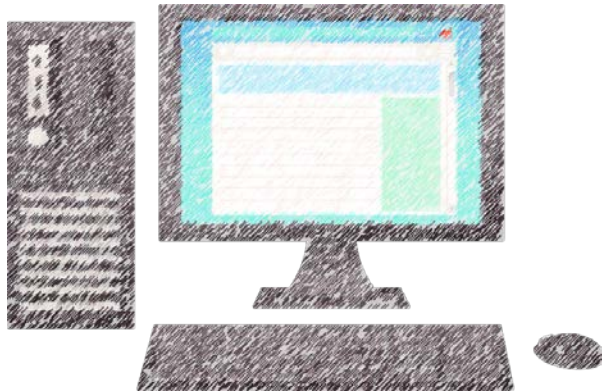


桃園市政府勞動局 112 年勞工學苑產業應用班  
Python 程式設計：從零基礎入門到進階

## 第 2 單元

# Python 快速上手



林柏江老師

元智大學 電機工程學系 助理教授

pclin@saturn.yzu.edu.tw

# 預計課程進度

週次	日期	上午課程內容 (09:00 ~ 12:00)	下午課程內容 (13:00 ~ 16:00)
1	2023/07/23	01. 運算思維簡介	02. Python 快速上手
2	2023/07/30	03. Python 基礎	04. Python 基本資料結構
3	2023/08/06	05. 字串	06. 字典
4	2023/08/13	07. 流程控制	08. 函式
5	2023/08/20	09. 模組與作用域	10. Python 程式檔
6	2023/08/27	11. 檔案系統的使用與檔案讀寫	12. 例外處理
7	2023/09/03	13. 類別與物件導向程式設計	14. 初探資料結構與演算法
8	2023/09/10	15. 陣列	16. 鏈結串列
9	2023/09/17	17. 堆疊與佇列	18. 圖形結構
	2023/09/24, 2023/10/01, 2023/10/08 (共三周) 放假		
10	2023/10/15	19. 樹狀結構	20. 分治法
11	2023/10/22	21. 動態規劃	22. 貪婪演算法
12	2023/10/29	23. 回溯	24. 分支定界法

# 課程大綱

---

1. 認識 Python
2. Python 程式開發環境安裝與設定
3. 執行與測試 Python 程式

# 課程大綱

---

1. 認識 Python
2. Python 程式開發環境安裝與設定
3. 執行與測試 Python 程式

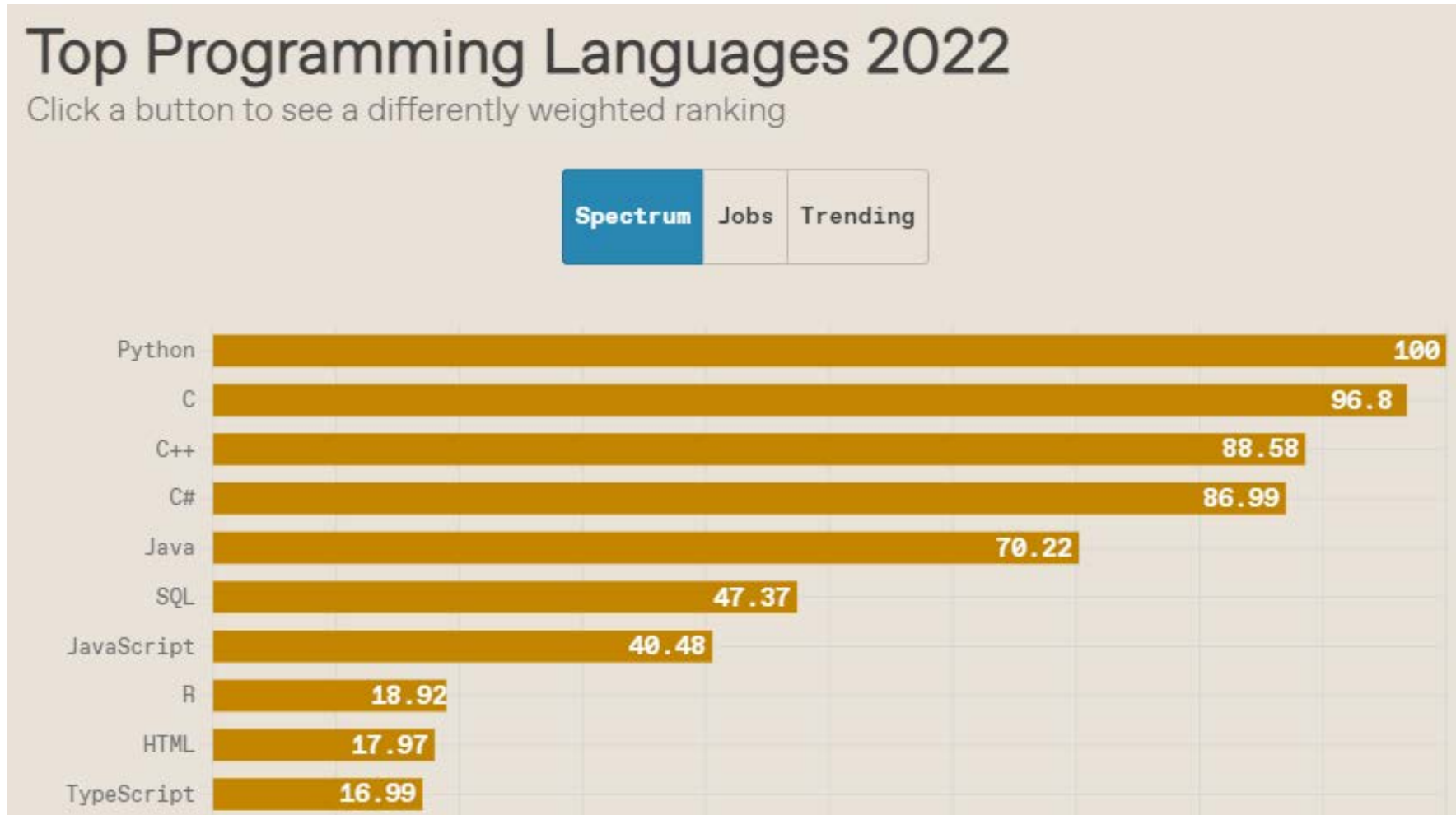
# 認識 Python

---

## Python 的特性

- 跨平台。
- 開放原始碼 (open source)，每個人皆可自由使用與貢獻。
- 簡單直觀，程式碼像英文一樣容易理解。
- 內建許多模組與套件，許多單位也開發更多模組與套件，功能強大且持續擴充。
- 適用於日常任務，開發時間短。
- 應用廣泛。

