Conda

Conda是一個開源套件管理系統和環境 管理系統,可以在Windows、macOS和 Linux 等作業系統上運行。Conda 可以快 速安裝、執行和更新套件。Conda在本 地端的電腦上輕鬆創建、保存、載入及 切換環境。它雖然是為Python 程式所創 建的,但它可以打包和發布為任何語言 的軟體。作為套件管理器可幫助您尋找及安裝套件。如果你需要不同版本 Python 來執行一些專案,你不需要切換到不同的環境管理器,因為 conda 本身就是一 個環境管理器,所以只需幾個命令,您 就可以設置一個完全獨立的環境來執行不同版本的Python,同時繼續在正常環境中直行常用版本的 Python。

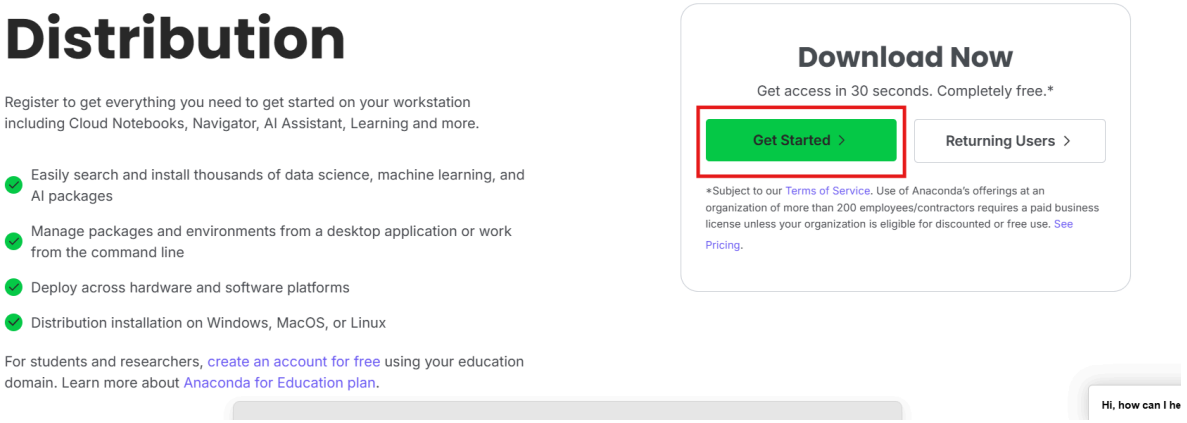
使用conda時,你可以進行建立 (create)、輸出(export)、列表(list)、移除(remove)和更新(update)環境於不同 Python 版本及套件 (Packages),同時也 可以分享你的虛擬環境。接下來筆者將 針對conda 如何建立及管理虛擬環境用 下面6個步驟來說明。



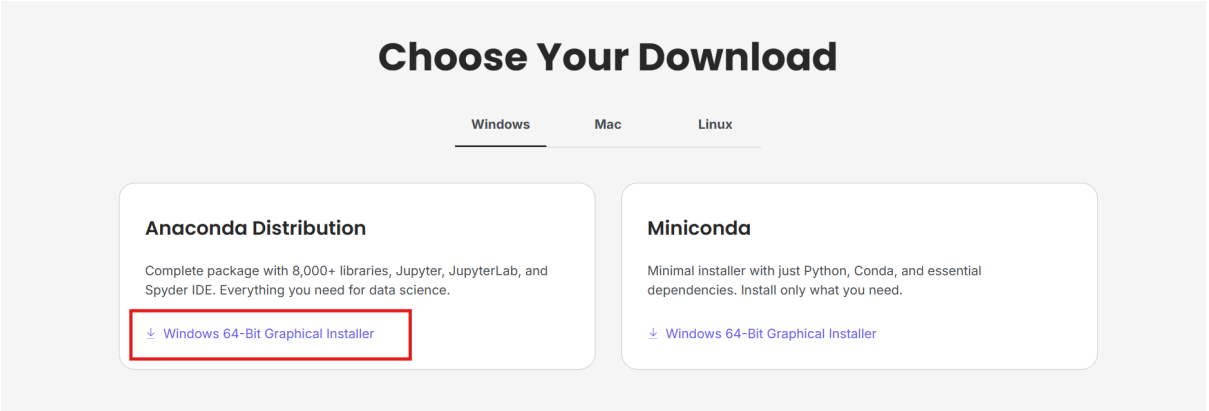
安裝及更新

第一階段：下載 (先把安裝檔抓下來)

1. 點這個網址：[Anaconda 官網點擊Get Started]



2. 看到網頁中間有一個藍色的「**Download**」按鈕，按下去。

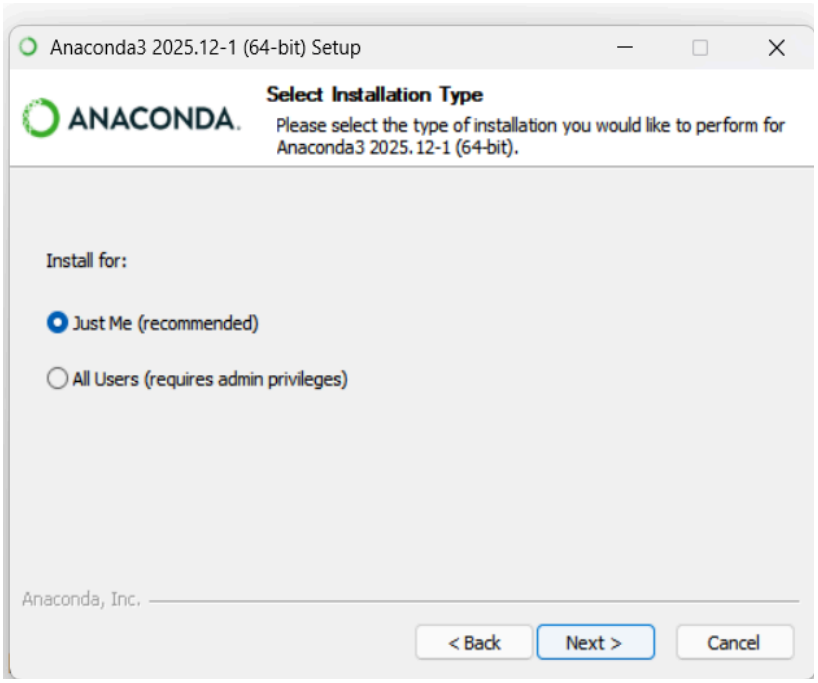


3. 電腦會開始下載一個檔案(大概1.1GB左右，要等一下)，下載完後點兩下打開它。

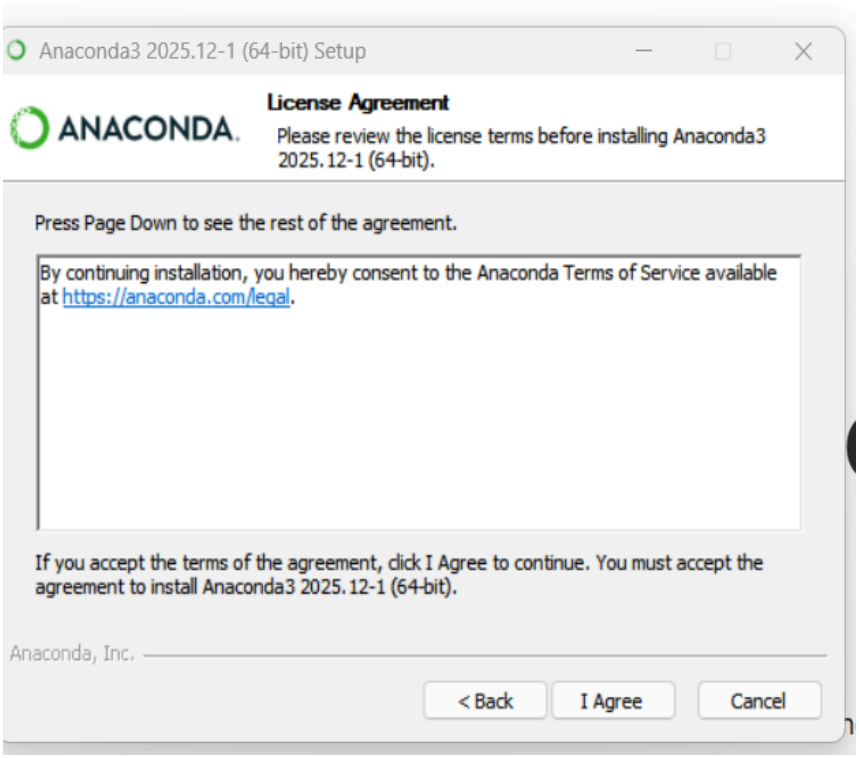
第二階段：安裝 (一直按 Next，只有這一步要停下來)

