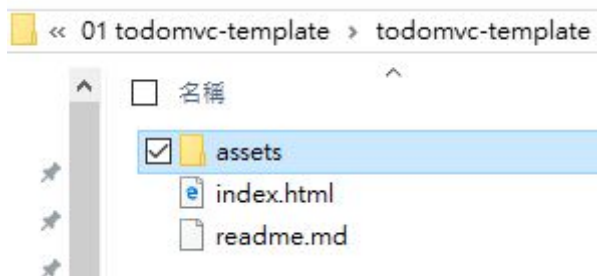
**{{ 變數 }}**

# 實做練習

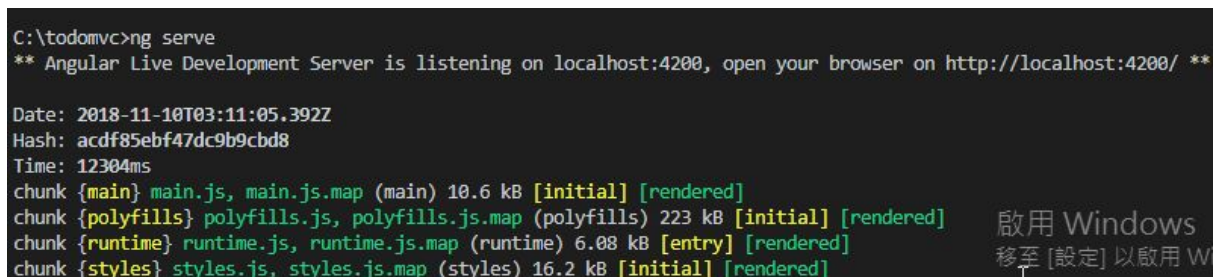
1. 先建立好一個專案並開啟，看到以下畫面代表成功



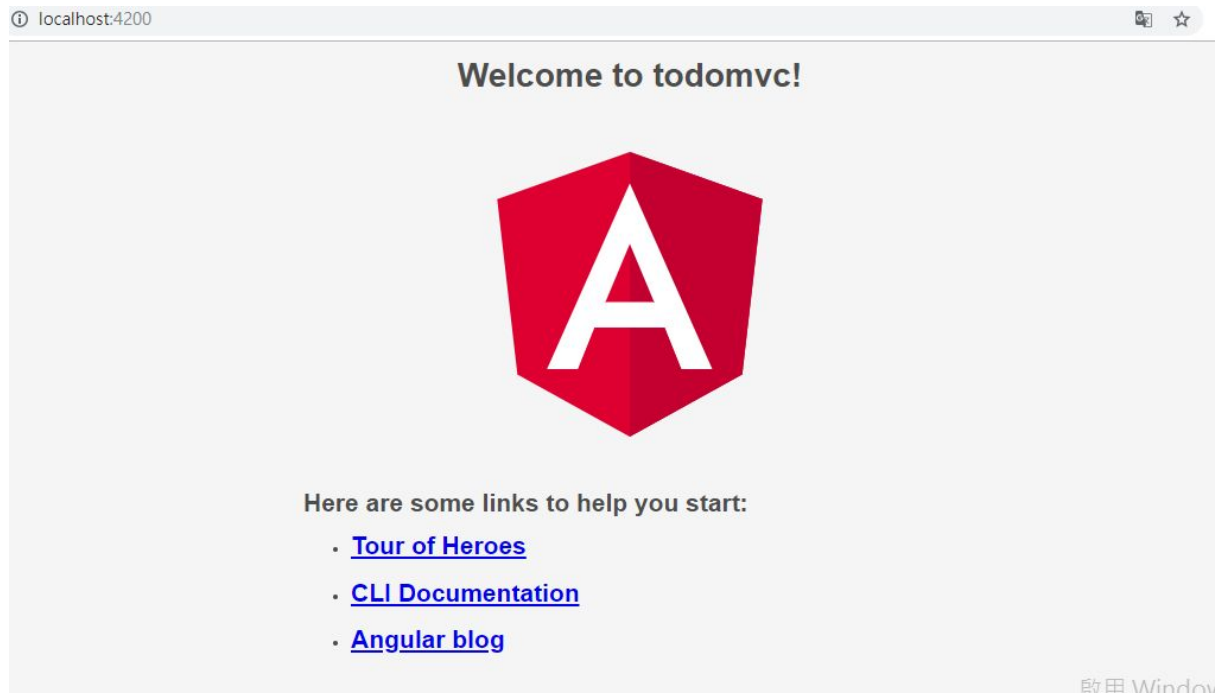
2. 將todomvc-template資料夾(教學用)內的assets資料夾覆蓋掉自己的src/app裡的assets，然後開啟自己以及todomvc-template內的index.html(注意不要覆蓋掉自己的index.html)