# Top Programming Languages 2022



Reference: <https://spectrum.ieee.org/top-programming-languages-2022>

# 認識 Python

---

- Python 是一種直譯式 (interpreted) 的程式語言。

## 直譯式程式語言

- 直譯器 (interpreter) 會直接把程式裡的指令一句一句轉換並執行。
  - 不需要先使用編譯器 (compiler) 進行編譯 (compile)，把整個程式轉換成機器指令，然後再執行。
- Python Documentation:
    - <https://docs.python.org/3/index.html>

# Python 的作者

## Guido van Rossum

- 1956 年出生於荷蘭。
- 1982 年畢業於阿姆斯特丹大學數學與電腦科學系，取得碩士學位。
- 1989 年創建 Python 並開放。
- 個人網頁：  
<https://gvanrossum.github.io>





# Python 2 與 Python 3

- 2000/10/16 Python 2.0 正式發表。
- 2008/12/03 Python 3.0 正式發表，並以每隔一年左右推出一個 3.x 版本前進。
- Python 3 在開發時，為了不受到先前 Python 2.x 版本的束縛，因此沒有考慮向後相容性。
- Python 2.7 是 Python 2.x 的最後一個版本，並且已於 2020 年停止支援。

版本	發表日期
3.0	2008/12/03
3.1	2009/06/27
3.2	2011/02/20
3.3	2012/09/29
3.4	2014/03/16
3.5	2015/09/13
3.6	2016/12/23
3.7	2018/06/27
3.8	2019/10/14
3.9	2020/10/05
3.10	2021/10/04
3.11	2022/10/24

# 課程大綱

---

1. 認識 Python
2. Python 程式開發環境安裝與設定
3. 執行與測試 Python 程式

# 建立 Python 程式開發環境

---

- 一般而言，到 Python 官方網站 (<https://www.python.org/>) 就可以找到安裝檔，然後下載安裝即可。
- 但是 ...
- 我們通常需要使用到許多 Python 套件。
- 許多套件之間有相依性。
- 許多套件的版本與日俱進，相依性也跟著變化。
- 我們可能會希望在一部電腦上有多種 Python 與各套件的版本，並且不希望互相干擾或混亂。
- 建議：使用套件管理系統，以及虛擬環境 (virtual environment)。

# Python 程式開發環境的安裝與設定

---

1. Anaconda 安裝

<https://youtu.be/hYRBpqiK-Fg>

2. 虛擬環境的建立、啟動、以及離開

<https://youtu.be/KEDw5iELpQU>

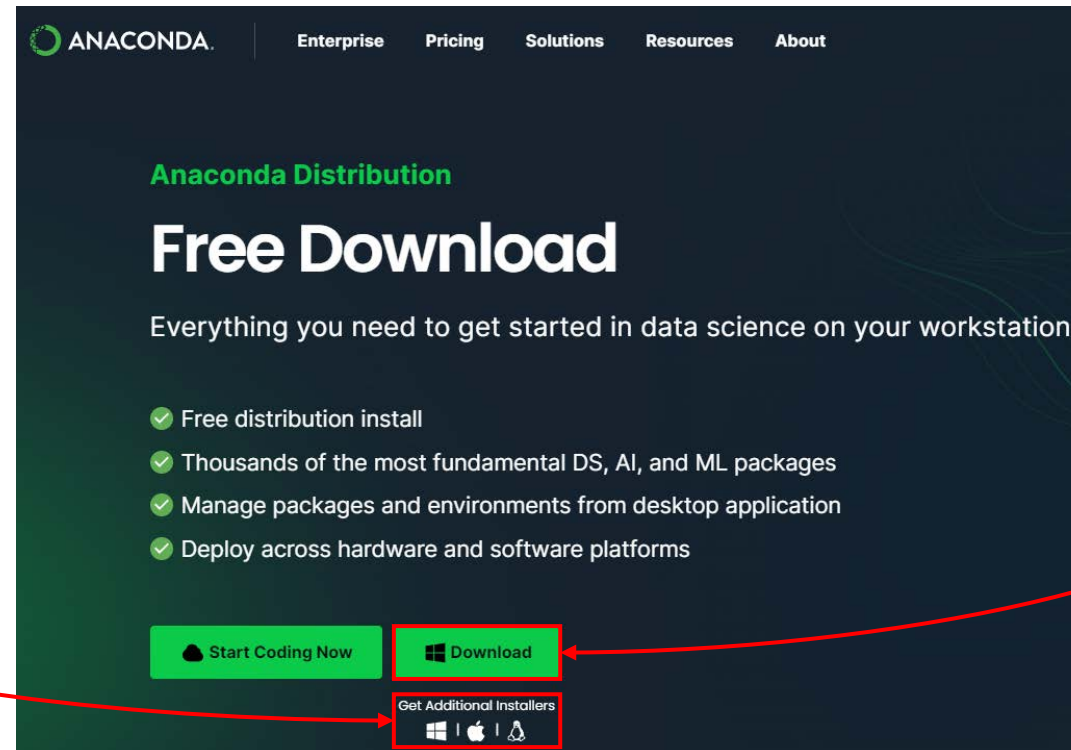
3. Visual Studio Code 的安裝、Python 相關設定、以及使用

[https://youtu.be/eXxMIle\\_NJE](https://youtu.be/eXxMIle_NJE)

# 下載並安裝 Anaconda

- Anaconda 是一套容易使用的 Python 發行版，著重於機器學習、資料科學的應用，並提供 conda 套件管理系統。
- 下載網址：<https://www.anaconda.com/download>