打開安裝檔後，請依序執行以下動作(如果英文看不懂沒關係，跟著位置點就好)：

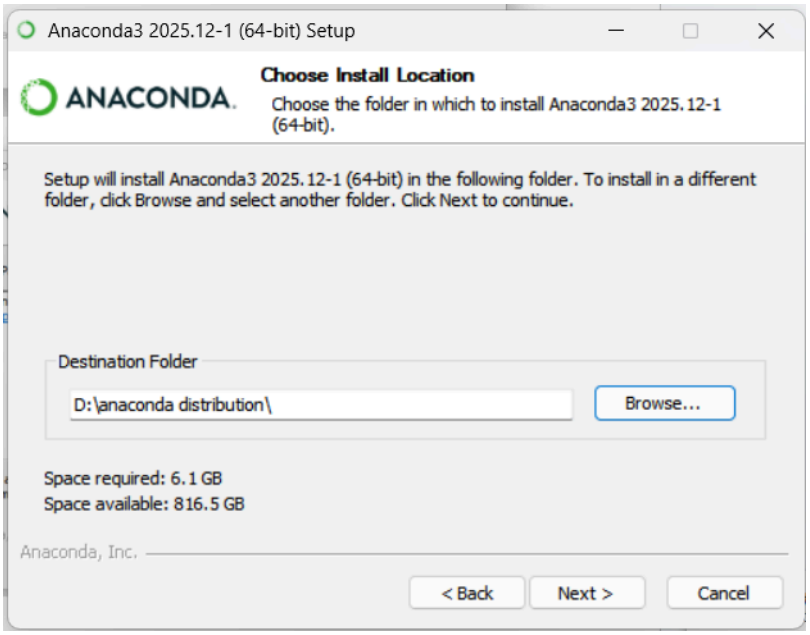
1. 歡迎畫面:點 [Next](下一步)。



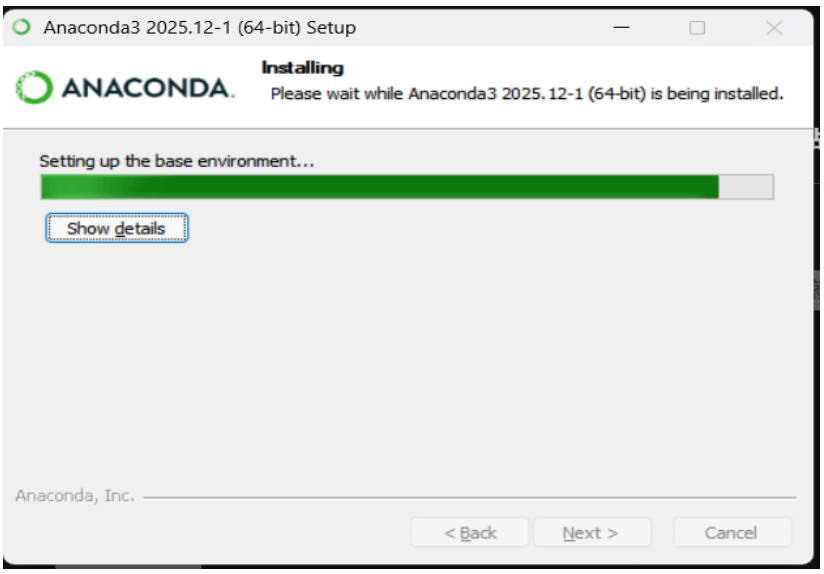
2. 同意條款:點 [I Agree](我同意)。



3. 選誰要用:
    - 直接點 **[Next]**。
4. 要裝在哪:
    - 不用改**(**看自己硬碟儲存量**)**, 直接點 **[Next]**。



5. 關鍵步驟 **(**這裡最重要！**)**:
    - 畫面會有四個的勾選框。
    - 「不要勾」(Add Anaconda3 to my PATH...) : 勾了可能會讓電腦其他程式怪怪的, 千萬別勾。
    - 「保持勾選」(Register Anaconda3...) : 預設就是勾好的, 不要動它。
    - 確認好後, 點 **[Install]** (開始安裝)。
6. 等待安裝:
    - 看著綠色條跑完, 這可能需要幾分鐘, 去喝口水。
    - 跑完後點 **[Next]**。



7. 廣告畫面:
    - 它會問你要不要用PyCharm之類的 不用理它, 點 **[Next]**。
8. 完成:
    - 點 **[Finish]**(完成)。

第三階段:檢查 (確定真的裝好了)

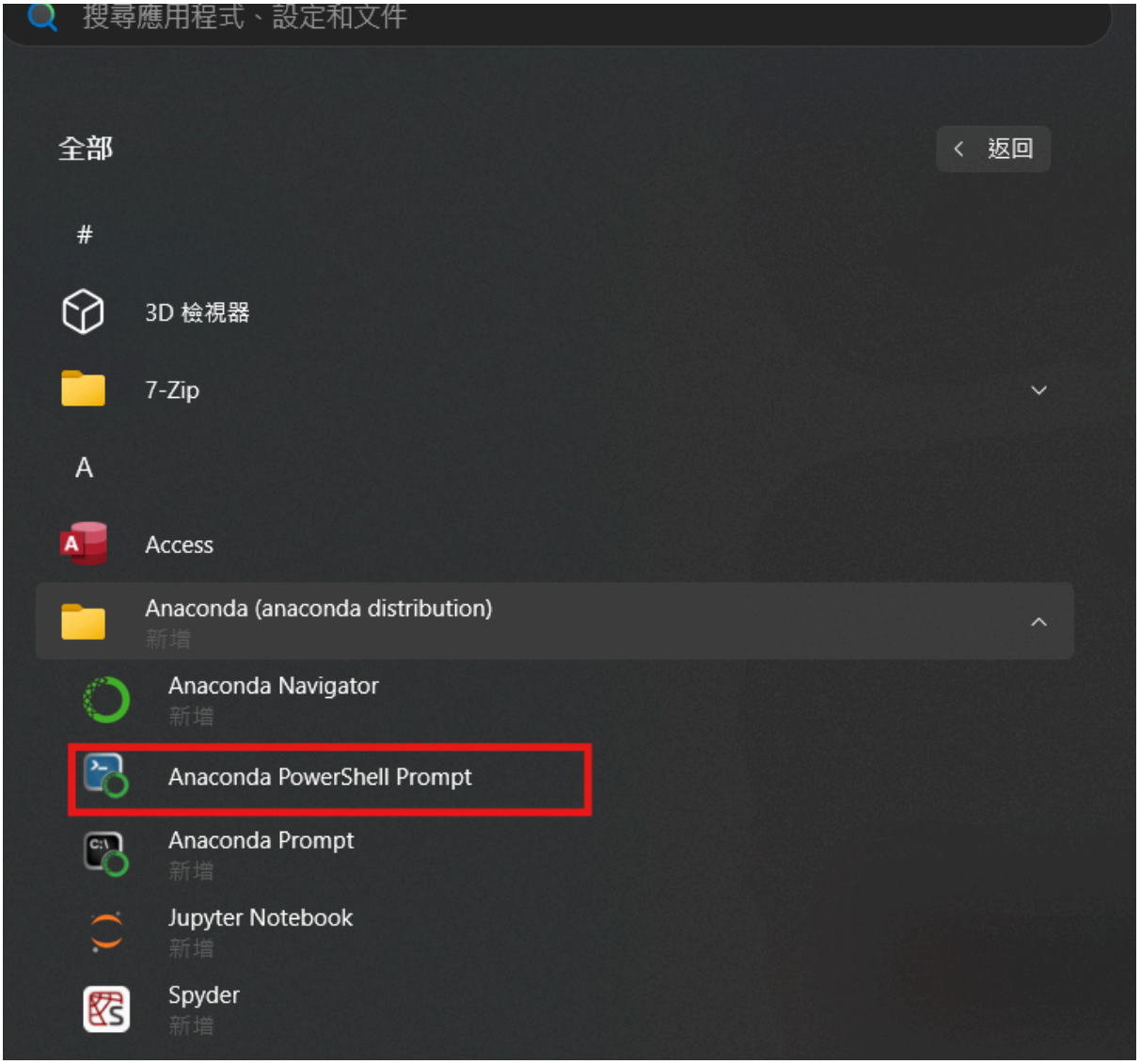
1. 點電腦左下角的 Windows「開始」按鈕。
2. 在清單裡找 A 開頭的, 找到一個資料夾叫「Anaconda 3(64-bit)」。
3. 點擊裡面的綠色圓圈圖示「Anaconda Navigator」。
4. 如果有一個綠綠的視窗跳出來(可能會有點慢, 要等它轉一下), 上面有很多按鈕(像 Jupyter Notebook, Spyder), 那就代表大功告成

針對 **Python 3.11** 的全新懶人包。

請把大腦關機，只要負責「複製、貼上、按 **Enter**」就好

第 1 步：叫出黑視窗(終端機)

- **Windows** 搜尋：打字輸入 `Anaconda Prompt`。
- 看到紅色的圖示：點下去打開它。



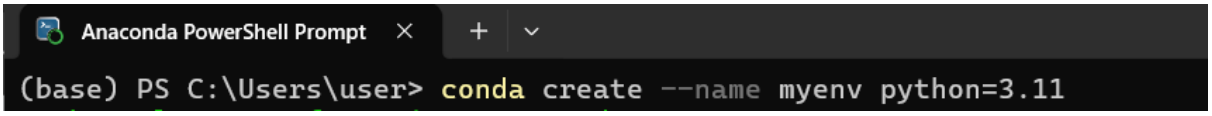
第 2 步:貼上這行指令 (建立環境)

請直接複製下面這行, 貼進黑視窗, 然後按 **Enter**:

conda create --name py311 python=3.11

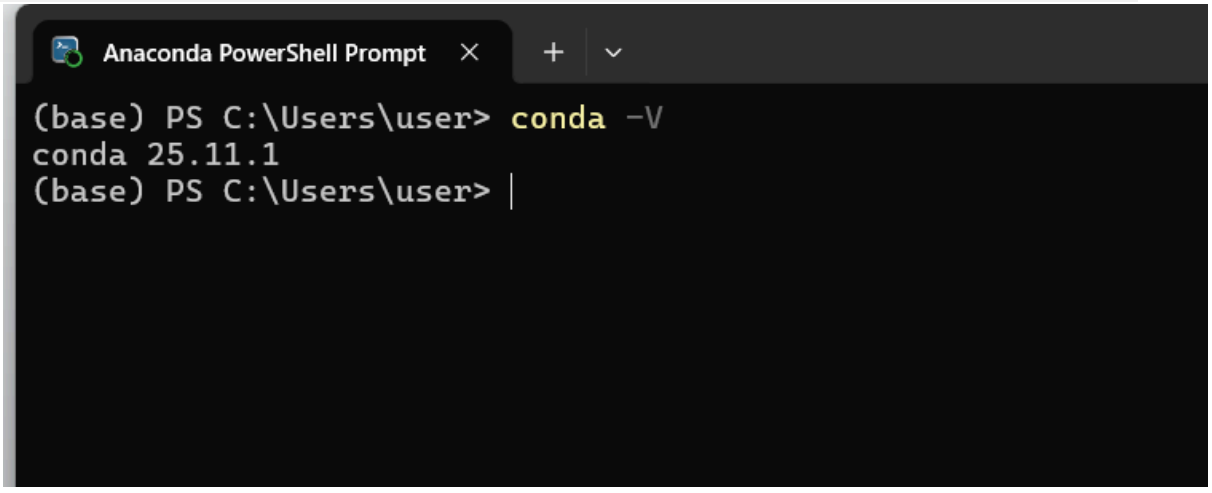
(這行指令的意思是:幫我蓋一個叫 py3.11 的房間, 裡面裝 Python 3.11)

