



多奇·教育訓練

Angular 13 開發實戰 新手實作篇

(題目本)

多奇數位創意有限公司

全端工程師 黃升煌 (Mike)

<https://fullstackladder.dev>





主軸1：Angular 專案基礎架構

任務01：建立專案基礎架構

- 請使用 `ng new` 指令，建立一個全新的 `Angular` 專案

任務02：現有模板切版，及元件規劃

- 下載模板

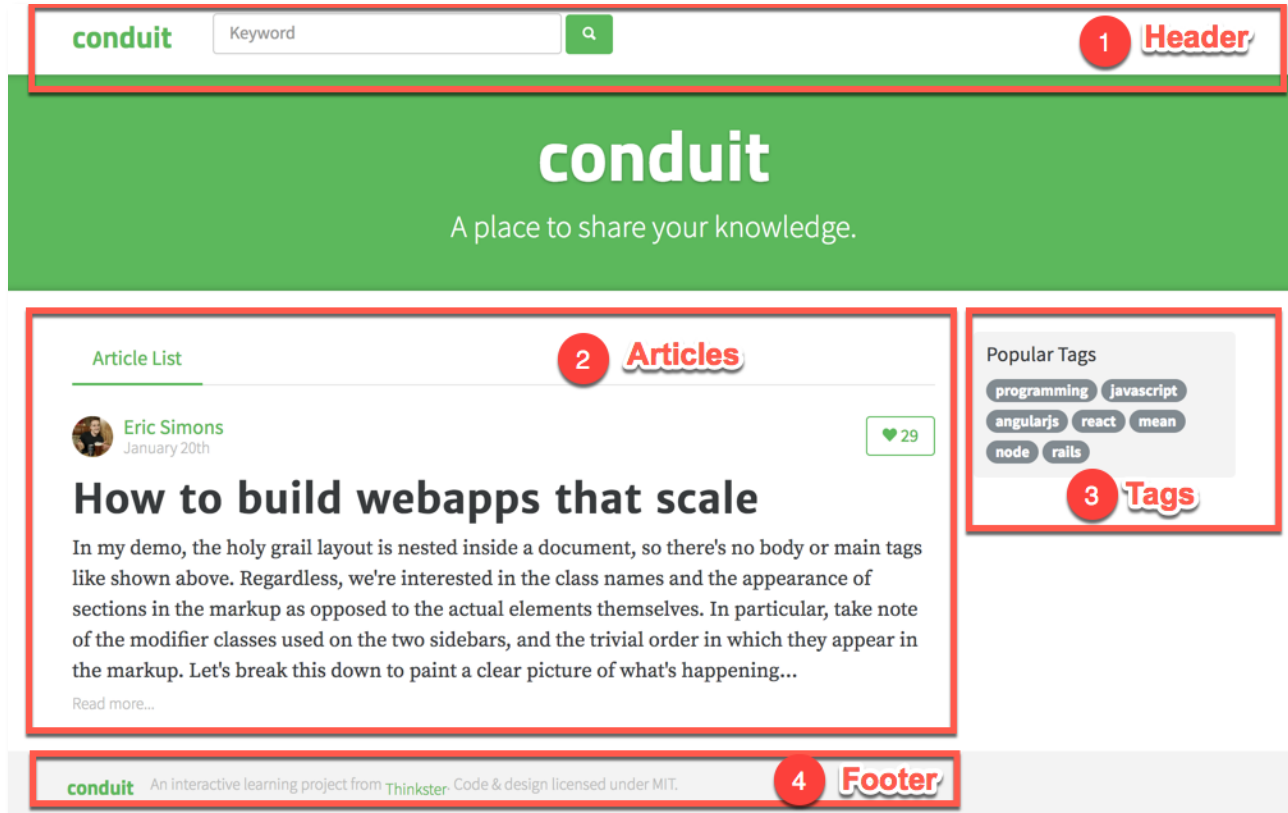
The screenshot shows the GitHub interface for the repository 'coolrare / realworld-basic-template'. At the top, there are buttons for 'Unwatch', 'Star' (0), and 'Fork' (0). Below this is a navigation bar with tabs for 'Code', 'Issues' (0), 'Pull requests' (0), 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area shows a message 'No description, website, or topics provided.' with an 'Edit' button. Below this, it lists '5 commits', '1 branch', '0 releases', and '1 contributor'. A bar shows the current branch 'master' and a 'New pull request' button. To the right are buttons for 'Create new file', 'Upload files', 'Find File', and a green 'Clone or download' button. A dropdown menu is open from the 'Clone or download' button, showing options to 'Clone with HTTPS' (with a help icon), 'Use SSH', and a text input field containing the URL 'https://github.com/coolrare/realworldc'. Below the URL is a clipboard icon. At the bottom of the dropdown are two buttons: 'Open in Desktop' and 'Download ZIP' (which is highlighted with a red rectangle). The repository files list shows 'assets' (Update style.css) and 'index.html' (移除不必要的內容). At the bottom, there is a blue box with the text 'Help people interested in this repository understand your project by adding a README.'