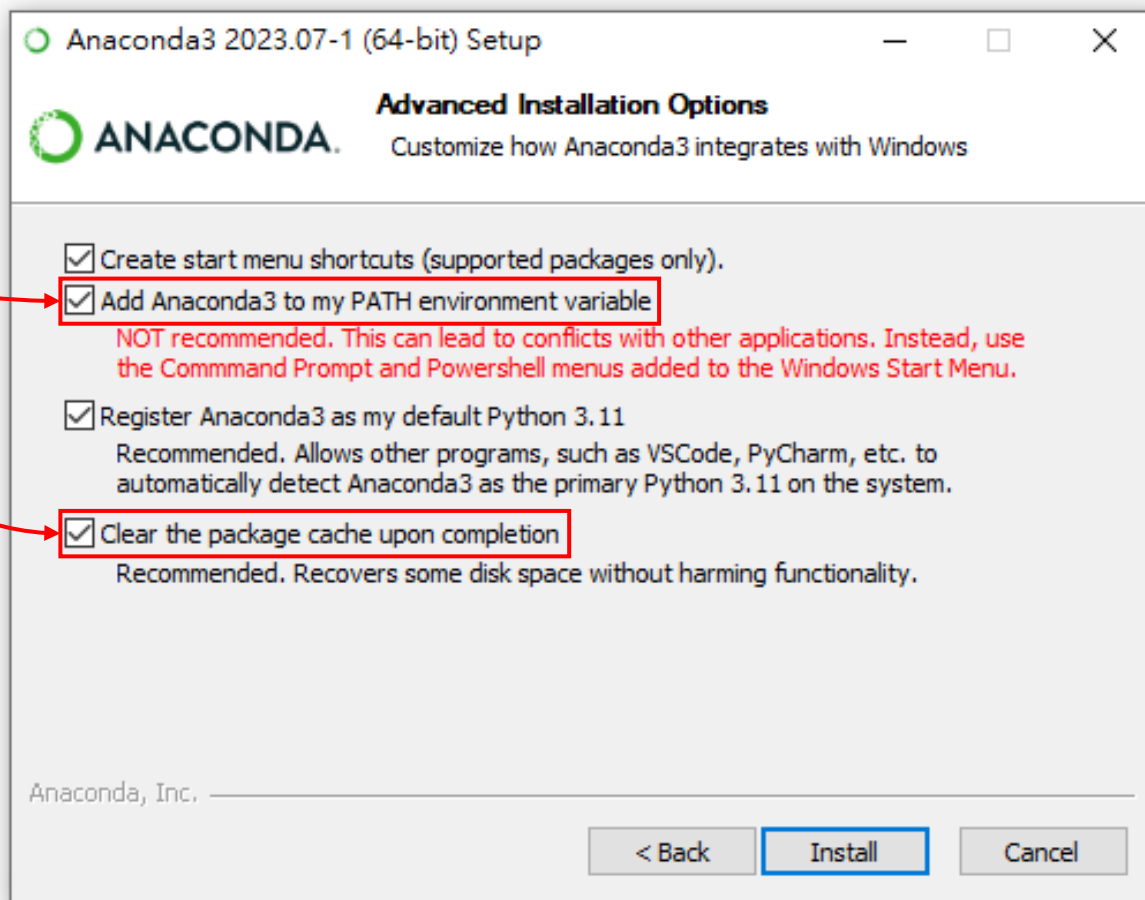
也可根據作業系統  
來選擇安裝檔



點這裡下載

# Anaconda 安裝過程的注意事項

建議勾選這兩項



# 建立與使用虛擬環境

虛擬環境的名稱 (可自行指定)

指定 Python 版本

- 使用命令提示字元建立一個新的虛擬環境 (只需要建立一次即可) :
  - `conda create -n python-course python=3`
- 顯示本機已建立的虛擬環境 :
  - `conda info -e`
- 啟用一個虛擬環境 :
  - `conda activate python-course`
  - 命令提示字元前方的小括號內會顯示正在使用的虛擬環境名稱。
- 離開目前的虛擬環境 :
  - `conda deactivate`
  - 小括號標註的虛擬環境名稱會消失，表示已離開虛擬環境。

要啟用的虛擬環境名稱

# 刪除虛擬環境

---

- 若有虛擬環境用不到了，想要刪除：
    - `conda remove -n python-course --all`
- 要刪除的虛擬環境名稱



# 安裝所需套件 (以 Numpy 為例)

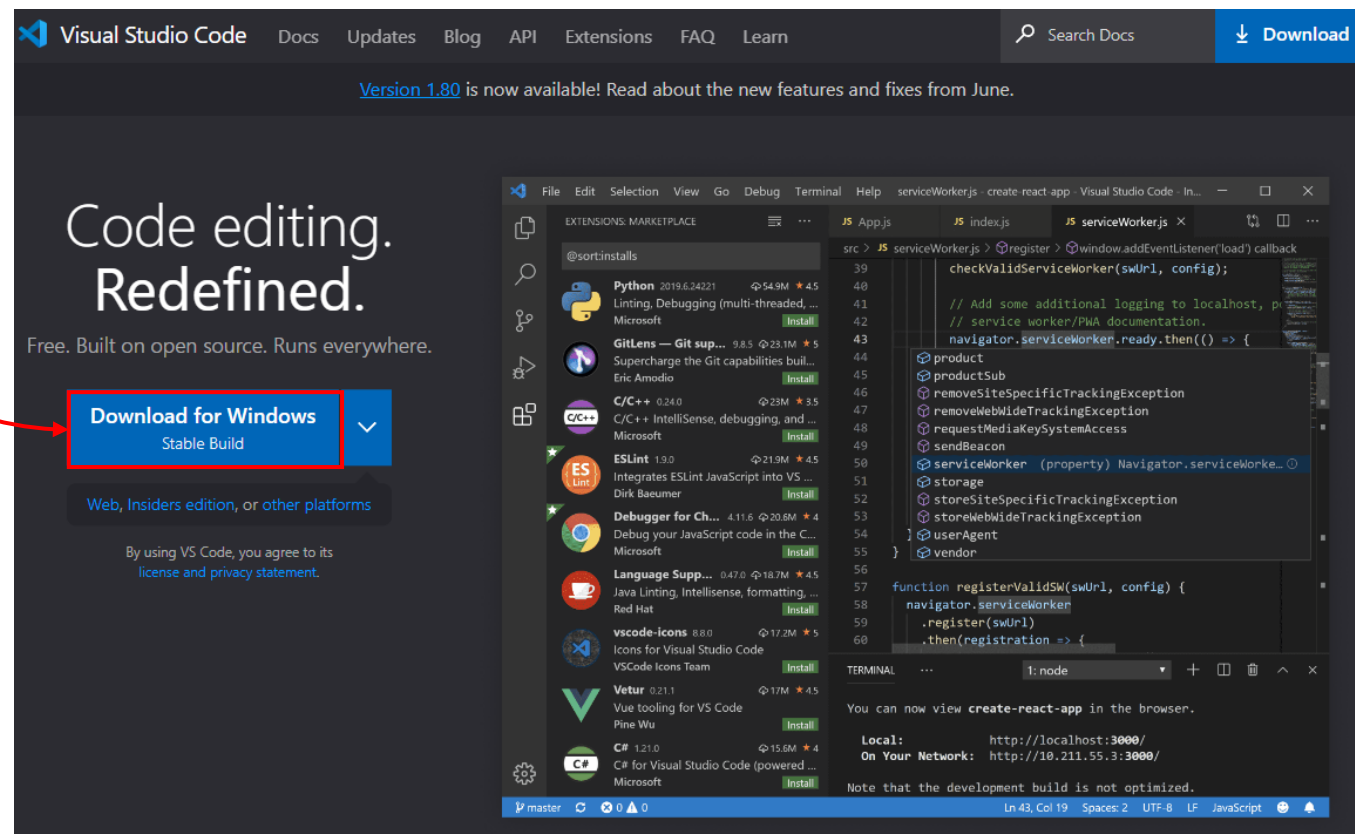
---

- 開啟命令提示字元。
- 啟動虛擬環境。
- 執行以下指令：
  - `conda install numpy`

# 下載並安裝 Visual Studio Code

- Visual Studio Code (簡稱 VS Code) 是一款免費、有效率的程式碼編輯器，支援多種程式語言、除錯等功能，並可安裝各種延伸模組 (extension)。
- 下載網址：<https://code.visualstudio.com>

