

AI



紀老師程式教學網

<https://www.facebook.com/teacherchi>



機器學習

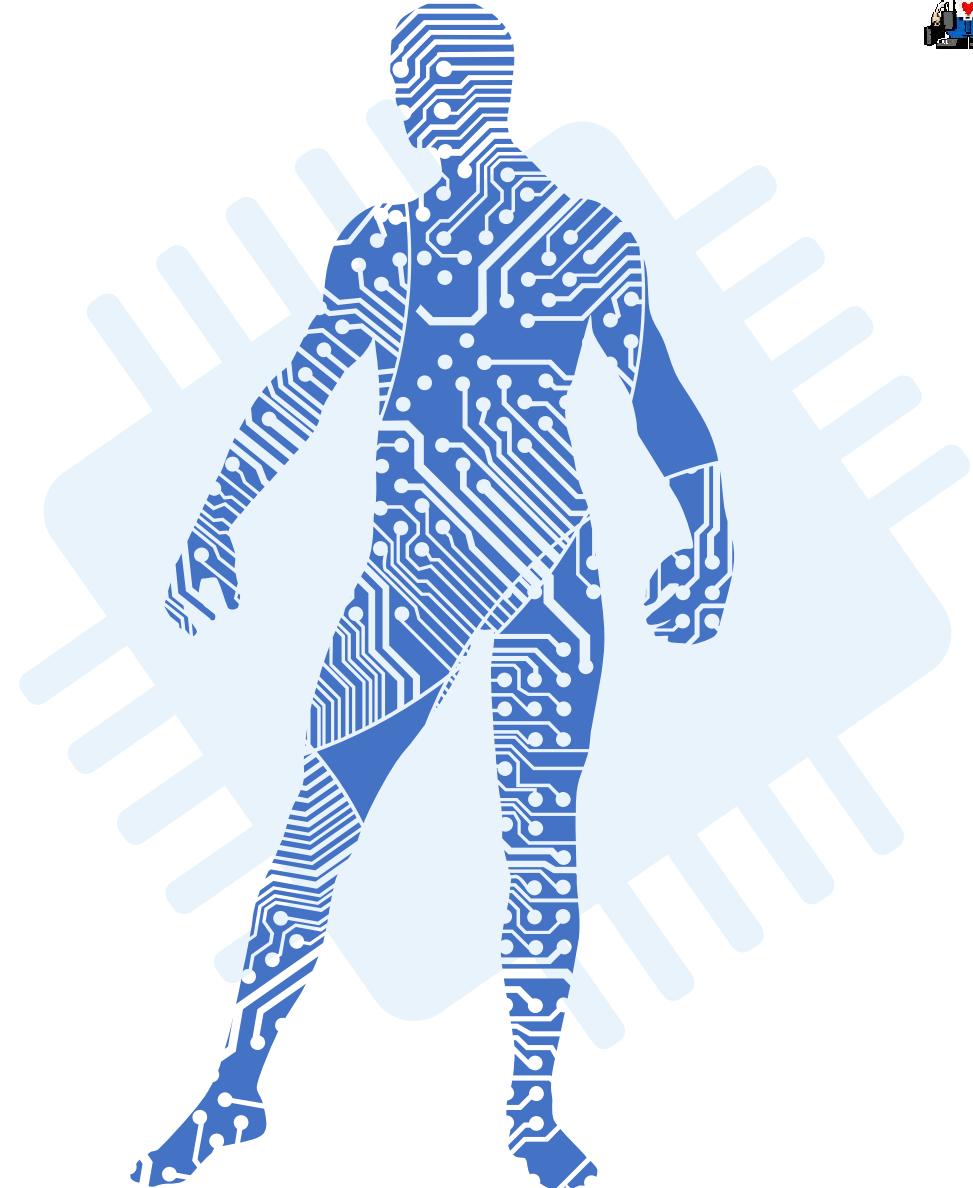
第 2 章 環境安裝

講師：紀俊男



本章大綱

- 機器學習所需工具
- 安裝 Anaconda
- Anaconda 環境介紹 & 試用
- Anaconda 除錯的方法
- 外掛套件的安裝方法





**機器學習
所需工具**



機器學習所需之「軟體工具」



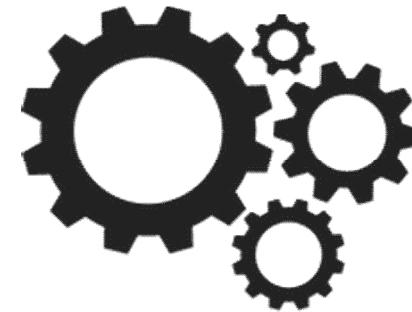
流程控制



工作環境



函式庫



NumPy, SciPy,
Pandas, SKLearn,
TensorFlow, Keras





安裝 Anaconda



Anaconda 的優點



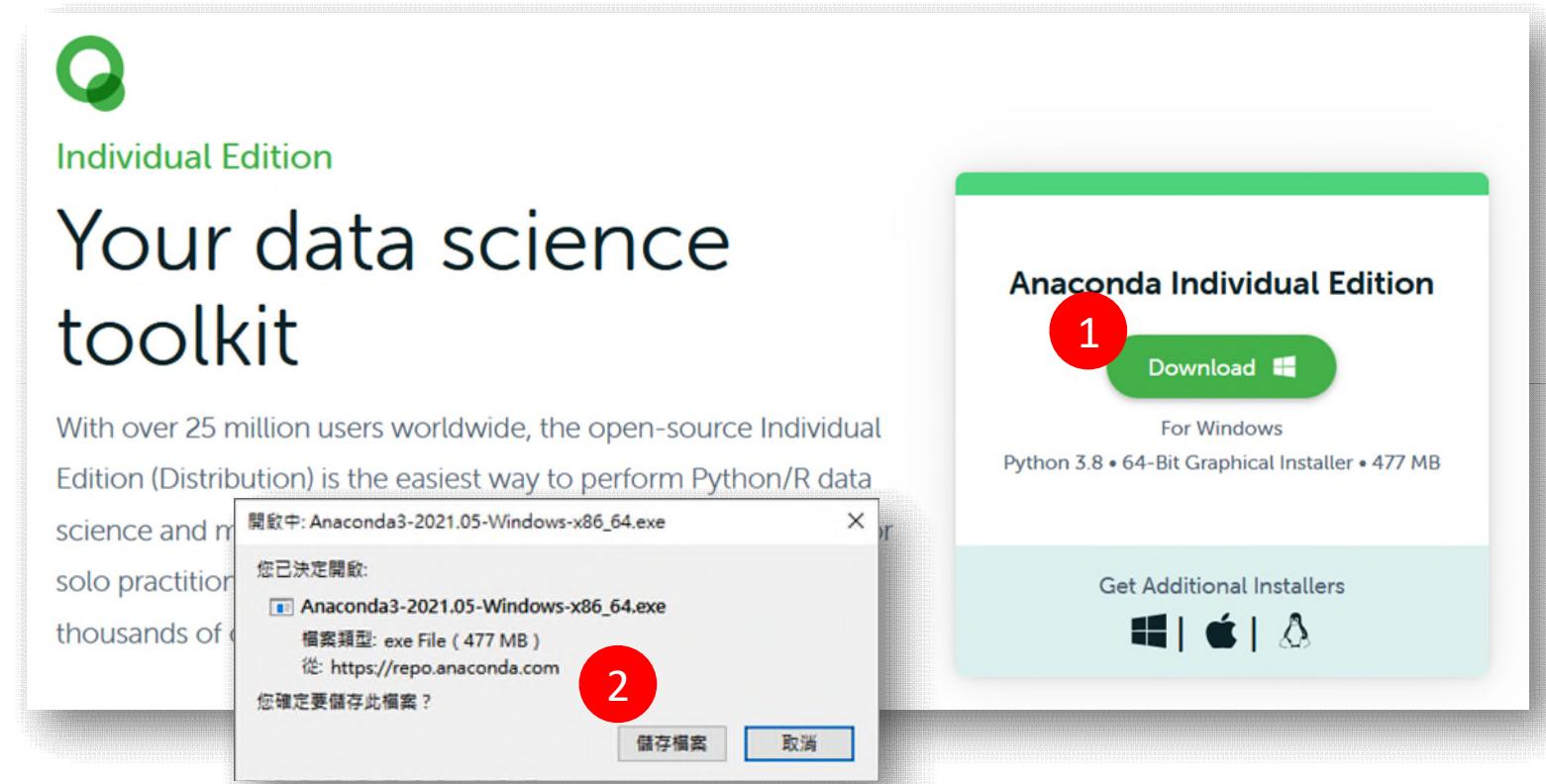
- 資料科學 & 人工智慧最佳工作環境
- 支援 Windows、MacOS、Linux
- 開源、免費
- 支援 Python 與 R 語言
- 內建常用的資料科學外掛套件
- 支援以 conda 指令，下載額外 1500+ 套件





Step 1：下載 Anaconda

- <https://www.anaconda.com/products/individual> (個人版首頁)



