



快樂樂學會 Angular 2 網站開發框架

適用 Angular 2.0.0-rc.5 版本



多奇數位創意有限公司

技術總監 黃保翕 (Will 保哥)

部落格：<http://blog.miniasp.com/>



今天晚上你回家開電腦一看



某前端套件又改版了



假的！



哎呀！我的眼睛業障重啊！

我們所認識的前端技術都是假的

千萬不要執著你之前學會的任何技術

忘掉一切的業障

才有重生的機會

今天我們要來 重新認識 Angular 2 開發框架

記得：今天是來**消除業障**的！

Angular 2 Introduction

ANGULAR 2 簡介



Angular 2 從框架轉向平台

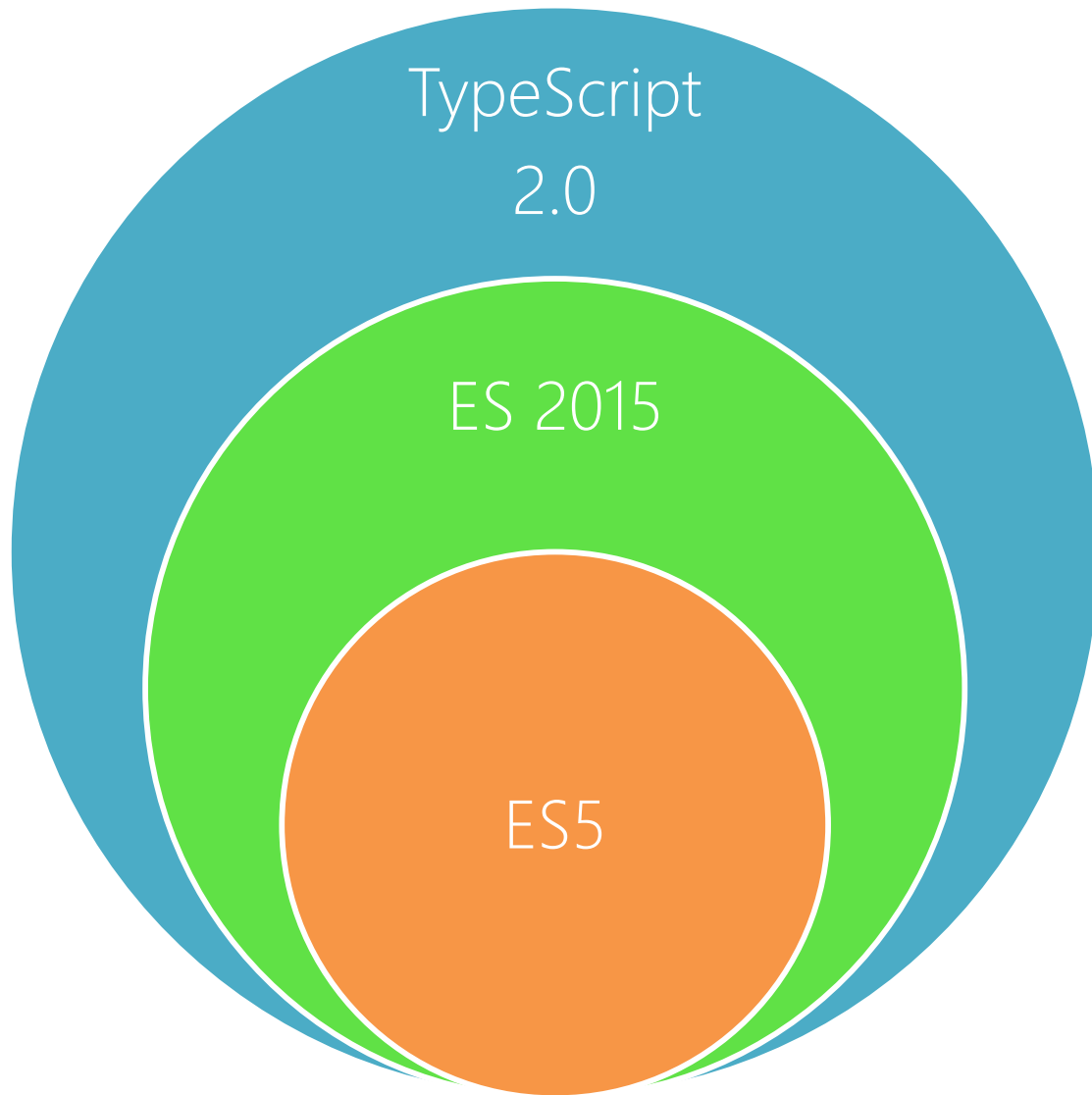
i18n	CLI	Language Services	Augury
Animation	Material	Mobile	Universal
Router	Compile	Change	Render
ngUpgrade	Dependency Injection	Decorators	Zones

