

開發環境與工具

測試伺服器環境、套件管理工具

現在都會建置一個開發用的伺服器環境，用來模擬程式執行，因為 Node 與 NPM 工具用於管理 JavaScript 的各模組相當方便，所以這是在實務上都會這樣使用的方式。一開始我們要先在電腦上安裝 Node，它是一套可以在伺服器環境執行 JavaScript 程式的系統，裡面就有包含 NPM 工具，NPM 工具則是專門用於管理 JavaScript 套件相依性的。下面是它的官方下載連結：

- [Node 官方網站](#)

在官方網站上，你可以看到有兩個下載的按鍵區塊，一個是像是「8.12.0 LTS」後面有 LTS 字詞的，但版本號碼可能不一樣，另一個是「10.10.0 Current」後面有 Current(目前最新) 字詞的。LTS 是長期維護版本的意思，如果你並沒有要開發或學習 Node，也就是伺服器端的 JavaScript 程式，下載這個 LTS 版本就可以。

Node 的安裝就像是一般你在作業系統上安裝軟體一樣，會有一個安裝的精靈程式，引導你一步步設定然後安裝，設定的部份使用預設就可以，這部份不會有什麼困難的地方。

麻煩的地方可能會出現在日後的維護和升級部份，對於 Windows 作業系統的開發者來說，如果要升級就直接下載新的版本，再重新安裝一遍就可以。macOS 作業系統的開發者來說，也可以用重新安裝的方式，或是為了日後方便起見，改用 Homebrew 來安裝 Node。不過，如果你也是要學習 Node(伺服器端的 JavaScript)要如何開發，那麼安裝多個 Node 版本的方式，是一定要學習的，至少你的作業系統裡要安裝 LTS 和 Current 兩種版本，這樣才能方便測試和使用新的功能，這個時候就需要用 n 或 nvm 這種套件(它們都是 NPM 的套件)，來管理和切換到不同的版本。

程式碼撰寫工具、相關套件

現在通常的程式碼撰寫工具，都使用 Visual Studio Code(VSC) 這套由微軟出品的軟體，它是開放原始碼的專案，而且可以在各種作業系統上使用，最棒的是它還提供了中文的操作介面。不過 Visual Studio Code 與微軟的另一個知名的開發整合式工具 Visual Studio 是完全不同的軟體，這個千萬不要搞混了。下面是它的官方下載連結：

- [Visual Studio Code](#)

在官方網站上有一個「Download for Mac」或「Download for Windows」字詞的下載區塊，按下後就可以下載這個開發工具。然後依照一般的軟體安裝方式來安裝到你的電腦中。

Visual Studio Code 的程式畫面如下，它只有幾個區域，所以很簡單就可以理解每個區域是作什麼用的：

Visual Studio Code 一開始安裝好之後，只有內建的功能和套件，所以我們還需要利用它的擴充套件功能，來安裝一些搭配的套件，以下是幾個常用的 JavaScript 套件，你從擴充套件中可以用名字來搜尋它們，然後安裝它們：

- Babel JavaScript - 支援最新的 JavaScript 語法和 API，可以高亮度顯示
- ESLint - 搭配 ESLint 套件使用
- vscode-icons - 不同檔案以特定圖示顯示
- Prettier - 程式碼格式化排版用

當然套件還有很多，這裡只是一些最常用的而已，你可以視需求再加裝其它的套件，例如搭配除錯器，或是特定 JavaScript 函式庫或框架使用的。有些套件是安裝後就會自動套用，有一些則是需要再進行設定才能正確運作，這部份就要看每個套件的說明文件。

Visual Studio Code 從下方的狀態列點按後，可以叫出主控台的區域，裡面就有可以輸入命令的終端機功能。或是從上方的選單中「檢視」->「整合式終端機」也可以達成。在專案中我們需要加入新的套件模組，或是啟動測試伺服器，都需要輸入指令，所以知道如何打開終端機的輸入命令的介面也是很重要的。

