

AI與程式語言-第3章 Anaconda簡介與安裝

大數據分析

- R/Python/Julia/SQL程式設計與應用
(R/Python/Julia/SQL Programming and Application)
- 資料視覺化 (Data Visualization)
- 機器學習 (Machine Learning)
- 統計品管 (Statistical Quality Control)
- 最佳化 (Optimization)



李明昌 博士

alan9956@gmail.com

<http://rwepa.blogspot.com/>



主題

第1章 課程介紹 (iClass)

第2章 AI與程式語言簡介

第3章 **Anaconda**簡介與安裝

第4章 資料匯入探索

第5章 資料處理探索

第6章 資料視覺化探索

第7章 資料整合應用探索

大綱：

3.1 Python 簡介與安裝

3.2 Anaconda 簡介與安裝

3.3 Spyder 軟體簡介

3.4 變數

3.5 資料型別與運算子

3.6 資料物件

3.7 認識模組

3.1 Python 簡介與安裝

Python 簡介

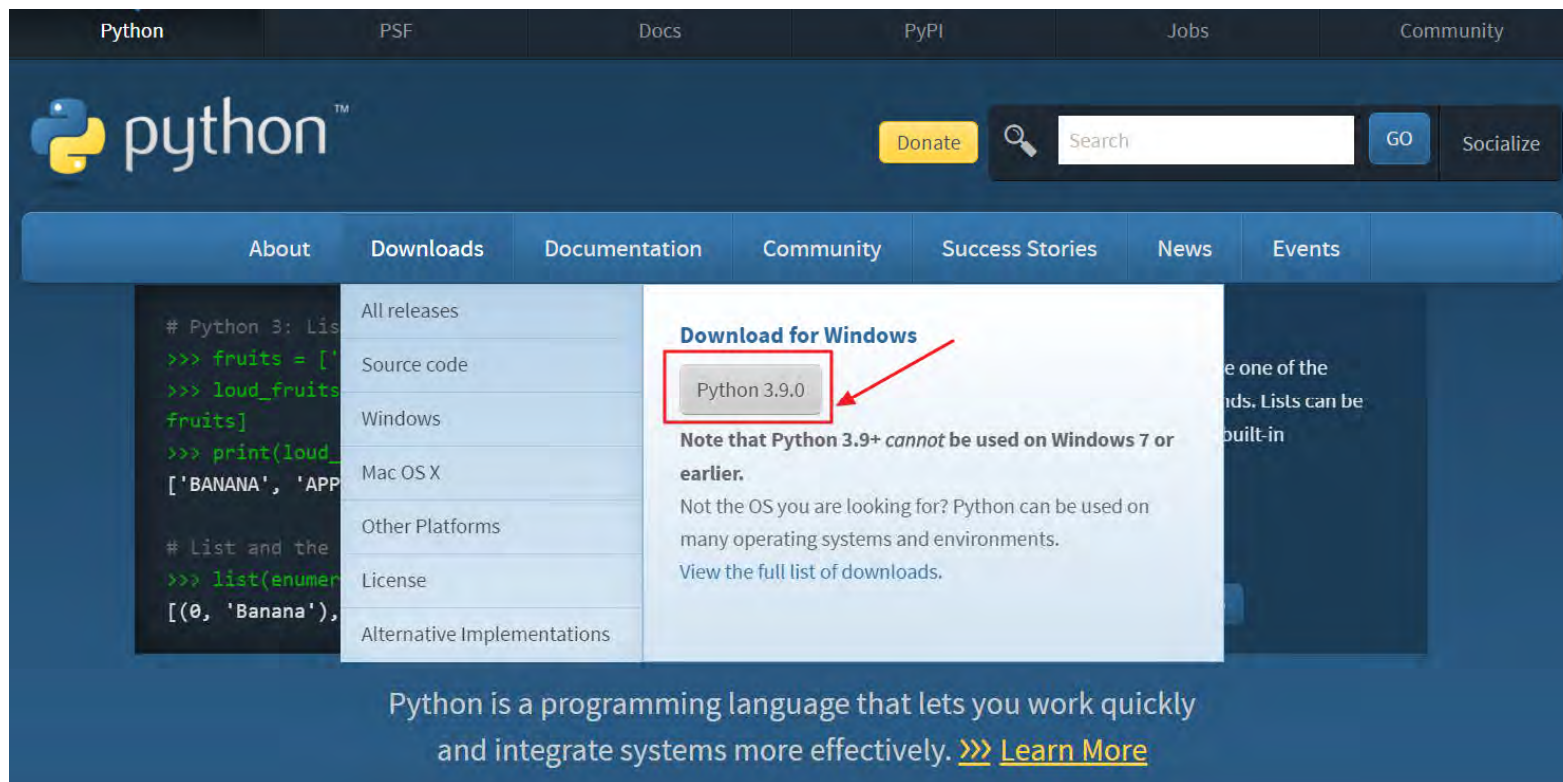
- 吉多·范羅蘇姆 (Guido van Rossum) 在1989年的聖誕節期間研發 Python 語言。
 - https://en.wikipedia.org/wiki/Guido_van_Rossum
 - Python 3.10.6
- 特性:
 - 跨平台
 - 開放性
 - 易讀性
 - 動態語言
 - 直譯語言
 - 豐富套件(模組)
 - 其他語言結合, 例: Cython 編譯成執行檔(.exe)

Python 下載

2022.9.6 最新版 3.10.6

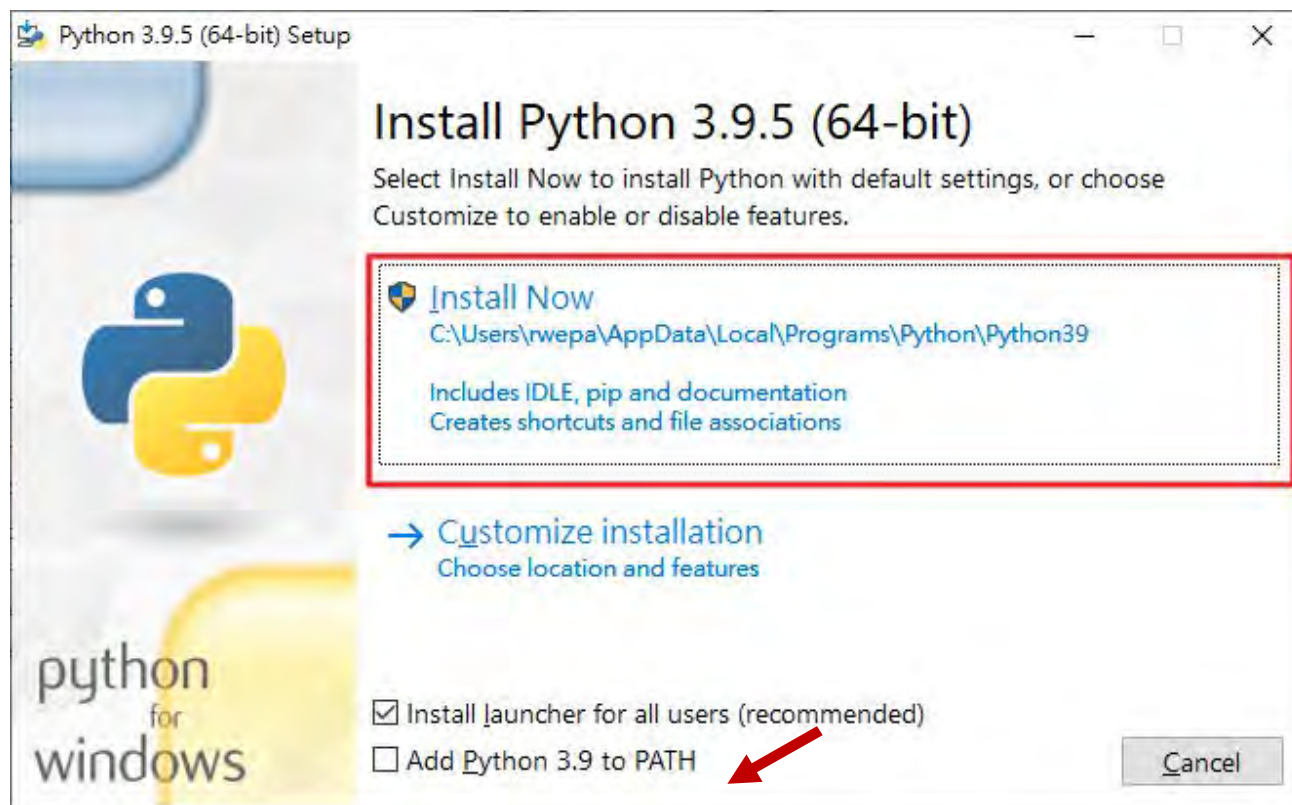
【注意】本課程使用
Anaconda 軟體，已經內含
Python程式，因此**不用下載**
Python 3.10.6

- <https://www.python.org/>

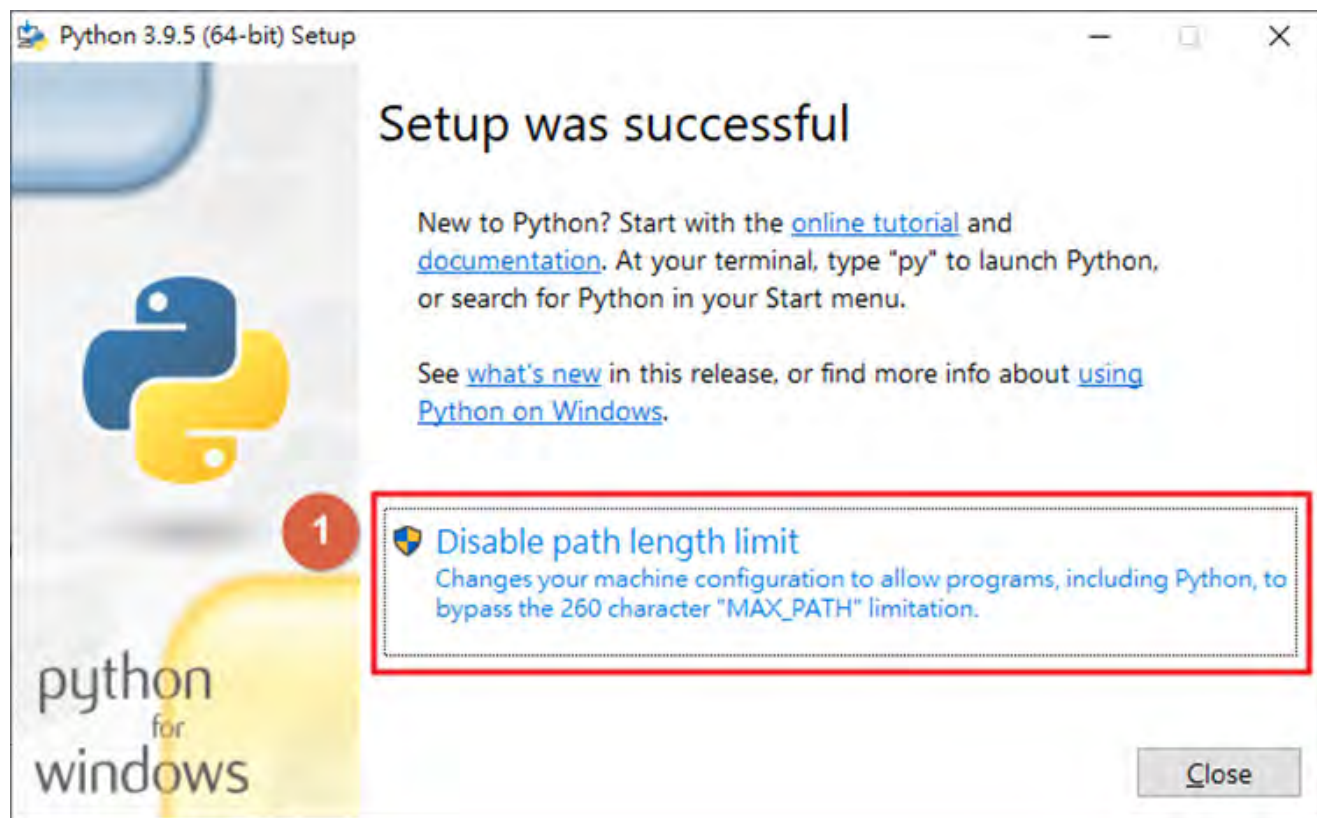


Python 安裝

- python-3.9.5-amd64.exe

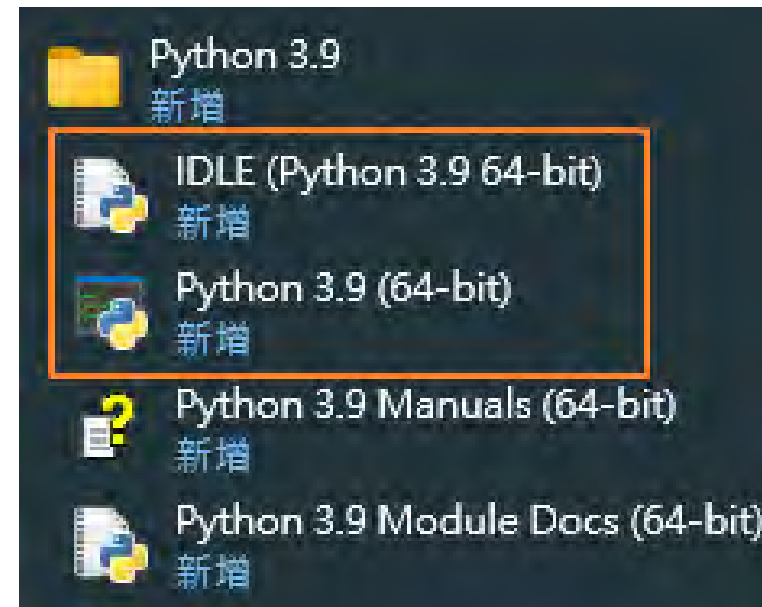
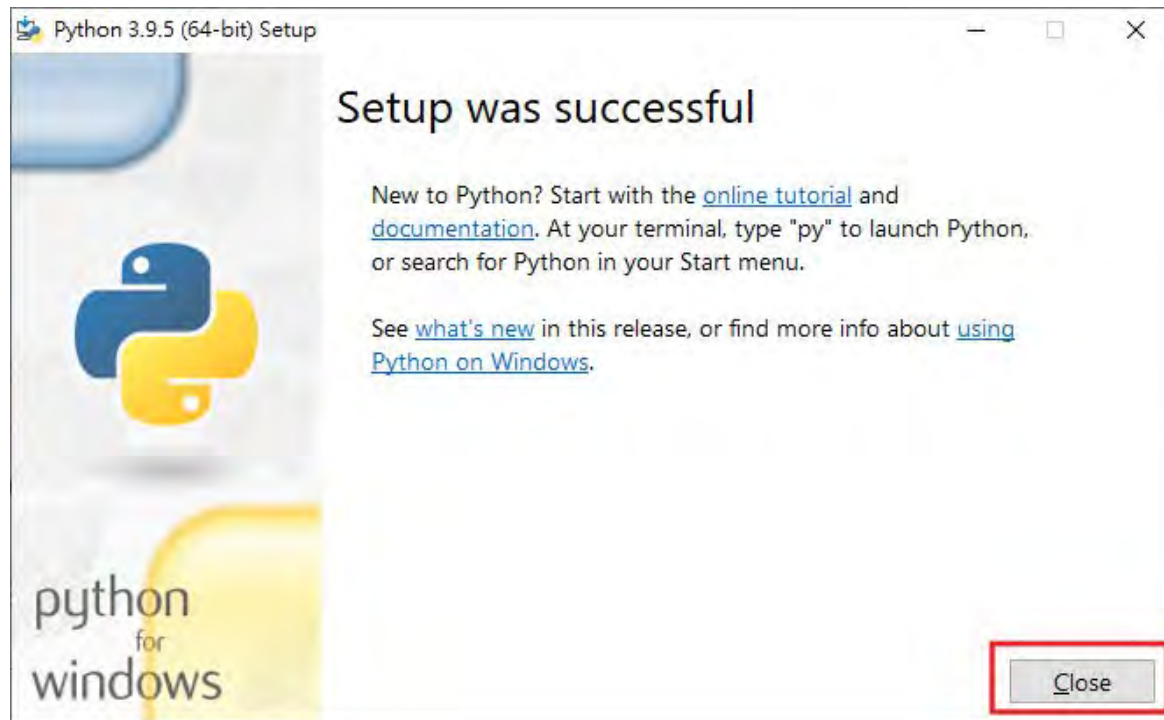