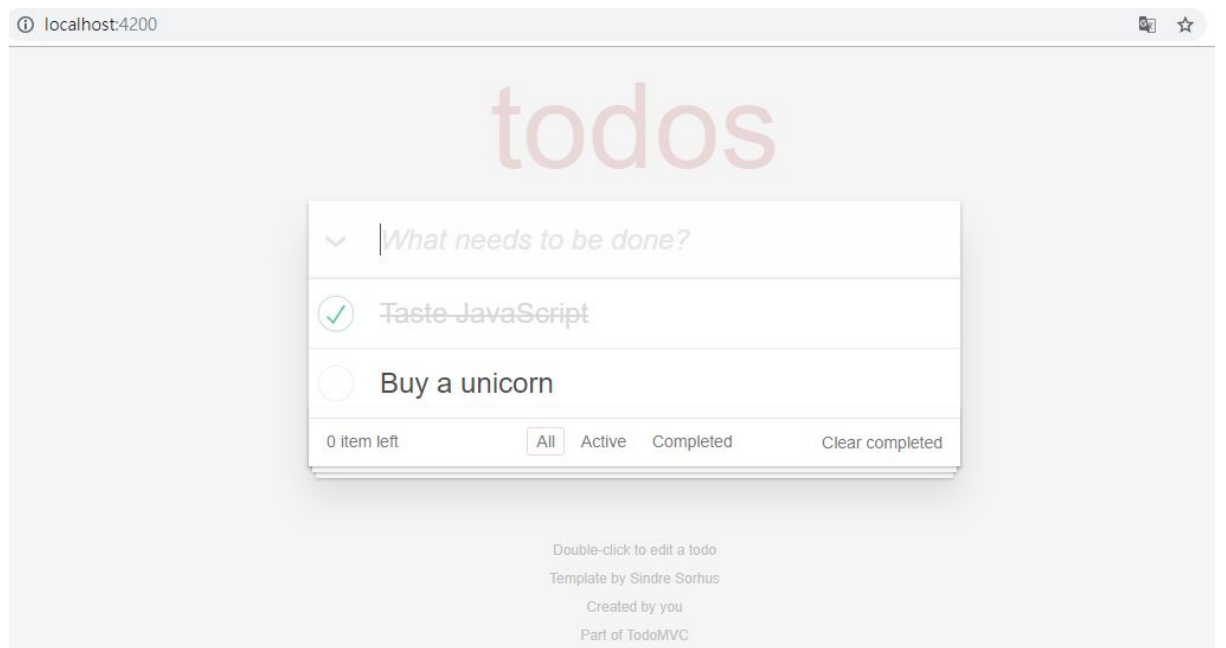
3. 將index.html(教學用)的css內容貼到我們的index.html  
`<link rel="stylesheet" href="assets/css/base.css">`  
`<link rel="stylesheet" href="assets/css/index.css">`
4. 之後開啟Terminal(工具列有，或是按Ctrl+` )，輸入ng serve(注意要記得裝npm，沒裝就在terminal輸入 npm install)
5. 成功畫面如下：



(網頁輸入localhost:4200)