上一章節我們裝好了 Node 與 NPM 工具，你可以試著在終端機輸入命令的介面，輸入以下的指令，看看 Node 與 NPM 工具是不是已經安裝好了，這兩行指令是要取得目前已安裝的版本編號：

```
node -v
```

```
npm -v
```

啟動專案工具(Starter Kits)或專案樣版文件(boilerplate)

現今在專案開發時，如果是較為大型的專案，通常都會由開發小組自己維護一個專案的樣版，裡面會先設定好所需的套件、相關的測試設定等等。然後其它有參與的開發者，會直接從這個專案樣版，來建立自己的開發用的專案。不過，這種專案樣版文件，通常都會針對某些特定的函式庫或框架來設計，在網路上也有很多這類的專案樣版文件，針對不同的需求而加入各種套件。

當然你也可以自己建立一套，不過因為專案樣版文件通常會包含以下幾個套件或工具，所以你也需要多花時間一一了解：

- babel
- webpack
- eslint
- hmr
- test
- css module

而且專案樣版文件麻煩的地方除了設定外，還有日後的更新與升級，所以我會建議初學者用現成的專案樣版文件即可，因為我對 React 開發較為熟悉，所以我會推薦直接使用下面這套專用於 React 學習與開發的專案樣版工具：

- [create-react-app](#)

要注意的是，這工具並不是直接從上面的網站中下載來安裝，它是用你剛剛安裝好的 NPM 工具，來安裝到你的電腦中。我們在終端機中輸入以下的指令(下面install 可以只簡寫為 i)：

```
npm install -g create-react-app
```

這行指令會安裝create-react-app工具程式到你的電腦中，-g的參數代表它是要安裝到電腦的全域，也就是你的電腦各地方都可以直接用這個工具。

安裝完成後，我們可以用這個指令create-react-app來建立新的專案，在終端機中輸入以下的指令：

```
create-react-app my-app
```

這行指令會在你目前終端機所在的目錄位置(通常是使用者各別的目錄中)，建立一個新的名為my-app資料夾，然後把相關的專案套件安裝在裡面。這需要一點時間來安裝整個模組套件，當然因為我們現在只是初學

JavaScript 而已，這裡面有一半以上的東西還不會用到，用這專案樣版工具來建立新專案只是圖個方便而已。

再來開啟一個新的 Visual Studio Code 的新視窗，然後點按"開啟資料夾"，指向你剛剛新建立的專案my-app 資料夾，在左邊的檔案總管區域應該會看到目前專案的所有檔案列表。接著再開啟在終端機中輸入以下的指令：

```
npm start
```

這個指令可以啟動測試伺服器，並且會自動開啟你的預設瀏覽器，連到<http://localhost:3000>這個網路位置，localhost 代表的是你自己這台電腦，3000 指的是埠號 3000，這是它的預設伺服器用的埠號。如果你能正確地看到執行的預設頁面，那麼就是這個專案樣版已經安裝成功了。

在專案的檔案中，我們在一開始學習時，只會用到兩個檔案，一個是 src 目錄裡的 index.js，另一個是 public 目錄中的 index.html 檔案，其它的檔案你可以先不用管它們。測試伺服器可以先啟動，然後把 index.js 先清除裡面的內容，開始練習撰寫你的程式碼，因為有熱模組抽換的功能，所以你在修改程式碼的同時，瀏覽器裡的測試網頁也會一起跟著重新執行出結果。

最後，如果你需要把寫好的程式，傳到正式的網站上執行，你可以在終端機中輸入以下的指令：

```
npm run build
```

上面這行指令，會將目前的程式碼進行編譯和壓縮，你會看到有一個 build 資料夾出現在你的專案目錄中，這裡面就是編譯過的 index.html 和 js 相關檔案，傳到伺服器上就可以執行。當然，有一些在 src 或 public 目錄中沒用到的檔案在編譯前最好先移出專案之外，以免造成編譯的檔案過大。