



NPM 基本指令 與套件維護

本篇教學使用iOS語法為例

Eason



NPM

NPM 是 Node.js 編寫的，
所以需要安裝 Node.js 才能使用。

01

基本簡寫說明

本篇教學使用iOS語法為例

基本簡寫說明

| 簡寫語法 | 說明 |
|------|--|
| -v | --version，查看目前npm 版本。 |
| -g | --global，表示安裝包時，視作全域的包。安裝之後的包將位於系統預設的目錄之下。 |
| -S | --save，表示安裝的包將寫入 package.json 裡面的 dependencies。 |
| -D | --save-dev，表示安裝的包將寫入 package.json 裡面的 devDependencies。 |
| -O | --save-optional，表示安裝的包將寫入 package.json 裡面的 optionalDependencies。 |
| -E | --save-exact，表示安裝的包的版本是精確指定的。 |
| -B | --save-bundle，表示安裝的包將寫入 package.json 裡面的 bundleDependencies。 |
| -f | --force，表示不管是否安裝過，都要強制重新安裝。 |

02

基本指令

本篇教學使用iOS語法為例

基本指令 (1)

- 檢查npm目前版本號
(檢測npm 是否正常運作的常用指令)

command line

```
$ npm -v
```

- 安裝最新版本npm

command line

```
$ npm i npm@latest -g
```

- 『npm i』 為 npm install 的縮寫
- 『@latest』 表示尋找線上目前最新版本
- 『-g』 為 --global，表示

基本指令 (2) - 初始化

- 新建package.json，npm 初始化
 1. 請先建立新專案資料夾 並 切換至專案根目錄。

command line

```
$ mkdir demo-project  
$ cd demo-project
```

2. npm 初始化

command line

```
$ npm init
```

3. 初始化後，進行相關設定。

| 關鍵字 | 說明 |
|-----------------------------|---|
| <code>name</code> | 專案名稱。 |
| <code>version</code> | 專案版本。 |
| <code>description</code> | 表示安裝的包將寫入 <code>package.json</code> 裡面的 <code>dependencies</code> 。 |
| <code>entry point</code> | 專案切入點。 |
| <code>test command</code> | 專案測試指令。 |
| <code>git repository</code> | 專案原始碼的版本控管位置。 |
| <code>keywords</code> | 專案關鍵字。 |
| <code>author</code> | 專案作者 (以 <code>author-name <author@email.com></code> 寫之)。 |
| <code>license</code> | 專案版權 |

- 亦可略過設定，生成 package.json 後，再進行修改。

command line

```
$ npm init -y
```

- 『-y』為 --yes 的縮寫
- 設定 init 預設值
(為 npm init 設置默認值，這些訊息會存放在用戶主目錄的 ~/.npmrc 文件)

command line

```
$ npm set init-author-name <Your name>  
$ npm set init-author-email <Your email>  
$ npm set init-author-url <Your Url>  
$ npm set init-license 'MIT'
```

- 加入模塊時，紀錄確切版本，而不是一個可選的版本範圍

command line

```
$ npm set save-exact true
```

基本指令 (3) - 查看套件資訊

- 查看本機安裝資訊

- 查看專案內引用的套件，若想查看全域，加上『-g』

command line

```
$ npm list
```

- 查看某個套件

command line

```
$ npm list grunt
```

- 查看套件線上資訊

command line

```
$ npm info <packageName>
```

基本指令 (4) - 套件管理

- 安裝套件 (若要鎖定版號安裝，請在套件名稱之後加上『@版號』，無空格，如 @1.0.3)

command line

```
$ npm i <packageName>
```

- 卸載套件

command line

```
$ npm un <packageName>
```

- 『npm un』 為 npm uninstall 的縮寫

- 更新套件

command line

```
$ npm update <packageName>
```

基本指令 (5) - 其他指令

- 搜索套件

command line

```
$ npm search <packageName>
```



03

npm i 和 npm ci



npm i

- 『 `npm install` 』 的縮寫。
- 按名稱搜索模塊。
- 下載並安裝模塊及其依賴項。
- 更新（或創建） `package-lock.json`。
該文件稱為lockfile，它列出了每個已安裝套件的 URL 和校驗
- 將模塊名稱和版本添加到 `package.json`。此文件稱為清單。
- 執行 `npm i`，通常會有兩個東西一起出現：
 1. `node_modules`
 2. `package-lock.json`
(`package-lock.json`，會詳細記錄套件的版本號，主要目的是為了鎖定套件安裝時的依賴版本。)

npm ci

- 『 `npm clean-install` 』 的縮寫，在 npm 5.7.1才推出。
- npm ci 是依據 package-lock.json 來安裝。
- 必須有 package-lock.json檔案。
- npm ci 只能還原全部套件，無法單一還原。
- 會先刪除 node_modules 並進行全新安裝。
- 安裝過程若發生依賴錯誤或與package.json不匹配，npm ci 不會更新 package.json，而是直接退出。
- npm ci 永遠不會更新 package.json，所以安裝的版本基本上是鎖定的。