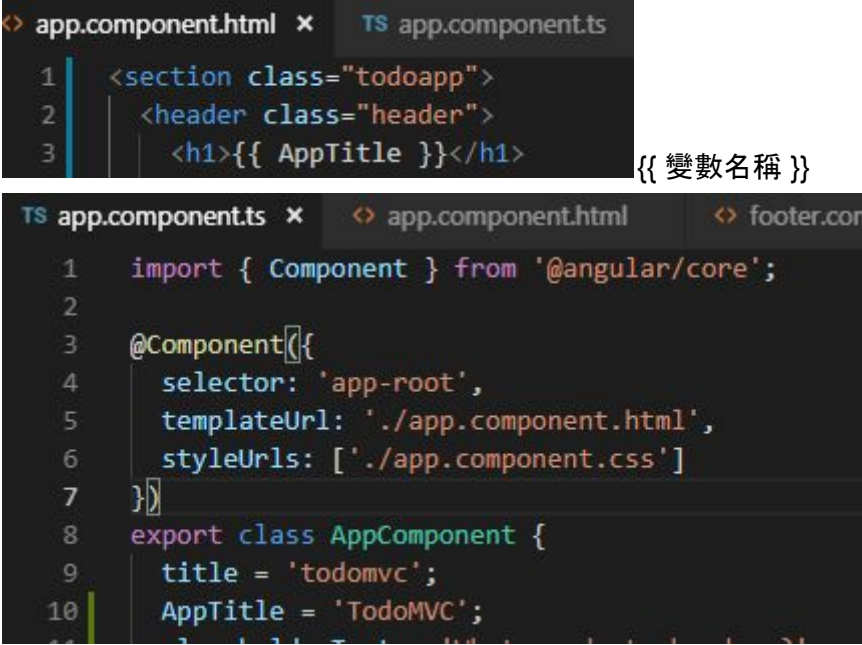


6. 將index.html(教學用)的body內容(不含body)覆蓋到我們的app.component.html





7. 用變數顯示title：



The screenshot shows two files in VS Code. The top file, `app.component.html`, contains the following HTML code:

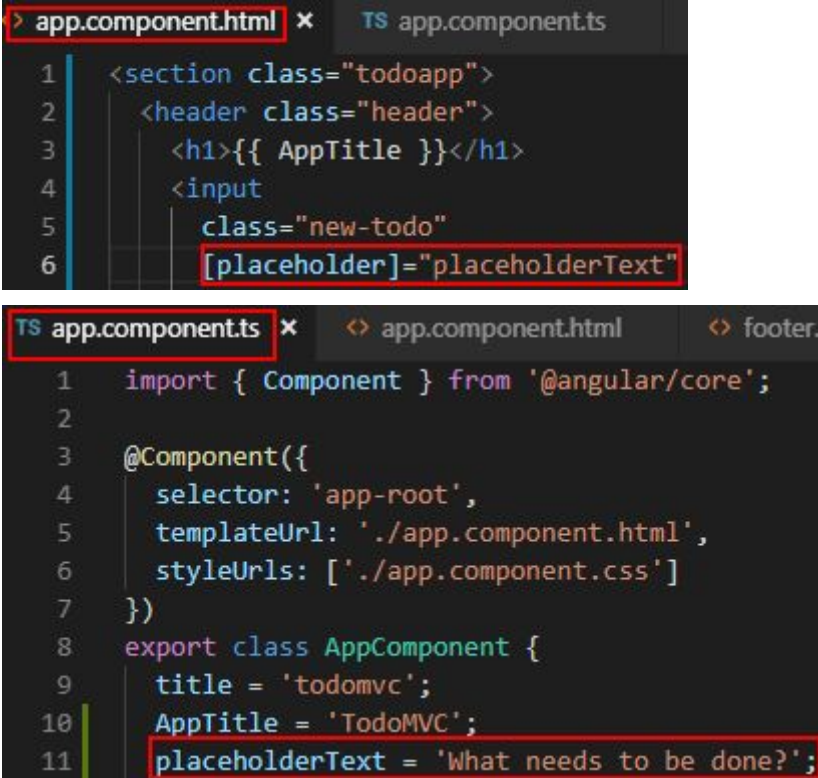
```
1 <section class="todoapp">
2   <header class="header">
3     <h1>{{ AppTitle }}</h1>
```