Python 安裝 – 取消字元長度限制





Python 安裝完成





Python 執行 <方法1> Python 3.9(64-bit)

```
Python 3.9 (64-bit)
Python 3.9.5 (tags/v3.9.5:0a7dcdbd, May 3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> help('print')
Help on built-in function print in module builtins:

print(...)
    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.

>>> print('Hello World - RWEPA')
Hello World - RWEPA
>>> quit()
```

Windows 環境變數 Path: 加入
;C:\Windows\System32

- `help(print)`
- `print('Hello world')`
- `quit()`

Python 執行 <方法2> 命令提示字元



```
命令提示字元
Microsoft Windows [版本 10.0.19042.928]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\rwepa>python
Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> quit()

C:\Users\rwepa>_
```

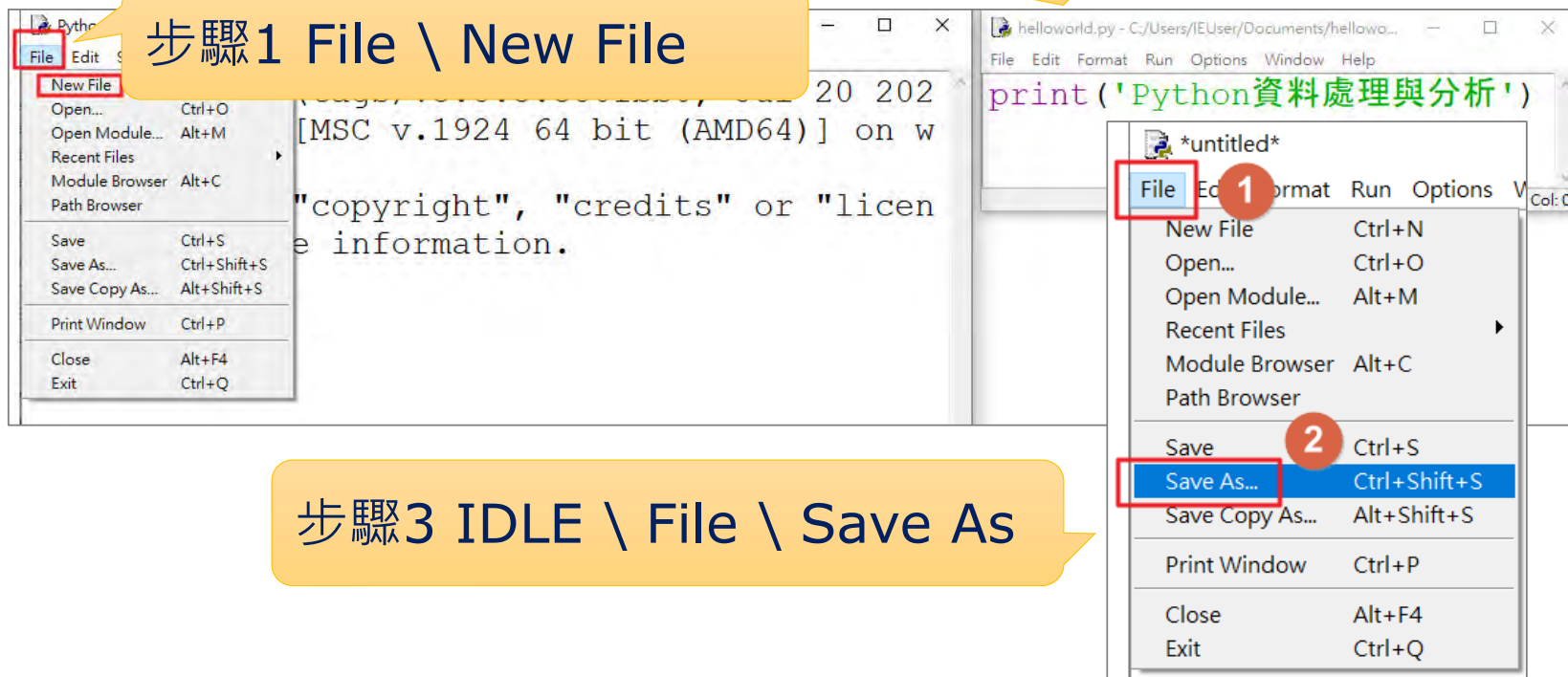
- python
- 1+2
- quit()

Python 執行 <方法3> IDLE模式

- IDLE (Integrated Development and Learning Environment)
可先將檔案編輯與儲存

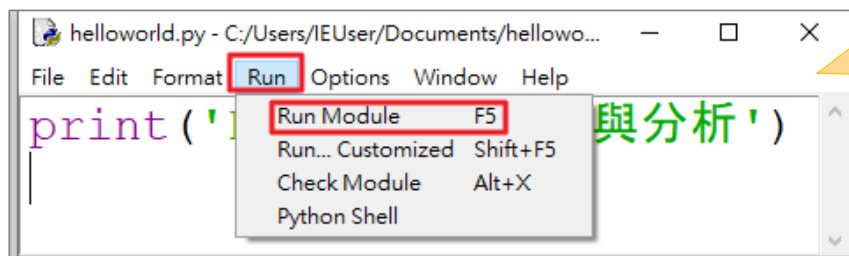
步驟2 IDLE \ 輸入程式碼

步驟1 File \ New File



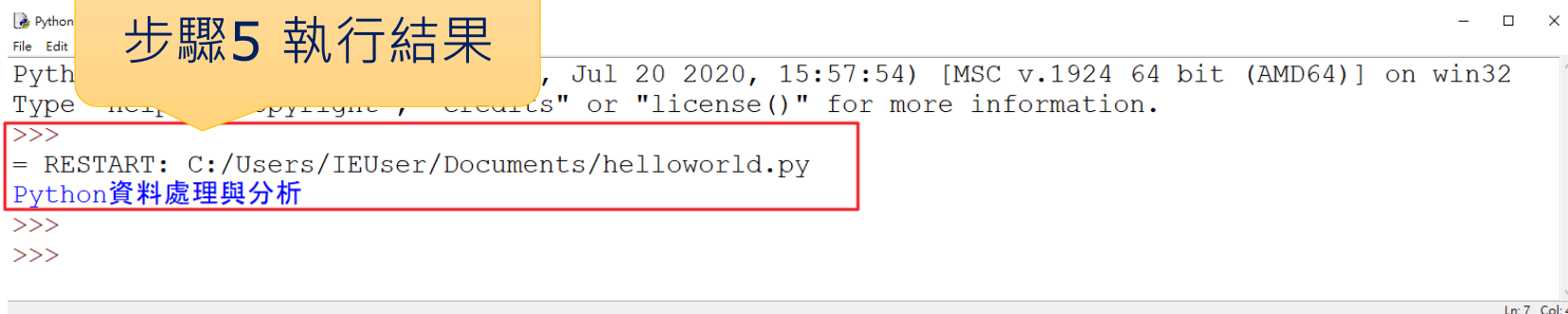


Python 執行 <方法3> IDLE模式 (續)



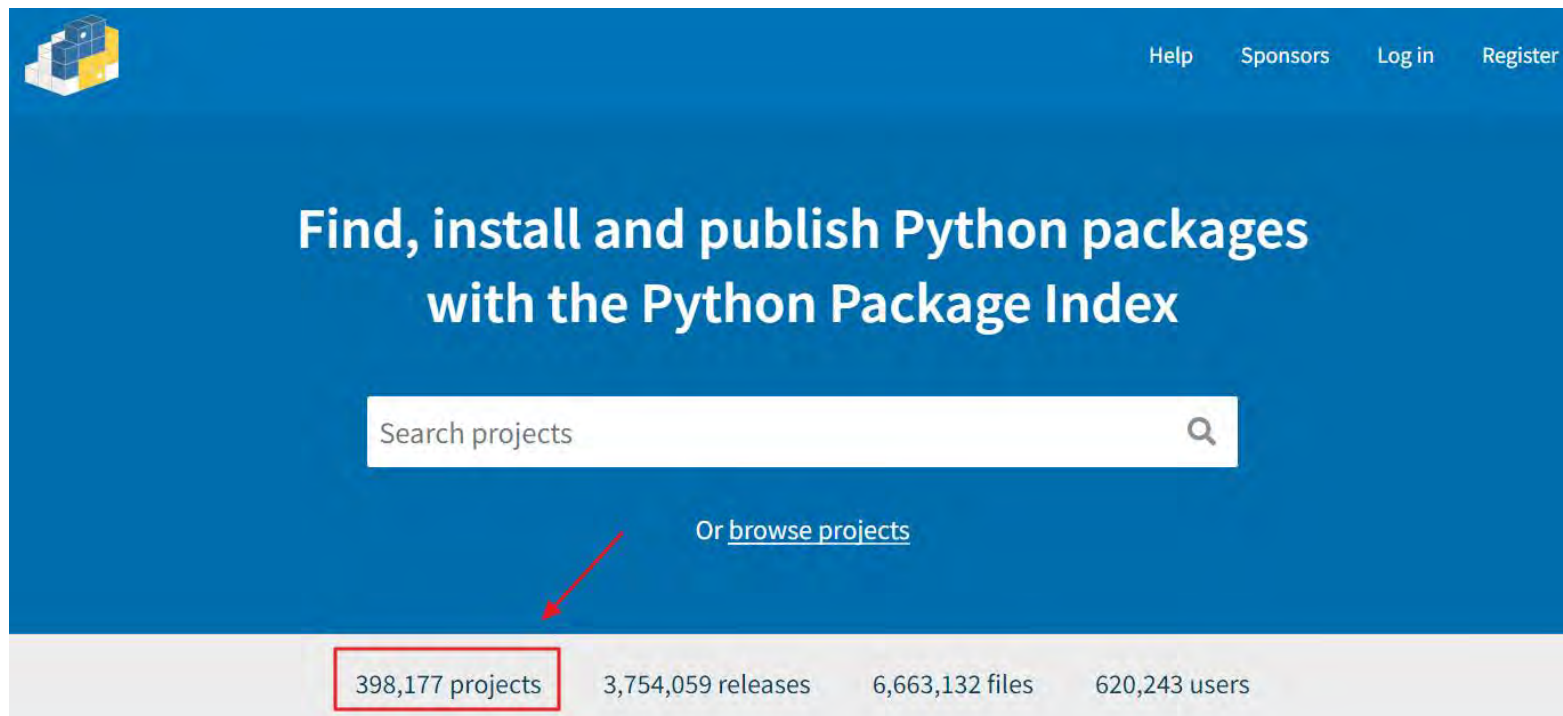
步驟4 Run \ Run Module

步驟5 執行結果



PyPI (Python Package Index)

- <https://pypi.org/>
- 39萬多專案總表



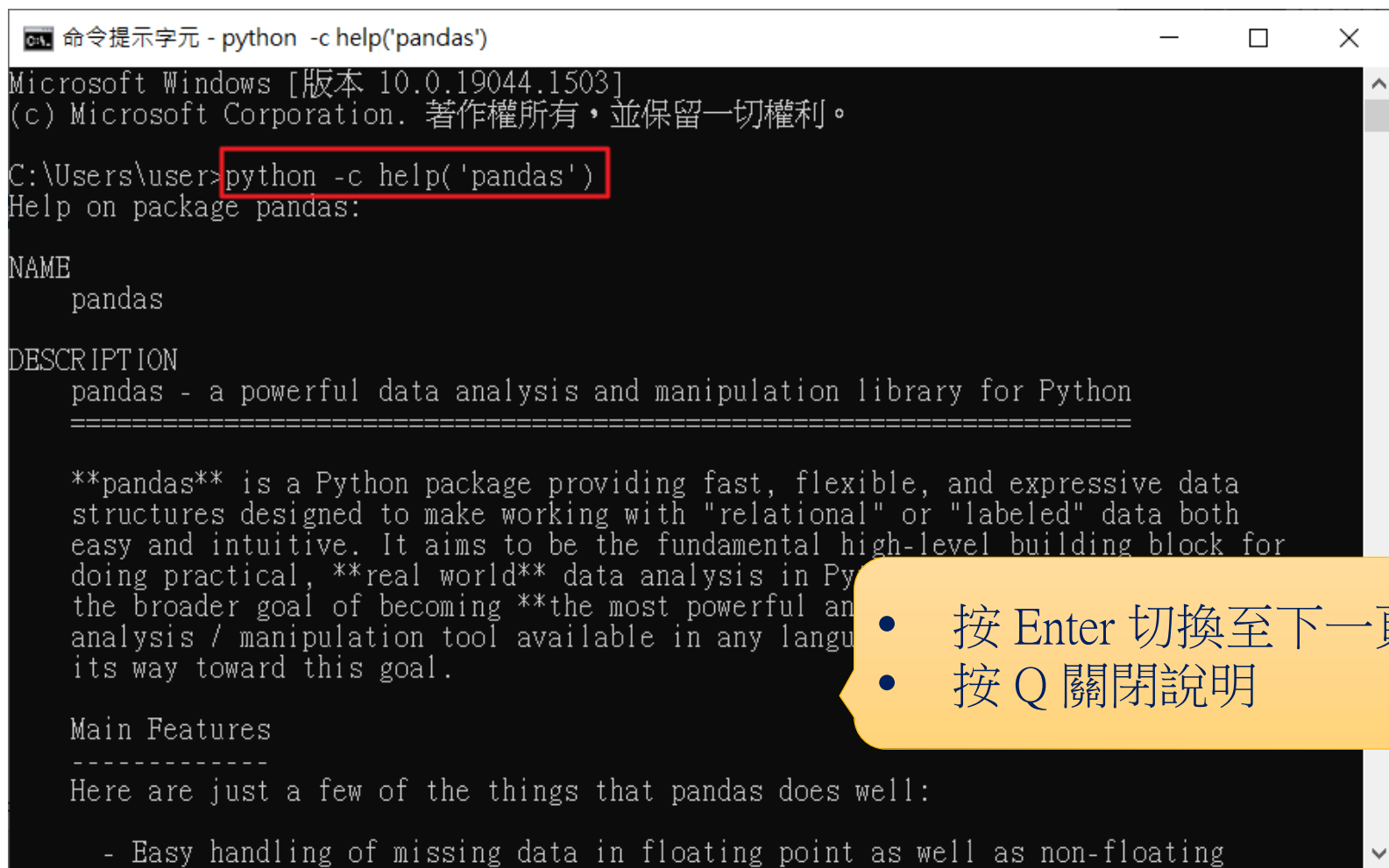
已安裝模組 pip list



```

命令提示字元
Microsoft Windows [版本 10.0.19043.1055]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\88697>pip list
Package                                Version
-----
absl-py                                0.12.0
appdirs                                1.4.4
astunparse                             1.6.3
cachetools                             4.2.1
certifi                                2020.12.5
chardet                                4.0.0
defusedxml                             0.7.1
distlib                                0.3.1
engineering-notation                    0.6.0
    
```


pandas 模組 - 線上說明 `python -c help('pandas')`



```
C:\> 命令提示字元 - python -c help('pandas')
Microsoft Windows [版本 10.0.19044.1503]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\user>python -c help('pandas')
Help on package pandas:

NAME
    pandas

DESCRIPTION
    pandas - a powerful data analysis and manipulation library for Python
    =====

    **pandas** is a Python package providing fast, flexible, and expressive data
    structures designed to make working with "relational" or "labeled" data both
    easy and intuitive. It aims to be the fundamental high-level building block for
    doing practical, **real world** data analysis in Python, with the broader goal
    of becoming **the most powerful and flexible** analysis / manipulation tool
    available in any language for its way toward this goal.

    Main Features
    -----
    Here are just a few of the things that pandas does well:

    - Easy handling of missing data in floating point as well as non-floating
```

- 按 Enter 切換至下一頁
- 按 Q 關閉說明



Python IDE

- Anaconda, 包括 **Spyder**, Jupyter notebook

- <https://www.anaconda.com/>

示範軟體 (Spyder)

- PyCharm:

- <https://www.jetbrains.com/pycharm/>

- WinPython:

- <http://winpython.github.io/>

- RStudio - Terminal 視窗

- <https://www.rstudio.com/products/rstudio/>

- Visual Studio Code

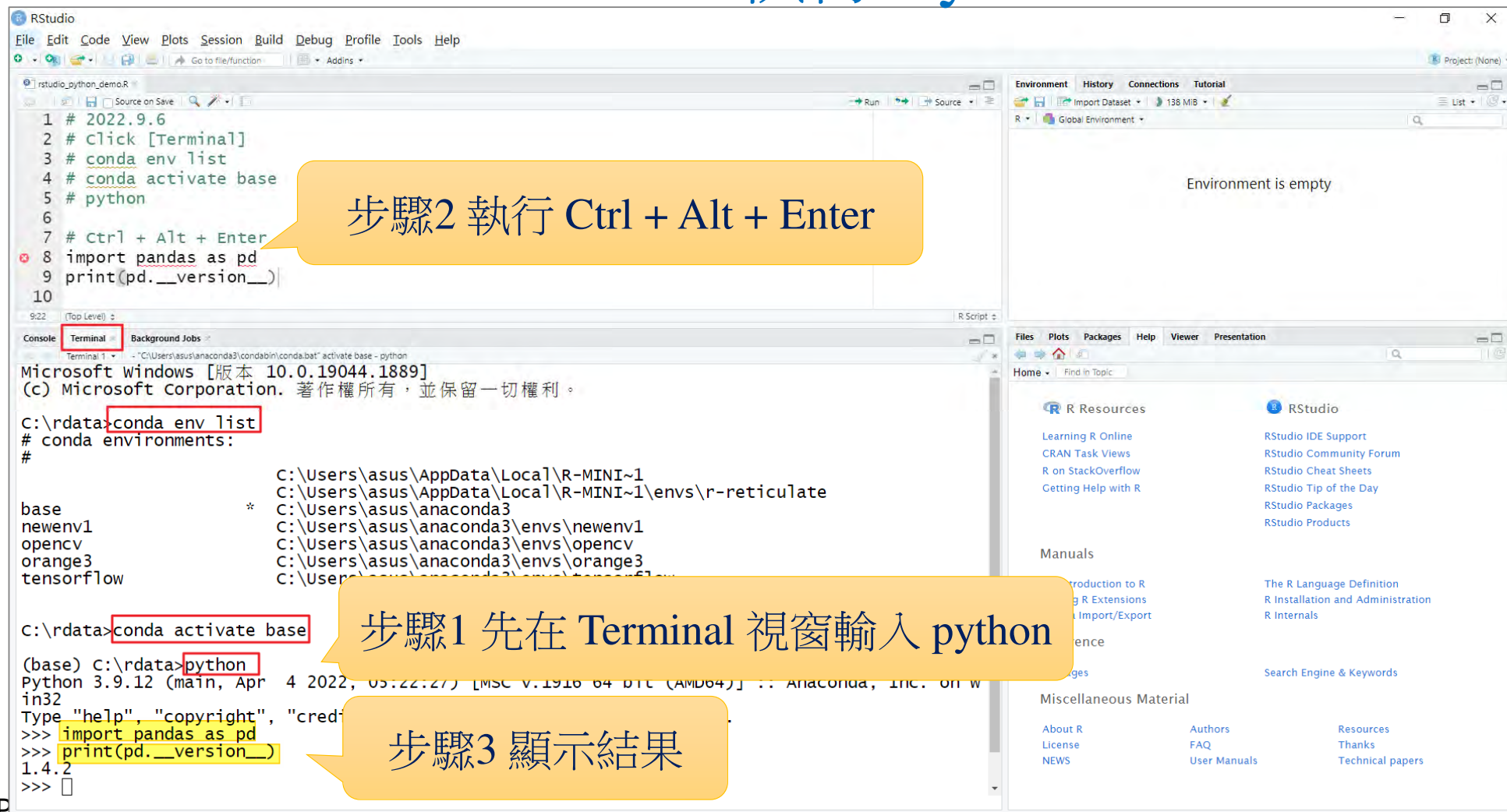
- <https://code.visualstudio.com/docs/python/python-tutorial>

- Google:

-  Google Colaboratory

IDE 整合開發環境-
Integrated
Development
Environment

RStudio – 執行 Python



步驟1 先在 Terminal 視窗輸入 python

```

C:\rdata>conda env list
# conda environments:
#
base                  * C:\Users\asus\AppData\Local\R-MINI~1
newenv1               C:\Users\asus\anaconda3
opencv                 C:\Users\asus\anaconda3\envs\opencv
orange3               C:\Users\asus\anaconda3\envs\orange3
tensorflow             C:\Users\asus\anaconda3\envs\tensorflow

C:\rdata>conda activate base
(base) C:\rdata>python
Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64-bit AMD64] :: Anaconda, INC. ON W
in32
Type "help", "copyright", "credits() or "help()" to get more help.
>>> import pandas as pd
>>> print(pd.__version__)
1.4.2
>>>
    
```

步驟2 執行 Ctrl + Alt + Enter

步驟3 顯示結果

3.2 Anaconda 簡介與安裝



Anaconda 特性

- Anaconda是一個免費、易於安裝/管理並支援Python語言
- 支援7500個以上的資料科學模組 (package)
- 支援模組的下載, 安裝, 更新
- 支援 Spyder (支援 Python IDE)
- 支援 jupyter notebook
- 支援機器學習模組 (scikit-learn)
- 支援深度學習模組 (Tensorflow, Keras)
- 支援 Windows、Mac OS X和Linux 跨平台

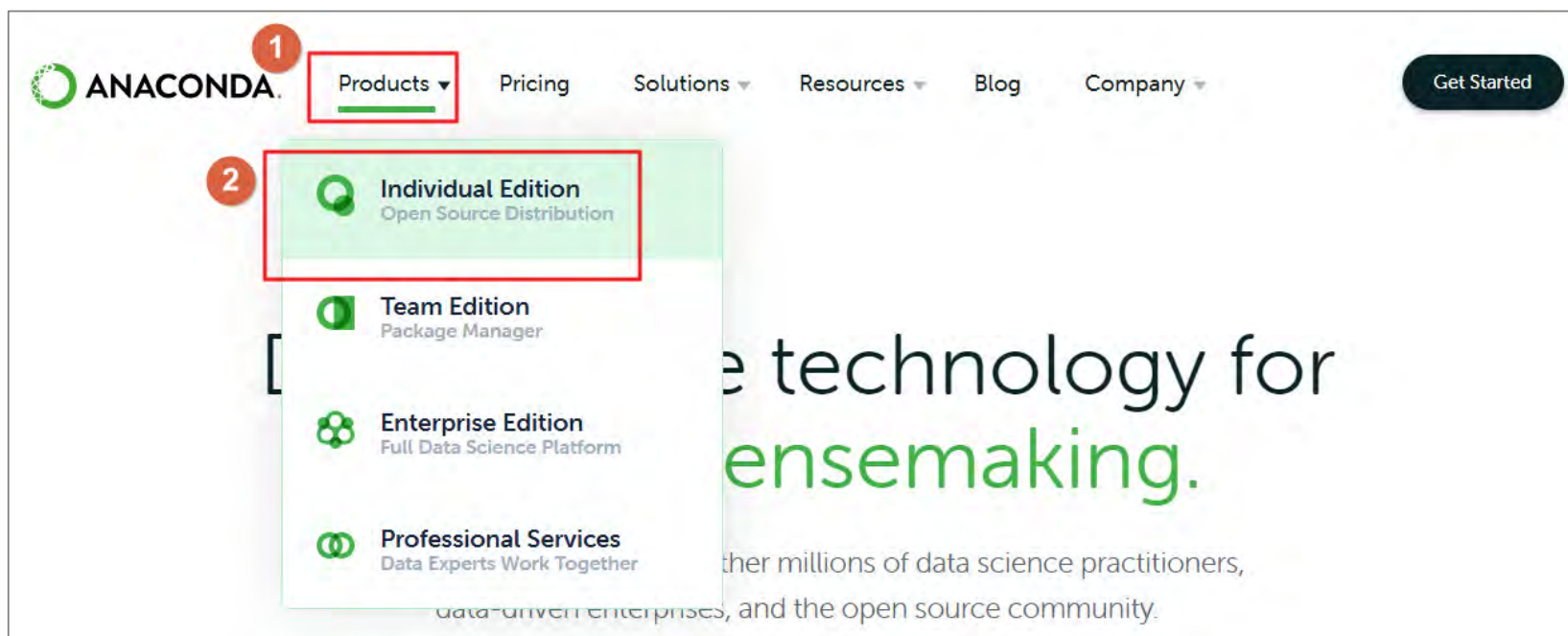
Anaconda 特性 (續)



source: <https://www.anaconda.com/distribution/>

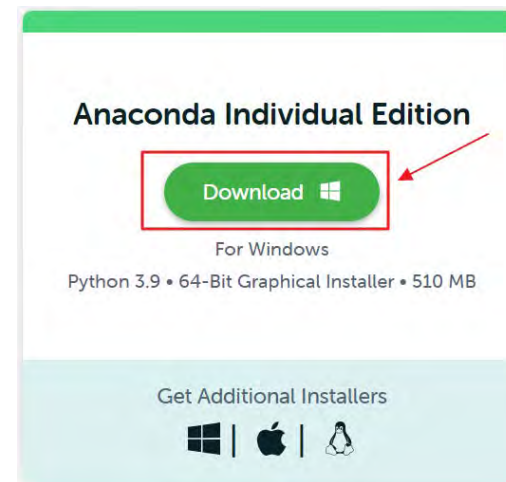
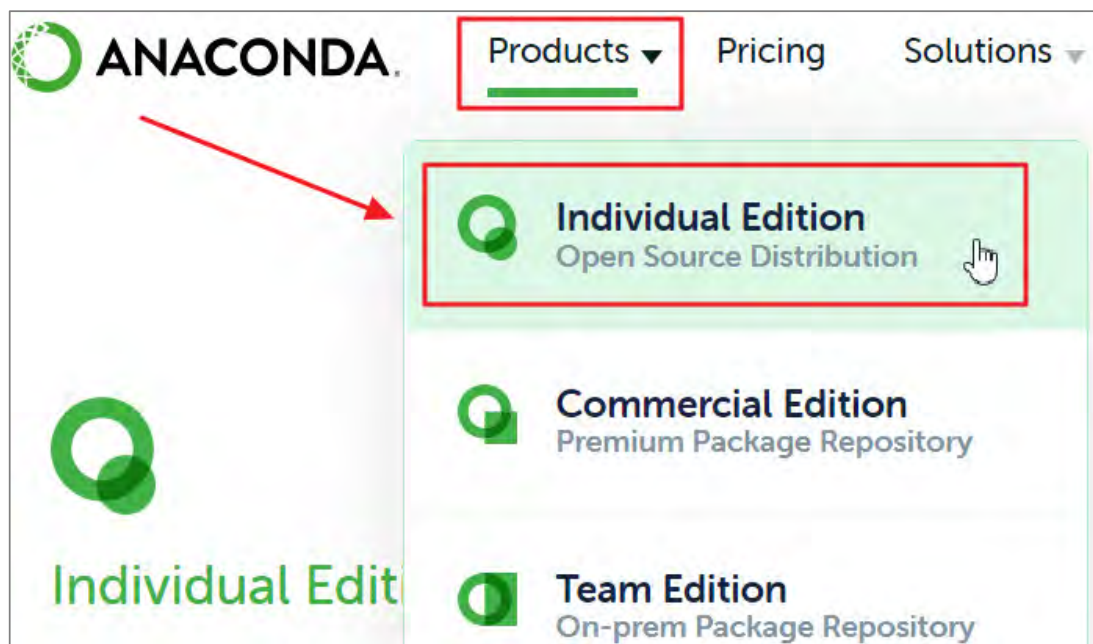
Anaconda 下載

- <https://www.anaconda.com/>





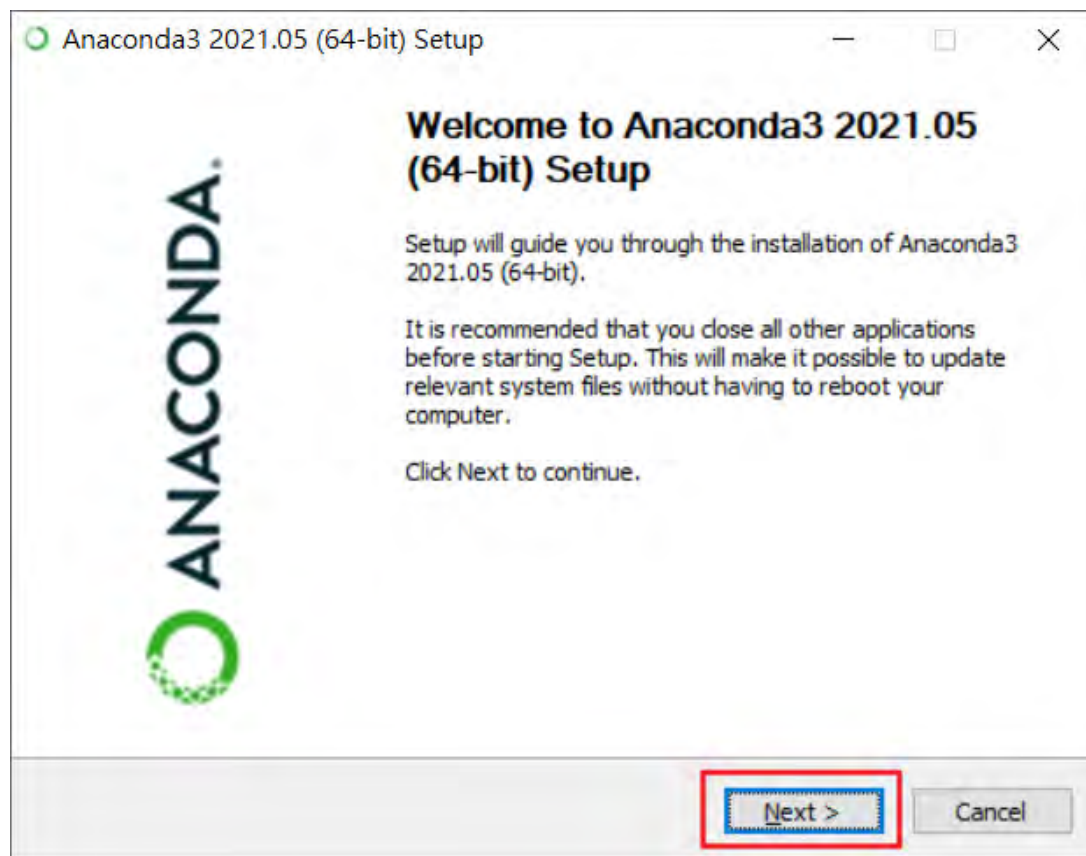
Anaconda 下載 (續)