點開之後會像是紅色框起來的地方 括號寫(base), 代表的是本機環境。



此時可以輸入下列命令來檢查目前版本。

```
conda -V
```



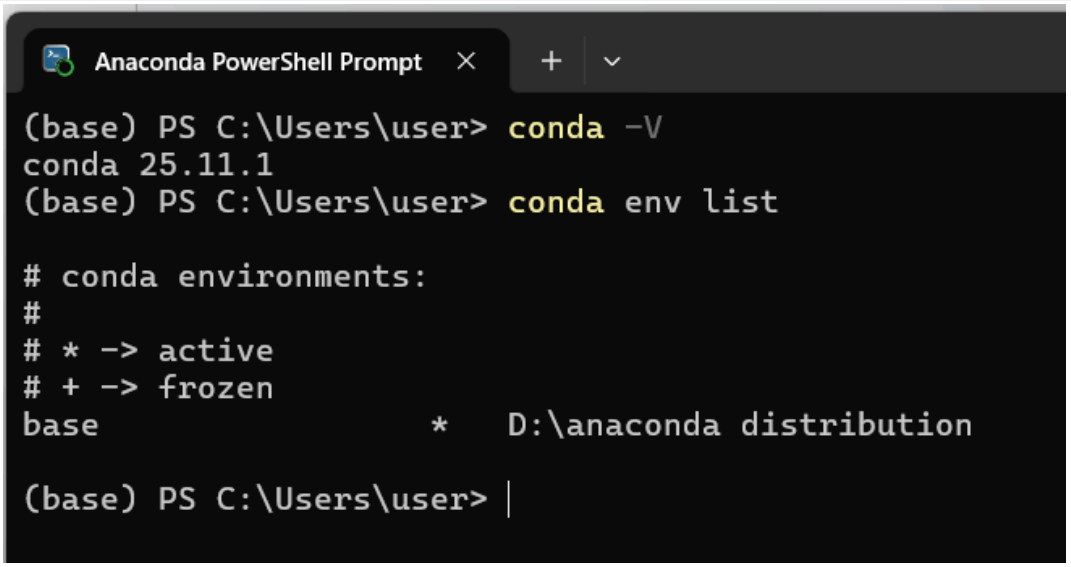
若想要進行更新, 可以輸入下列命令

```
conda update conda
```

Step 1:建立虛擬環境

你可以輸入下面命令看目前系統已經安裝幾個虛擬環境。

```
conda env list
```



假設我們要建立一個叫做myenv的虛擬環境, 並且是安裝python 3.11的版本, 那我們可以鍵入下面的命令。

```
conda create --name myenv python=3.11
```

這邊會提示預設要安裝的庫有哪些, 輸入y按下Enter同意

```
## Package Plan ##

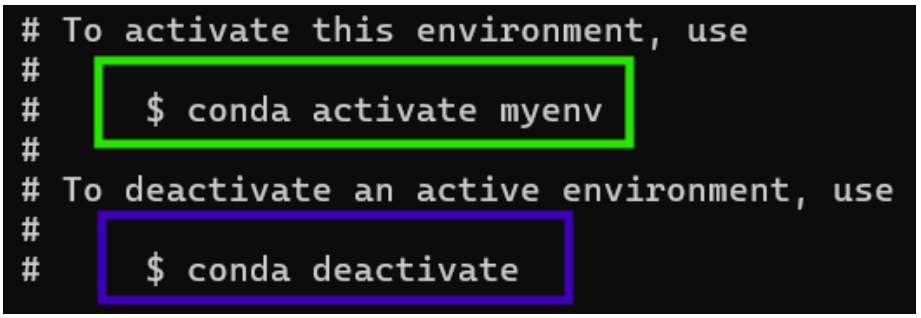
  environment location: D:\anaconda distribution\envs\myenv

  added / updated specs:
    - python=3.11

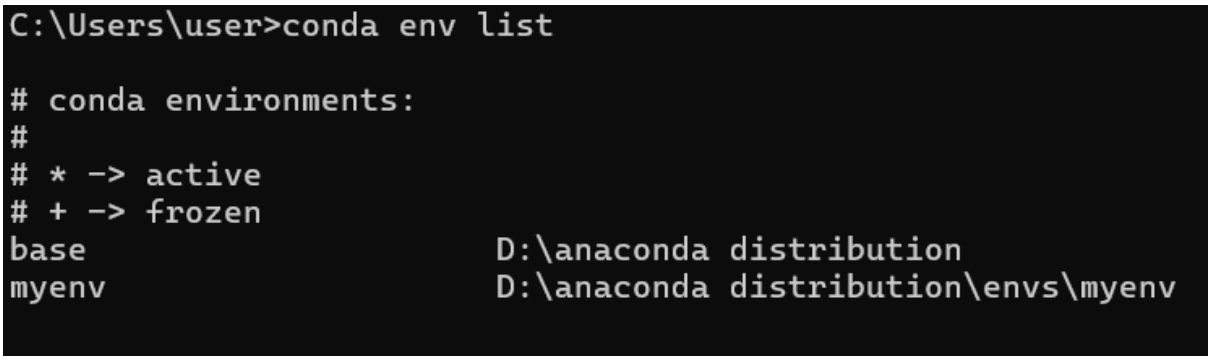

The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    bzip2-1.0.8                |       h2bbff1b_6          90 KB
    expat-2.7.3                |       h885b0b7_4          31 KB
    libexpat-2.7.3             |       h885b0b7_4         119 KB
    libffi-3.4.4               |       hd77b12b_1         122 KB
    libzlib-1.3.1              |       h02ab6af_0          64 KB
    openssl-3.0.18             |       h543e019_0         6.8 MB
    pip-25.3                   |     pyhc872135_0         1.1 MB
    python-3.11.14             |       h981015d_0        17.7 MB
    setuptools-80.9.0          |  py311haa95532_0         1.9 MB
    sqlite-3.51.0              |       hda9a48d_0         917 KB
    tk-8.6.15                  |       hf199647_0         3.5 MB
    tzdata-2025b               |       h04d1e81_0         116 KB
    ucrt-10.0.22621.0          |       haa95532_0         620 KB
    vc-14.3                    |      h2df5915_10          19 KB
    vc14_runtime-14.44.35208   |      h4927774_10         825 KB
    vs2015_runtime-14.44.35208 |      ha6b5a95_10          19 KB
    wheel-0.45.1               |  py311haa95532_0         182 KB
    xz-5.6.4                   |       h4754444_1         280 KB
    zlib-1.3.1                 |       h02ab6af_0         113 KB
    ------------------------------------------------------------
                                           Total:        34.5 MB
```

安裝完後會出現如下圖所示
提醒啟動(綠)與關閉(藍)虛擬環境的用法。



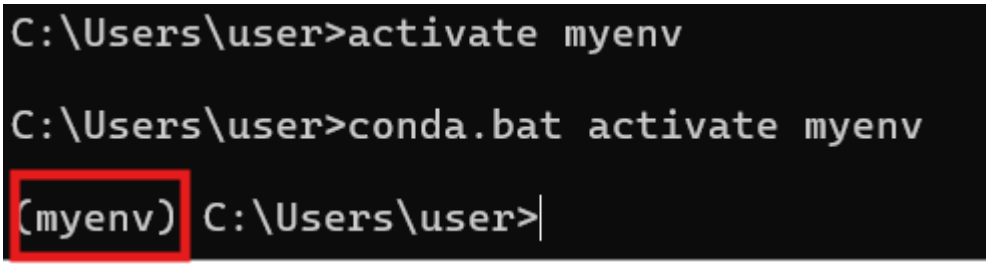
我們試著再下命令conda env list, 列出目前虛擬環境狀況, 將會看到多了一個剛建立的虛擬環境myenv。