點這裡下載



# 把 Visual Studio Code 改為中文介面

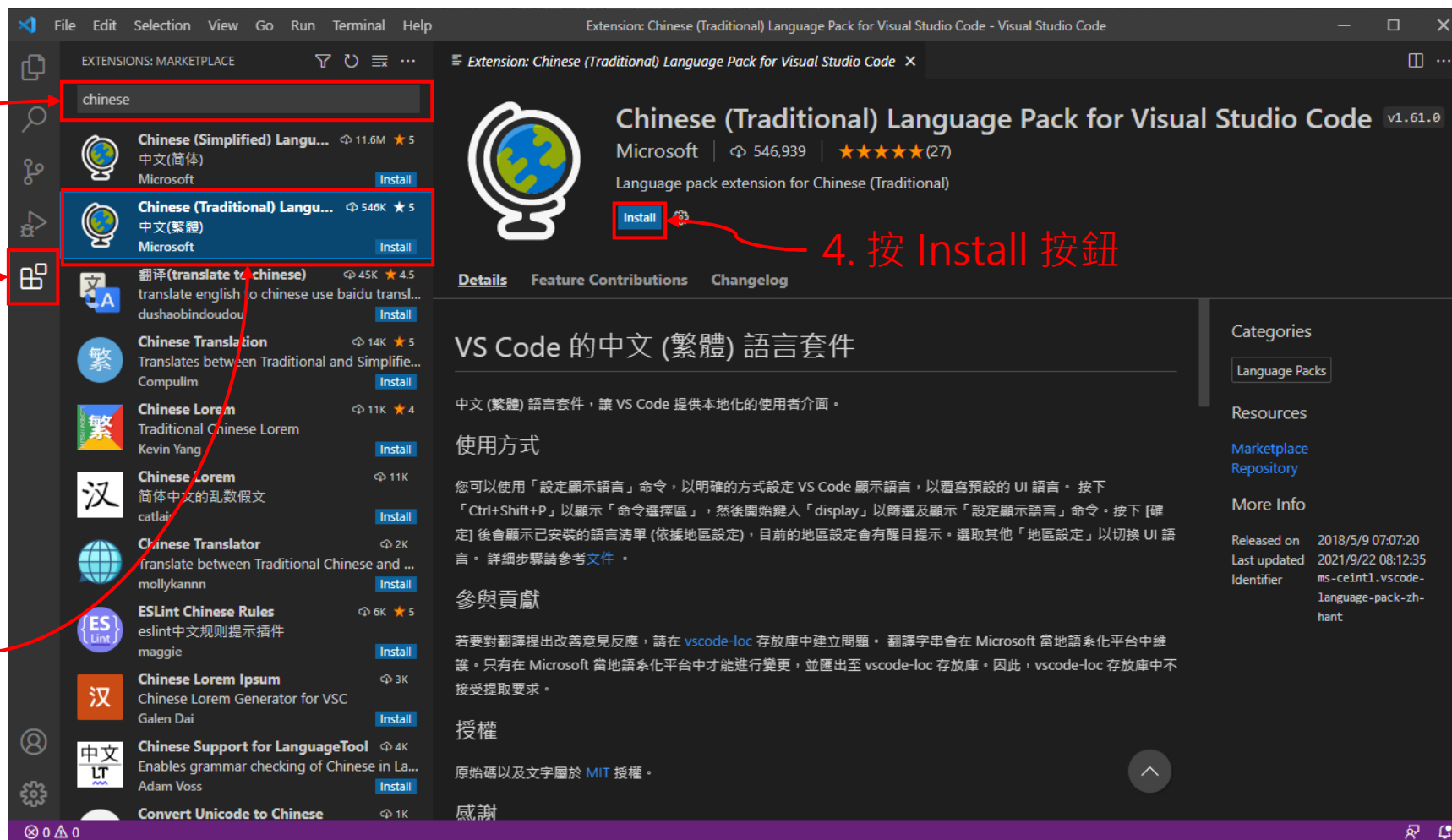
2. 在搜尋欄位  
輸入 chinese

1. 按 Extensions  
按鈕

3. 選擇  
Chinese  
(Traditional)  
Language  
Pack

4. 按 Install 按鈕

5. 重新啟動  
VS Code



根據經驗，中文介面偶而會導致 VS Code 出現奇怪錯誤。  
可試著把此中文介面解除安裝，看看錯誤是否消失。

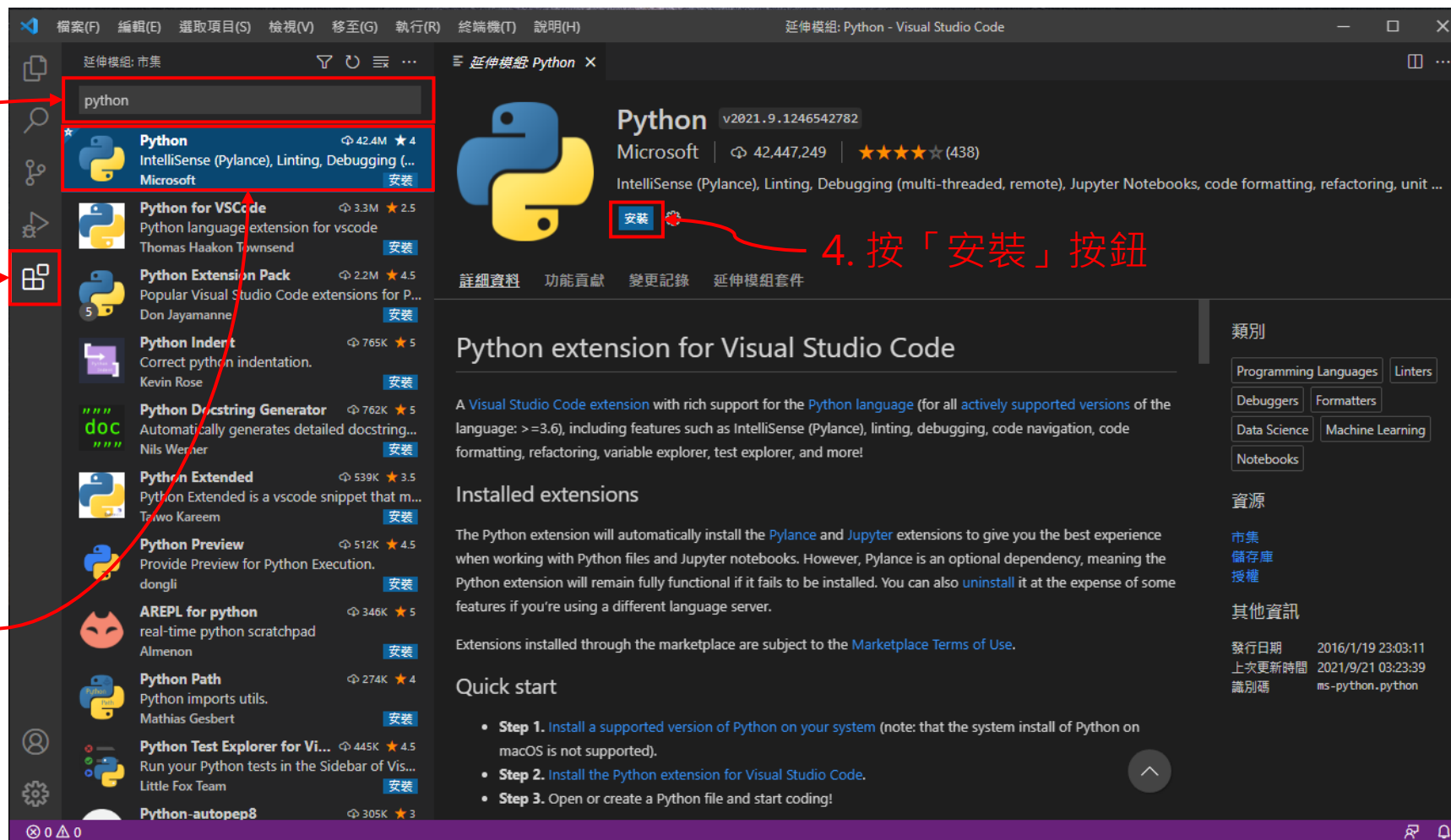
# 安裝 Python Extension

2. 在搜尋欄位  
輸入 python

1. 按 Extensions  
按鈕

3. 選擇 Python  
(請確認發行者  
名稱是  
Microsoft)

4. 按「安裝」按鈕



# 課程大綱

---

1. 認識 Python
2. Python 程式開發環境安裝與設定
3. 執行與測試 Python 程式

# 開始撰寫程式

---

有兩種方式來撰寫並執行程式：

## 方式一：在 Python Shell 互動模式輸入指令並觀察結果

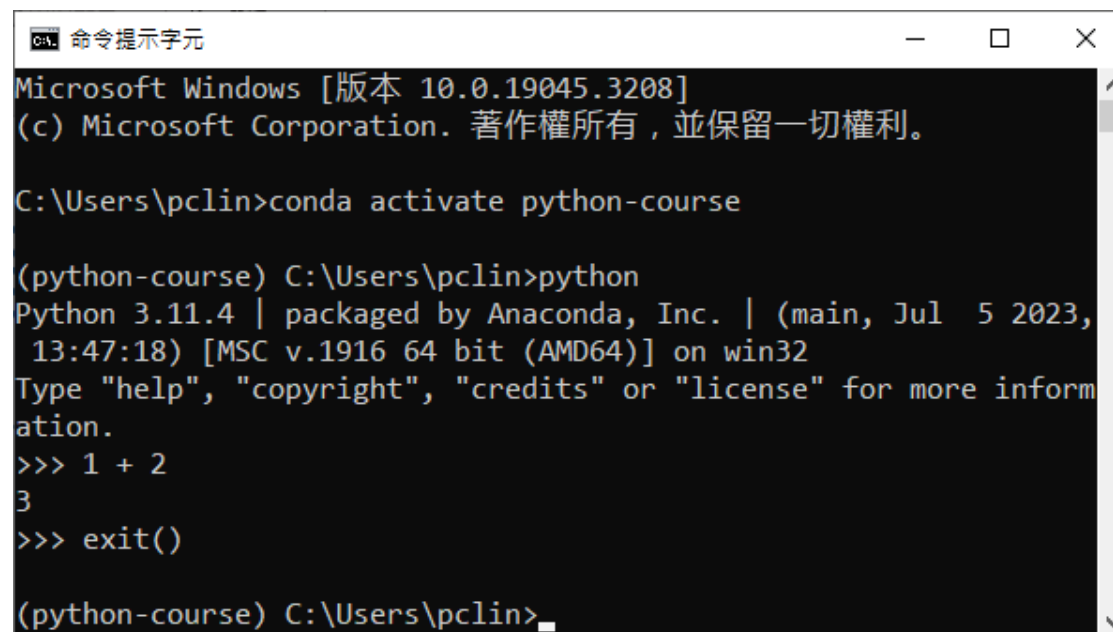
- 簡單、即時、互動式的程式設計環境。
- 每輸入一行指令，就會立刻執行，並顯示結果。
- 方便使用，適合學習與測試。
- 也稱為 REPL (Read-Eval-Print Loop)。

## 方式二：以直譯器執行 Python 程式碼檔案

- 先在文字編輯器或整合式開發環境中撰寫程式碼，並存檔。
- 再使用直譯器執行此程式碼檔案。
- 可開發較大的程式，並且易於除錯。
- 常見的副檔名為 .py。

# 在 Python Shell 輸入指令並觀察結果

- 在命令提示字元裡啟動 Python 虛擬環境。
- 輸入 `python`，進入 Python Shell。
- Python Shell 的提示符號是 `>>>`。
- 輸入 Python 指令並觀察結果。
- 若要離開 Python Shell，可以採用以下任一種方法：
  - `exit()`
  - `quit()`
  - `Ctrl+z` 再按 `Enter` 鍵



```
命令提示字元
Microsoft Windows [版本 10.0.19045.3208]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\pclin>conda activate python-course

(python-course) C:\Users\pclin>python
Python 3.11.4 | packaged by Anaconda, Inc. | (main, Jul  5 2023,
13:47:18) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 1 + 2
3
>>> exit()

(python-course) C:\Users\pclin>
```