The bottom file, `app.component.ts`, contains the following TypeScript code:

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'todomvc';
10  AppTitle = 'TodoMVC';
```

Annotations: The text `{{ 變數名稱 }}` is placed next to the HTML binding. The text `在class定義值` is placed next to the `AppTitle` property in the TypeScript class.

8. 也可以用Property binding顯示：**[屬性名稱]="變數名稱"**



The screenshot shows two files in VS Code. The top file, `app.component.html`, contains the following HTML code:

```
1 <section class="todoapp">
2   <header class="header">
3     <h1>{{ AppTitle }}</h1>
4     <input
5       class="new-todo"
6       [placeholder]="placeholderText">
```

The bottom file, `app.component.ts`, contains the following TypeScript code:

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.css']
7 })
8 export class AppComponent {
9   title = 'todomvc';
10  AppTitle = 'TodoMVC';
11  placeholderText = 'What needs to be done?';
```

Annotations: Red boxes highlight the `[placeholder]` attribute in the HTML and the `placeholderText` property in the TypeScript class.

9. Event binding(事件繫結)：在元件加入事件如點擊事件等。

語法：(事件)="處理事件名稱(變數名稱)"

範例：(keyup)="newTodo(todo)"

上面的todo代表某個元件，故要在某個元件加上#todo

```
<> app.component.html • TS app.component.ts
1  <section class="todoapp">
2    <header class="header">
3      <h1>{{ AppTitle }}</h1>
4      <input
5        class="new-todo"
6        [placeholder]="placeholderText"
7        #todo
8        (keyup)="newTodo(todo)"
9        autofocus
10     />
11  </header>
```

```
TS app.component.ts • <> app.component.html
7  })
8  export class AppComponent {
9    title = 'todomvc';
10   AppTitle = 'TodoMVC';
11   placeholderText = 'What needs to be done?';
12   newTodo(event, inputElement) {
13     if (event.code === 'Enter') {
14       console.log(inputElement.value);
15       // console.dir(event);
16     }
17   }
}
```

或是直接在html那輸入

```
<> app.component.html x TS app.module.ts TS s
1  <section class="todoapp">
2    <header class="header">
3      <h1>{{ AppTitle }}</h1>
4      <input
5        class="new-todo"
6        [placeholder]="placeholderText"
7        #todo
8        (keyup.enter)="newTodo(todo)"
9        autofocus
10     />
11  </header>
```

```
TS app.component.ts • <> app.component.html
23   newTodo(inputElement) {
24     console.log(inputElement.value);
25   }
```

\*input值改變事件：change    button按下按鈕事件：click



10. 建立component：以下以footer為例子

```
C:\todomvc>ng g c footer
CREATE src/app/footer/footer.component.html (25 bytes)
CREATE src/app/footer/footer.component.spec.ts (628 bytes)
CREATE src/app/footer/footer.component.ts (269 bytes)
CREATE src/app/footer/footer.component.css (0 bytes)
UPDATE src/app/app.module.ts (458 bytes)
```

# import module

加入Module步驟(以下用formModule為例子)：

```
TS app.module.ts x  <> app.component.html  <> section-footer.component.html  TS section-footer.comp
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3  import { FormsModule } from '@angular/forms';
4  import { AppComponent } from './app.component';
5  import { FooterComponent } from './footer/footer.component';
6  import { DataService } from './data.service';
7  import { SectionFooterComponent } from './section-footer/section-footer.component';
8
9  @NgModule({
10   declarations: [AppComponent, FooterComponent, SectionFooterComponent],
11   imports: [BrowserModule, FormsModule],
12   providers: [DataService],
13   bootstrap: [AppComponent]
14 })
15 export class AppModule {}
16
```

參考：

<https://ithelp.ithome.com.tw/articles/10193913>

<https://github.com/chgc/20181110-angular-beginner>