Step 2:啟動虛擬環境

啟動一個新的虛擬環境可以利用下面命令。

```
activate myenv
```



這時候cmd模式下前面會有一個(myenv), 表示你目前是處於此虛擬環境中, 這時候你就可以在此虛擬環境中, 開始安裝你所需要的各種package。

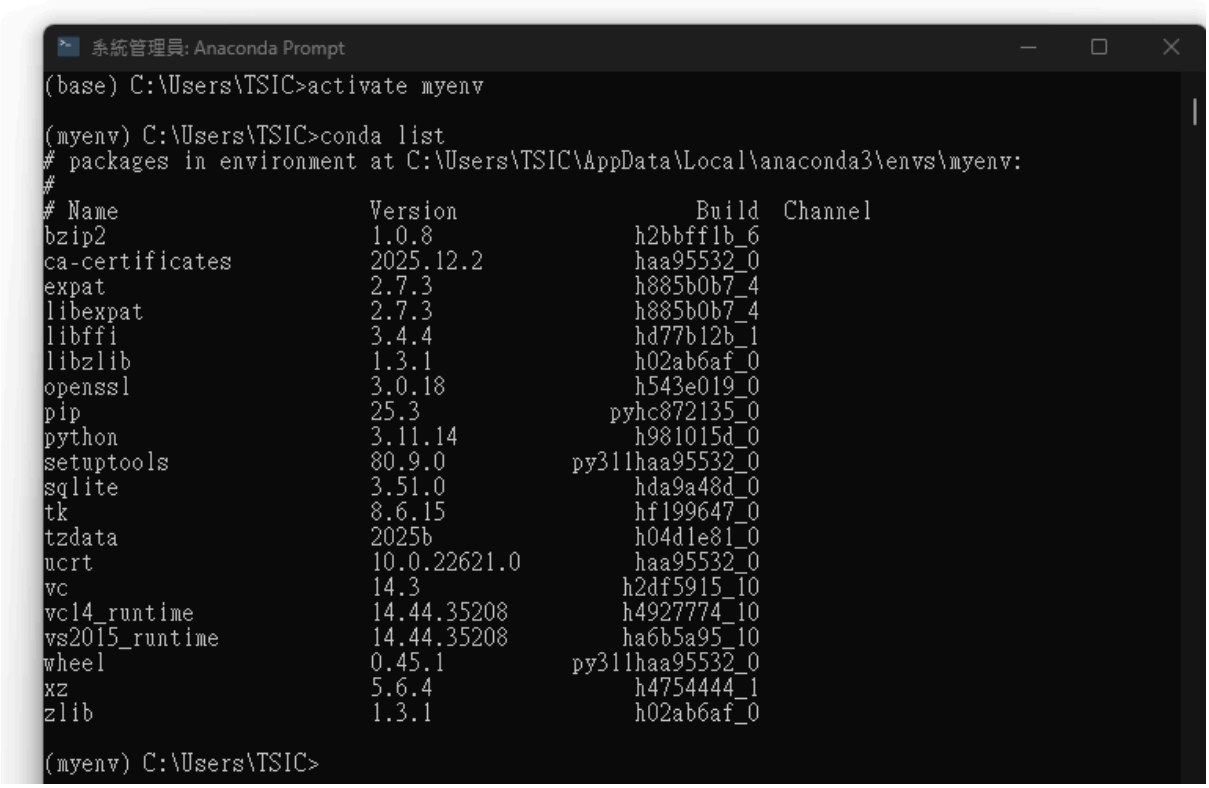
如果你是LINUX或macOS, 那你所需輸入啟動的方式將會是

```
source activate myenv
```

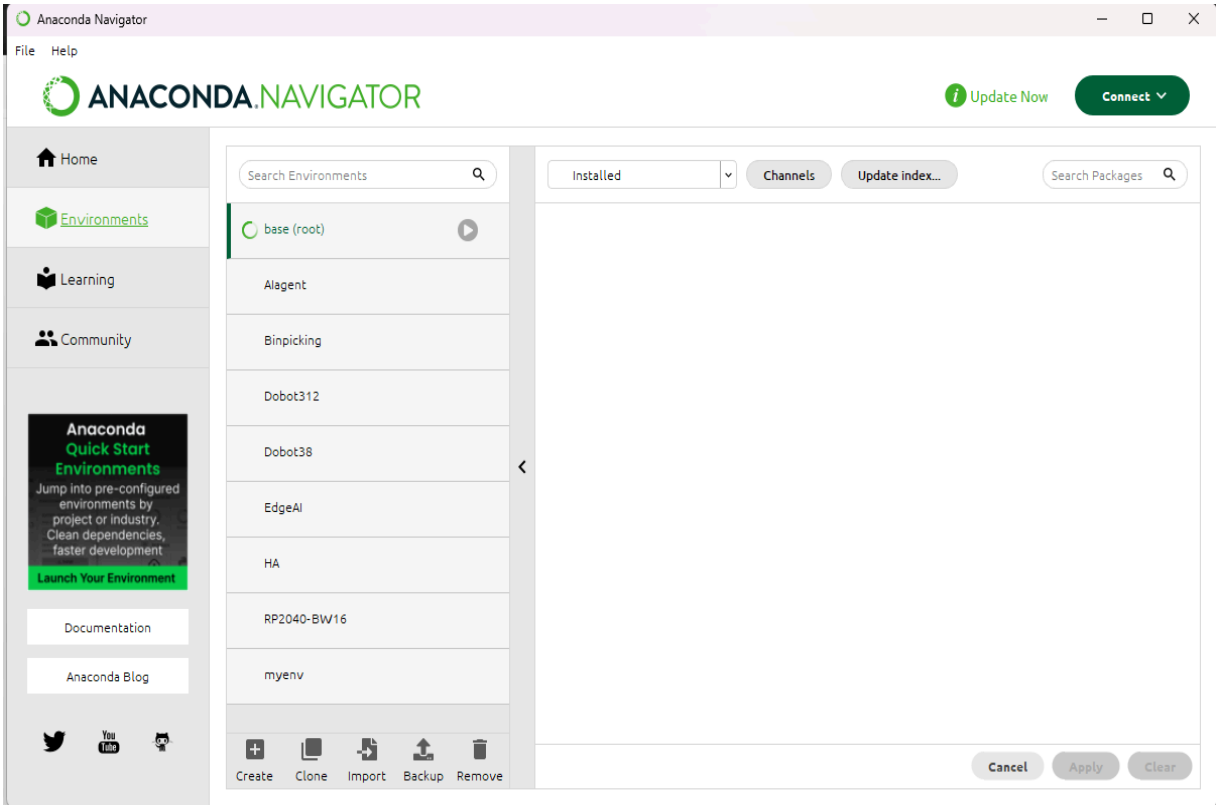
你可以利用下面命令來查看, 目前此虛擬環境中已經先安裝了那些東西。

```
conda list
```