# 在 Python Shell 輸入指令並觀察結果

- 可直接輸入算術運算，並觀察結果。



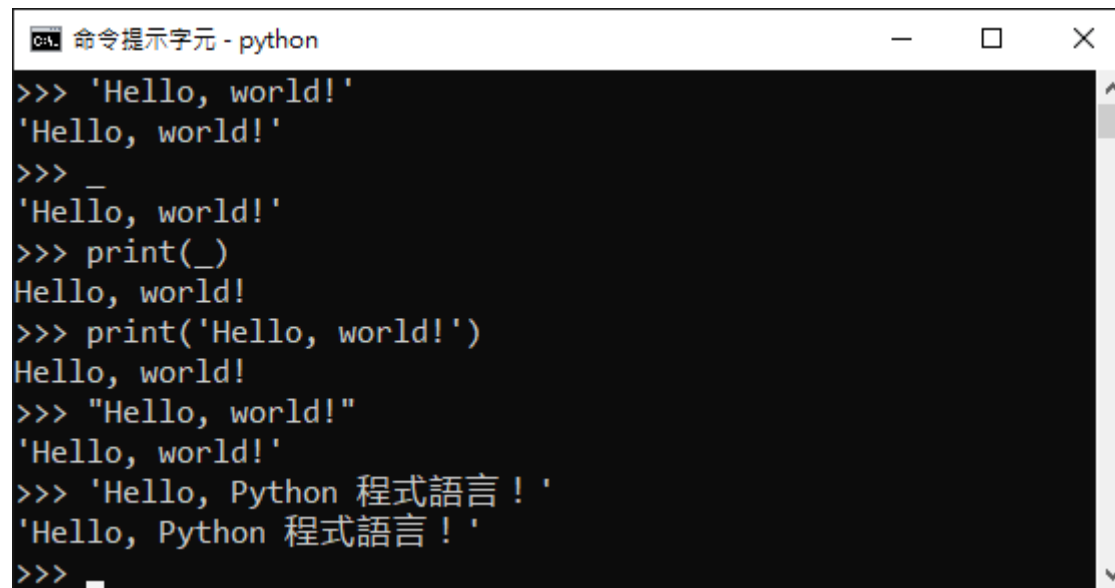
```
命令提示字元 - python
>>> 1 + 2
3
>>> _
3
>>> 1 + _
4
>>> _
4
>>> _
```

底線符號\_表示在 Python Shell 環境裡上一次運算結果，方便直接取用。



# 在 Python Shell 輸入指令並觀察結果

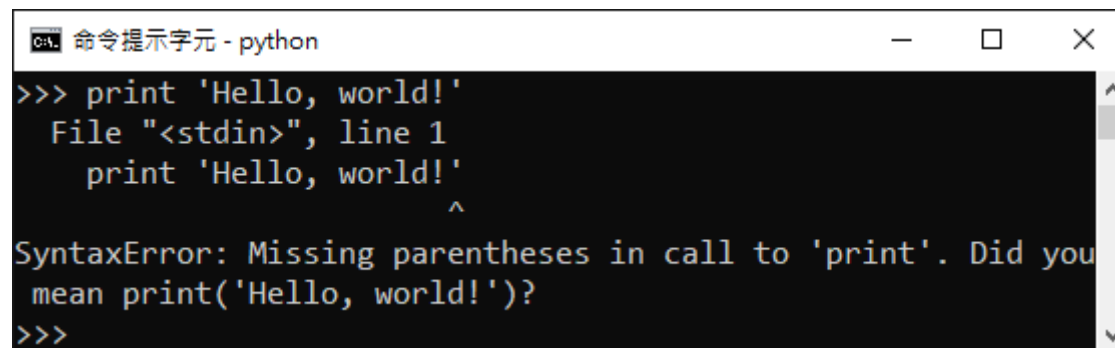
- 一對單引號 `' '` 或是雙引號 `" "` 框起來的文字是一個字串。
- 字串內的文字中英文皆可。

A screenshot of a Windows command prompt window titled "命令提示字元 - python". The window has a black background with white text. It shows a series of Python commands and their outputs. The commands are: `>>> 'Hello, world!'`, `>>> _`, `>>> print(_)`, `>>> print('Hello, world!')`, `>>> "Hello, world!"`, and `>>> 'Hello, Python 程式語言!'`. The outputs are: `'Hello, world!'`, `'Hello, world!'`, `Hello, world!`, `Hello, world!`, `'Hello, world!'`, and `'Hello, Python 程式語言!'`. The cursor is at the end of the last command.

```
>>> 'Hello, world!'
'Hello, world!'
>>> _
'Hello, world!'
>>> print(_)
Hello, world!
>>> print('Hello, world!')
Hello, world!
>>> "Hello, world!"
'Hello, world!'
>>> 'Hello, Python 程式語言!'
'Hello, Python 程式語言!'
>>> 
```

# 在 Python Shell 輸入指令並觀察結果

- 若輸入的指令有錯誤，Python Shell 會產生錯誤訊息。

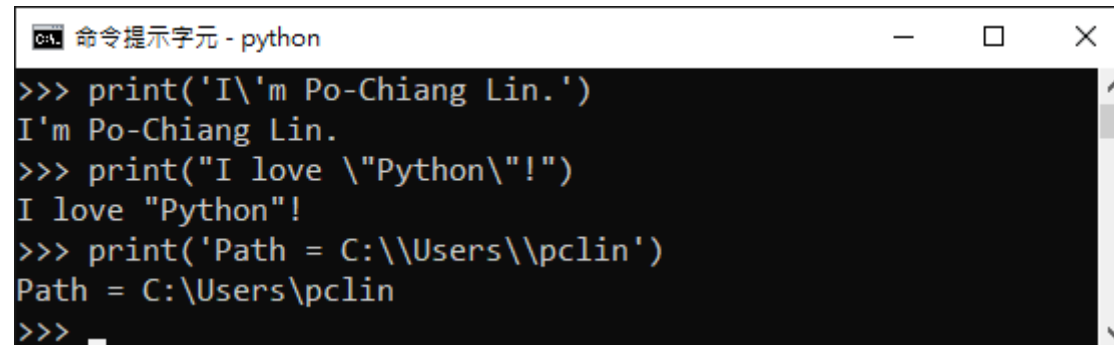


A screenshot of a Python Shell window titled "命令提示字元 - python". The window has a black background with white text. The prompt is ">>>". The user has entered the command `print 'Hello, world!'`. The shell has responded with an error message: `File "<stdin>", line 1`, `print 'Hello, world!'`, and a caret (^) pointing to the space before the closing quote. Below this, the full error message is displayed: `SyntaxError: Missing parentheses in call to 'print'. Did you mean print('Hello, world!')`. The prompt `>>>` is shown again at the bottom.

```
>>> print 'Hello, world!'
File "<stdin>", line 1
    print 'Hello, world!'
          ^
SyntaxError: Missing parentheses in call to 'print'. Did you
mean print('Hello, world!')?
>>>
```