任務02：現有模板切版，及元件規劃

- 將 assets 目錄內容搬到專案的 src/assets 中
- 將 index.html 內容搬到 src/index.html
- index.html 的 `<head></head>` 區塊中
 - 加入 `<base href="/">`
- 將 index.html 內容的 body 區塊
 - 剪下移到 app.component.html
- 在 index.html 的 body 區塊中
 - 加上 `<app-root></app-root>`

任務02：現有模板切版，及元件規劃

- 將 `app.component.html` 的內容，依照下圖指示切割成 4 個元件

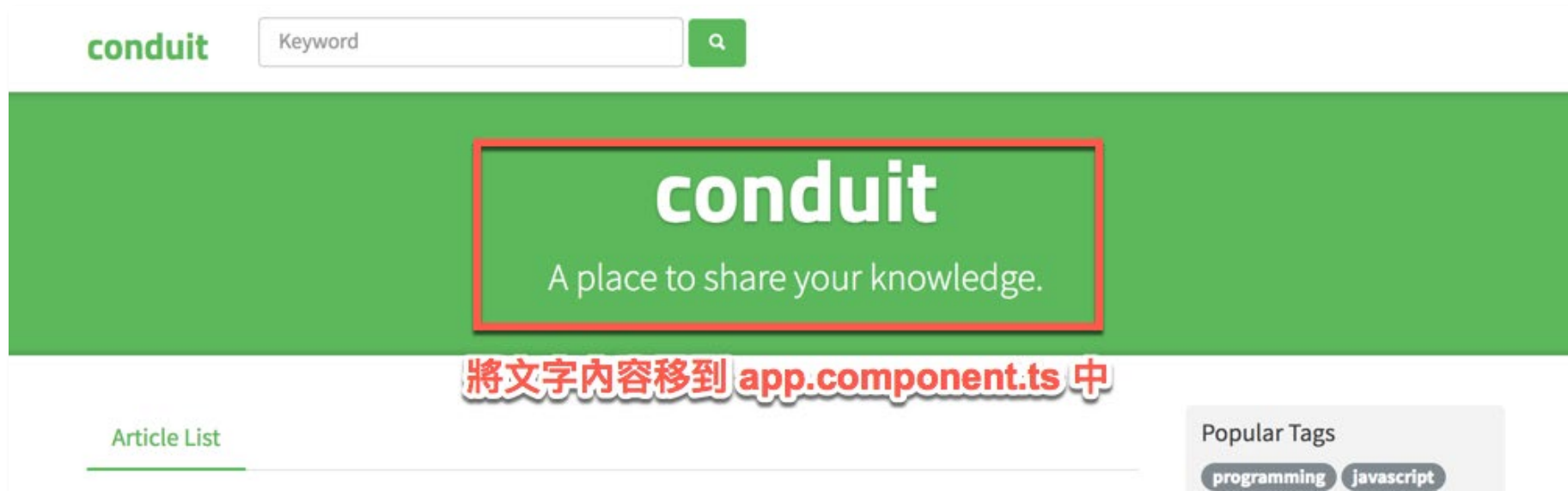




主軸2：Angular 畫面呈現與應用

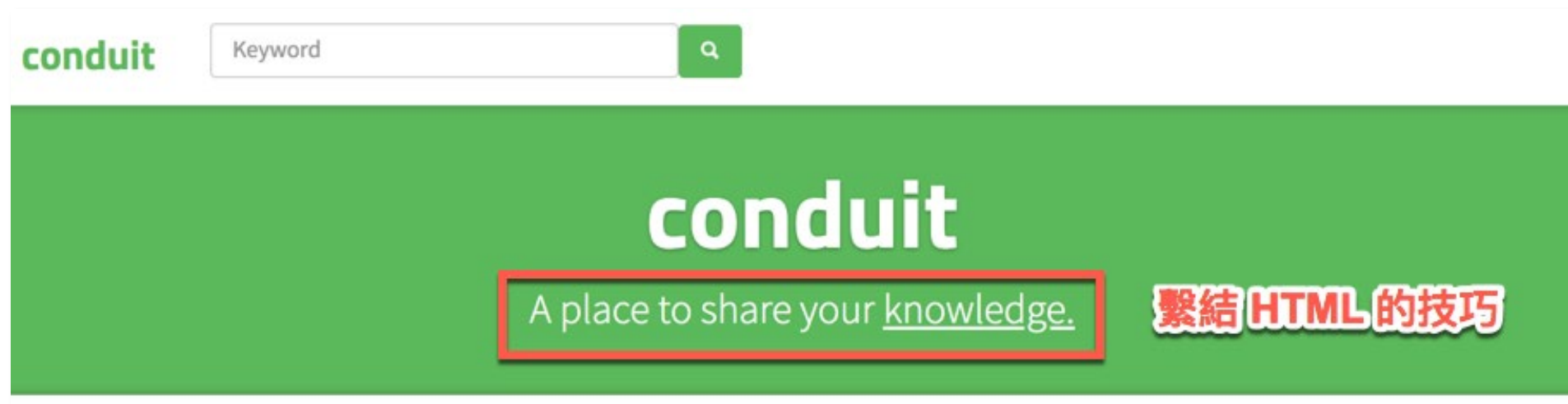
任務03：使用內嵌繫結顯示資料

- 說明：
 - 請將圖片中紅色框框內的文字內容(title 與 subtitle)，移動到 `app.component.ts` 中
 - 使用內嵌繫結 (Interpolation) 的方式，顯示在畫面上



任務04：使用屬性繫結顯示資料

- 目標：
 - 改使用屬性繫結將資料呈現在畫面上
 - 請嘗試呈現包含 HTML 標籤的資料

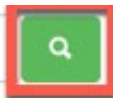


任務05：使用事件繫結

- 目標：
 - 使用事件繫結來得知按鈕按下的事件
 - 請自行嘗試在按下按鈕後改變元件內的一些行為

conduit

Keyword



conduit

A place to share your <u>knowledge.</u>

任務06：使用雙向繫結

- 目標：
 - 在輸入框使用雙向繫結(two way binding)綁定資料
 - 將綁定的資料顯示在其他地方
 - 透過內嵌繫結或屬性繫結

conduit

test



test

conduit

A place to share your knowledge.

任務07：使用屬性指令(Attribute Directive)

- 目標：
 - 每次按下按鈕時，使用 `[class]`，替文字切換 `highlight` 樣式
 - 自行在元件中加入 `highlight` 樣式
 - 每次按下按鈕時，使用 `[style]`，放大文字大小

conduit

Keyword