你也可以進入Anaconda Navigator看到你目前環境及所對應的安裝項目，與Anaconda Prompt底下所看到

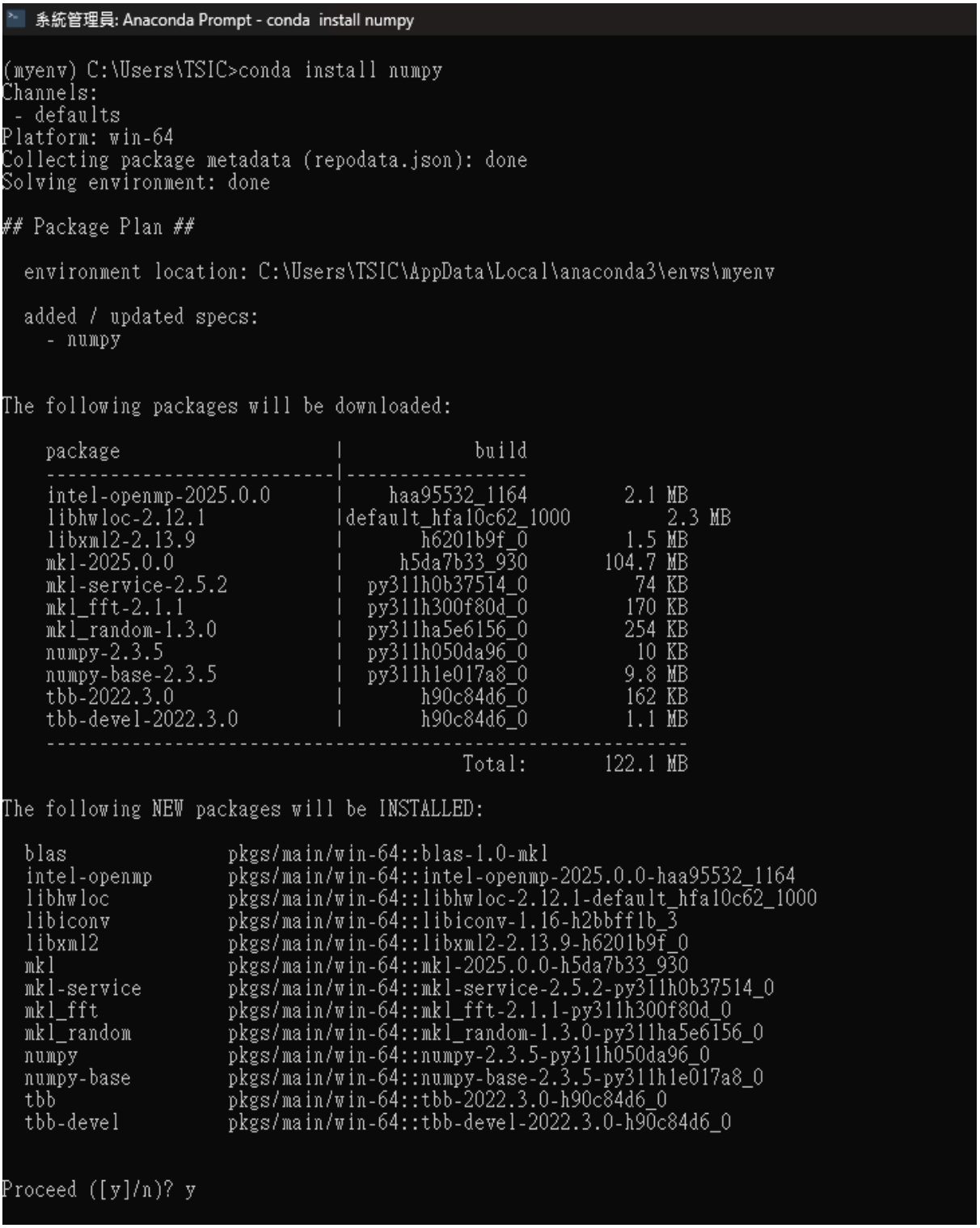


的是一樣的。



如果要在此虛擬環境下安裝所需套件, 例如numpy那只需要輸入下令命令即可。

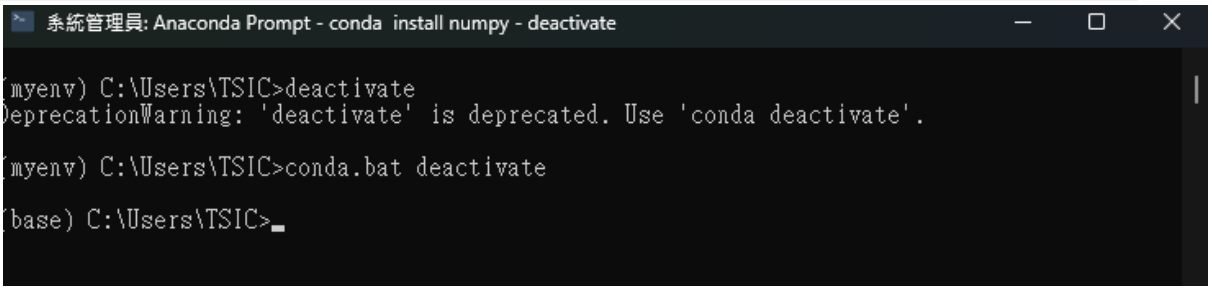
```
conda install numpy
```



Step 3:離開虛擬環境

若要關閉虛擬環境, 在windows中可使用下列命令

```
deactivate
```



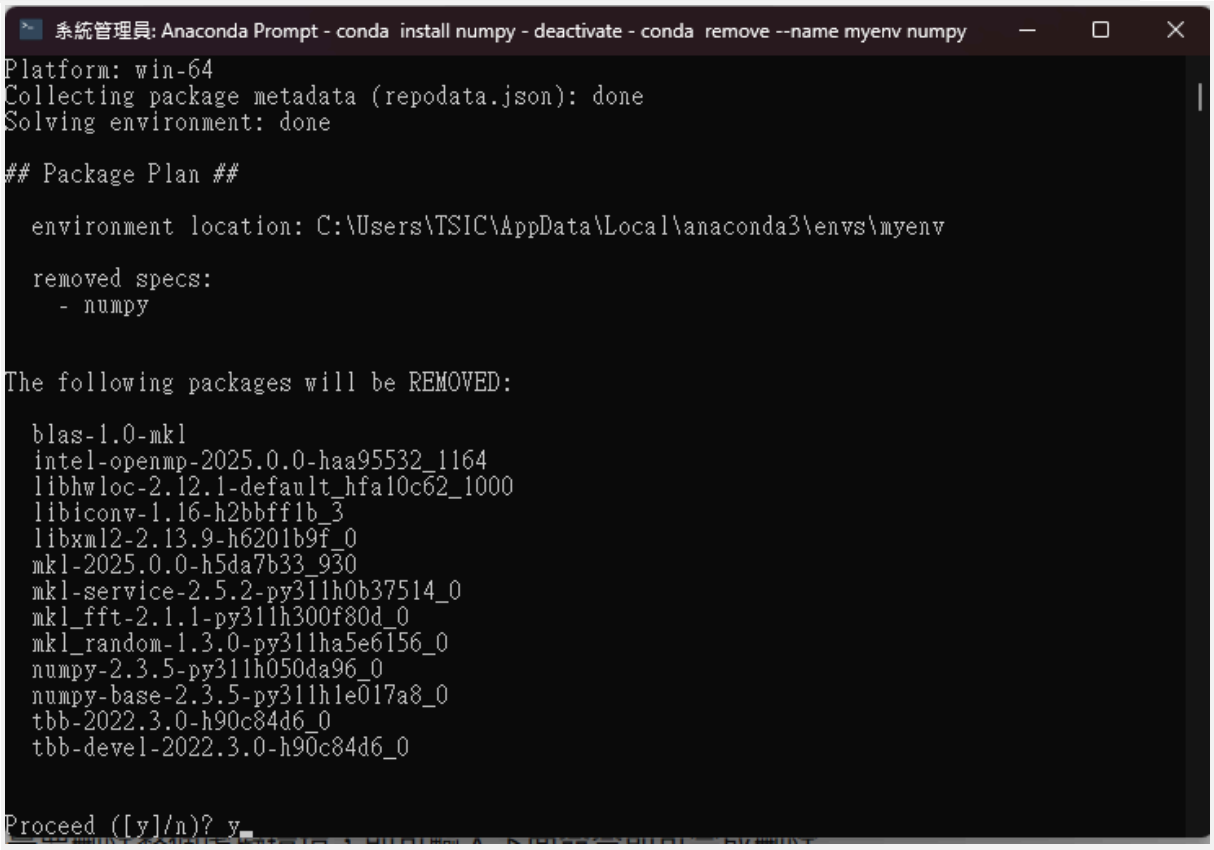
而在macOS或LINUX則可以使用

```
source deactivate
```

Step 4:刪除虛擬環境或package

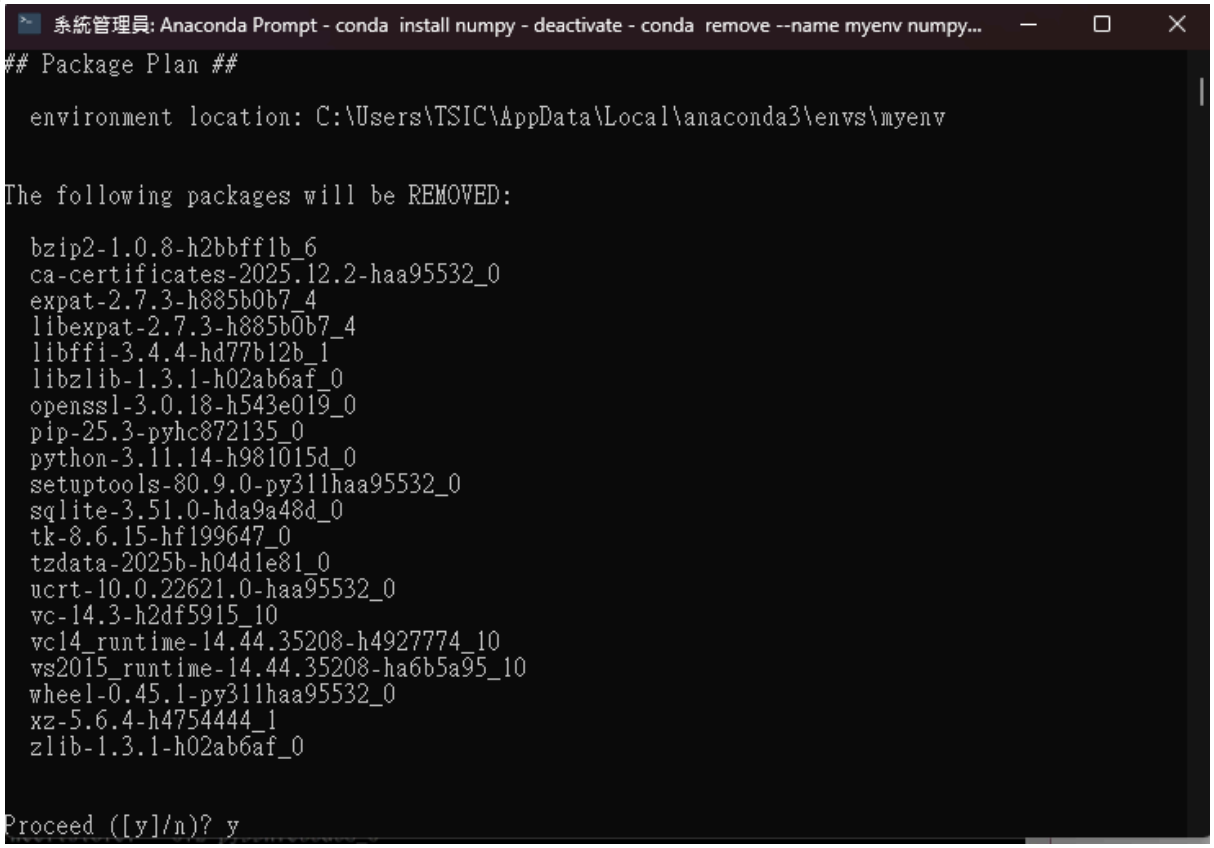
若要刪除虛擬環境中某個package(例如在剛剛建立的虛擬環境myenv中的numpy), 那可以輸入下面命令

```
conda remove --name myenv numpy
```



如果是要刪除整個虛擬環境, 則可輸入下面命令即可完成刪除

```
conda env remove --name myenv
```

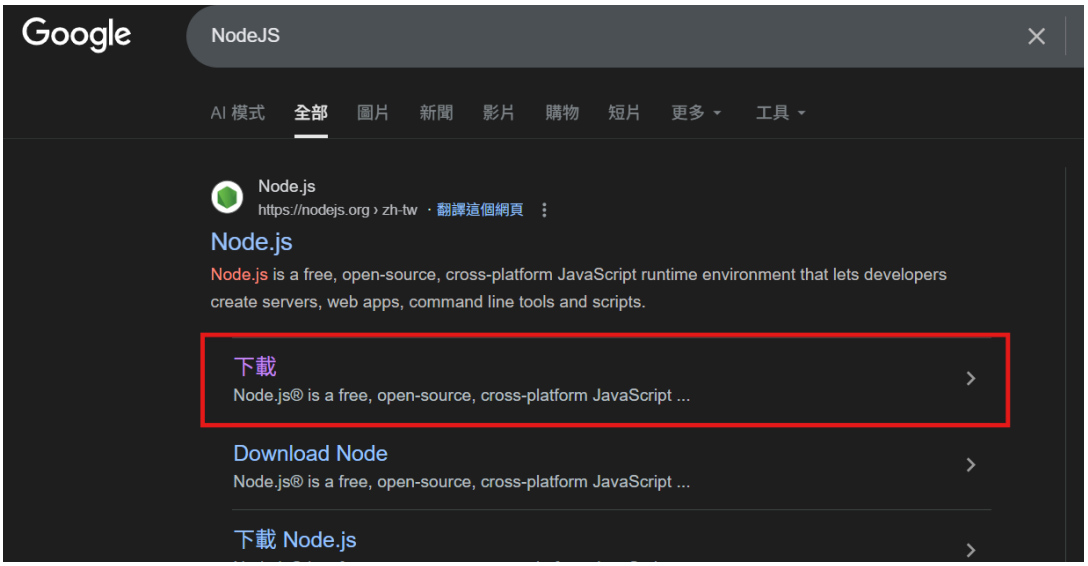


結語

為每一個不同需求的專案建立一個獨立適合的虛擬環境是一個很好的習慣, 因為它並不會去影響其他的系統配置而產生不預期問題。若在配置上產生了問題只要輕易的移除某個package或是虛擬環境再重新建構它即可。