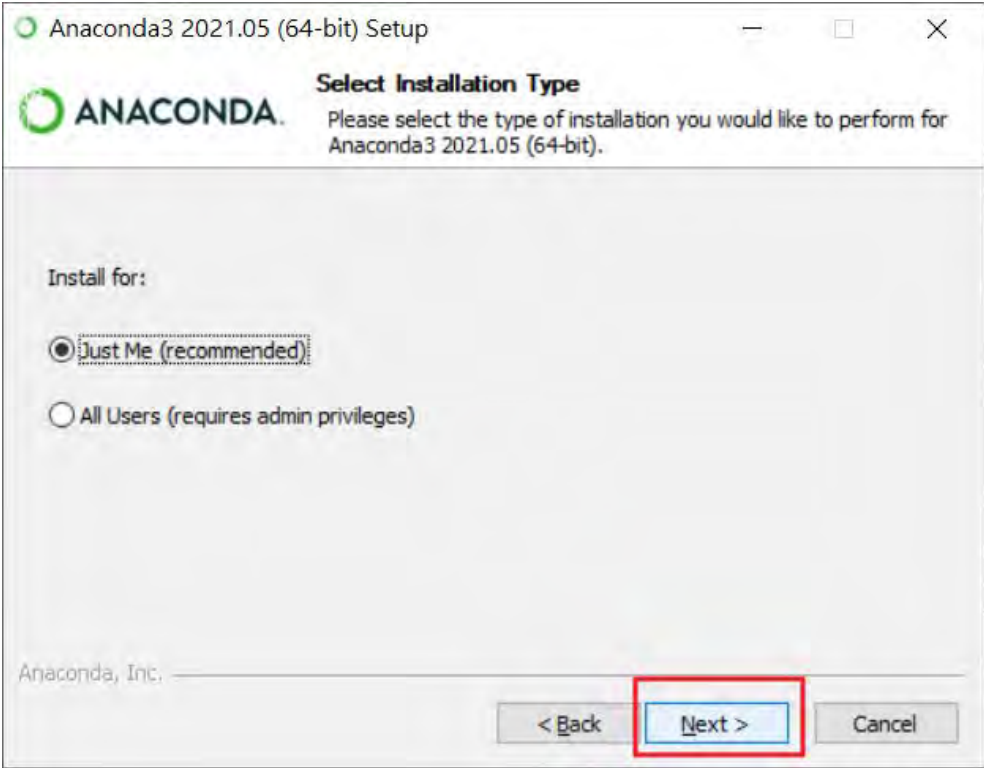
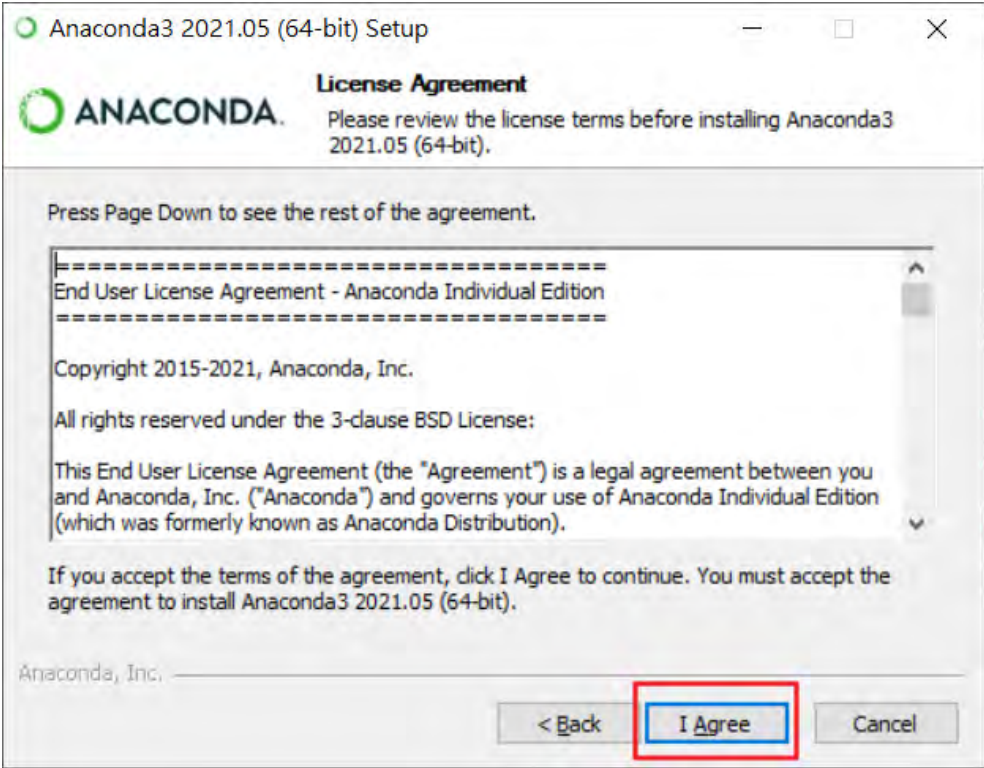
Anaconda3-2021.11-Windows-x86_64.exe

下載 64位元,
510 MB

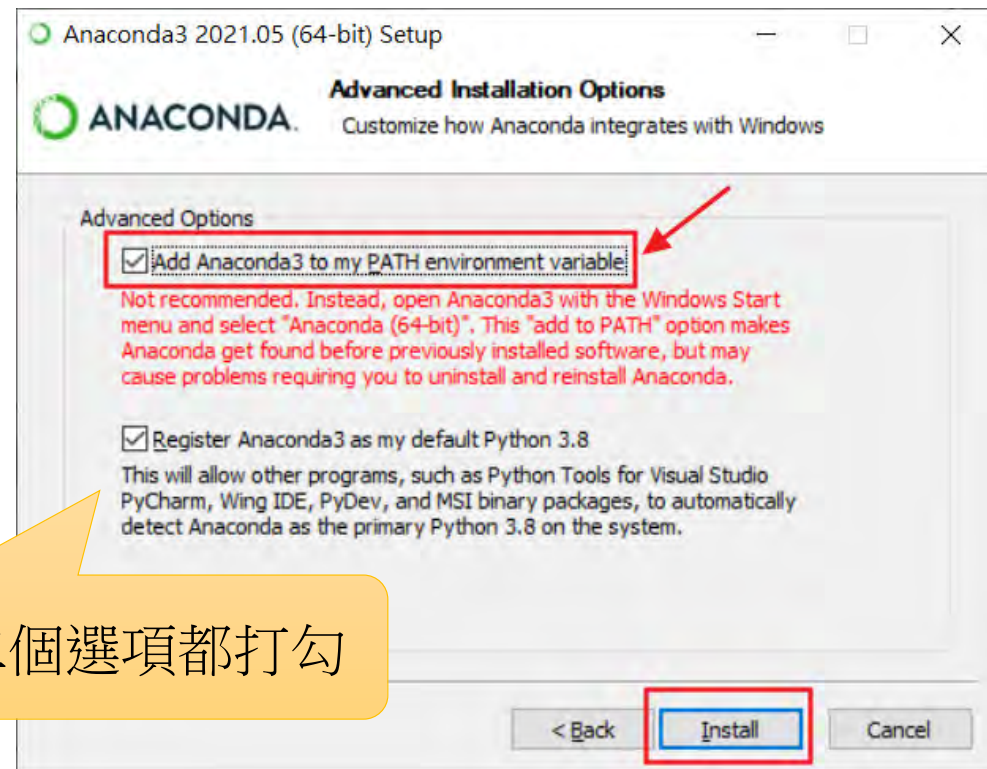
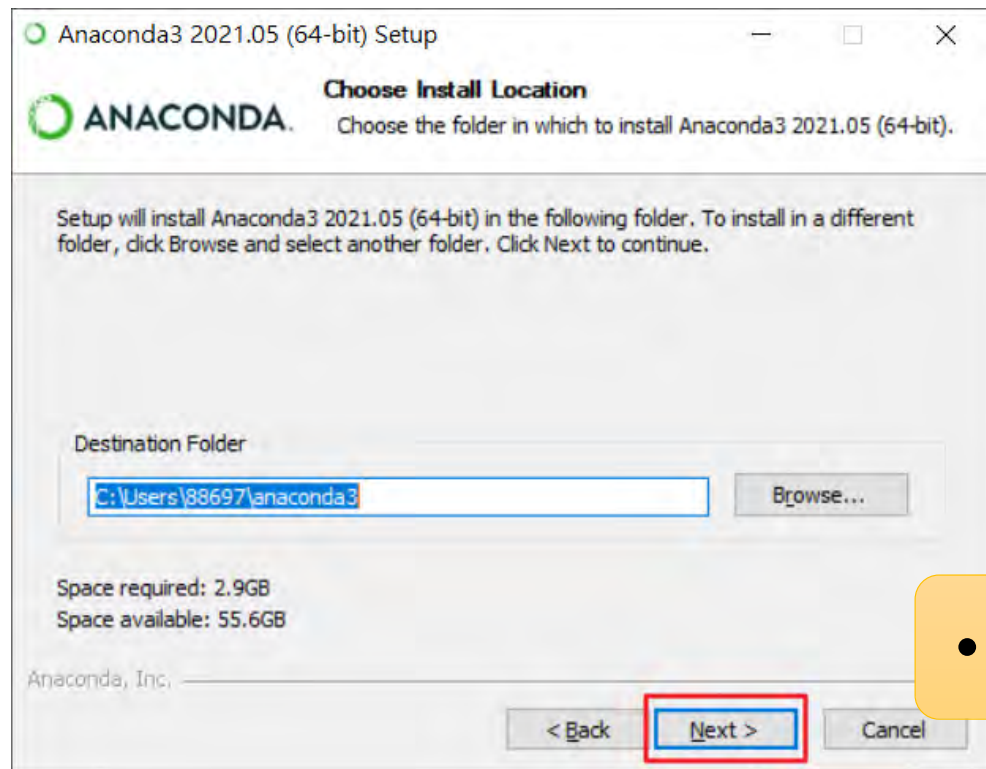
安裝 Anaconda



安裝畫面

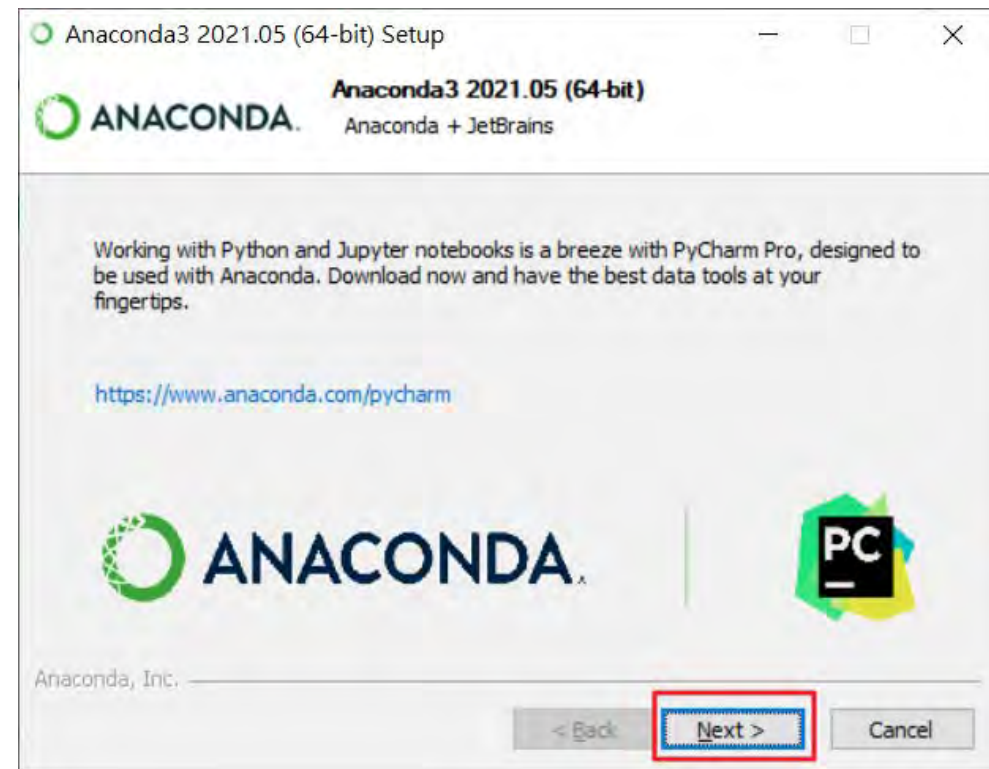
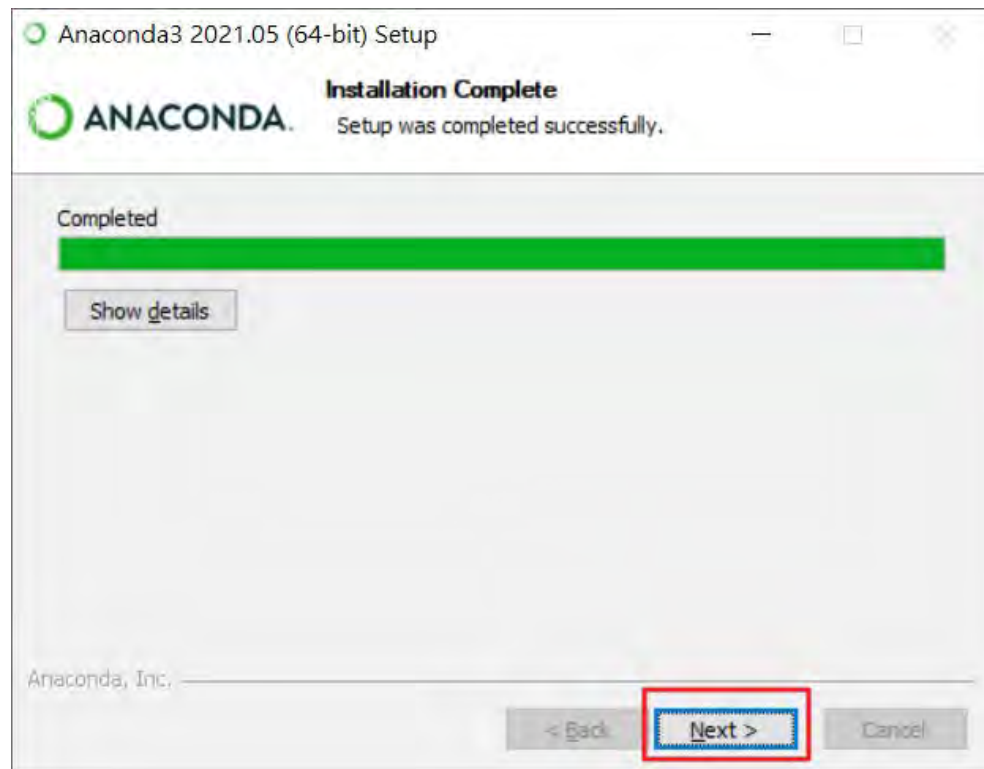


