

第 1 章 - 開發環境與專案建置

開發環境

- JS 執行環境/npm 管理程式: [Node.js](#) - 下載與安裝 LTS 版本
- 程式碼編輯程式 - [Visual Studio Code\(vscode\)](#) - 下載與安裝對應作業系統
- Chrome 瀏覽器擴充外掛 - [React Developer Tools](#) - 安裝後請重開瀏覽器，確認有啟動在工具列上

註: 安裝完請確認先啟動一次各軟體，以及在終端機(命令列)視窗輸入 `node -v`，檢查是否都已經安裝完成與啟動。

Next 樣版專案安裝步驟

專案位置在: <https://github.com/mfee-react>

1. 複製儲存庫(參考[步驟文件](#))
2. 安裝專案套件

切換到複製下來的專案目錄後，輸入以下指令

```
npm i
```

3. 安裝 Visual Studio Code 擴充套件

前三者為必要，第四個是中文語言操作介面，可視需求安裝：

- Babel JavaScript
- ESLint
- Prettier - Code formatter
- Chinese (Traditional) Language Pack

{自訂安裝} Next 專案安裝 (使用課堂用樣版專案可以略過)

```
npx create-next-app@latest
```

執行指令後會出現以下關於專案建立的問答:

```
What is your project named? my-app
Would you like to add TypeScript with this project? Y/N
Would you like to use ESLint with this project? Y/N
Would you like to use Tailwind CSS with this project? Y/N
Would you like to use the `src/` directory` with this project? Y/N
What import alias would you like configured? `@/*`
```

初次建立，可依照以下輸入後，按下 Enter 依序回應，之後會開始建立專案：

```
next-course
N
Y
N
N
N
```

以下為各問答項目的對照中文說明與建議：

```
project name (專案名稱。請使用英文全小寫加連接符號(`-`)，例如`next-230808`或`next-project`): next-course
TypeScript (TypeScript 支援。目前暫不需要。): N
ESLint (ESLint 支援。先安裝內建基本的，以下還需額外安裝其它搭配套件。): Y
Tailwind CSS (Tailwind CSS 支援。可以先不用裝，如果有需要可以之後再加上即可。): N
`src/` directory (`src/` 目錄使用。會多出一層src目錄，視專案團隊需求，可加可不加，不影響專案。): N
import alias: N (導入模組時從根目錄導入的別名。預設為`@`，除非有需要，否則使用預設即可。)
```

註：如果問答項目回應錯誤，或是在建立專案(從伺服器下載套件)時發生錯誤中斷，通常是網路問題造成，可以刪除專案再重新建立新的專案。

{自訂安裝} ESLint/Prettier 額外安裝 (使用課堂用樣版專案可以略過)

第 1/4 步驟

在終端機(命令列)視窗安裝額外的 `eslint` 與 `prettier` 模組：

```
npm install -D eslint eslint-plugin-react eslint-plugin-react-hooks
eslint-plugin-jsx-ally prettier eslint-plugin-prettier eslint-config-
prettier eslint-import-resolver-alias eslint-plugin-import
```

第 2/4 步驟

拷貝 `.vscode` 資料夾(其中有一個 `settings.json` 檔案)，到專案根目錄，裡面有相關的針對專案的設定值。

拷貝以下四個檔案到專案根目錄，其中 `eslinttrc.json` 需要覆蓋原本剛建立的的檔案：

- `.eslinttrc.json`
- `.eslintignore`
- `.prettierrc.json`
- `.prettierignore`

第 3/4 步驟

安裝以下的 Visual Studio Code 擴充外掛(註: 前三者為必要，第四個是中文語言操作介面，可視需求安裝)：

- Babel JavaScript
- ESLint
- Prettier - Code formatter
- Chinese (Traditional) Language Pack

安裝完成後，請整個重新啟動 Visual Studio Code 程式。關閉所有應用程式視窗再重新啟動即可。

第 4/4 步驟

檢查是否正常運作。

開啟 `pages/_app.js` 檔案，在最後加入以下的測試程式碼：

```
const a = 1
a = 2
```

觀察 vscode 編輯畫面是否會進行自動排版，也有正常回報錯誤 (錯誤處會有明顯的紅色下底波浪線)。

如果沒有正常運作，請從 1/4 步驟再檢查是否有安裝正確。

啟動專案命令

啟動專案(開發模式)

```
npm run dev
```

打包專案

注意: next 專案會進行最佳化與檢查後才會打包輸出檔案，如果發生錯誤或 ESLint 檢查出問題會無法正確地執行完成

```
npm run build
```

啟動專案(營運模式)

注意: 要先 `build` 後，next 專案會進行最佳化與檢查後才能使用此指令，如果發生錯誤或檢查出問題會無法啟動

```
npm run start
```

Visual Studio Code 設定

建議設定

1. 設定自動儲存檔案

在選單中勾選 "檔案 > 自動儲存" (選單英文語言介面為: "File > Auto Save")

2. 登入 Github 帳戶，同步編輯器的設定與 github 連結

3. 中文字體

Windows 系統可以在字體設定("editor.fontFamily")中，加入 '微軟正黑體' (通常是第二個)

或是在電腦中安裝 [Noto Sans Traditional Chinese](#) 繁中字體，在字體設定("editor.fontFamily")中，加入 'Noto Sans TC' (通常是第二個)

其它擴充外掛

- Nextjs snippets(用片段縮寫快速建立程式碼樣版，[對照表](#))
- ES7+ React/Redux/React-Native snippets(用片段縮寫快速建立程式碼樣版，[對照表](#))
- vscode-icons - 區分不同檔案的圖示和顏色
- Todo Tree - 加上 todo(待解決或記錄點)
- Blockman - 區塊標明確畫出區塊
- Better Comments - 註解更多顏色呈現
- Console Ninja - 直接在程式碼中呈現 console 結果
- styled-jsx - 高亮度呈現 style jsx 語法

額外設定

自訂開啟視窗顏色

在 `.vscode` 資料夾中開啟 `setting.json` 檔案，依三種顏色色碼調整：

```
"workbench.colorCustomizations": {  
  "activityBar.background": "#332C2D",  
  "titleBar.activeBackground": "#483D3F",  
  "titleBar.activeForeground": "#FBFAFA"  
}
```

其它

更多資訊

- 要提高專案建立時的速度或效率，可以安裝與使用 yarn 或 pnpm

改變啟動 port 號

註: 以下為 dev 修改，start 修改一樣

第 1 種方式(適合長期改動): `package.json` 中的 dev 啟動指令改為以下:

```
"dev": "next dev -p 3005"
```

第 2 種方式(適合暫時改動): 啟動時使用以下的指令

```
npm run dev -- -p 3002
```