NodeJS安裝

現在許多AI工具在CLI上執行，比較多都是透過npm指令安裝，同時也方便，因此推薦安裝NodeJS環境方便處理，您可以到Google搜尋NodeJS第一個結果就是安裝的連結。



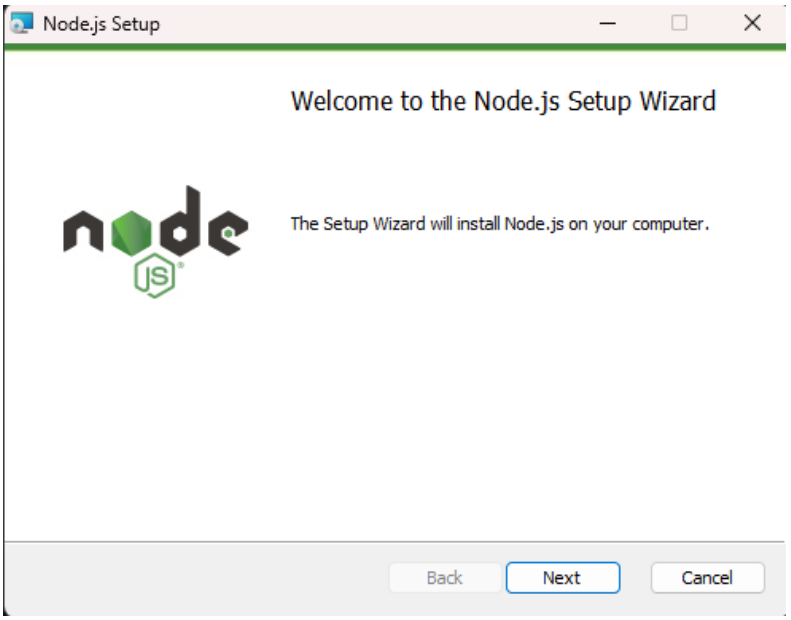
下載

點選Download進去後就會進入到下圖畫面，依照您的作業系統選擇，Windows安裝則如下畫面點選msi方式做安裝，之後瀏覽器會下載一個.msi的檔案，點開進行安裝。:



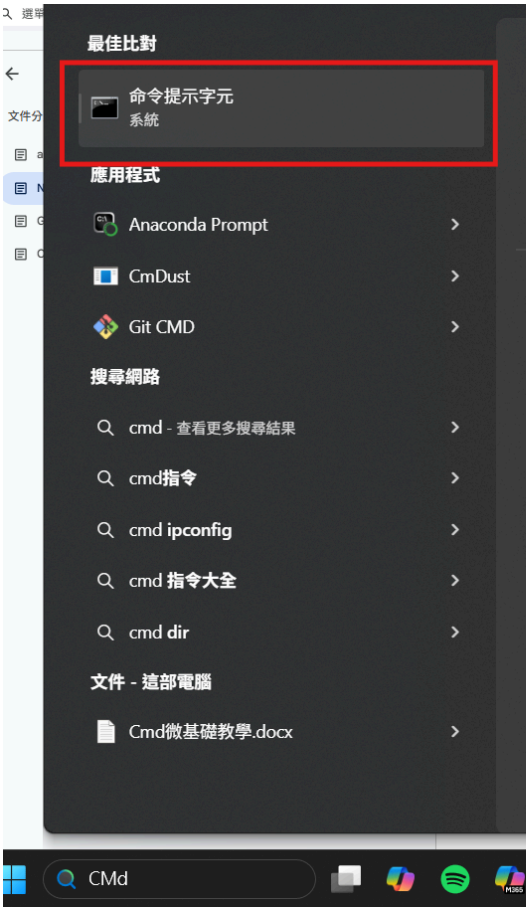
安裝

安裝的步驟基本上非常簡易，
安裝時就只需要不斷的同意(NEXT)下一步按到底即可。



安裝成功驗證

結束之後**Windows** 搜尋:打字輸入CMD



打開這CMD(終端機)輸入

```
npm
```

如果出現下圖資訊代表成功安裝。

```
Microsoft Windows [版本 10.0.26100.7462]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\TSIC>npm
npm <command>

Usage:

npm install        install all the dependencies in your project
npm install <foo>  add the <foo> dependency to your project
npm test           run this project's tests
npm run <foo>      run the script named <foo>
npm <command> -h   quick help on <command>
npm -l             display usage info for all commands
npm help <term>    search for help on <term> (in a browser)
npm help npm       more involved overview (in a browser)

All commands:

    access, adduser, audit, bugs, cache, ci, completion,
    config, dedupe, deprecate, diff, dist-tag, docs, doctor,
    edit, exec, explain, explore, find-dupes, fund, get, help,
    help-search, hook, init, install, install-ci-test,
    install-test, link, ll, login, logout, ls, org, outdated,
    owner, pack, ping, pkg, prefix, profile, prune, publish,
    query, rebuild, repo, restart, root, run-script, sbom,
    search, set, shrinkwrap, star, stars, start, stop, team,
    test, token, uninstall, unpublish, unstar, update, version,
    view, whoami

Specify configs in the ini-formatted file:
    C:\Users\TSIC\.npmrc
or on the command line via: npm <command> --key=value

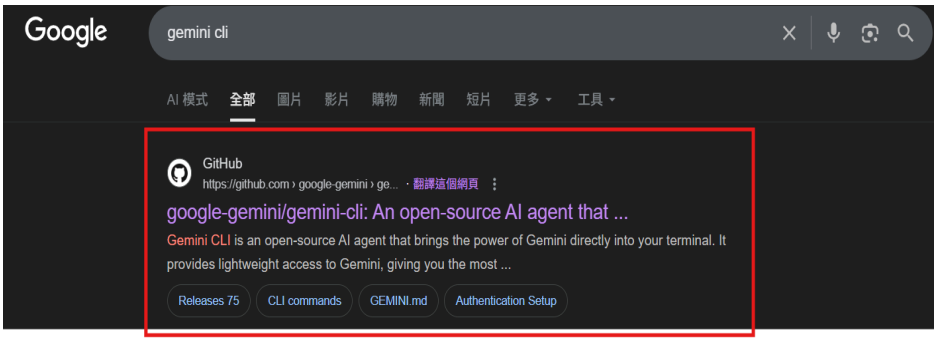
More configuration info: npm help config
Configuration fields: npm help 7 config

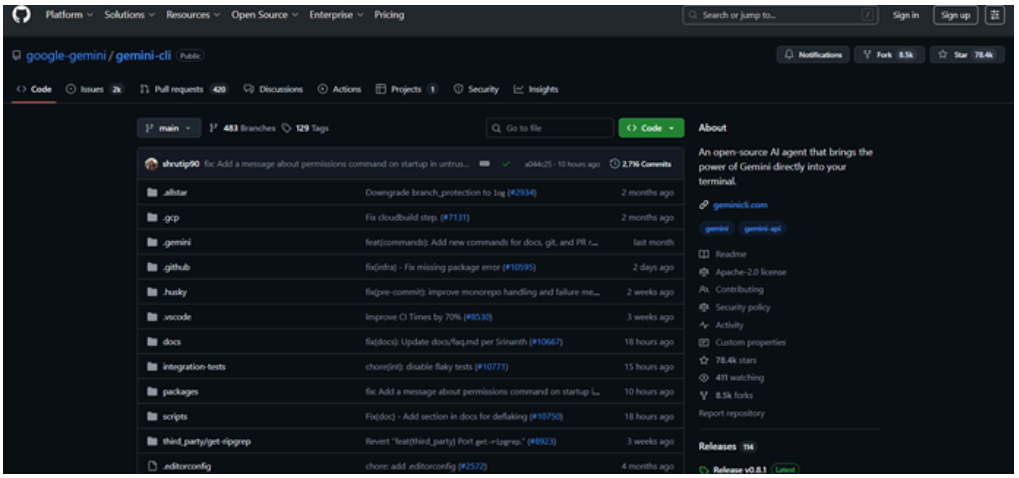
npm@10.9.2 C:\Program Files\nodejs\node_modules\npm

C:\Users\TSIC>
```

Gemini cli

網路搜尋gemini cli
點擊他的github



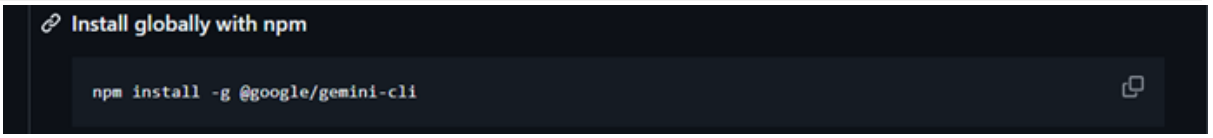


用滑鼠中間滾輪往下滑動到

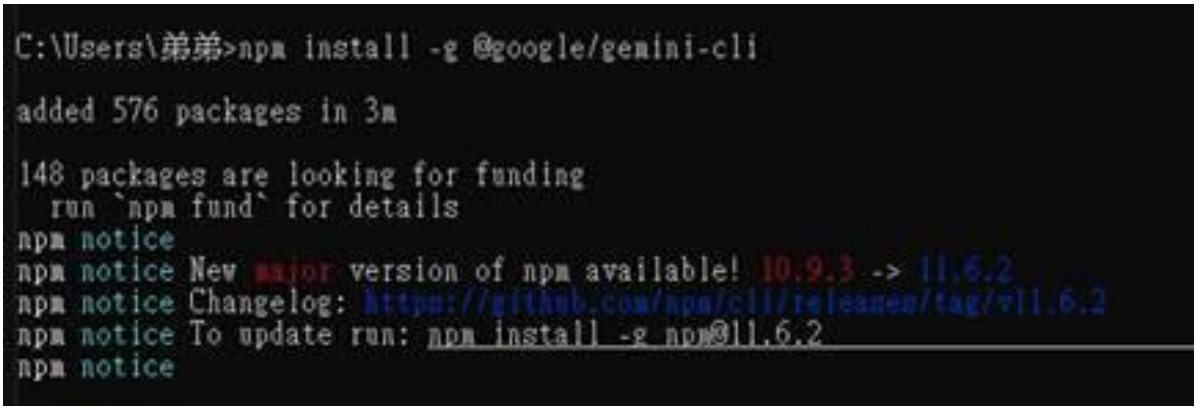


點擊

```
npm install -g @google/gemini-cli
```



複製 ; 回到cmd(終端機)貼上



之後再打gemini就會跳出



後面就可以搜尋(你對哪個github專案有興趣你就可以去clone下來，利用github desktop 建立一個檔案在你找的到文檔檔名語路徑都要英文)