1. 點擊「Download」
2. 儲存檔案即可！



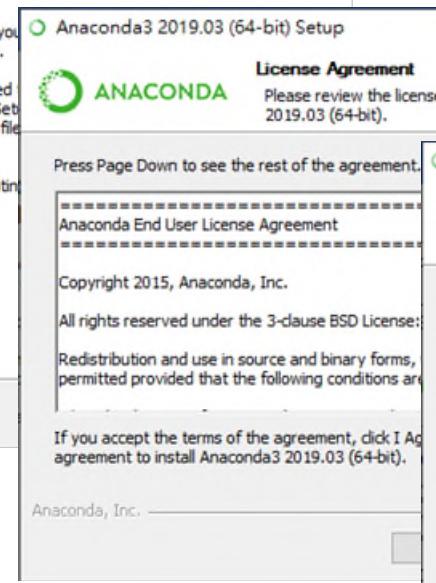
Step 2：執行安裝精靈



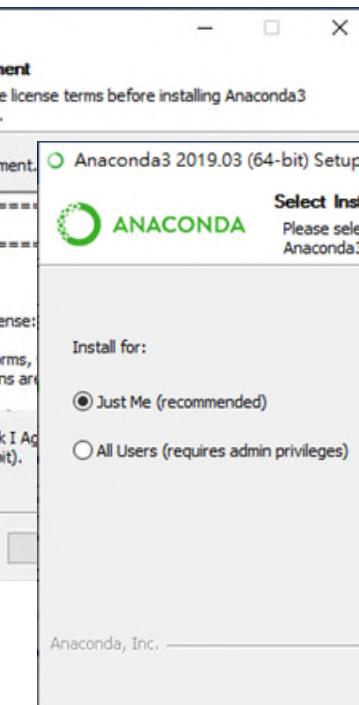
Anaconda3-201
9.03-Windows-x
86_64



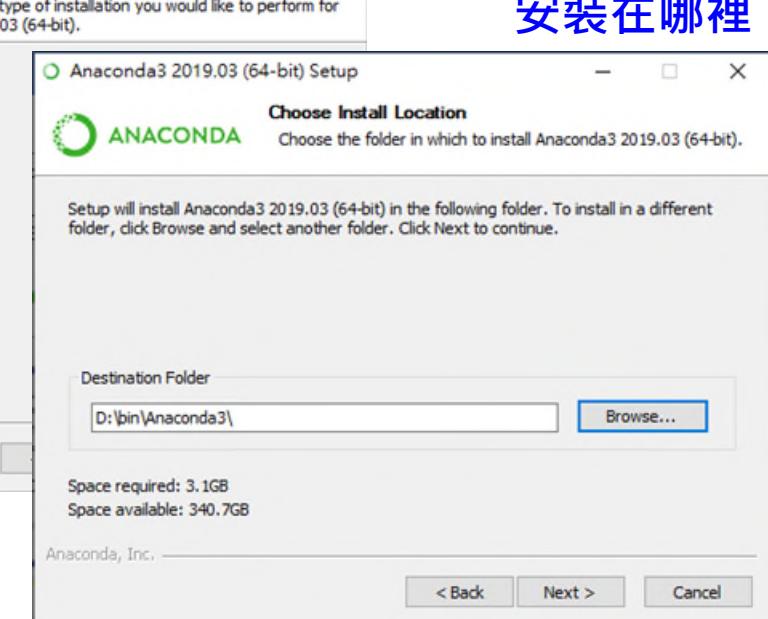
歡迎畫面



許可協議



安裝給誰用



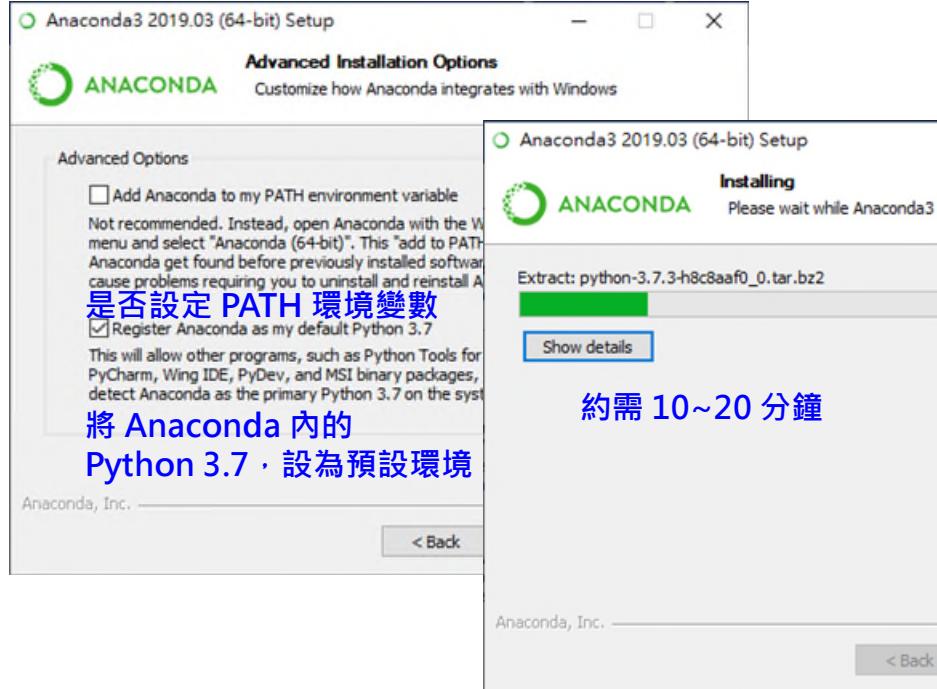
安裝在哪裡



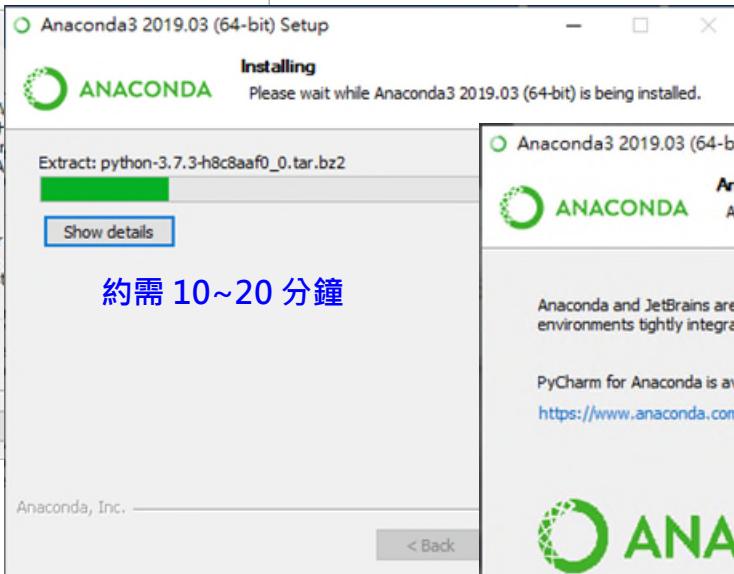


Step 2：執行安裝精靈（續）

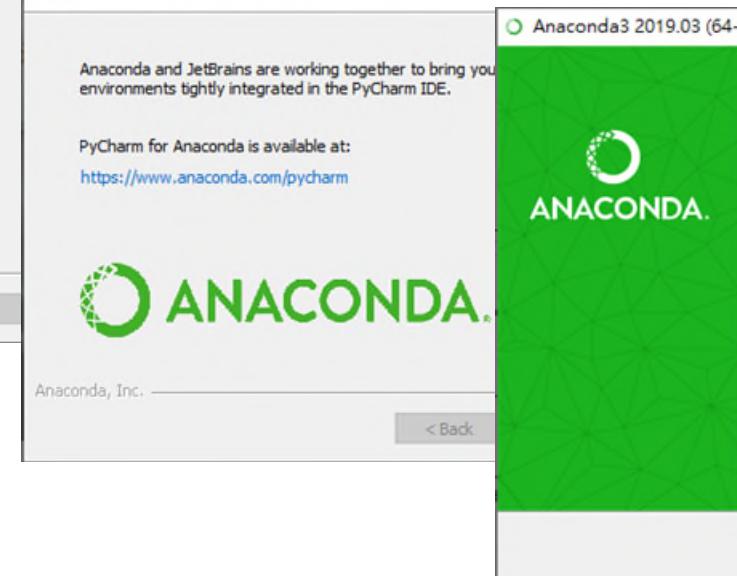
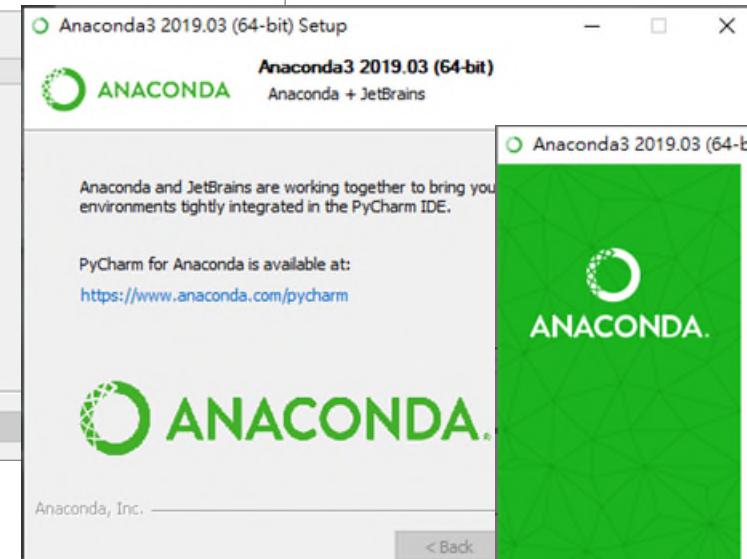