前端工程的夢幻逸品：Angular 2 開發框架介紹

Angular 2 比 Angular 1 更好

- 速度更快
 - 偵測變更：比 ng1 快 10 倍
 - 渲染速度：比 ng1 快 5 倍 (Render & Re-render)
 - 範本編譯：支援 Template 預先編譯機制
 - 更小的 Library Size (透過 webpack 自動合併)
 - 支援延遲載入機制與伺服器渲染機制 (Node.js & [ASP.NET](#))
- 學習曲線更低
 - 採用 TypeScript 語法**更簡潔** (類別、介面、屬性、方法)
 - 移除超過 40+ 個 directives
- 平台支援更強大
 - 支援 Browser, Node.js, NativeScript, React Native, ...

Angular 2 的開發語言



Angular 2 的開發語言

- ES5
 - 傳統 JavaScript 程式語言 (IE9+)
- [ES 2015](#)
 - 此版本為 ES5 的「超集合」
 - 具有新穎的 JavaScript 語言特性 (let, const, for-of, ...)
 - 可透過 [Babel](#) 轉譯器將瀏覽器不支援的語法轉為 ES5 版本
- [TypeScript](#)
 - 此版本為 ES 2015 的「超集合」
 - 具有強型別特性、內建 ES5 轉譯器 (Transpiler)、更好的工具支援
- [Dart](#)
 - 非 JavaScript 家族的程式語言
 - 具有強型別特性

Angular 2 的開發工具

- [Visual Studio Code](#) (推薦)
- [Visual Studio 2015](#)
- [Sublime Text](#)
- [WebStorm](#)
- [Atom](#)
- [Plunker](#)

Angular 2 頁面的組成

應用程式元件 + 樣板 + 樣式
(AppComponent)

頁首元件 + 樣板 + 樣式
(HeaderComponent)

子選單
元件 + 樣板 + 樣式
(AsideComponent)

主要內容
元件 + 樣板 + 樣式
(ArticleComponent)

Angular 2 應用程式的組成

模組

- AppModule

元件

- App Component

元件

- Child Component

元件

- Services Component

元件

- Pipe Component

Angular 2 元件的組成

範本 (Template)

- HTML 版面配置
- HTML 部分片段
- 資料繫結 (Bindings)
- 畫面命令 (Directives)

類別 (Class)

- 建構式 (Constructor)
- 屬性 (Properties)
- 方法 (Methods)

中繼資料 (Metadata)