二個選項都打勾

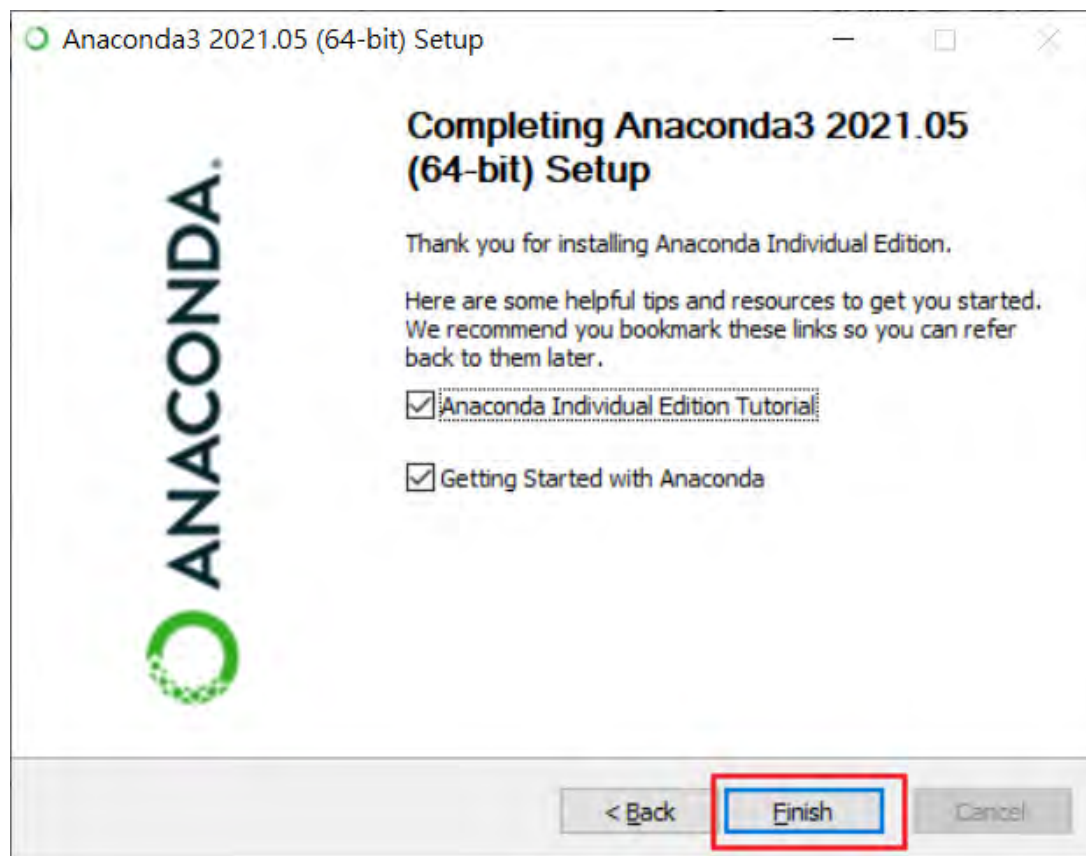


- 二個選項都打勾

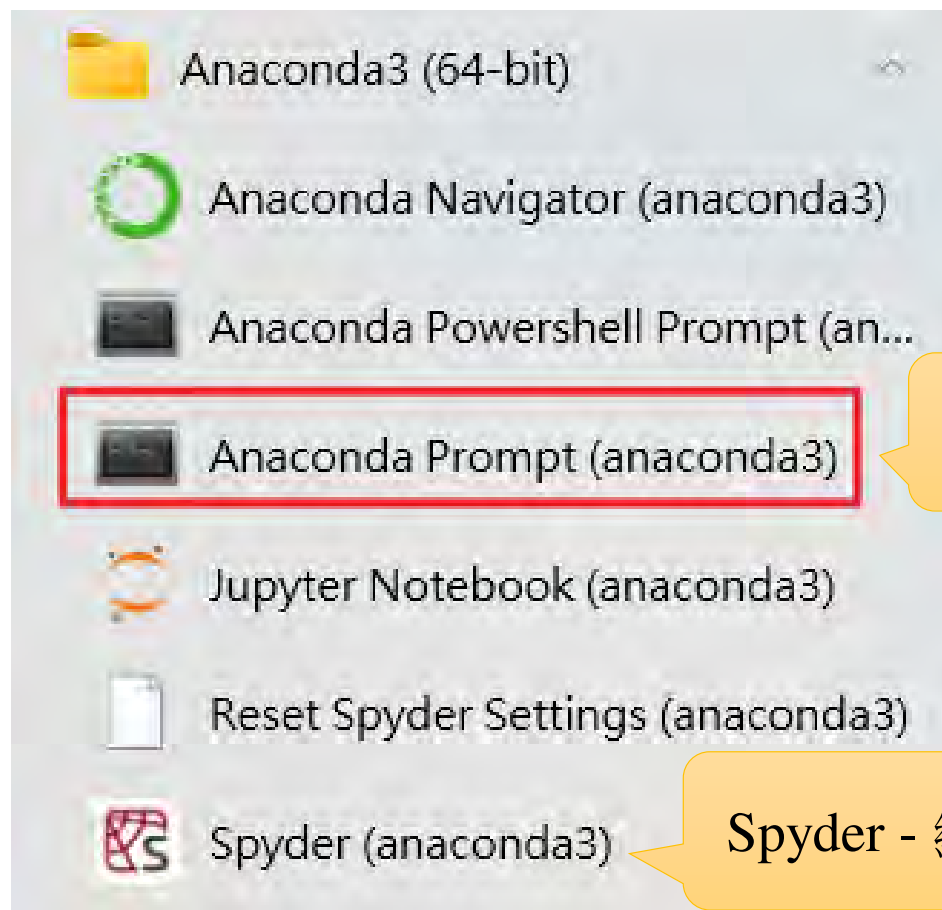
安裝畫面



Anaconda 安裝完成



Anaconda3 (64-bit) - 已安裝6個元件



安裝模組

Spyder - 編輯Python 程式

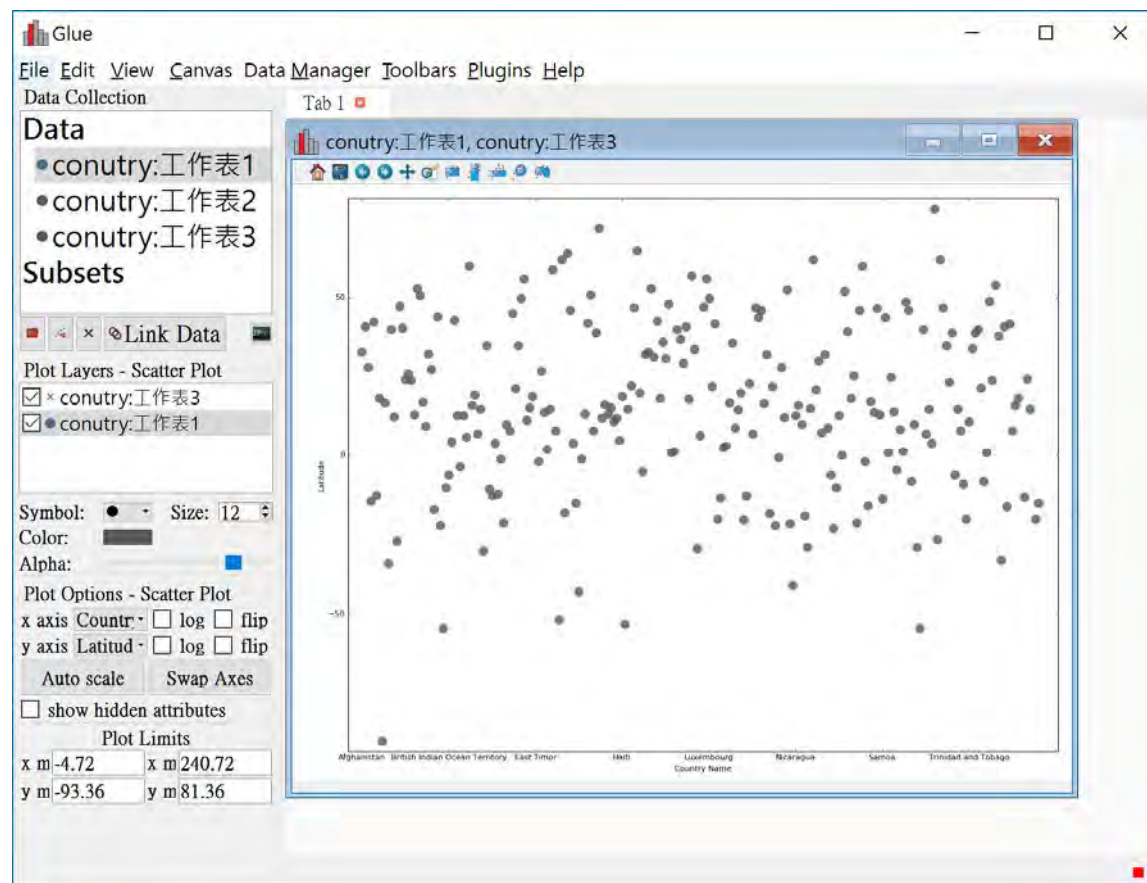
Anaconda Navigator

不同電腦, 安裝模組可能有差異

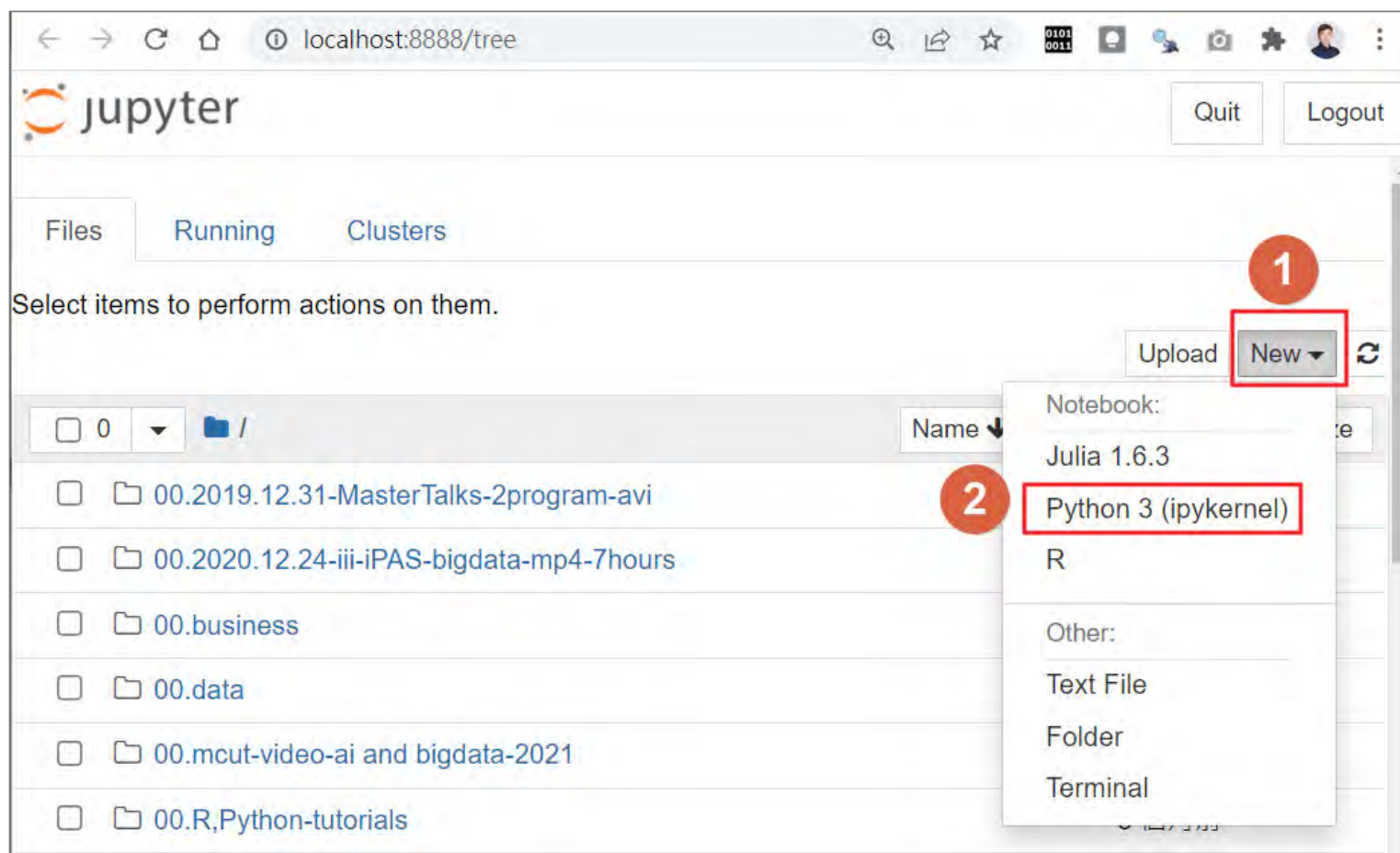
注意: 不要在此安裝 RStudio, 可能會有問題?

- Launch 可直接啟動
- Install 須安裝

glueviz 視覺化



jupyter notebook

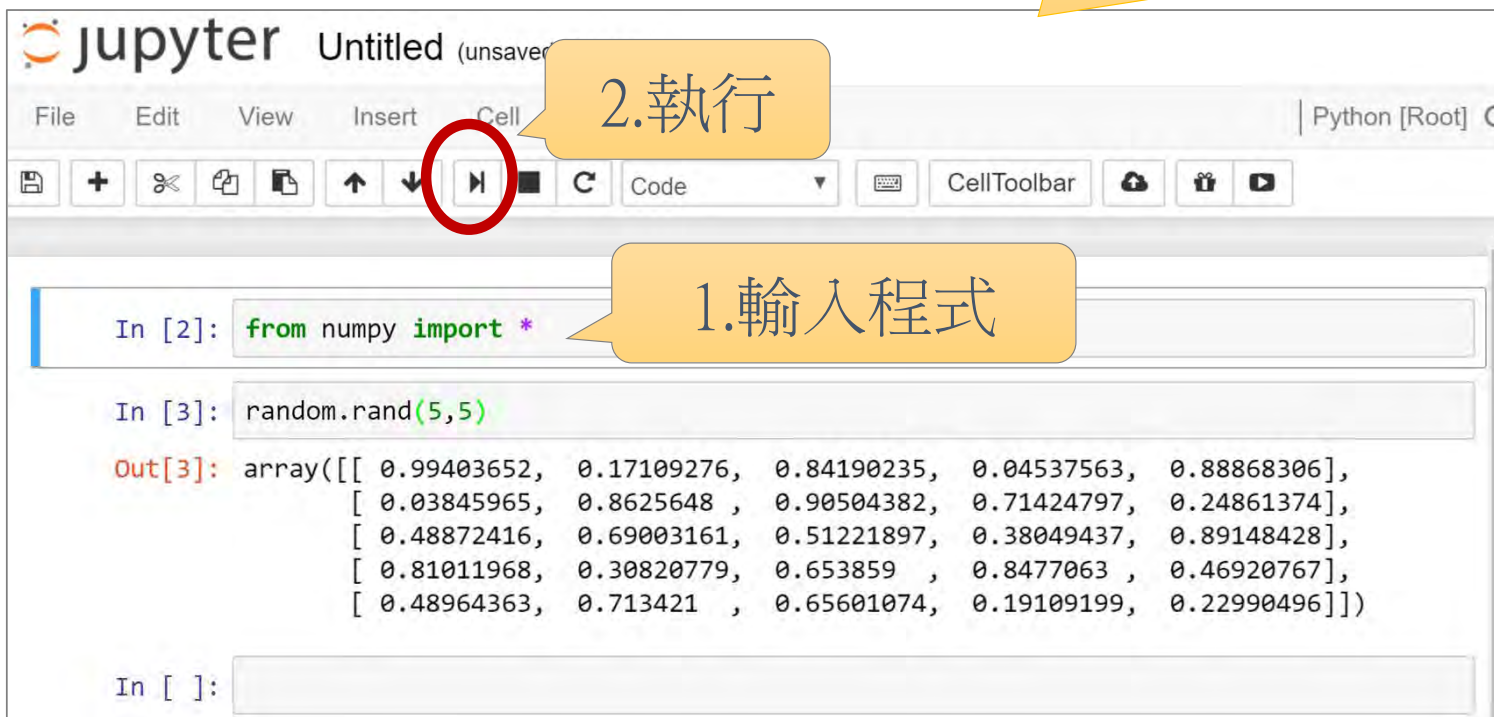




jupyter notebook (續)

理解重新命名方式

實作練習1



The screenshot shows the Jupyter Notebook interface. The title bar says "jupyter Untitled (unsaved)". The menu bar includes "File", "Edit", "View", "Insert", and "Cell". The toolbar contains icons for saving, adding, deleting, and running cells. A red circle highlights the "Run" button (a play icon). Below the toolbar, there are three input cells. The first cell contains the code `from numpy import *` and is annotated with "1.輸入程式". The second cell contains the code `random.rand(5,5)` and is annotated with "2.執行". The output of the second cell is displayed below it: `Out[3]: array([[0.99403652, 0.17109276, 0.84190235, 0.04537563, 0.88868306],`
`[0.03845965, 0.8625648 , 0.90504382, 0.71424797, 0.24861374],`
`[0.48872416, 0.69003161, 0.51221897, 0.38049437, 0.89148428],`
`[0.81011968, 0.30820779, 0.653859 , 0.8477063 , 0.46920767],`
`[0.48964363, 0.713421 , 0.65601074, 0.19109199, 0.22990496]])`. The third cell is empty and labeled "In []:".

jupyter notebook – 更改預設目錄

- 程式集 \ Windows 系統 \ 命令提示字元
- `cd C:\`
- `jupyter-notebook`




```

命令提示字元 - jupyter-notebook
Microsoft Windows [版本 10.0.19044.1503]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\user>cd C:\
C:\>jupyter-notebook
[I 2022-02-06 14:25:10.590 LabApp] JupyterLab extension loaded from C:\Users\user\anaconda3\lib\site-packages\jupyterlab
[I 2022-02-06 14:25:10.590 LabApp] JupyterLab application directory is C:\Users\user\anaconda3\share\jupyter\lab
    
```



Jupyter Notebook 快速鍵

- 按 [Esc] cell旁邊為藍色:
 - 按 **x** : 刪除當前選擇的cell
 - 按 **a** : 在當前選擇的上方新增一個cell
 - 按 **b** : 在當前選擇的下方新增一個cell
 - 按 **Shift + Enter** : 執行當前的cell並且選到下一個cell
 - 按 **Ctrl + Enter** : 執行當前cell
 - 按 **M** : 轉成markdown模式，可以看到紅色框框內容從code變成markdown