額外設定



安裝中



與 PyCharm 相容

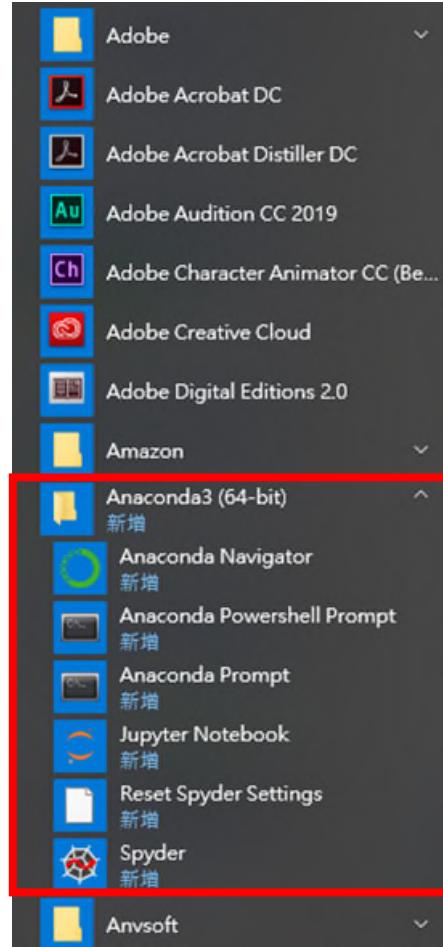


安裝結束





Step 3：確認安裝完成



- **Anaconda Navigator**
 - 通往各種整合工具的「門戶」
 - 可通往「Jupyter」、「Spyder」...等等
- **Anaconda Powershell Prompt / Anaconda Prompt**
 - Anaconda 的命令列工具
 - 安裝額外套件時很常用
- **Jupyter Notebook**
 - 一個「網頁式」的 Python 開發環境
- **Reset Spyder Settings**
 - 將 Spyder 的設定，還原成預設值
- **Spyder**
 - 視窗型的 Python 開發環境（本課程主力環境）