# 在 Python Shell 輸入指令並觀察結果

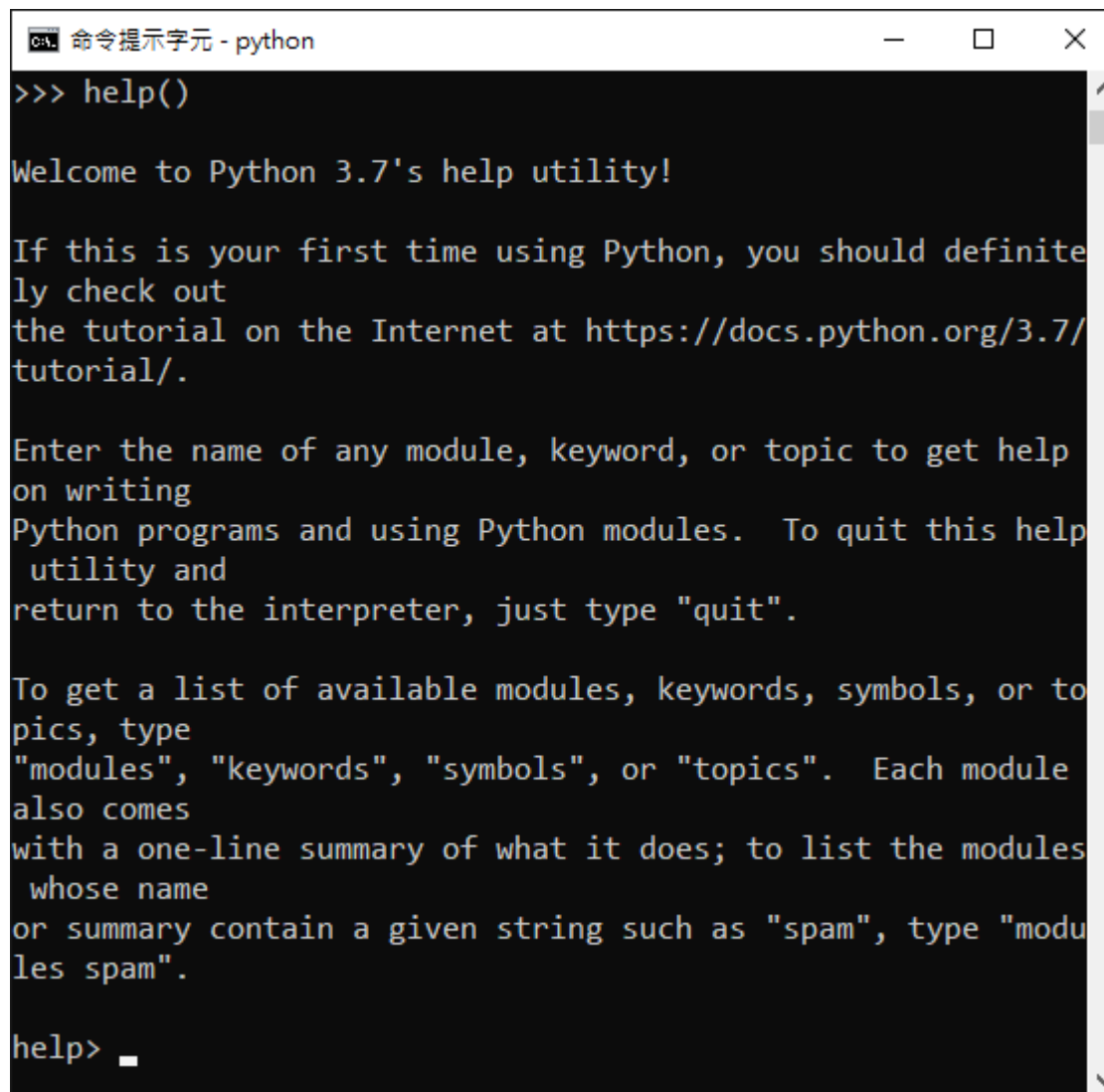
- 以單引號包起來的字串，如果在字串資料中遇到單引號時，則必須以 \ 來跳脫 (escape) 處理，避免被誤認為字串結束。
- 同理，以雙引號包起來的字串，如果在字串資料中遇到雙引號時，則必須以 \ 來跳脫 (escape) 處理，避免被誤認為字串結束。
- 若是字串中有 \ 字元時，通常必須再用一個 \ 跳脫。



```
命令提示字元 - python
>>> print('I\'m Po-Chiang Lin.')
I'm Po-Chiang Lin.
>>> print("I love \"Python\"!")
I love "Python"!
>>> print('Path = C:\\Users\\pclin')
Path = C:\Users\pclin
>>> _
```

# 在 Python Shell 輸入指令並觀察結果

- 若想要取得協助訊息，可輸入 `help()` 進入說明頁面。
- 提示符號變成 `help>`。
- 可輸入要查詢的訊息，例如 `modules`、`keywords`。
- 也可輸入要查詢的函式，例如 `print`。
- 若要結束此說明頁面，可輸入 `quit`。