Claude Code 概述

了解 Claude Code，Anthropic 的代理編碼工具，它存在於您的終端中，幫助您比以往更快地將想法轉化為代碼。

30 秒內開始使用
前置條件：
一個 Claude.ai（推薦）或 Claude Console 帳戶

步驟 1：安裝 Claude Code

```
npm install -g @anthropic-ai/claude-code
```

需要 Node.js 18+
或是Windows本地安裝
Windows PowerShell：

```
irm https://claude.ai/install.ps1 | iex
```

Windows 命令提示字元：

```
curl -fsSL https://claude.ai/install.cmd -o
install.cmd && install.cmd && del install.cmd
```

macOS、Linux、WSL：

```
curl -fsSL https://claude.ai/install.sh | bash
```

開始使用 Claude Code：

```
cd your-project
claude
```

教程：Visual Studio Code 入門

在本教程中，你將瞭解 Visual Studio Code 的關鍵功能，以幫助你快速開始編碼。你將瞭解使用者介面的不同元件以及如何自定義它以滿足你的喜好。然後，你將編寫一些程式碼並使用內建的程式碼編輯功能，如 IntelliSense 和程式碼操作，你還將學習如何執行和除錯你的程式碼。透過安裝語言擴充套件，你可以為不同的程式語言新增支援。

提示

如果你更喜歡透過影片學習，可以觀看 入門影片，該影片涵蓋了本教程的相同步驟。

先決條件

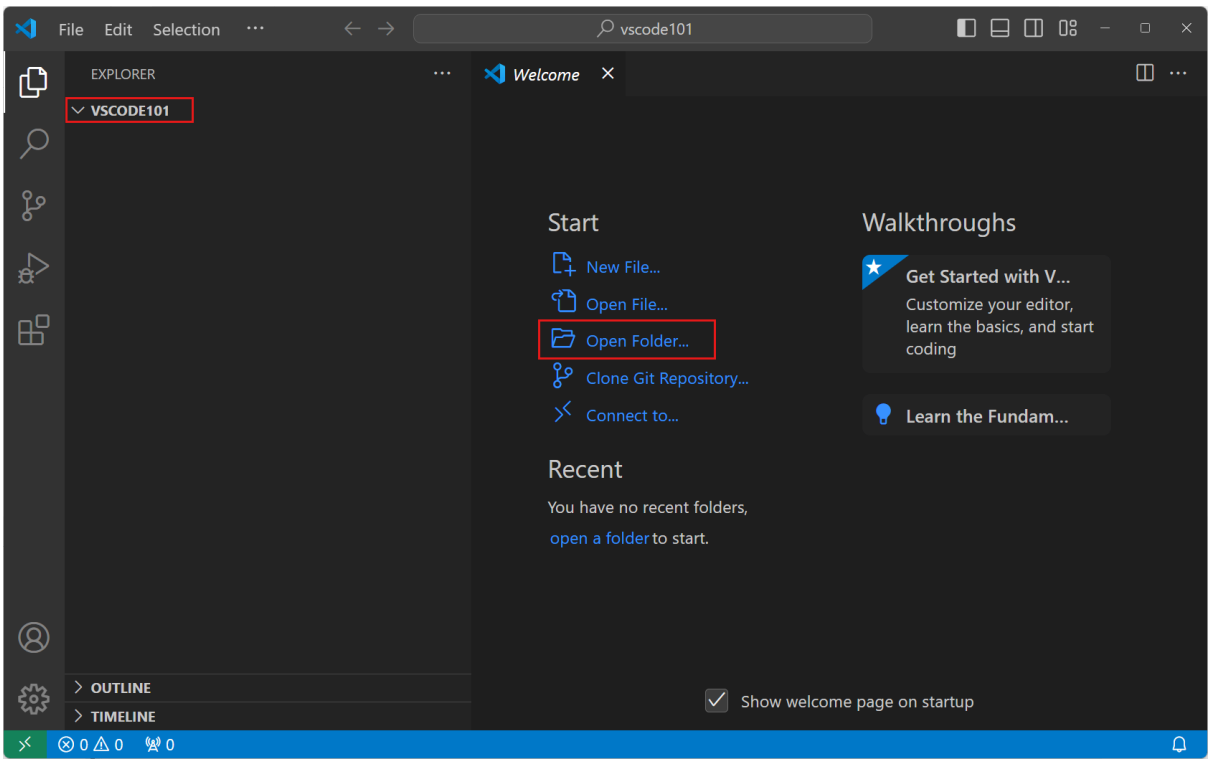
- 在你的計算機上下載並安裝 Visual Studio Code

在 VS Code 中開啟資料夾

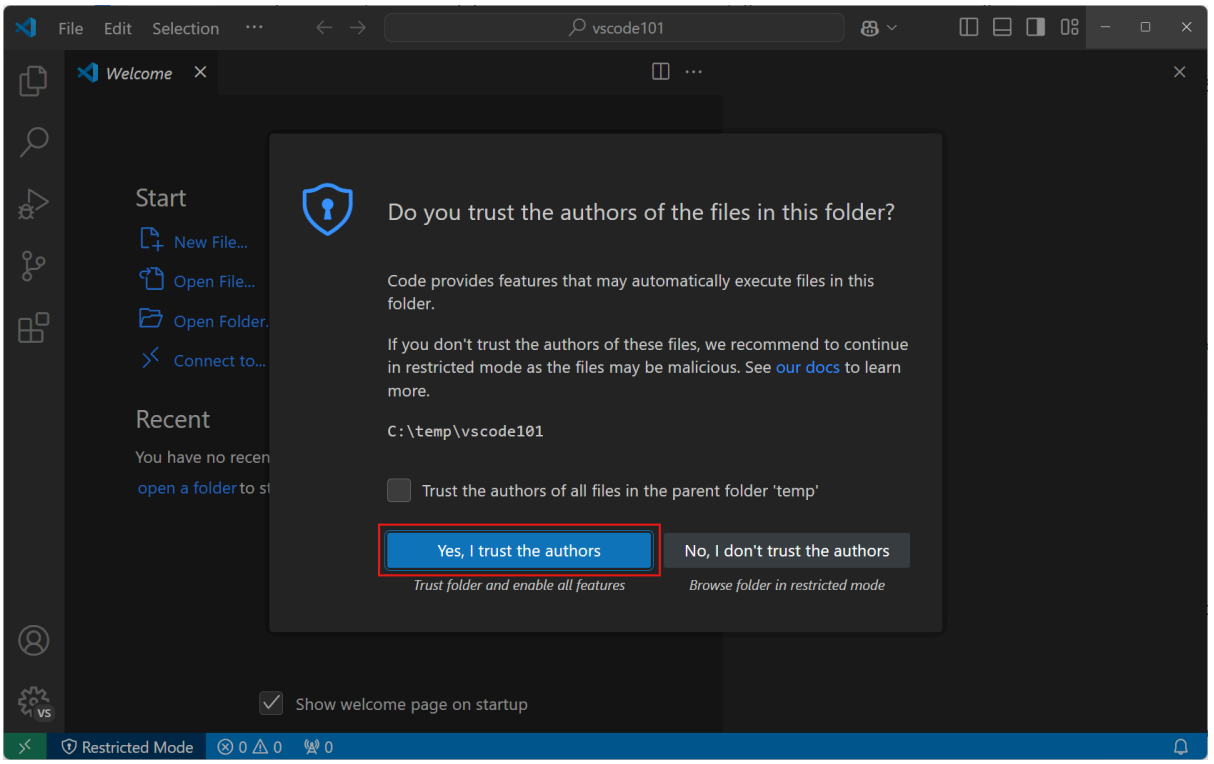
你可以使用 VS Code 處理單個檔案進行快速編輯，也可以開啟一個資料夾，也稱為 _工作區_。

讓我們首先建立一個資料夾並在 VS Code 中開啟它。你將在整個教程中使用此資料夾。

1.開啟 Visual Studio Code。
首次開啟 VS Code 時，你應該會看到 **歡迎** 頁面，其中包含不同的入門操作。
2.從選單中選擇 **檔案** > **開啟資料夾...** 來開啟資料夾。



3.選擇 **新建資料夾**，建立一個名為 vscode101 的新資料夾，然後選擇 **選擇資料夾**（macOS 上為 **開啟**）。
你建立的資料夾是你工作區的根目錄。
4.在 **工作區信任** 對話方塊中，選擇 **是，我信任作者** 以啟用工作區中的所有功能。