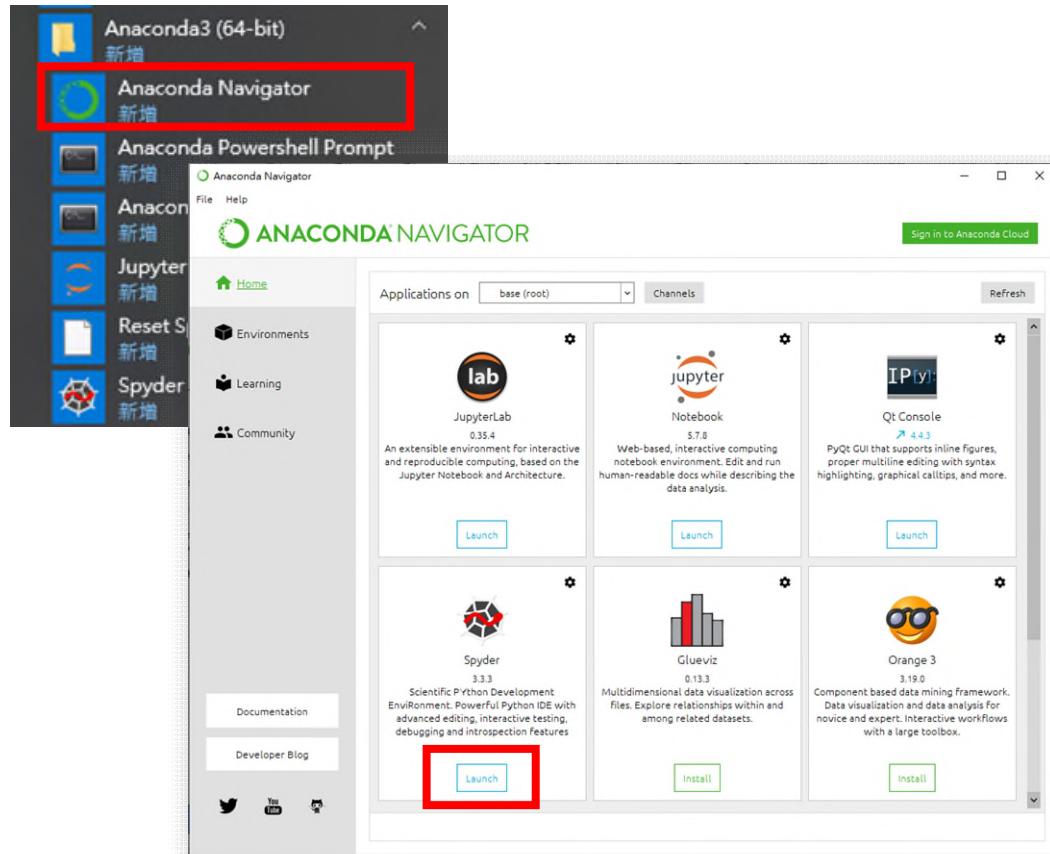


Anaconda 環境介紹 & 試用

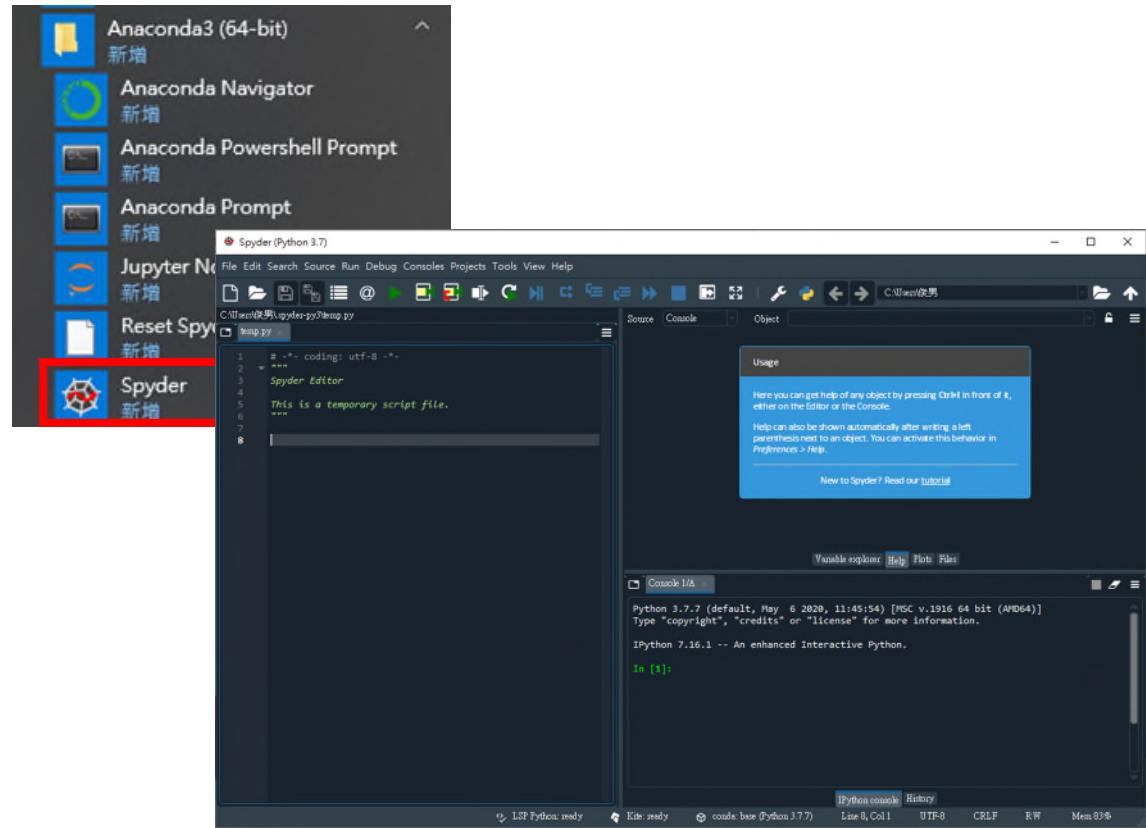


啟動 Anaconda 內的 Spyder

- Navigator > Spyder



- 直接開啟 Spyder





隨堂練習：開啟 Spyder

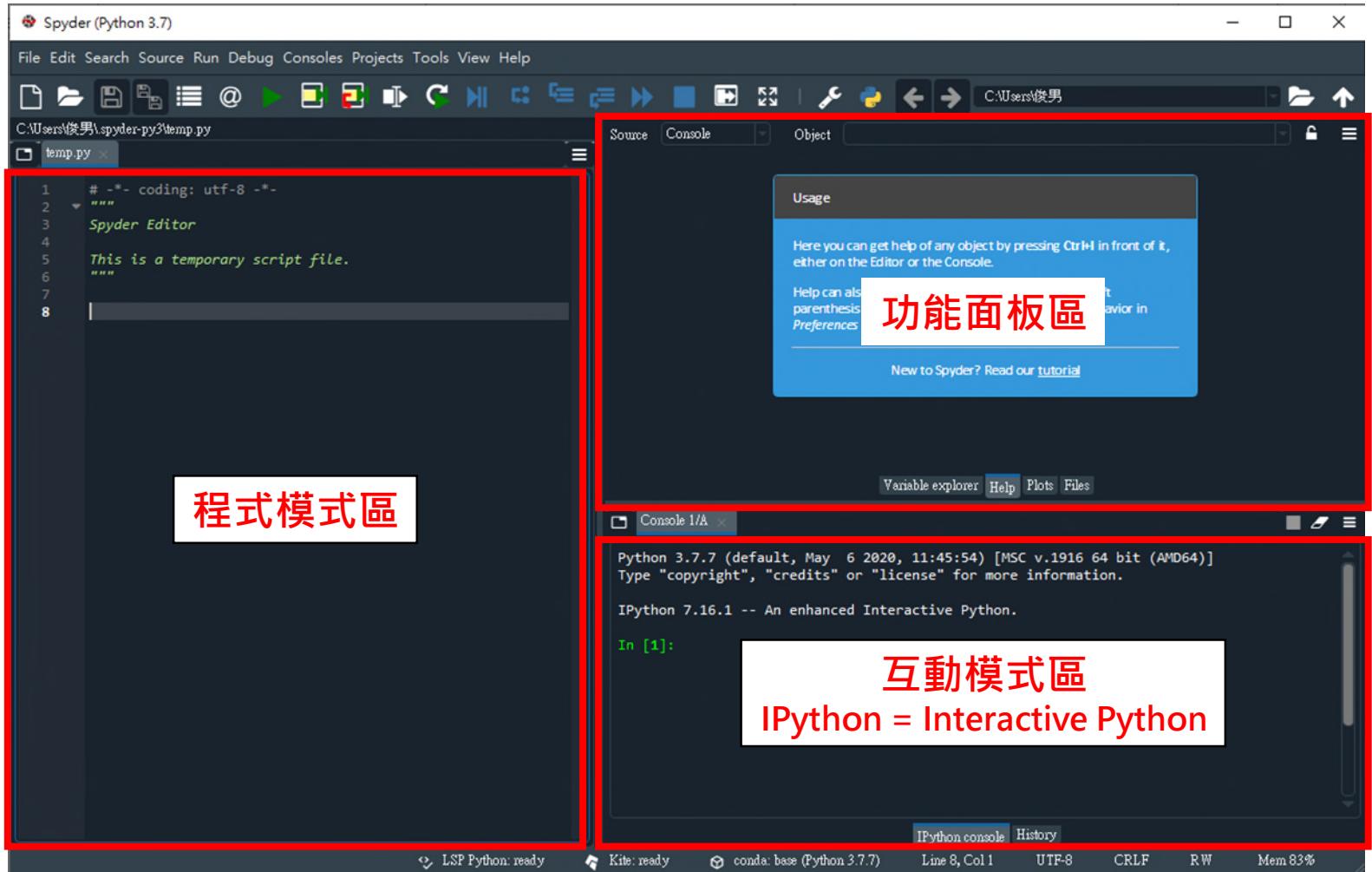


- 請先點擊 Anaconda Navigator，並且從 Anaconda Navigator，啟動 Spyder 看看。
- 關掉 Spyder。這次請從「開始功能表」，直接點擊 Spyder 的圖示，來啟動 Spyder 看看。





Spyder 開發環境介紹





互動模式區 (IPython)

In [1]: 第一筆輸入

In [2]: 第二筆輸入

In [3]: & Out[3]:
第三筆輸入 & 輸出

```
Console 1/A
```

```
Python 3.7.7 (default, May 6 2020, 11:45:54) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]: x = 3
In [2]: y = 5
In [3]: x + y
Out[3]: 8
In [4]: |
```

The screenshot shows an IPython console window titled "Console 1/A". It displays the Python version and license information, followed by three input lines: "In [1]: x = 3", "In [2]: y = 5", and "In [3]: x + y". The result of the third input, "Out[3]: 8", is shown in red. The bottom of the window has tabs for "IPython console" and "History".





隨堂練習：使用互動模式



- 請開啟 Spyder
- 請在 IPython 區域，練習下列程式碼，以熟悉「互動模式」的用法：

```
Python 3.7.7 (default, May  6 2020, 11:45:54) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.16.1 -- An enhanced Interactive Python.

In [1]: x = 3
In [2]: y = 5
In [3]: x + y
Out[3]: 8
```





程式模式區

2 產生新檔案

3 撰寫程式碼

4 存檔 5 執行

1 設定工作路徑

觀看當前工作路徑

首次執行設定

6 觀看執行結果

The screenshot shows the Spyder IDE interface. On the left, there's a code editor with a Python file named 'temp.py' containing the following code:

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jul 6 13:18:18 2020
@author: 後男
"""

x = 3
y = 5
print(x + y)
```

Below the code editor is a 'Run settings for FirstTest.py' dialog box. It has several sections:

- Console**:
 - Execute in current console
 - Execute in a dedicated console
 - Execute in an external system terminal
- General settings**:
 - Remove all variables before execution
 - Directly enter debugging when errors appear
 - Command line options: []
- Working Directory settings**:
 - The directory of the file being executed
 - The current working directory
 - The following directory: []
- External system terminal**:
 - Interact with the Python console after execution
 - Command line options: []
- Always show this dialog on a first file run

At the bottom right of the dialog is a 'Run' button.

To the right of the code editor is a 'Files' tab in the navigation bar. Below it is a 'Console 1/A' window showing the output of the script execution:

```
In [1]: x = 3
In [2]: y = 5
In [3]: x + y
Out[3]: 8

In [4]: runfile('C:/Users/後男/OneDrive/工作/深度學習課程/DemoCode/Ch02/untitled0.py',
              wdir='C:/Users/後男/OneDrive/工作/深度學習課程/DemoCode/Ch02')
8