透過按鈕切換文字樣式

conduit

A place to share your knowledge.

任務08：使用結構指令(Structural Directive)

- 目標 1：當未輸入查詢文字時，使用 `*ngIf` 提示使用者

conduit

Keyword



請輸入查詢條件

當未輸入關鍵字時，顯示提示文字 (*ngIf)

conduit

A place to share your knowledge.

任務08：使用結構指令(Structural Directive)

- 目標 2：使用 *ngFor 顯示清單資料
 - 資料來源





主軸3：Angular 中元件與元件傳遞 資料的技巧

任務09：使用 @Input() 傳遞資料給子元件

- 目標：
 - 將文章清單資料移動到 AppComponent 中
 - 替 ArticlesComponent 加入 @Input() 接收外部傳來的資料

任務10：使用 @Output() 接受元件輸出的資料

- 目標：完成基本搜尋功能
 - 替 HeaderComponent 加入 @Output() 自訂事件
 - 在 AppComponent 中取得 HeaderComponent 中的搜尋結果，並過濾清單資料





主軸4：服務 (service) 概念與應用

任務11：自訂服務元件

- 目標：
 - 將文章清單與查詢功能都移動到一個 Service 統一管理
 - 各個元件中不再存放資料與查詢程式，而是統一透過 Service 處理

任務12：使用 HttpClient 服務

- 目標：
 - 使用 HttpClient 服務串接真實的 RESTful API
 - API 規格
 - 測試用的 API Endpoint
 - 自行建立 JSON Server



主軸5：其他 Angular 功能特色

任務13：使用Angular內建的Pipe

- 目標：
 - 使用 DatePipe 來改變日期顯示格式
 - 格式：yyyy-MM-dd



改變日期顯示格式

Markdown Test

Testing markdown syntax

[Read more...](#)

任務14：自訂 Pipe

- 目標：
 - 建立一個自訂的 `FilterArticlePipe`
 - 將過濾文章的程式邏輯移動到 `FilterArticlePipe` 中
 - 頁面透過 `FilterArticlePipe` 來過濾文章



多奇·教育訓練

THANK YOU!

Q&A