開啟 → Python程式設計-李明昌.ipynb, 瀏覽加入數學式主題

- <http://rwepa.blogspot.com/2020/02/pythonprogramminglee.html>

實作練習2

主題: **Python** 程式設計-李明昌 - ipynb

檔名: Python_Programming_Lee_ipynb.zip

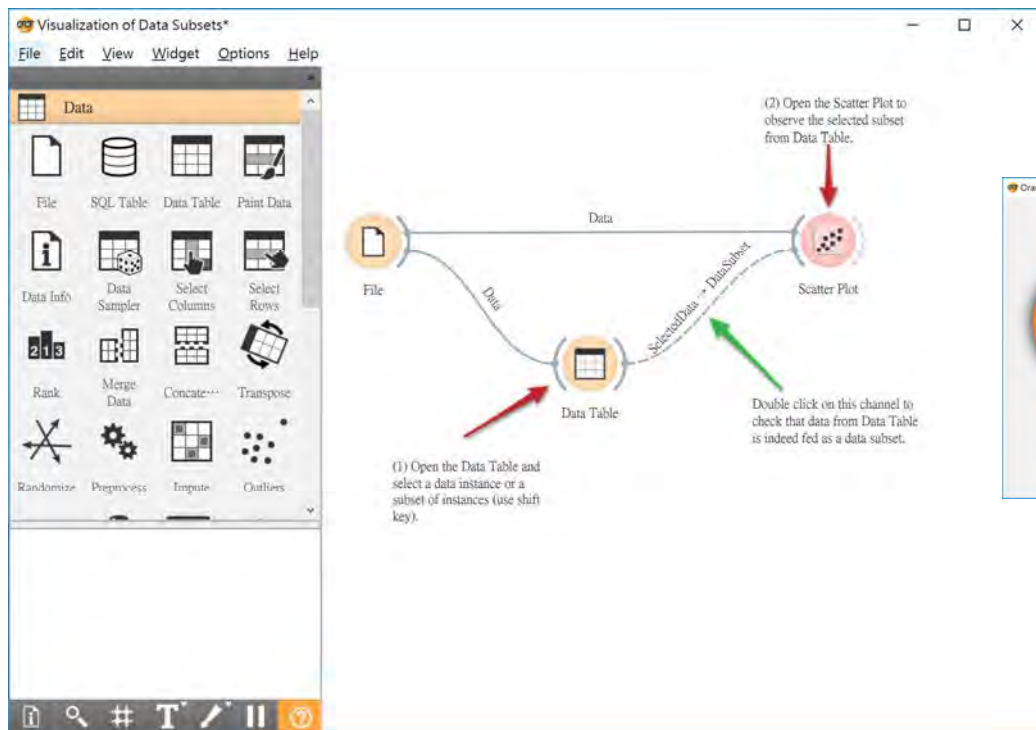
包括 Python 程式設計-李明昌電子書的原始 ipynb 檔案, 圖檔, 部分資料集

下載: https://github.com/rwepa/DataDemo/blob/master/Python_Programming_Lee_ipynb.zip

Python_Programming_Lee_ipynb.zip > python.book.lee >

名稱	類型
.ipynb_checkpoints	檔案資料夾
data	檔案資料夾
img	檔案資料夾
Python程式設計-李明昌.ipynb	IPYNB 檔案

Orange



```
# 安裝 Orange
# conda install -c conda-forge orange3

# 使用命令提示列 開啟 Orange
# python -m Orange.canvas
```



Python Orange - 關聯規則教學

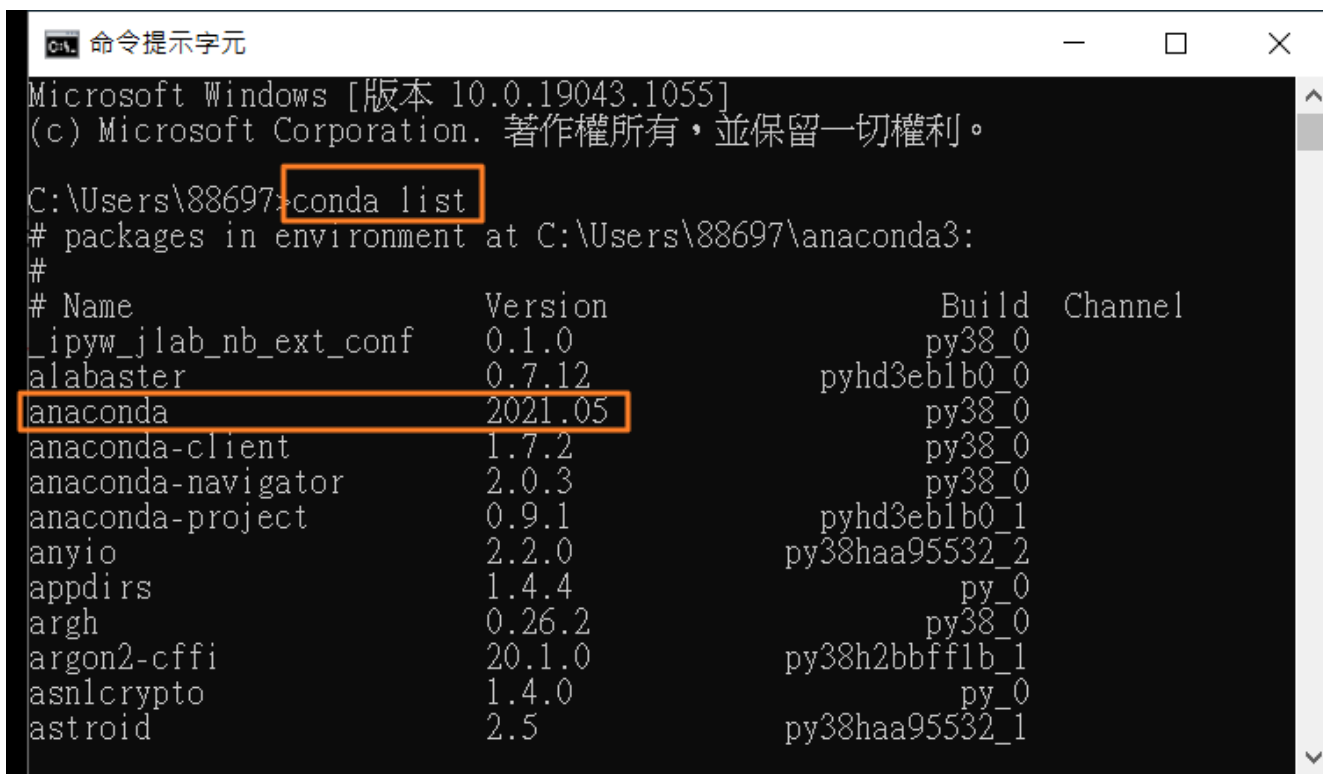
- <http://rwepa.blogspot.com/2022/07/python-orange-associate-tutorial.html>
- <https://youtu.be/rh5GxJamtNg>

Orange - Scatter Plot



Anaconda 模組管理

- 顯示已安裝模組
conda list



```
命令提示字元
Microsoft Windows [版本 10.0.19043.1055]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\88697>conda list
# packages in environment at C:\Users\88697\anaconda3:
#
# Name                                Version                                Build                                Channel
_ipyw_jlab_nb_ext_conf                0.1.0                                py38_0
alabaster                             0.7.12                              pyhd3eb1b0_0
anaconda                              2021.05                              py38_0
anaconda-client                       1.7.2                                py38_0
anaconda-navigator                    2.0.3                                py38_0
anaconda-project                      0.9.1                                pyhd3eb1b0_1
anyio                                 2.2.0                                py38haa95532_2
appdirs                              1.4.4                                py_0
argh                                  0.26.2                              py38_0
argon2-cffi                           20.1.0                              py38h2bbff1b_1
asn1crypto                            1.4.0                                py_0
astroid                               2.5                                  py38haa95532_1
```

Anaconda 模組管理(續)

- 搜尋官網可以下載模組版本

`conda search matplotlib`

- 安裝模組

`conda install 模組名稱`

- 更新模組

`conda update 模組名稱`



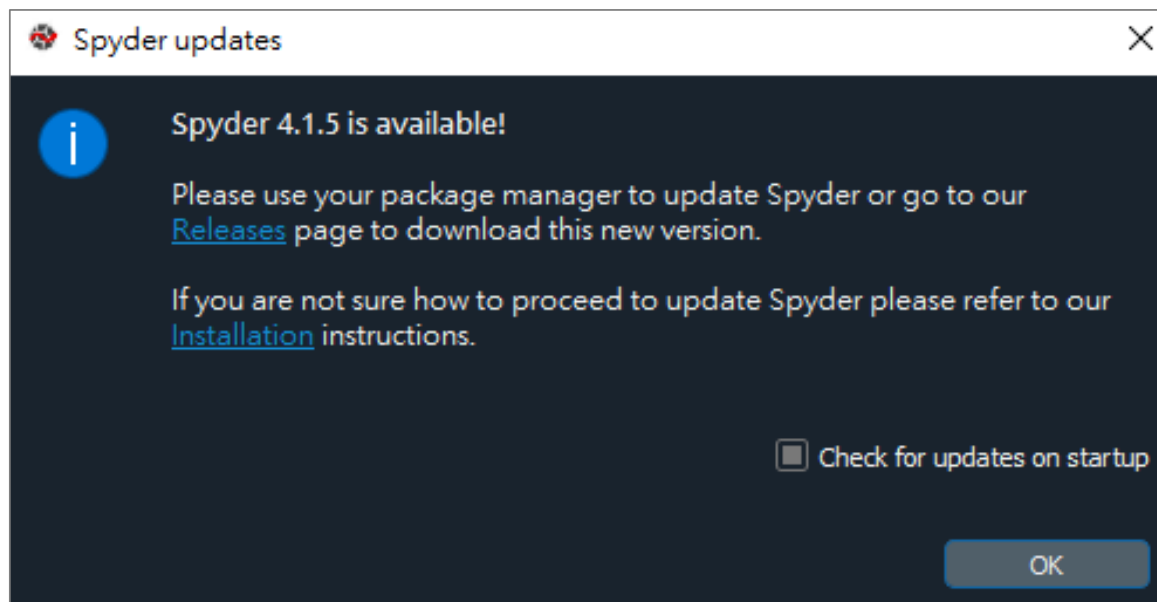
conda 虛擬環境

- 檢視所有虛擬環境清單 **conda env list**
- 啟用 myenv 虛擬環境 **conda activate myenv**
- 關閉虛擬環境 **conda deactivate**
- 建立 myenv 虛擬環境 **conda create --name myenv**
- 建立特定 python 版本的虛擬環境
 - **conda create -n myenv python=3.9**
- 建立特定 scipy 模組版本的虛擬環境
 - **conda create -n myenv scipy=1.9.0**

參考 <https://github.com/rwepa/DataDemo/blob/10c8ab50a0871a6d30b212cef3fc865248532a39/iPAS-python-program.py#L2658>

3.3 Spyder 軟體簡介

Spyder 更新



- Anaconda 整體更新
- Spyder 更新

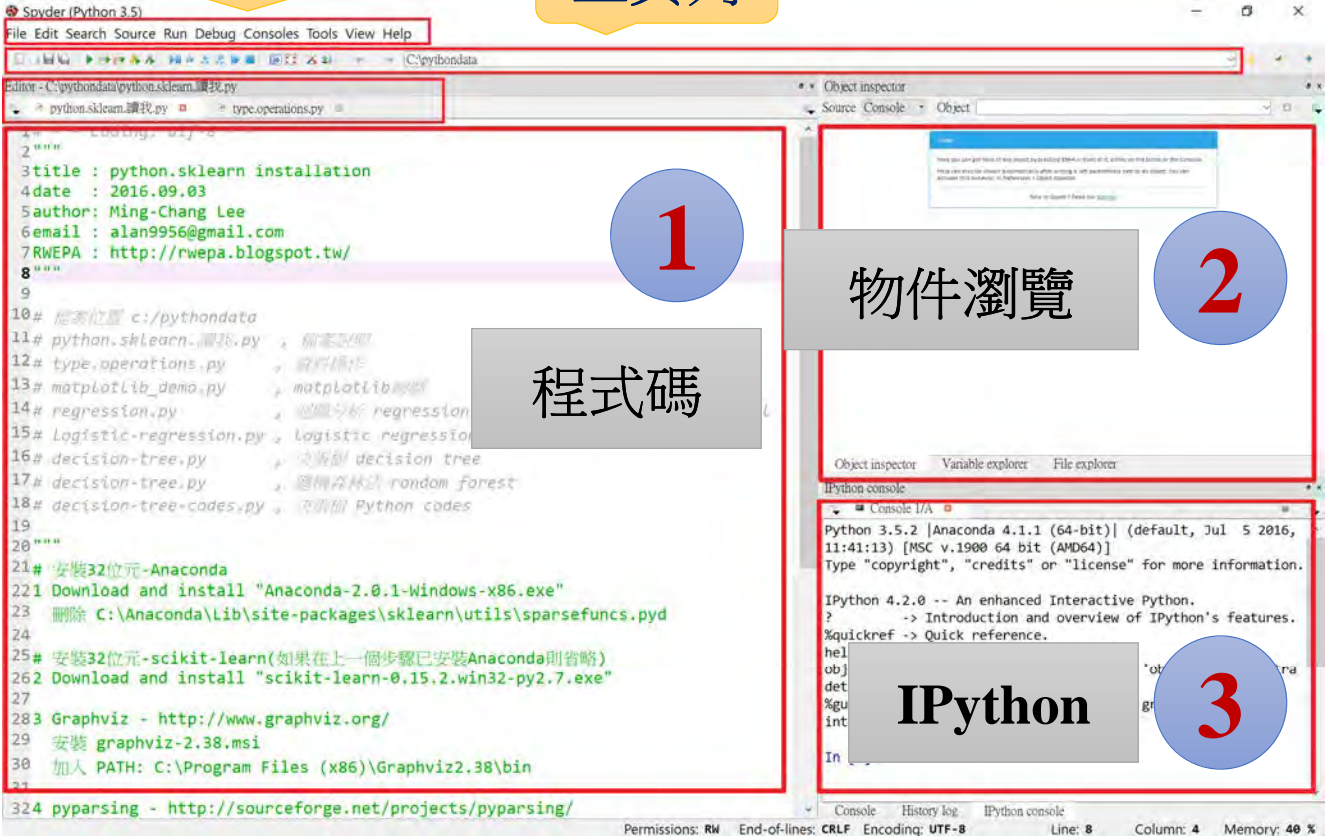
`conda update anaconda`
`conda update spyder`