In [5]:
```





隨堂練習：使用程式模式



- 請先於 Spyder 設定「工作路徑」
- 點擊「File explorer」頁籤，觀看工作路徑目前有哪些檔案？
- 點擊「New file」鈕，產生一個新檔案
- 撰寫如下的程式碼
- 存檔，命名為「FirstTest.py」
- 執行，接受「首次執行設定」視窗的預設值
- 於 IPython 視窗中觀看執行結果

```
x = 3  
y = 5  
print(x + y)
```



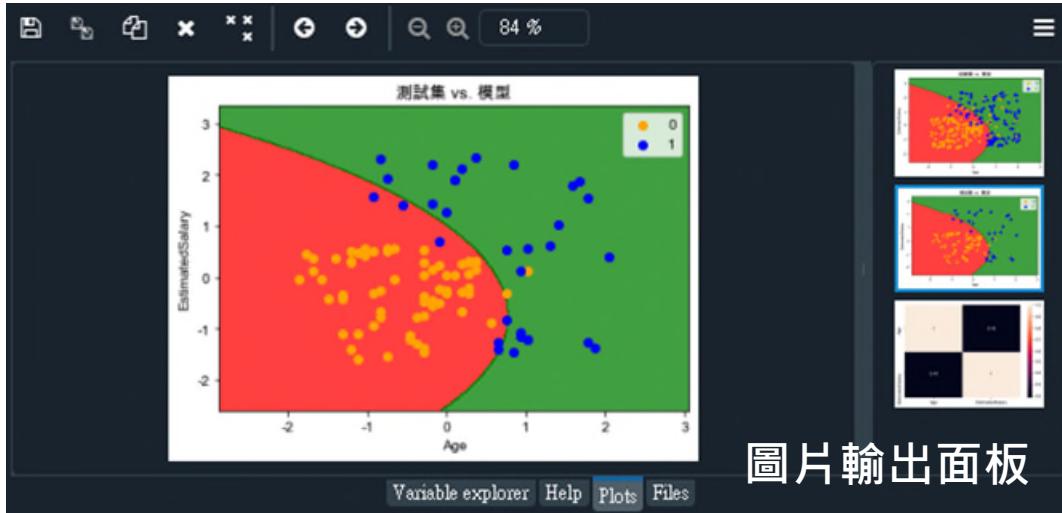


各類面板介紹

Name	Type	Size	Value
x	int	1	3
y	int	1	5

Variable explorer Help Plots Files

變數觀察面板



Source Editor Object

print

Definition: print(*values: object, sep: optional[Text]=..., end: optional[Text]=..., file: Optional[_Writer]=..., flush: bool=...) -> None

```
print(value, ..., sep='', end='\n', file=sys.stdout, flush=False)
```

Prints the values to a stream, or to sys.stdout by default. Optional keyword arguments: file: a file-like object (stream); defaults to the current sys.stdout. sep: string inserted between values; default a space. end: string appended after the last value; default a newline. flush: whether to forcibly flush the stream.

Variable explorer Help Plots Files

求助訊息面板

Name	Size	Type	Date Modified
Diabetes.csv	23 KB	csv File	2019/8/3 下午 10:34
Homework-Diabetes.py	2 KB	py File	2019/8/10 上午 10:15
NaiveBayes.py	3 KB	py File	2020/5/6 下午 04:28
Social_Network_Ads.csv	10 KB	csv File	2019/7/16 下午 09:54

Variable explorer Help Plots Files

檔案管理面板





隨堂練習：求助面板 & 變數觀察面板



- 「**求助面板**」練習
 - 請將游標停駐在「print」指令上
 - 按下 Ctrl-I
 - 瀏覽「求助面板」上的資訊
- 「**變數觀察面板**」練習
 - 請點擊「變數觀察面板」
 - 瀏覽面板上的變數 & 變數值





執行「選取區域」程式碼



The screenshot shows the Spyder Python 3.7 IDE interface. In the code editor (temp.py), lines 8 and 9 are selected: `x = 30` and `y = 50`. A red arrow points from the text "1 x=30; y=50" to these lines. A red circle labeled "3 Run Selection (F9)" highlights the run button in the toolbar. The variable explorer shows variables `x` and `y`. A red box highlights the row for `y`, with a red arrow pointing to the text "只有 y 被改變" (Only y is changed). In the IPython console, the command `In [8]: y = 50` is entered, followed by `...: print(x + y)`, resulting in output `53`. A red circle labeled "4 只有框住的那兩行被執行" (Only the selected two lines are executed) highlights the console output.

1 x=30; y=50

2 只框住這兩行

3 Run Selection (F9)

只有 y 被改變

4 只有框住的那兩行被執行

好用之處：
1. 測試部分程式碼
2. 花大把時間跑出來的程式碼不用再跑

```
In [7]: runfile('C:/Users/後男/OneDrive/工作/深度學習課程/DemoCode/Ch02/untitled0.py', wdir='C:/Users/後男/OneDrive/工作/深度學習課程/DemoCode/Ch02')
Reloaded modules: robert.preprocessor, robert.classification, robert.performance, robert.model_drawer, robert.criteria
8
In [8]: y = 50
...: print(x + y)
53
In [9]:
```





隨堂練習：執行選取區塊



- 先將前一個練習的 x 與 y ，做如下修改：
 - $x = 3 \rightarrow 30$
 - $y = 5 \rightarrow 50$
- 只選取下列兩行程式碼
 - $y = 50$
 - `print(x + y)`
- 按下「Run Selection」鈕，或是 F9
- 觀察 IPython 視窗，看看是否只執行了選中的程式碼
- 再觀察「變數面板」，看看是否只有 y 被修改成 50？





執行「游標停駐區塊」程式碼



1 程式碼以
In[1], #In[2]...
隔出「程式區塊」

其實不用編號，
寫 #In[] 就可以。

好用之處：
1. 分段除錯時，
很好用！

3 Run Current Cell (Ctrl-Enter)

2 游標停在某區塊上

可以用「變數面板」
檢查 x, y 的現值

4 只有該區塊被執行

The screenshot shows the Spyder Python 3.7 IDE interface. The code editor displays a script named 'temp.py' with the following content:

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jul  6 13:18:18 2020
@author: 後男
"""

# In[1]
x = 30
y = 50

# In[2]
print(x + y)
```

Line 10 is highlighted with a red arrow and labeled '2 游標停在某區塊上'. The variable explorer panel on the right shows the values of 'x' and 'y':

Name	Type	Size	Value
x	int	1	30
y	int	1	50

A red box highlights this table and an arrow points from it to the text '可以用『變數面板』檢查 x, y 的現值'. The IPython console at the bottom shows the execution of the code. The command 'In [9]: runcell([1], 'C:/Users/後男/OneDrive/工作/深度學習課程/DemoCode/Ch02/untitled0.py')' is highlighted with a red box and labeled '4 只有該區塊被執行'. The status bar at the bottom indicates 'LSP Python: ready', 'Kite: ready (no index)', 'conda: base (Python 3.7.7)', and memory usage 'Mem 76%'. A red circle with the number '3' is positioned above the title 'Run Current Cell (Ctrl-Enter)'.