A screenshot of a Windows command prompt window titled "命令提示字元 - python". The window shows the execution of the `help()` command in a Python shell. The output text is as follows:

```
>>> help()

Welcome to Python 3.7's help utility!

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at https://docs.python.org/3.7/tutorial/.

Enter the name of any module, keyword, or topic to get help
on writing
Python programs and using Python modules. To quit this help
utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, symbols, or to
pics, type
"modules", "keywords", "symbols", or "topics". Each module
also comes
with a one-line summary of what it does; to list the modules
whose name
or summary contain a given string such as "spam", type "modu
les spam".

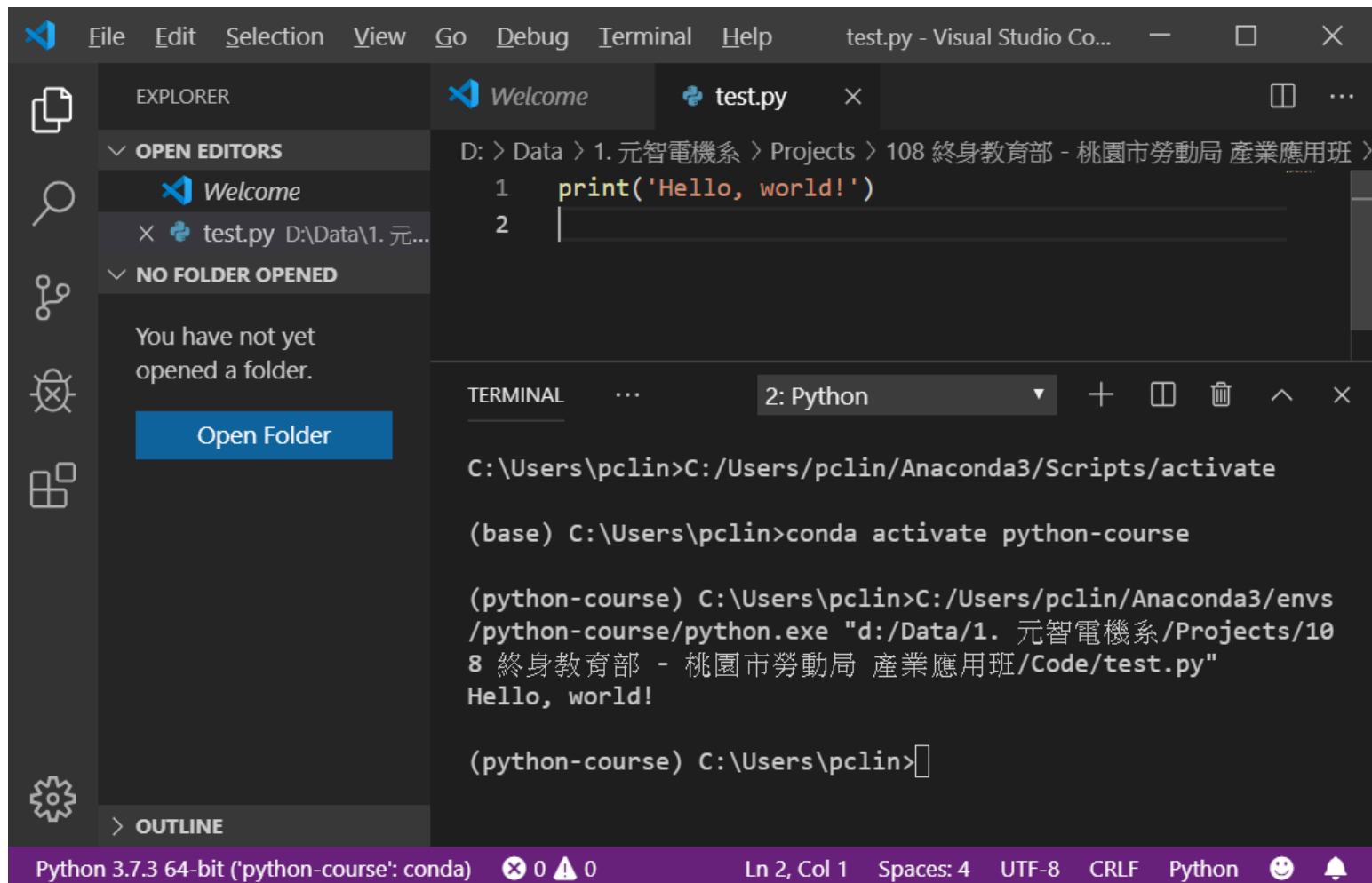
help> _
```

# 以直譯器執行 Python 程式碼檔案

---

- Python 程式碼檔案為純文字檔，可以使用任何文字編輯器來開啟、修改內容、並儲存。
- 我們使用 Visual Studio Code (簡稱 VS Code)。
  - 開啟 VS Code。
  - 按 File -> New File，開啟一個新的 Python 程式碼檔案。
  - 按 File -> Save As，確認存檔類型是 Python，再輸入你想要的檔名，並選擇儲存的資料夾位置。
  - 輸入程式碼。
  - 在左下角確認目前的 Python 直譯器是你需要的版本，不是的話就點選並修改。
  - 點擊右上角的 Run Python File 三角形按鈕，執行程式碼。
  - 下方會開啟一個 Terminal，並出現執行結果。

# 以直譯器執行 Python 程式碼檔案



The screenshot displays the Visual Studio Code interface. The Explorer panel on the left shows the file structure with 'test.py' open. The Editor panel shows the code: `print('Hello, world!')`. The Terminal panel at the bottom shows the execution process, including activating the conda environment and running the script, resulting in the output 'Hello, world!'.

```
File Edit Selection View Go Debug Terminal Help test.py - Visual Studio Co...  
EXPLORER  
OPEN EDITORS  
Welcome  
test.py D:\Data\1. 元智電機系 > Projects > 108 終身教育部 - 桃園市勞動局 產業應用班 >  
NO FOLDER OPENED  
You have not yet opened a folder.  
Open Folder  
OUTLINE  
D: > Data > 1. 元智電機系 > Projects > 108 終身教育部 - 桃園市勞動局 產業應用班 >  
1 print('Hello, world!')  
2 |  
2: Python  
C:\Users\pclin>C:/Users/pclin/Anaconda3/Scripts/activate  
(base) C:\Users\pclin>conda activate python-course  
(python-course) C:\Users\pclin>C:/Users/pclin/Anaconda3/envs/python-course/python.exe "d:/Data/1. 元智電機系/Projects/108 終身教育部 - 桃園市勞動局 產業應用班/Code/test.py"  
Hello, world!  
(python-course) C:\Users\pclin>  
Python 3.7.3 64-bit ("python-course": conda) 0 0 Ln 2, Col 1 Spaces: 4 UTF-8 CRLF Python
```

# 以直譯器執行 Python 程式碼檔案

- 另一種執行 Python 程式碼檔案的方法：
  - 先在文字編輯器撰寫程式並儲存，副檔名為 .py。
  - 在命令提示字元裡啟動 Python 虛擬環境。
  - 輸入 `python filename.py`，其中 `filename` 是 Python 程式碼檔案的主檔名。
  - 畫面上會出現執行結果。



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19045.3208]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

D:\Data\1. 元智電機系\Projects\112 終身教育部 - 桃園市政府勞動局勞工學苑產業應用班\Code>conda activate python-course

(python-course) D:\Data\1. 元智電機系\Projects\112 終身教育部 - 桃園市政府勞動局勞工學苑產業應用班\Code>python test.py
Hello, world!

(python-course) D:\Data\1. 元智電機系\Projects\112 終身教育部 - 桃園市政府勞動局勞工學苑產業應用班\Code>_
```