- 裝飾器 (Decorator)
 - 針對類別
 - 針對屬性
 - 針對方法
 - 針對參數

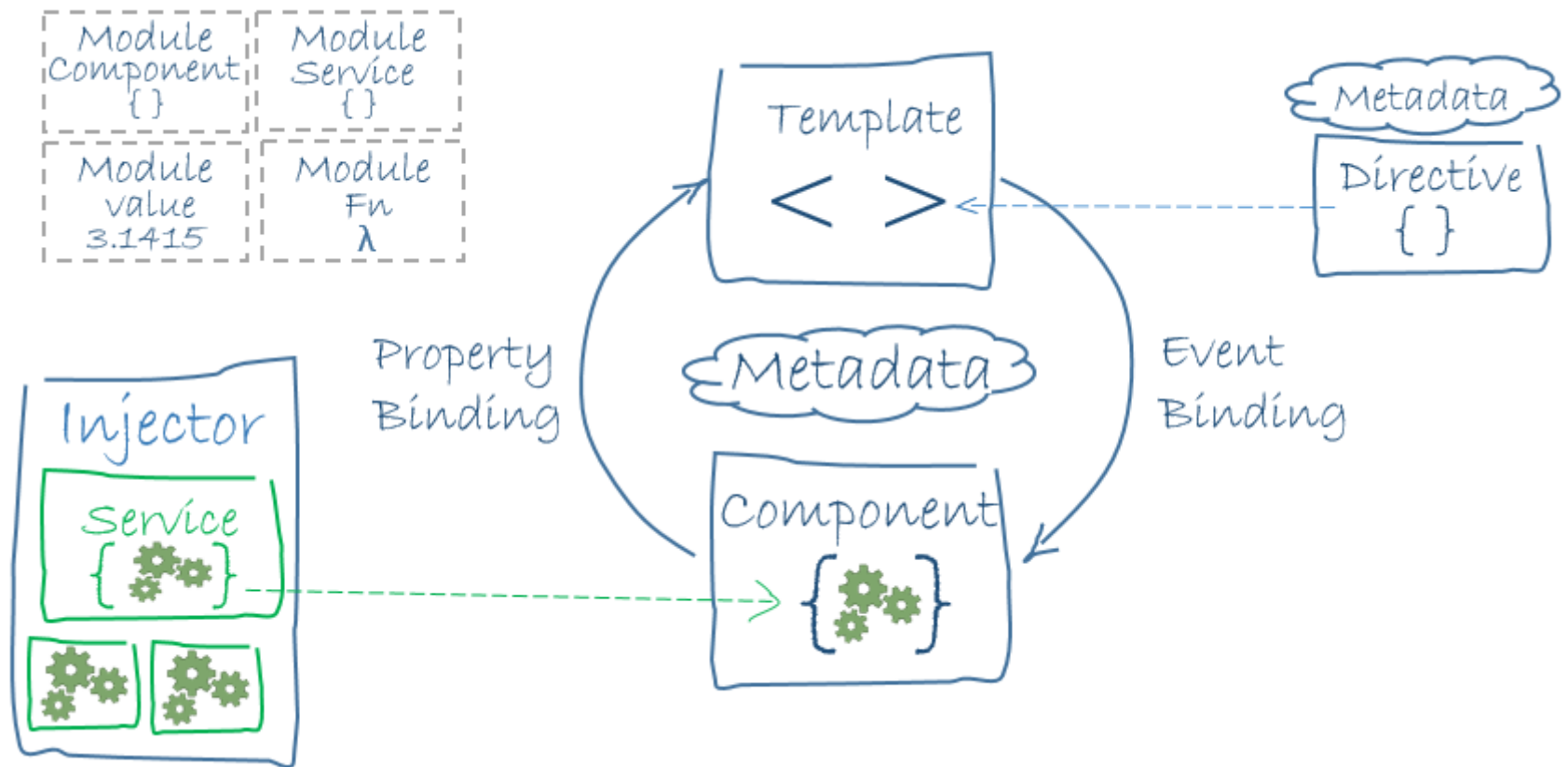
認識 Angular 元件的程式碼結構

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

```
@Component({  
  selector: 'app-root',  
  templateUrl: 'app.component.html',  
  styleUrls: ['app.component.css']  
})
```

```
export class AppComponent {  
  title = 'app works!';  
}
```

Angular 2 結構剖析



Angular 2 結構剖析

- [Module](#) 應用程式被切分成許多「模組」
- [Component](#) 每個模組下有許多「元件」
- [Template](#) 每個元件都可能有自己的「樣板」
- [Metadata](#) 每個元件都可以標示「中繼資料」
- [Data Binding](#) 樣板與元件屬性、方法可以進行綁定
- [Directive](#) 將 DOM 轉換為多功能的「宣告命令」
- [Service](#) 由「服務」集中管理資料與運算邏輯
- [Dependency Injection](#) 由「相依注入」機制管理物件生命週期



Build your own Angular 2 Application

快快樂樂建立 ANGULAR 2 應用程式

準備 Angular 2 開發環境

- [架設 Angular 2 開發環境說明文件](#)
- [如何修改 Visual Studio Code 內建的 TypeScript 版本](#)
- [關於 TypeScript 2.0 之後的模組定義檔 \(*.d.ts \)](#)

使用 Angular CLI 建立專案範本

- 請務必在 NTFS 檔案系統執行相關命令 (Windows)
- `ng new demo1`
 - 初始化 Git 儲存庫
 - 安裝 npm 相依套件
- `cd demo1`
- `ng serve`
 - <http://localhost:4200>

了解專案結構

- 首頁 HTML 與 Angular 2 主程式
 - src/index.html 預設網站首頁
 - src/main.ts 預設 Angular 2 啟動器 (主程式)
- 公用檔案資料夾
 - public/ 放置網站的相關資源檔案 (CSS,Image,Fonts,...)
- 根元件
 - src/app/app.module.ts 應用程式的NgModule定義檔
 - src/app/app.component.ts 根元件主程式
 - src/app/app.component.html 根元件範本檔 (Template)
 - src/app/app.component.css 根元件樣式檔 (CSS)
 - src/app/app.component.spec.ts 單元測試定義檔
 - src/app/shared/index.ts 根元件的相關服務元件
 - src/app/environments/environment.ts 環境變數設定
 - src/app/index.ts 載入根元件的預設檔

src/index.html

index.html



```
1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <title>Demo1</title>
6   <base href="/">
7
8   <meta name="viewport" content="width=device-width, initial-scale=1">
9   <link rel="icon" type="image/x-icon" href="favicon.ico">
10
11 </head>
12 <body>
13   <app-root>Loading...</app-root>
14 </body>
15 </html>
```