隨堂練習：執行程式區塊



- 請先以 `# In[]` 語法，標好程式區塊如下。
- 點擊「Run Current Cell」或按下「Ctrl-Enter」
- 觀察是否只執行了第一區塊？
- 點擊「Run Current Cell & Go Next」，或按下「Shift-Enter」
- 觀察是否執行了第一區塊後，自動跳往第二區塊？

```
8 # In[1]
9 x = 30
10 y = 50
11
12 # In[2]
13 print(x + y)
```





Anaconda
除錯的方法

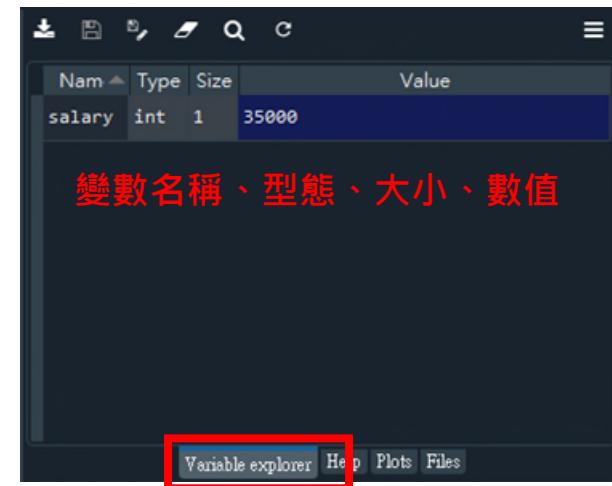


除錯機制



中斷點 (單擊) →

```
8 ● salary = eval(input("請輸入您的月薪："))
9                                     ↴ 邏輯錯誤之處
10    if salary <= 0:               (因為 >= )
11        print("您的薪資是：{}".format(salary))
12    else:
13        print("薪資不可以是負數！請重新輸入！")
```



變數觀察面板



隨堂練習：除錯機制



- 請先開啟一個新檔案「DebugTest.py」。
- 撰寫好如下的程式碼。
- 單擊左側的行號，製造中斷點。
- 點擊「Debug File」鈕（或 Ctrl-F5），執行至第一個中斷點停下。
- 點擊「Run Current Line」鈕（或 Ctrl-F10），一行行單步追蹤。
- 觀察右上角的「Variable Explorer」面板，監視變數值是否與你預期的相同？
- 找出錯誤點，並修改它。

```
8 salary = eval(input("請輸入您的月薪："))
9
10 if salary <= 0:
11     print("您的薪資是：{}".format(salary))
12 else:
13     print("薪資不可以是負數！請重新輸入！")
```

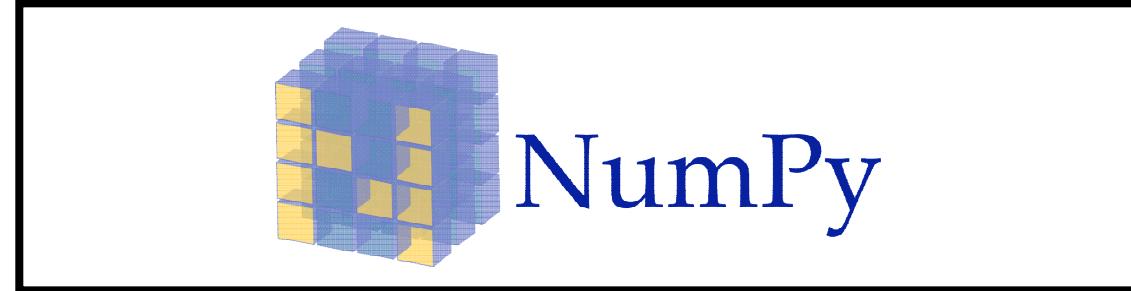




外掛套件的 安裝方法



機器學習常用的「外掛套件」



- **NumPy**
 - 陣列處理函式庫
- **pandas**
 - 外部資料讀取 & 處理函式庫
- **SciPy**
 - 數學、科學處理函式庫
 - 如：微積分、線性代數...
- **Matplotlib**
 - 圖形繪製函式庫
- 其它函式庫
 - **SciKit-Learn**：機器學習演算法
 - **TensorFlow**、**Keras**：深度學習演算法

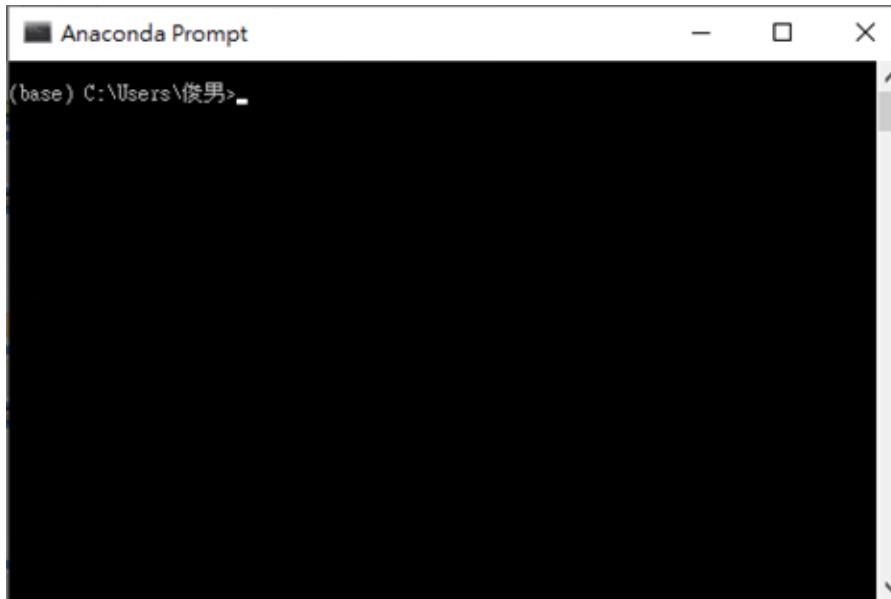




如何得知已安裝了哪些外掛套件



- 打開 Anaconda Prompt



已安裝套件

- 輸入「`conda list`」命令

Name	Version	Build	Channel
ipyw_jlab_nb_ext_conf	0.1.0		py37_0
alabaster	0.7.12		py37_0
anaconda	2019.03		py37_0
anaconda-client	1.7.2		py37_0
anaconda-navigator	1.9.7		py37_0
anaconda-project	0.8.2		py37_0
asn1crypto	0.24.0		py37_0
astroid	2.2.5		py37_0
astropy	3.1.2		py37he774522_0
atomicwrites	1.3.0		py37_1
attrs	19.1.0		py37_1
babel	2.6.0		py37_0
backcall	0.1.0		py37_0
backports	1.0		py37_1
backports.os	0.1.1		py37_0
backports.shutil_get_terminal_size	1.0.0		py37_2
beautifulsoup4	4.7.1		py37_1
bitarray	0.8.3		py37hfa6e2cd_0

