

打造個性化的 Neovim 編輯器

居正中 Alan Jui

May 15, 2022

Abstract

Neovim 0.5 的發行，令 Neovim 推進到一個更加高階的里程碑。對於有心打造符合自己需求，個人專用編輯器的使用者，Neovim 0.5 支援 LSP 及 Lua Script 的新功能，讓懷抱著這個夢想的人們，可以有機會美夢成真。

【註】：Neovim Lanage Server protocol (LSP) 主要功能說明，請參考本文件在 6.1 處之內容。

當 Neovim 0.5 還在 Beta 版，便已開始使用，如今 Neovim 0.6 都已推出了，個人對 Neovim 的掌握度，仍是一知半解。想要擺脫這狀態，所以需要進行深入的研究；對於研究該如何開展、如何檢驗是否達標，所以需要設定目標。

最後，將個人在 Django Project 應用的需要，彙集成這個研究的「專案需求」如下：

- 程式編碼可透過自動補全，加快輸入及避免打字錯誤
- 程式碼在呼叫某 method/function 時，能「顯示用法」，提示該輸入的引數順序，及應使用的資料類型 (Data Type)
- 可使用 snippets，加速編碼工作及避免打字錯誤
- 原先在 VS Code 已建置的 Snippets，能於 Neovim 環境套用
- 自動檢閱程式碼，確保沒有語法的錯誤
- 程式碼已被檢閱到的錯誤，可提示：「統計總數」、「標示位置」
- 適用於 Django 開發專案
- 能依據語法標準（如：autopep8），自動調整程式碼的排版格式
- 可編輯及預覽 Markdown 文件，以便可以與 VuePress 整合，作為「技術文件」編輯器
- 可使用如 PlantUML 的工具，以文字描述，繪製 UML 圖形 (Diagrams)
- 可以透過 DAP 與 Neovim 整合，讓 Neovim 可像 VS Code 一樣，當作除錯 (Debug) 工具來使用

Neovim 在執行時期，對於設定檔存放目錄及插件存放目錄，有其預設如下：

- 設定 (Configuration) 存放目錄路徑： `/.config/nvim/`
- 執行 (Runtime) 存放目錄路徑： `/.local/share/nvim/`
- 插件存放目錄路徑： `/.local/share/nvim/site/pack/packer/start/`

目前網路各大高手及高高手所分享的 Neovim 設定，幾乎都是遵循上述預設而成。可是，對於我這種入門新手，個人的期望是：「在鑽研 Neovim 的過程中，需要不斷參考，各個高人們的心得成果，然後自行實作、驗證自己是否理解，最後決定是否要採用，納入本專案的產出：my-nvim」。所以，我需要兩個相互不影響的「工作空間」，一個為參考用；另一個則為實作用。

基於上述的這個需求，my-nvim 被設計成不會佔用 Neovim 的預設目錄路徑：`/.config/nvim` 及 `/.local/share/nvim`。這個設計，固然能帶來各自獨立的好處；但也有副作用的麻煩：您得改變以 nvim 指令啟動 Neovim 的習慣。

Contents

1	安裝前準備作業 (Prerequisites)	1
1.1	完成作業系統 Shell 環境設定	1
1.2	完成 Python 開發環境設定	1
1.3	完成 Node.js 開發環境設定	1
1.4	完成 Lua 開發環境設定	1
2	安裝與設定作業 (Setup process)	2
3	後續規劃 (Todos)	3
4	快捷鍵 (Bindings)	4
5	使用插件 (Plugins)	5
6	參考資料 (References)	6
6.1	Neovim Language Server Protocol (LSP)	7

1 安裝前準備作業 (Prerequisites)

安裝 Neovim 之前，以下四大開發作業環境最好先備妥：

1. 作業系統 Shell 環境
2. Python 開發環境
3. Node.js 開發環境
4. Lua 開發環境

Figure 1: Neovim 0.5 為文字編輯器開創應用新紀元

- 1.1 完成作業系統 Shell 環境設定
- 1.2 完成 Python 開發環境設定
- 1.3 完成 Node.js 開發環境設定
- 1.4 完成 Lua 開發環境設定

2 安裝與設定作業 (Setup process)

3 後續規劃 (Todos)

4 快捷鍵 (Bindings)

5 使用插件 (Plugins)

6 參考資料 (References)

6.1 Neovim Language Server Protocol (LSP)

Neovim LSP 主要功能摘要：

- Go to definition：定義跳轉
- (auto)completion：自動補全
- Code Actions (automatic formatting, organize imports, ...)：程式碼操作，如：自動調整排版、組織 import 順序... 等
- Show method signatures：顯示用法
- Show/go to references：顯示/跳轉引用處
- snippets：程式碼片段