咦？沒有載入任何 JavaScript 函式庫？

根元件的 directive 宣告



src/main.ts

```
main.ts x
1  import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
2  import { enableProdMode } from '@angular/core';
3  import { AppModule, environment } from './app/';
4
5  if (environment.production) {
6    enableProdMode();
7  }
8
9  platformBrowserDynamic().bootstrapModule(AppModule);
10
```

啟用 Production 模式 (提升執行速度)

設定 AppModule 為啟動模組

src/app/app.module.ts

app.module.ts x

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule, ApplicationRef } from '@angular/core';
3 import { CommonModule } from '@angular/common';
4 import { FormsModule } from '@angular/forms';
5 import { AppComponent } from './app.component';
6
7 @NgModule({
8   declarations: [ ← 宣告跟 View 有關的元件
9     AppComponent
10  ],
11  imports: [ ← 宣告要匯入此模組的外部模組
12    BrowserModule,
13    CommonModule,
14    FormsModule
15  ],
16  providers: [], ← 宣告要註冊的服務元件
17  entryComponents: [AppComponent], ← 幾乎用不到此設定
18  bootstrap: [AppComponent] ← 宣告根元件
19 })
20 export class AppModule {
21 |
22 }
```

src/app/app.component.ts

app.component.ts ✕

```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root', ← 指令 (directive) 選擇器
5    templateUrl: 'app.component.html', ← 元件網頁範本
6    styleUrls: ['app.component.css'] ← 元件 CSS 樣式
7  })
8  export class AppComponent { ← TypeScript 類別
9    title = 'app works!'; ← 類別中的屬性 (Property)
10
11    changeTitle(title: string) { ← 類別中的方法 (Method)
12      this.title = title;
13    }
14  }
15
```

Angular 2 實戰開發

LIVE DEMO



資料繫結的四種方法 (Binding syntax)

- 內嵌繫結 (interpolation)
{{property}}
- 屬性繫結 (Property Binding)
[property]='statement'
- 事件繫結 (Event Binding)
(event)='someMethod(\$event)'
- 雙向繫結 (Two-way Binding)
[(ngModel)]='property'

範本參考變數 (Template reference variables)

- 在範本中任意 HTML 標籤套用 **#name** 語法
 - 會在範本內建立一個名為 **name** 的區域變數
 - 該 **name** 區域變數將只能用於目前元件範本中
 - 該 **name** 區域變數將會儲存該標籤的 DOM 物件
 - 你可以透過「事件繫結」將任意 DOM 物件中的任意屬性傳回元件類別中 (Component class)
- 以下這兩種是完全相等的語法 (使用 **#** 是語法糖)
 - **#name**
 - **ref-name**

三種 Angular 指令 (Directives)

<https://github.com/miniasp/ng2demo>

- 元件型指令
 - 預設「元件」就是一個含有樣板的指令 (最常見)
- 屬性型指令
 - 這種指令會修改元素的外觀或行為
 - 例如內建的 [NgStyle](#) 或 [NgClass](#) 指令就可讓你自由的變更樣式
- 結構型指令 ([Structure Directives](#))
 - 這種指令會透過新增和刪除 DOM 元素來改變 DOM 結構
 - 例如內建的 [NgIf](#)、[NgFor](#) 或 [NgSwitch](#) 就可以用來控制 DOM 結構
 - 請注意 ngSwitch 前面不要加上 * 星號
 - 請注意 ngIf 與 ngFor 與 ngSwitchDefault 與 ngSwitchCase 前面要加上 * 星號

匯入 Http 模組與使用 Http 服務元件

- 匯入 HttpModule
- 注入 Http 服務元件
- 發出 HTTP GET 要求
- 訂閱 HTTP GET 執行結果

會不會覺得 Angular 2 很帥？

假的！

假的！(才怪)

相關連結

- [Angular 2 官網](#) ([官方簡體中文翻譯](#))
- [Angular 2 風格指南](#) (官方版)
- [Angular 2 學習資源](#) (官方版)
- [Angular 2 學習資源](#) (社群版)
- [ng-conf 2016 – YouTube](#)

- [TypeScript - JavaScript that scales.](#)
- [TypeScript Handbook \(中文版\)](#)

- [前端工程的夢幻逸品：Angular 2 開發框架介紹](#)

- [Angular 2 Fundamentals | AngularClass](#) (免費 ng2 課程)

聯絡資訊

- The Will Will Web

記載著 Will 在網路世界的學習心得與技術分享

– <http://blog.miniasp.com/>

- Will 保哥的技術交流中心 (臉書粉絲專頁)



– <http://www.facebook.com/will.fans>

- Will 保哥的噗浪

– <http://www.plurk.com/willh/invite>

- Will 保哥的推特

– https://twitter.com/Will_Huang