conda list

套餐名稱 套件版本 Python 版本

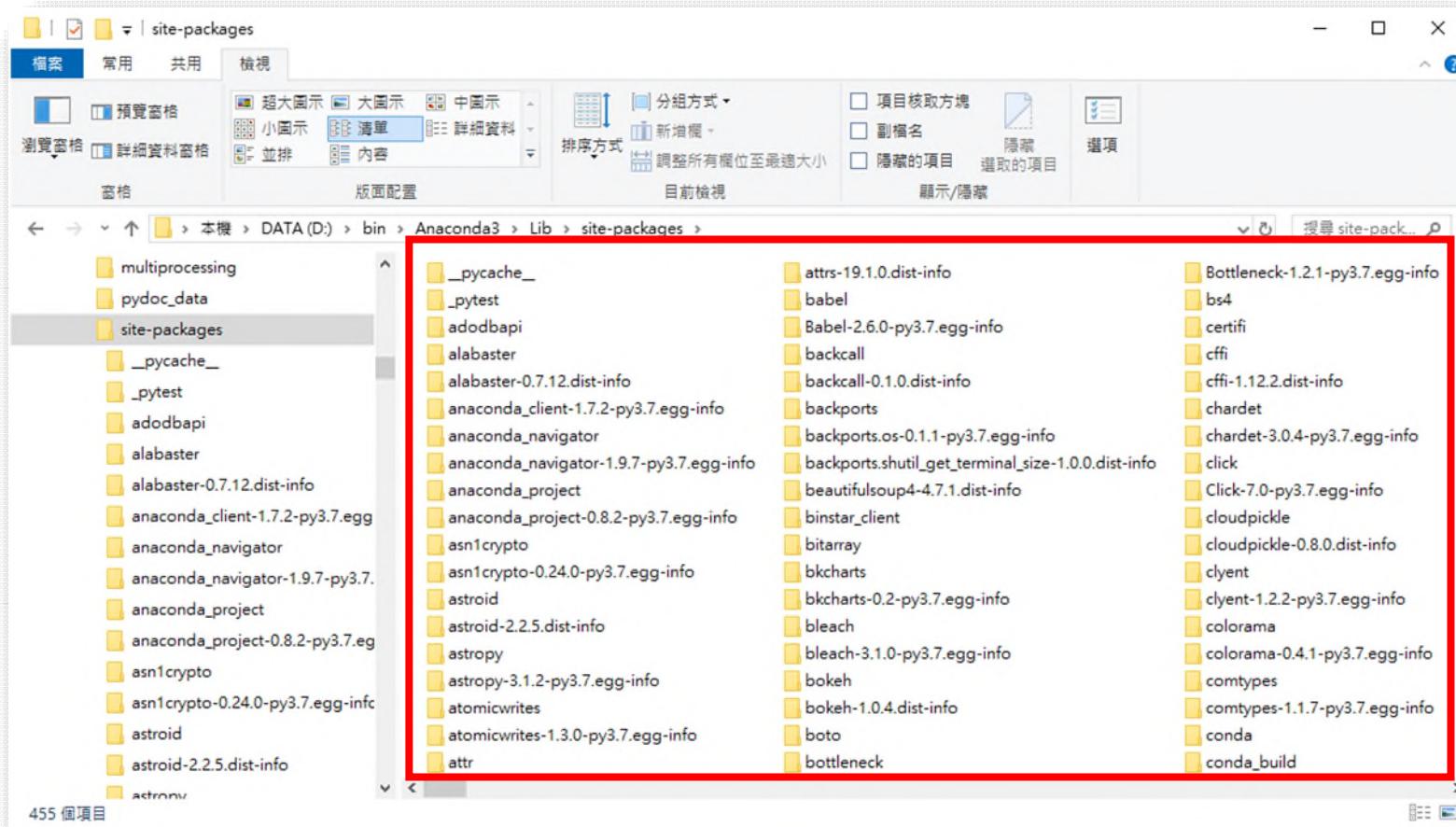




外掛套件存放在哪裡？



- <Anaconda安裝路徑>/Lib/site-packages/





隨堂練習：找尋已安裝套件



- 用 conda list 找尋已安裝套件
 - 打開 Anaconda Prompt
 - 輸入「`conda list`」
 - 依序找尋有無「matplotlib」、「numpy」、「pandas」、「scipy」
- 打開檔案總管
 - 前進到「<Anaconda 安裝路徑>/Lib/site-packages/」資料夾下
 - 依序找尋有無「matplotlib」、「numpy」、「pandas」、「scipy」





安裝、升級、移除外掛套件



安裝

```
conda install <外掛套件名稱>
```

升級

```
conda update <外掛套件名稱>
```

--all : 升級所有已安裝的外掛套件

移除

```
conda remove <外掛套件名稱>
```





隨堂練習：安裝、升級、移除套件

- 以「**scrapy**」這個「網路爬蟲」專用套件為例
- 請用下列指令，試著安裝「**scrapy**」套件：
 - `conda install scrapy`
- 請用下列指令，確認 **scrapy** 已經安裝成功了：
 - `conda list | findstr "scrapy"`
- 請用下列指令，練習更新 **scrapy** 套件：
 - `conda update scrapy`
- 請用下列指令，移除 **scrapy** 套件：
 - `conda remove scrapy`





本章總結



- 知道安裝 Anaconda 的方法。
- 了解 Spyder **程式區**、**面板區**、**互動區**各區塊任務。
- 會用**求助面板**、**變數觀察面板**、以及**檔案面板**。
- 能以「**選取區塊**」、「**程式區塊**」，執行部分程式。
- 會用 Spyder 除錯。
- 知道 **NumPy**、**pandas**、**SciPy**、**Matplotlib** 這些常用函式庫的作用。
- 能查閱、安裝、升級、移除各種外掛套件。





課後作業：安裝、熟悉開發環境



- 請在你家裡的機器上，**安裝** Anaconda。
- 寫一個 Hello! World! 的程式，確定自己熟悉 Spyder 的開發流程。
- 依照本章講義，練習「選取區塊」與「程式區塊」的**部分執行**功能。
- 練習**除錯**功能。
- 用 scrapy 套件，練習如何「**查閱、安裝、移除**」一個套件。
- 本作業自行練習，不計分、不用繳交。