Spyder 畫面

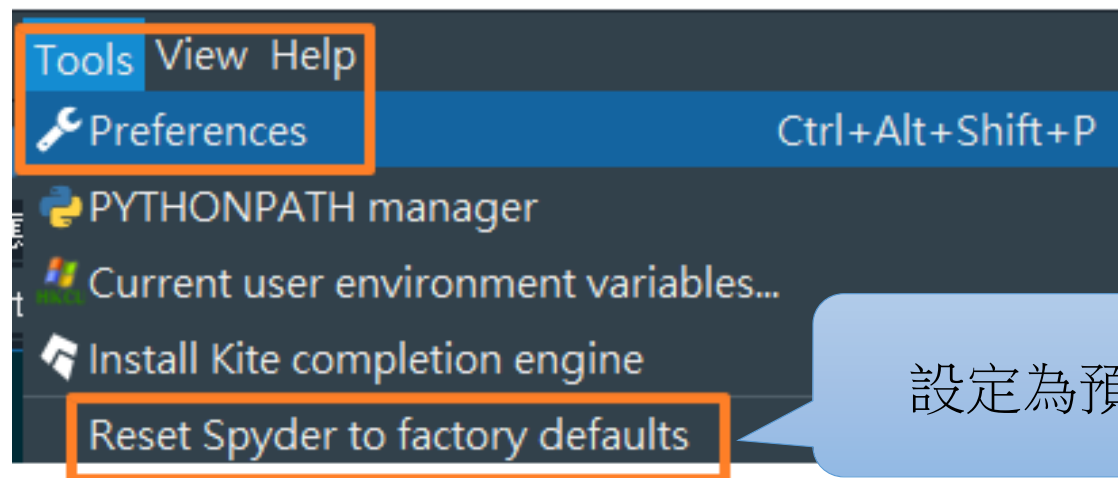
功能表列

工具列

切換窗格

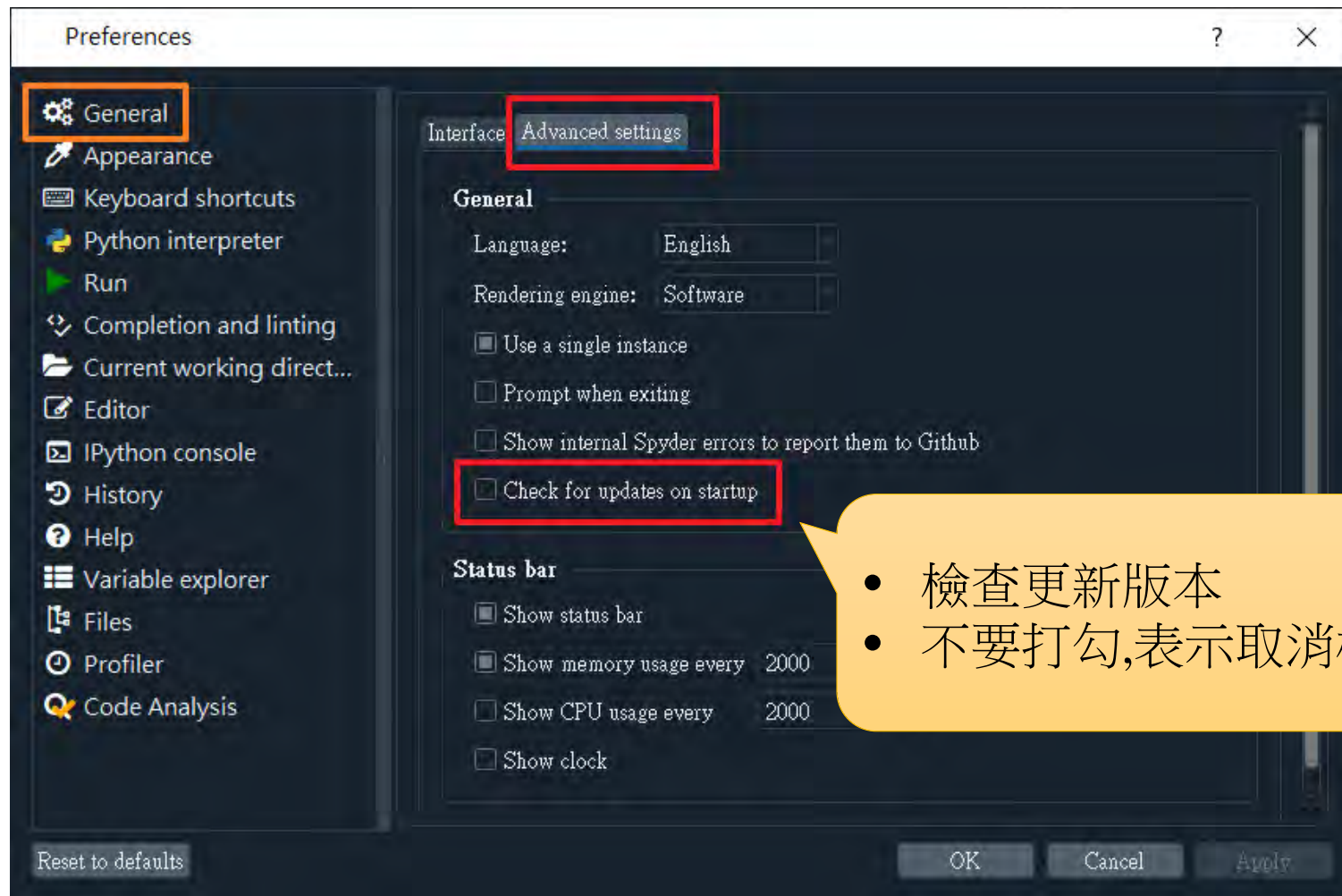


喜好設定 Tools\Preferences



設定為預設值

Preferences \ General \ Advanced setting



Preferences \ Appearance



The screenshot shows the 'Preferences' dialog box for Spyder, specifically the 'Appearance' tab. The left sidebar lists various settings categories, with 'Appearance' selected. The main area is divided into three sections: 'Main interface', 'Syntax highlighting theme', and 'Fonts'. The 'Main interface' section shows 'Interface theme' set to 'Automatic', 'Qt windows style' set to 'WindowsVista', and 'Icon theme' set to 'Spyder 3'. The 'Syntax highlighting theme' section shows 'Spyder' selected in a dropdown menu, with buttons for 'Edit selected scheme', 'Reset to defaults', 'Create new scheme', and 'Delete scheme'. The 'Fonts' section shows 'Plain text' set to 'Consolas' and 'Rich text' set to 'Calibri', both with a font size of '12'. A 'Preview' window on the right shows a code editor with syntax-highlighted Python code. Three yellow callout boxes with orange borders point to specific settings: '佈景主題 (Spyder)' points to the 'Spyder' theme selection, '客製化顏色' points to the 'Edit selected scheme' button, and '字型大小' points to the font size '12' for 'Plain text'.

佈景主題 (Spyder)

客製化顏色

字型大小

Preferences \ Keyboard shortcuts

Preferences

General

Appearance

Keyboard shortcuts

Python interpreter

Run

Completion and linting

Current working directory

Editor

IPython console

History

Help

Variable explorer

Files

Profiler

Code Analysis

Here you can browse the list of all available shortcuts in Spyder. You can also customize them by double-clicking on any entry in this table.

Context	Name	Shortcut
editor	re-run last cell	Alt+Return
-	run	F5
editor	run cell	
editor	run cell and advance	Shift+Return
editor	run selection	Ctrl+Return
profiler	run profiler	F10
pylint	run analysis	F8
-	debug step return	Ctrl+Shift+F11
-	spyder documentation	F1

Search: run

Reset to default values

Reset to defaults

OK

Cancel

Apply

1

2

3

4

• 原為 F9

• 改為 Ctrl + Return






Spyder 好用的快速鍵

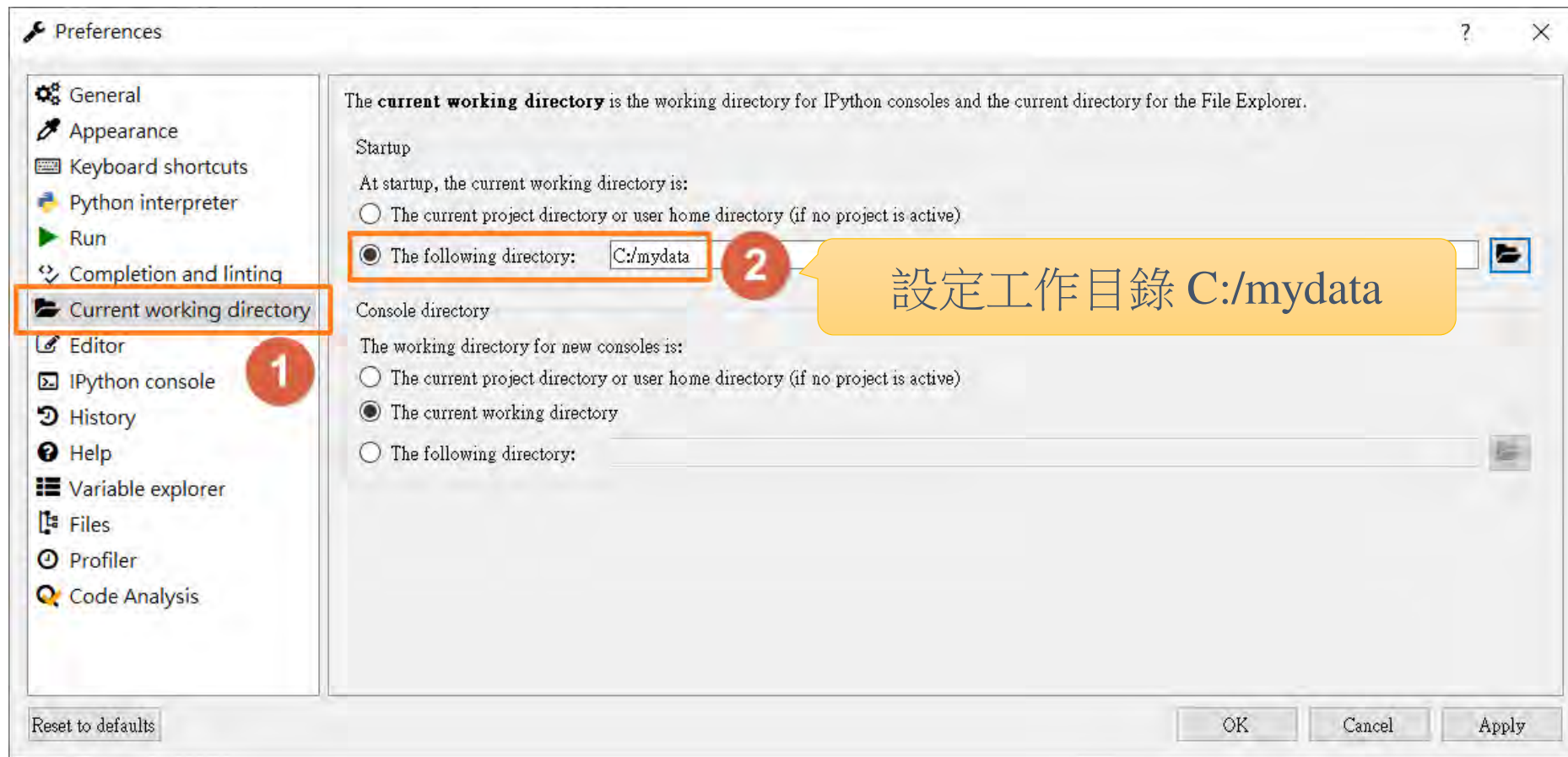
- | | |
|-------------------------------|-------------------------------------|
| • 執行選取程式碼 | Ctrl + Enter (原為 F9) |
| • 註解切換 | Ctrl + 1 |
| • 尋找 | Ctrl + F |
| • 取代 | Ctrl + R |
| • 切換至編輯視窗 | Ctrl + Shift + E |
| • 切換至 Ipython Console | Ctrl + Shift + I |
| • 檔案切換 | Ctrl + P |
| • 重新啟動 Ipython Console | Ctrl + . |
| • 檢視放大 | Ctrl + “+” |
| • 檢視縮小 | Ctrl + “-” |
| • 重新啟動 Spyder | Alt + Shift + R |

Ipython Console 視窗

- 清空: **Ctrl + L**
- 清空: **%clear**

	Split vertically (上下垂直分割)	Ctrl+{
	Split horizontally (左右水平分割)	Ctrl+_
	Close this panel	Alt+Shift+W

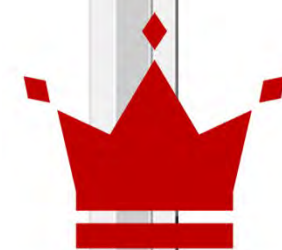
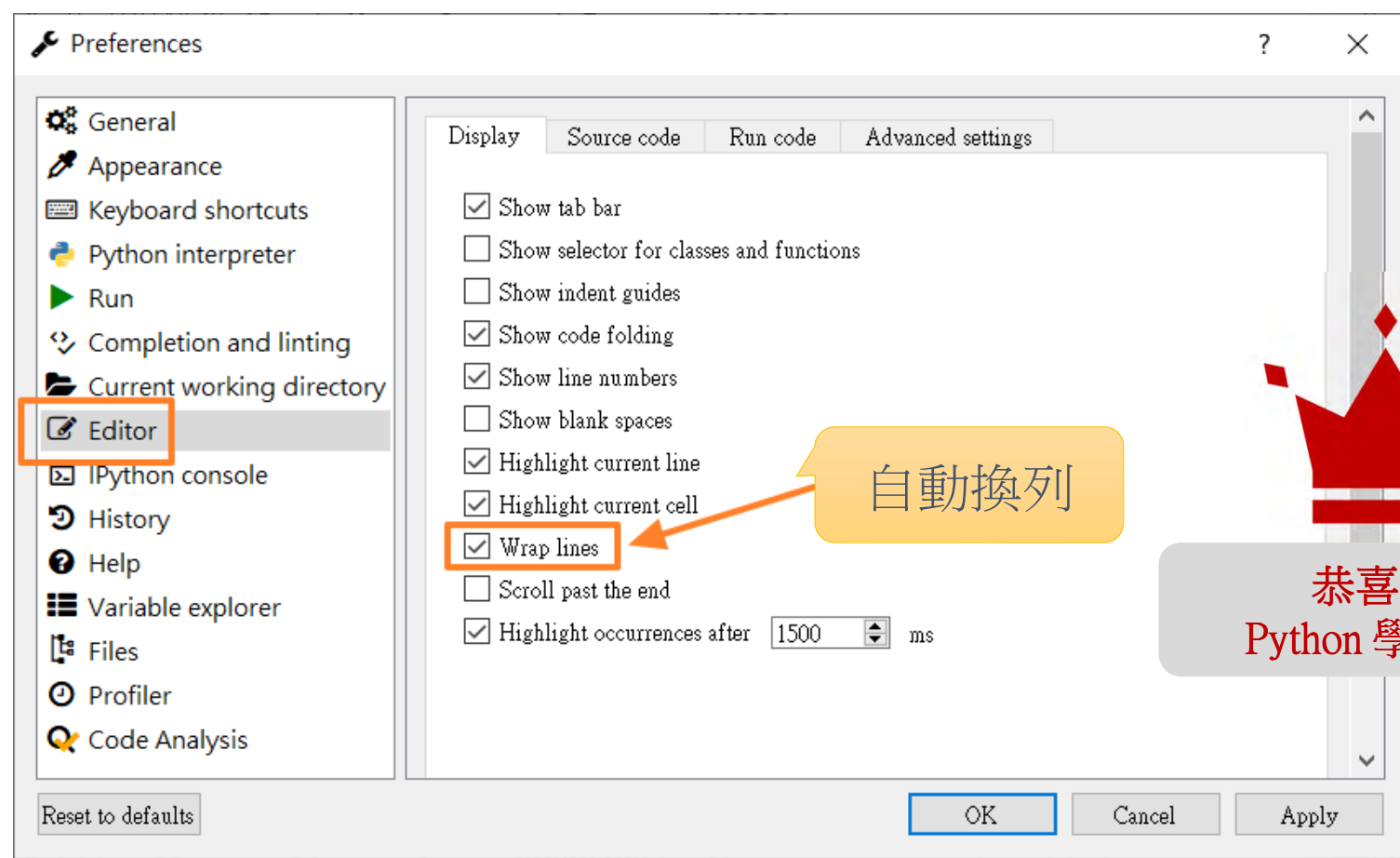
Preferences \ Current working directory





Preferences \ Editor \ Wrap lines

實作練習3



恭喜您, 開啟
Python 學習之旅 ^_^



Python 執行 方法1.命令提示列

- 建立 C:\mydata\helloworld.py

```
C:\mydata\helloworld.py
helloworld.py x
1  # -*- coding: utf-8 -*-
2  """
3  Created on Tue Jun 22 23:06:49 2021
4
5  @author: rwepa
6  """
7
8  print("大數據Python應用")
```

python helloworld.py

```
命令提示字元
C:\mydata>python --version
Python 3.8.8

C:\mydata>dir
磁碟區 C 中的磁碟是 WIN10
磁碟區序號: B428-7C96

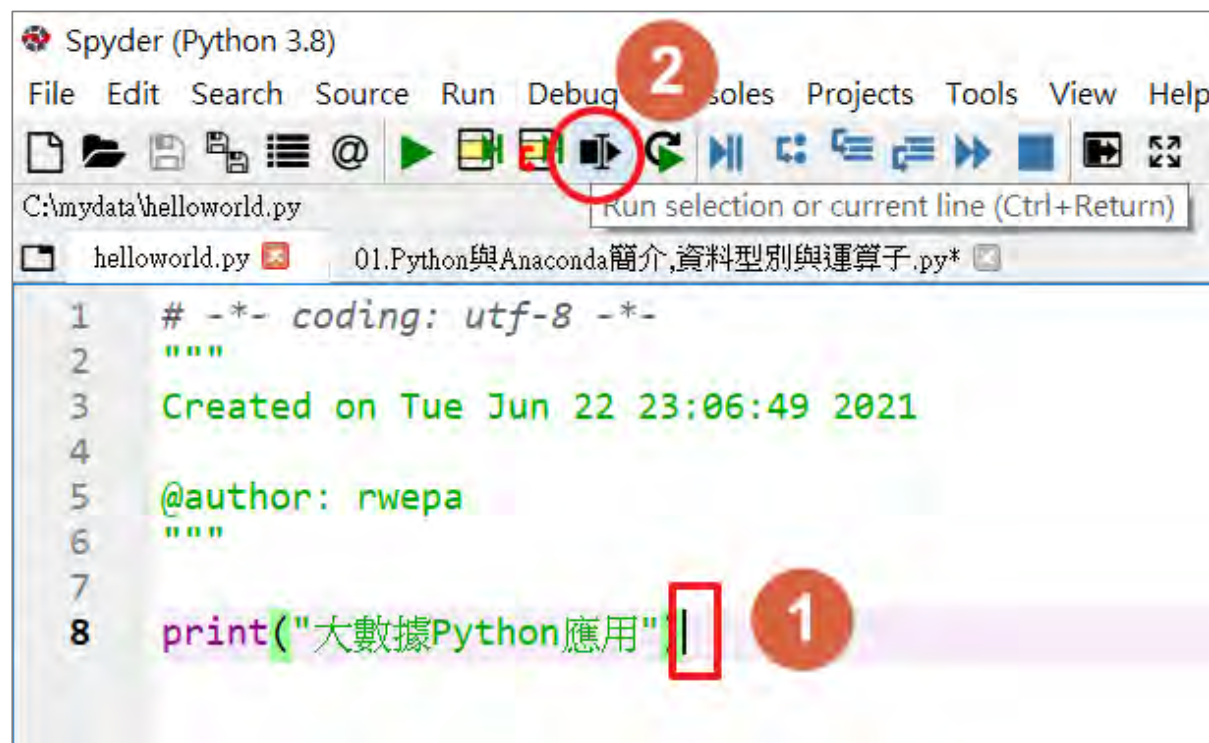
C:\mydata 的目錄
2021/06/22 下午 11:10 <DIR> .
2021/06/22 下午 11:10 <DIR> ..
2021/06/22 下午 11:08 122 helloworld.py
                  1 個檔案      122 位元組
                  2 個目錄    62,354,751,488 位元組可用

C:\mydata>python helloworld.py
大數據Python應用

C:\mydata>
```

Python 執行 方法2.使用 Spyder

- 選取資料列
- 按 [Run selection or current line]



3.4 變數



合法的變數名稱

- 必須以字母或下底線字符開頭
- 不能以數字開頭
- 只能包含字母、數字和下底線（A-z、0-9 和 _）
- 區分大小寫
 - 例: `customer`、`Customer` 和 `CUSTOMER` 是不同的變數
- 雙下底線開頭與結尾的名稱,部分已經由Python保留, 例: `__init__`

合法變數 vs. 不合法變數

合法變數

大數據 = 1 # 中文亦可, 建議不要使用

CustomerSaleReport = 1

_CustomerSaleReport = 1

Customer_Sale_Report = 1

customer_sale_report = 1

- 命名為中文, OK?
- 命名為Python保留字, OK?