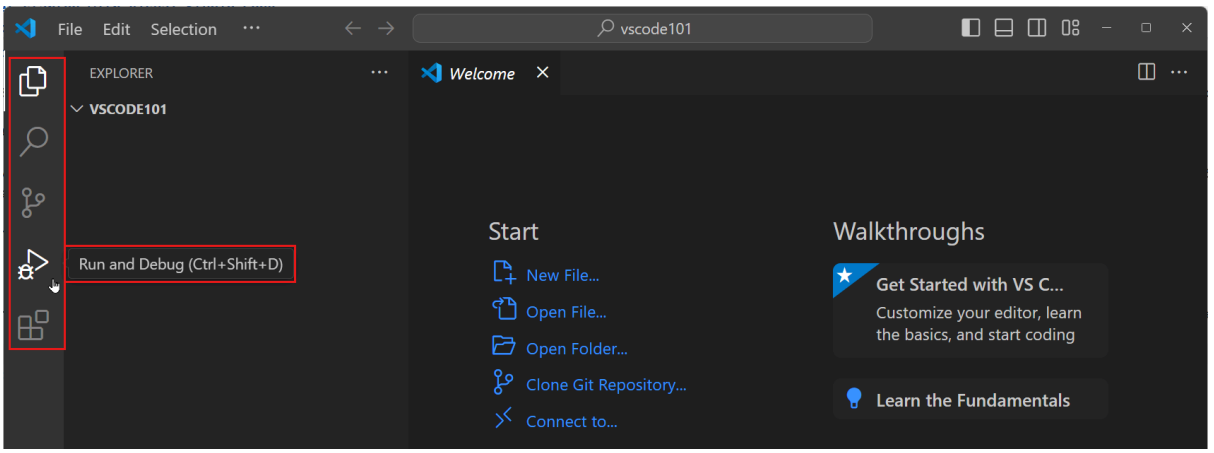
由於你在計算機上建立了該資料夾，因此你可以信任該資料夾中的程式碼。
重要
工作區信任允許你決定專案資料夾中的程式碼是否可以由 VS Code 執行。當你從 Internet 下載程式碼時，應首先對其進行審查，以確保其執行安全。有關 工作區信任 的更多資訊。
5.現在，你應該在左側看到 **資源管理器** 檢視，顯示資料夾的名稱。
你將使用資源管理器檢視來檢視和管理工作區中的檔案和資料夾。

探索使用者介面

現在你已經在 VS Code 中打開了一個資料夾，讓我們快速瀏覽一下使用者介面。

透過活動欄切換檢視

1.使用活動欄在不同檢視之間切換。



提示

將滑鼠懸停在活動欄上，即可看到每個檢視的名稱和相應的鍵盤快捷鍵。你可以透過再次選擇檢視或按鍵盤快捷鍵來開啟和關閉檢視。

2 當你選擇活動欄中的檢視時，**主側邊欄** 會開啟以顯示檢視特定的資訊。

例如，執行和除錯檢視允許你配置和啟動除錯會話。

透過編輯器檢視和編輯檔案

1.在活動欄中選擇 **資源管理器** 檢視，然後選擇 **新建檔案...** 按鈕在你的工作區中建立新檔案。
2.輸入檔名 `index.html`，然後按 `Enter`。

檔案將新增到你的工作區，並且編輯器將在視窗的主區域中開啟。

如何將介面的語言從中文切換成英文？

藉由這個例子，來說明如何透過「命令選擇區」(Commamd Palette) 來呼叫VS Code的指令 (command)或擴充套件的指令來執行工作。

- 首先按下快捷鍵`Ctrl+Shift+P`，就會跳出「命令選擇區」，它有`>`符號的輸入欄和指令選單
- 輸入欄敲入 `display` 來過濾指令，選擇指令選單的「Configure Display Language」

  
- 然後選取要切換的語言，如 “English”

對於新手，可以打開「命令選擇區」來看看VS Code或擴充套件有那些指令可以使用。

怎麼有些程式碼躲起來？

在編輯很多行數的檔案時，你可以把區塊內的程式碼摺疊 (code folding) 來增加程式碼的可讀性。

- 當滑鼠移到程式碼的左側行號附近，在行號的右邊，你會看到帶有小三角形 (向下) 的按鈕，它表示可以摺疊這個程式碼區域，點擊它可以進行摺疊。

  當區塊內的程式碼已經被摺疊，你可以點擊小三角形 (向右) 的按鈕來取消摺疊。
- 那麼如何一次取消所有的摺疊呢？一個一個點擊小三角形 (向右)很累喔。

很簡單，只要按下快捷鍵 `Ctrl + K, Ctrl + J`

如果你想要把整個檔案內可摺疊的程式碼都摺疊，就只要按下快捷鍵 `Ctrl + K, Ctrl + 0`

值得一提的是，程式碼摺疊功能是依賴於這個程式語言的支援擴充套件。如果你安裝了支援程式碼摺疊的語言擴充套件，你就會在該程式語言下看到額外的程式碼摺疊選項。

例如，Python 擴充套件提供了針對使用空白縮排的程式碼區塊的摺疊圖示。

而 C語言檔案是根據 `{` 和對應的 `}` 來判斷為程式碼區塊。

對於純文字檔案，當有一段縮排的文字時，VS Code 會將這段縮排的文字視為一個程式碼區塊，並提供程式碼摺疊的圖示。

甚麼是大綱 (OUTLINE) 和時間表 (TIMELINE)？

使用VS Code超過半年多時，我才想去瞭解在VS Code的檔案總管 (Explorer) 底下兩個區塊「大綱 (OUTLINE)」和「時間表 (TIMELINE)」。這是平時不常使用，但有想要導航或管理程式碼時，就覺得很方便的功能呢。從我的觀點，可以簡單認為這是從空間和時間兩個緯度來導航和管理程式碼。這裡只概要的說明主要功能：

- 「大綱」提供當下編輯檔案的樹狀視圖，會顯示出程式碼中的主要節點，像是函數、類別、變數等，展示出程式碼的階層結構，或是可以快速地導航到程式碼的不同部分。這對處理大型程式碼檔案特別有用。

  你也可將大綱的樹狀視圖從「主要側邊欄」的檔案管理處，拖曳到「次要側邊欄」內，這裡有比較大的顯示空間。

  如何打開「次要側邊欄」，只要按下快捷鍵 `Ctrl + Alt + B` 就可以了。
- 「時間表」提供本地端版本控制系統，它會記錄每次檔案儲存 (saved)、復原/重做 (redo/undo) 的快照條目，並比較對檔案所做的更改 (diff)。並

可以在需要時恢復到之前的版本。

更多和Git版本控制整合的功能，可以查閱下面傳送門的參考資料。

VS Code 操作的傳送門

- 🌐 Command Palette - VS Code User Interface
- 🌐 Outline view - VS Code User Interface
- 🌐 Timeline view - VS Code March 2020
- 🌐 VS Code Timeline — Your local version control system

如何開啟或建置新專案 (project)？喔，沒有專案！?

VS Code 沒有「專案」相關指令，而是使用「工作區」(workspace) 的方式來管理整個專案。「工作區」通常是指所有檔案的根目錄，工作區設定也儲存在這根目錄底下 .vscode 資料夾的 settings.json 檔案中。

那麼如何開啟或關閉「工作區」呢？就像在 VS Code 中打開或關閉單一資料夾一樣，同時開啟或關閉工作區。

通常你可選擇以下任一指令，來開啟工作區：

- 檔案 > 開啟資料夾
- 檔案 > 開啟最近的檔案 > (先前曾開啟的資料夾)

而選擇以下指令，來關閉工作區

- 檔案 > 關閉資料夾 (Ctrl + K F)
  說明: Ctrl + K F 是先同時按 Ctrl鍵和K鍵，放開後再按 F 鍵

如何管理工作區內的檔案？

- VS Code 左側的檔案總管可以讓你方便地瀏覽、管理和操作工作區的資料夾和檔案。你隨時可以按下快捷鍵 `Ctrl + Shift + E` 切換至檔案總管。
- 創建新檔案：
  - 在檔案總管中，滑鼠右鍵點擊你想要創建檔案的位置 (如在某一資料夾，或空白處)，然後選擇「新增檔案 ...」。然後，為新檔案指定一個名稱，並以適當的檔案擴展名（如 `.c`、`.py` 等）結尾。
    類似的方式，可以選擇「新增資料夾 ...」來創建新資料夾。

    - 另一方式，在檔案總管中，當滑鼠移到「工作區」名稱時，就會出現小圖示，你可以點擊去新增檔案 或 新增資料夾。
    - 另外，可如同專業工程師般，在 VS Code 的終端機內敲入 `code sample.c`，就可以在當下的資料夾內，新增檔案 “sample.c”，並開始編輯，而不需要使用滑鼠。
- 拖放檔案或資料夾：

  就像Windows檔案總管，你可以直接在VS Code 檔案總管中，拖放檔案或資料夾，以移動或複製它們。
- 在VS Code 檔案總管內刪除一個檔案後，怎樣恢復救回來？

  如果沒有使用 Git 版本控制功能，可以嘗試以下三種方式：
    - 先點選在VS Code 檔案總管後，選擇指令 `編輯 > 復原` 或是 按下快捷鍵 `Ctrl + Z`
    - 資源回收筒：刪除檔案後，又做很多操作，無法復原刪除檔案。這時，可以檢查Windows 資源回收筒中是否還有被刪除的檔案。
    - 在原來資料夾內重新產生被刪除的檔案 (空的檔案)，然後在VS Code的檔案總管的「時間表 (TIMELINE)」查看是否有先前儲存的版本。在先前的版本按下滑鼠右鍵，選擇「還原內容」。

哪裡還有學習操作資源？

你可以選擇指令選單 `說明` > `歡迎` 開啟「歡迎頁面」。這是非常有用和貼心的導引起點，提供了常用任務導引、教學資源和進一步學習的連結。

而VS Code 主畫面配置 (或稱作 使用者介面 user interface) 可以參考文章後面的傳送門，或是其他教學文章，它們都有提供清楚的介面截圖和基本使用說明。

VS Code 幾乎每個月都有更新功能，可以到它的更新網頁查閱 🌐 Update - VS Code 。

使用者介面的傳送門

🌐 Visual Studio Code User Interface