不合法變數

\$CustomerSaleReport = 1 # SyntaxError: invalid syntax

2020_sale = 100 # SyntaxError: invalid decimal literal

break = 123 # SyntaxError: invalid syntax

內建保留字

- `dir(__builtins__)`
- `len(dir(__builtins__))` # 160

```
[in] len(dir(__builtins__)) # 160  
[out] 160
```

Python 註解

- 使用一個 # → 用於1行註解
- 使用二個 `"""` → 用於超過1行註解或函數之說明文件

```
1  """  
2  file      : 01.Python與Anaconda簡介,資料型別與運算子.py  
3  author   : Ming-Chang Lee  
4  email    : alan9956@gmail.com  
5  RWEPA    : http://rwepa.blogspot.tw/  
6  GitHub   : https://github.com/rwepa  
7  Encoding : UTF-8  
8  """
```



內縮4個空白鍵之語法

```
with open('sampleinput1.txt', 'r') as f:
    doc = [x.strip() for x in f.readlines()] # 使用 strip 刪除換行符號
    docname = []
    corpus = []
    for index in range(len(doc)):
        if index == 0:
```

注意：with 結尾加冒號

內縮4個空白鍵

3.5 資料型別與運算子

資料型別 (Data Types)

- 數值型別

- 整數 `int`
- 長整數 `long`
- 浮點數 `float`
- 複數 `complex`

- 布林值 **bool**

- `True`
- `False`

- 空值 **None** → 類似 **NULL**

- 字串 (**String**)

參考: <https://docs.python.org/3/library/stdtypes.html>



資料型別 – 範例

整數 *int*

x1 = 1

type(x1)

浮點數 *float*

x2 = 1.234

type(x2)

複數 *complex*

x3 = 1+2j

type(x3)

布林值 (*Boolean*)

x4 = True

type(x4)

x4 + 10

- 轉換為整數 *int*
- 轉換為浮點數 *float*
- 轉換為複數 *complex*

整數亂數

```
In [1]: import random
...: random.seed(168)
...: myrandom = random.randrange(1, 100)
...: myrandom
Out[1]: 96
```

亂數種子 random.seed()

```
In [2]: random.seed(168)
...: myrandom = random.randrange(1, 100)
...: myrandom
Out[2]: 96
```




運算子

運算子	功能
**	次方
*	乘法
/	除法
//	整除
%	餘數
+	加法
-	減法
	或運算子 OR
^	互斥運算子 XOR
&	且運算子 AND
<<	左移運算子
>>	右移運算子

- Excel: =2^3
- R: 2^3
- Python: 2**3
- SQL: **SELECT POWER**(2, 3)

運算子－範例

```
In [1]:  
...: 3 + 5  
Out[1]: 8
```

```
In [2]: 3 + (5 * 4)  
Out[2]: 23
```

```
In [3]: 3 ** 2  
Out[3]: 9
```

```
In [4]: "Hello" + "World"  
Out[4]: 'HelloWorld'
```

```
In [5]: 1 + 1.234  
Out[5]: 2.234
```

```
In [6]: 7 / 2  
Out[6]: 3.5
```

```
In [7]: 7 // 2  
Out[7]: 3
```

```
In [8]: 7 % 2  
Out[8]: 1
```

```
In [9]: 2 ** 10  
Out[9]: 1024
```

```
In [10]: 1.234e3 - 1000  
Out[10]: 234.0
```

```
In [11]: x5 = 1 == 2
```

```
In [12]: x5  
Out[12]: False
```

```
In [13]: x5 + 10  
Out[13]: 10
```

指派運算子

指派運算子	範例	結果
=	x = 9	x = 9
+=	x += 2	x = x + 2
-=	x -= 3	x = x - 3
*=	x *= 4	x = x * 4
/=	x /= 5	x = x / 5
%=	x %= 6	x = x % 6
//=	x //= 7	x = x // 7
**=	x **= 8	x = x ** 8
&=	x &= 3	x = x & 3
=	x = 3	x = x 3
^=	x ^= 3	x = x ^ 3
>>=	x >>= 3	x = x >> 3
<<=	x <<= 3	x = x << 3

```
In [1]:
...:
...: x = 9
...: x+=2
...: print(x)
11
```

3.6 資料物件

資料物件



- 四大容器型別 (Container type)
 - tuple → () 表示, 資料不可修改 (序列/元組)
 - list → [] 表示, 資料可以修改 (串列/清單)
 - set → { } 表示, 執行集合運算 (集合)
 - dict → {'key1': value1, ... } 表示, 可藉由key查value (字典)

Tuple



Tuple

- 建立序列

```
f = (2,3,4,5)           # A tuple of integers
g = ()                  # An empty tuple
h = (2, [3,4], (10,11,12)) # A tuple containing mixed objects
```

- Tuples操作

```
x = f[1]                # Element access. x = 3
y = f[1:3]              # Slices (切片) y = (3,4)
z = h[1][1]            # Nesting. z = 4
```

取出 y 的第1個索引到 4-1=3

- 特色

- 與list類似，最大的不同tuple是一種唯讀且不可變更的資料結構
- 不可取代tuple中的任意一個元素，因為它是唯讀不可變更的
- Tuple 是具有 ordered 特性
- Python 的索引(指標)從0開始



Tuple - loop 處理

tuple - loop 處理

```
fruits = ("apple", "banana", "cherry")
```

方法1. tuple - 取出元素, 使用for

```
for x in fruits:
    print(x)
```

方法2. tuple - 取出元素, 使用while

```
i = 0
while i < len(fruits):
    print(fruits[i])
    i = i + 1
```

方法3. tuple - 取出元素, 使用指標 range, len

```
for i in range(len(fruits)):
    print(fruits[i])
```

長度 len

- 三個方法結果皆相同,何者較快?

In [3]:

```
...: for i in range(len(fruits)):
...:     print(fruits[i])
```

```
apple
banana
cherry
```



List



基本型態 – 串列(List)

- 建立串列

```
In [1]: a = [2, 3, 4]           # 整數串列
...: b = [2, 7, 3.5, "Hello"] # 混合資料串列
...: c = []                    # 空串列
...: d = [2, [a, b]]           # 巢狀串列
```

```
In [2]: a
Out[2]: [2, 3, 4]
```

```
In [3]: b
Out[3]: [2, 7, 3.5, 'Hello']
```

```
In [4]: c
Out[4]: []
```

```
In [5]: d
Out[5]: [2, [[2, 3, 4], [2, 7, 3.5, 'Hello']]]
```

1

2

串列操作

```
In [7]: a  
Out[7]: [2, 3, 4]
```

```
In [8]: a[1]          # 取得第2個元素  
Out[8]: 3
```

```
In [9]: a[-1]         # 取得最後一個元素  
Out[9]: 4
```

```
In [10]: b[1:3]       # 串列篩選  
Out[10]: [7, 3.5]
```

```
In [11]: d[1][0][2]   # 巢狀串列操作  
Out[11]: 4
```

```
In [12]: b[0]         # 2  
Out[12]: 2
```

```
In [13]: b[0] = 42    # 修改元素值
```

```
In [14]: b[0]         # 42  
Out[14]: 42
```

```
a  
[2, 3, 4]  
  
b  
[2, 7, 3.5, 'Hello']  
  
d  
[2, [[2, 3, 4], [2, 7, 3.5, 'Hello']]]
```

b[1:3] 相當於 b[1], b[2]

```
In: # 串列長度
```

```
In: len(b)  
Out: 4
```

Set

基本型態 – 集合(Set)

- 集合與字典相似, 但集合沒有key, 只有值
- 集合內容不可以修改
- 集合是 `unordered`
- 集合是 `unindexed`
- 集合會忽略重複的值

集合 - set

```
In [1]: a = set() # 空集合
```

```
In [2]: type(a)
```

```
Out[2]: set
```

```
In [3]: b = {"台北市", "新北市", "桃園市", "台中市", "台北市", "新北市", "高雄市"}
```

```
In [4]: b # {'台中市', '台北市', '新北市', '桃園市', '高雄市'}
```

```
Out[4]: {'台中市', '台北市', '新北市', '桃園市', '高雄市'}
```

```
In [5]: # b[0] = 1 # TypeError: 'set' object does not support item assignment
```

```
In [6]: # b[0]      # TypeError: 'set' object is not subscriptable
```

```
In [7]: len(b)
```

```
Out[7]: 5
```

Dict

基本型態 – 字典(Dict)

- 字典與集合相似, 但字典有key, 有值
- 字典內容可以修改
- 字典是 ordered (Python 3.6/早期版本 字典是*unordered*)
- 字典是 indexed
- 字典不可以有重複的key

字典 - 宣告

```
In [1]: mydict = {  
...:     "language": "Python",  
...:     "designer": "Guido van Rossum",  
...:     "year": 1991  
...: }
```

```
In [2]: print(mydict)  
{'language': 'Python', 'designer': 'Guido van Rossum', 'year': 1991}
```

```
In [3]: type(mydict) # dict  
Out[3]: dict
```



字典 - 重複 key, 只保留1個

```
In [4]: mydict1 = {
...:     "language": "Python",
...:     "designer": "Guido van Rossum",
...:     "year": 1991,
...:     "year": 2021
...: }
...:
...: print(mydict1)
{'language': 'Python', 'designer': 'Guido van Rossum', 'year': 2021}
```

保留重複新 key - value

字典存取元素 – keys, values

```
In [1]:
...: b = {
...:     "uid": 168,
...:     "login": "marvelous",
...:     "name" : 'Alan Lee'
...: }
...: b
Out[1]: {'uid': 168, 'login': 'marvelous', 'name': 'Alan Lee'}
```

In [2]: # dict 取得所有 keys

```
In [3]: mykeys = b.keys()
...: print(mykeys)
dict_keys(['uid', 'login', 'name'])
```

In [4]: # dict 取得所有 values

```
In [5]: myvalues = b.values()
...: print(myvalues)
dict_values([168, 'marvelous', 'Alan Lee'])
```

字典取得key的值

```
In [6]: u = b["uid"] # 168
...: print(u)
168
```

```
In [7]: # dict 更新值
```

```
In [8]: b.update({"uid": 123})
...: print(b)
{'uid': 123, 'login': 'marvelous', 'name': 'Alan Lee'}
```

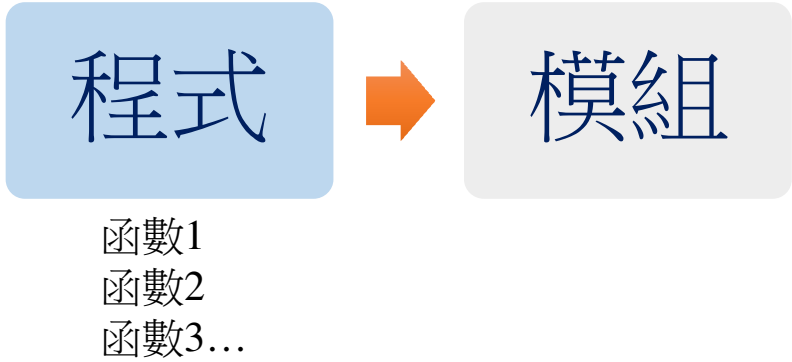
```
In [9]: # dict 新增元素
```

```
In [10]: b["shell"] = "/bin/sh"
...: print(b)
{'uid': 123, 'login': 'marvelous', 'name': 'Alan Lee', 'shell': '/bin/sh'}
```

3.7 認識模組

模組

- Python 可以將一個主程式，分割成多個檔案，此分割的檔案可以形成模組 (module)。
- 主檔名表示模組的名稱，即檔案名稱 = 「**模組名稱.py**」。
- 模組中包含 Python 函數和語法的檔案，可由其他專案使用。
- 模組中的函數可以被 `import` 到其他模組中，或是被 `import` 至Python程式。
- 在模組中，模組的名稱（字串）是全域變數 `__name__` 的值。
- 第1次執行時，原始程式會編譯成「**.pyc檔案**」之位元碼(**bytecode**)，以增進執行效率。
- 模組下載：
 - `pip install moduleName`
 - `conda install moduleName`



使用模組

- 匯入模組的所有函數
 - `import` 模組名稱
 - `import` 模組名稱 `as` 別名
- 匯入特定函數
 - `from` 模組名稱 `import` 函數名稱1,...
 - `from` 模組名稱 `import *`
- 使用模組內的特定函數
 - 模組名稱.函數()

```
In [1]: import math
```

```
In [2]: math.sqrt(9)
```

```
Out[2]: 3.0
```

```
In [3]: from math import sqrt
```

```
In [4]: sqrt(9)
```

```
Out[4]: 3.0
```




切換工作目錄

- `os.getcwd()`
- `os.chdir("C:/")`

```
# 切換工作目錄
import os
os.getcwd() # 讀取工作目錄
os.chdir("C:/") # 變更工作目錄
os.getcwd()
os.listdir(os.getcwd()) # 顯示檔案清單
```

模組的搜尋路徑

```
In [1]: import sys
```

```
In [2]: sys.path
```

```
Out[2]:
```

```
['C:\\\\Users\\88697\\anaconda3\\python38.zip',  
 'C:\\\\Users\\88697\\anaconda3\\DLLs',  
 'C:\\\\Users\\88697\\anaconda3\\lib',  
 'C:\\\\Users\\88697\\anaconda3',  
 '',  
 'C:\\\\Users\\88697\\anaconda3\\lib\\site-packages',  
 'C:\\\\Users\\88697\\anaconda3\\lib\\site-packages\\loket-0.2.1-py3.8.egg',  
 'C:\\\\Users\\88697\\anaconda3\\lib\\site-packages\\win32',  
 'C:\\\\Users\\88697\\anaconda3\\lib\\site-packages\\win32\\lib',  
 'C:\\\\Users\\88697\\anaconda3\\lib\\site-packages\\Pythonwin',  
 'C:\\\\Users\\88697\\anaconda3\\lib\\site-packages\\IPython\\extensions',  
 'C:\\\\Users\\88697\\.ipython']
```



實作練習4

自訂模組

- 自訂模組計算商數與餘數
檔案名稱為 **numberscompute.py**

```
# numberscompute.py
def divide(a,b):
    q = a/b
    r = a - q*b
    return q,r
```

q 商
r 餘

q = a/b 須要修改

- 練習檔案名稱為 **mycompute.py**

```
import numberscompute
x,y = numberscompute.divide(42,5)
```

模組名稱.函數()

參考資料

- RWEPA
 - <http://rwepa.blogspot.com/>
- Python 程式設計-李明昌 <免費電子書>
 - <http://rwepa.blogspot.com/2020/02/pythonprogramminglee.html>
- R入門資料分析與視覺化應用教學(付費)
 - <https://mastertalks.tw/products/r?ref=MCLEE>
- R商業預測與應用(付費)
 - <https://mastertalks.tw/products/r-2?ref=MCLEE>

謝謝您的聆聽

Q & A



李明昌

EMAIL: alan9956@gmail.com

WEB: <http://rwepa.blogspot.com/>