

Python

技術者們 實踐!

第 0 章 Python 簡介與安裝

本投影片（下稱教用資源）僅授權給採用教用資源相關之旗標書籍為教科書之授課老師（下稱老師）專用，老師為教學使用之目的，得摘錄、編輯、重製教用資源（但使用量不得超過各該教用資源內容之80%）以製作為輔助教學之教學投影片，並於授課時搭配旗標書籍公開播放，但不得為網際網路公開傳輸之遠距教學、網路教學等之使用；除此之外，老師不得再授權予任何第三人使用，並不得將依此授權所製作之教學投影片之相關著作物移作他用。

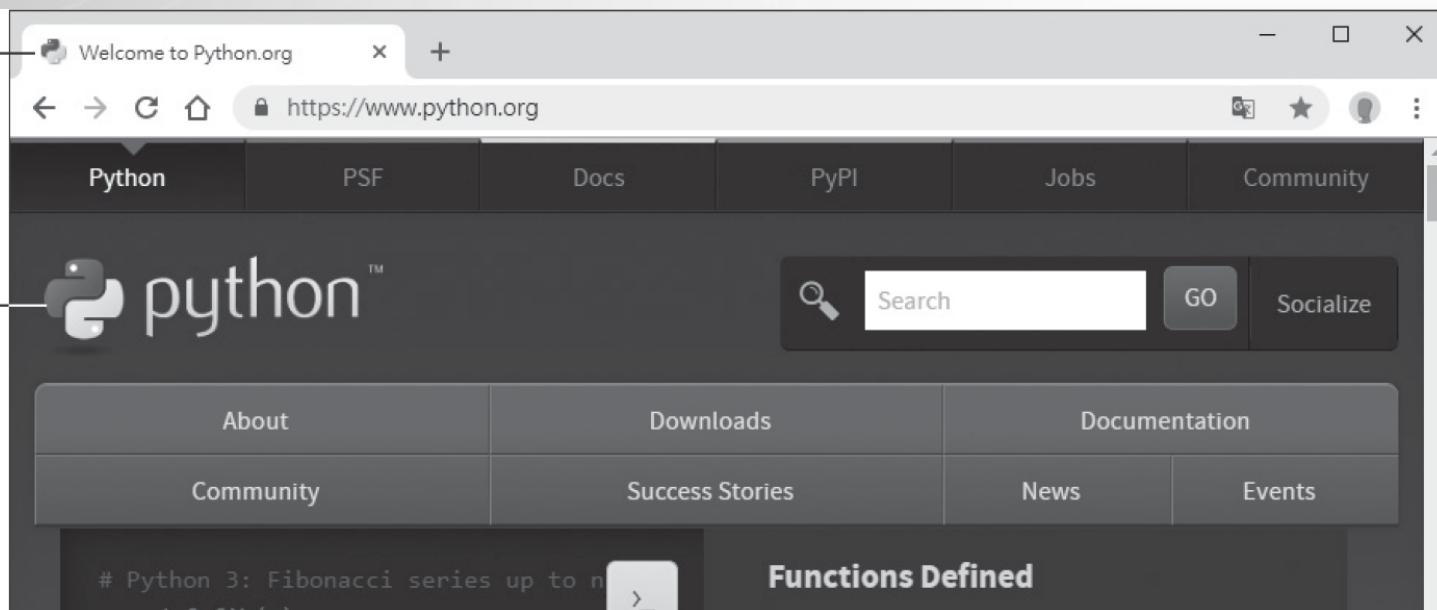
0-0 Python 簡介

- Python 是由荷蘭程式設計師 Guido van Rossum 於 1989 年所創建, 由於他是英國電視短劇 Monty Python's Flying Circus (蒙提·派森的飛行馬戲團) 的愛好者, 因此選中 Python (大蟒蛇) 做為新語言的名稱

Python



Python 的
蟒蛇標誌



Python



Python 的特色

- 易學易用
- 快速開發
- 高可攜性 (跨平台性)
- 容易擴充
- 直譯式語言
- 免費且開源

Python



Python 的 2.x 與 3.x 版

- Python 於 2008 年發佈 3.0 版, 但由於做了一些較大的改變而導致無法和 2.x 版的程式相容, 因此目前仍然提供 2.x 與 3.x 二種版本的支援與更新。
- 不過目前越來越多人人都已轉用 3.x 版, 而且幾乎所有的第三方套件也都升級到 3.x 版了, 因此強烈建議讀者選用 3.x 版

Python



Python 的開發環境

- 在開始撰寫 Python 程式之前, 要先建置 Python 的開發環境, 包括 Python 的直譯器、內建函式庫、以及相關的檔案和環境設定等

Python



0-1 安裝 Anaconda

- Anaconda 是目前廣受歡迎的 Python 整合開發套件, 其內容包含
 - Anaconda Navigator
 - Spyder
 - Jupyter Notebook
 - 預先安裝 200 多個常用的科學、數學、資料分析等領域的第三方套件。

Python



Downloads - Anaconda x +

← → ↻ 🏠 🔒 https://www.anaconda.com/download/ 🔍 ☆ 🧑

Documentation Blog Contact 🔍

ANACONDA. What is Anaconda? Products Support Resources About **Downloads**

Windows macOS Linux

Anaconda 5.2 For Windows Installer

Python 3.6 version *

↓ Download

[64-Bit Graphical Installer \(631 MB\)](#) ⓘ
[32-Bit Graphical Installer \(506 MB\)](#)

Python 2.7 version *

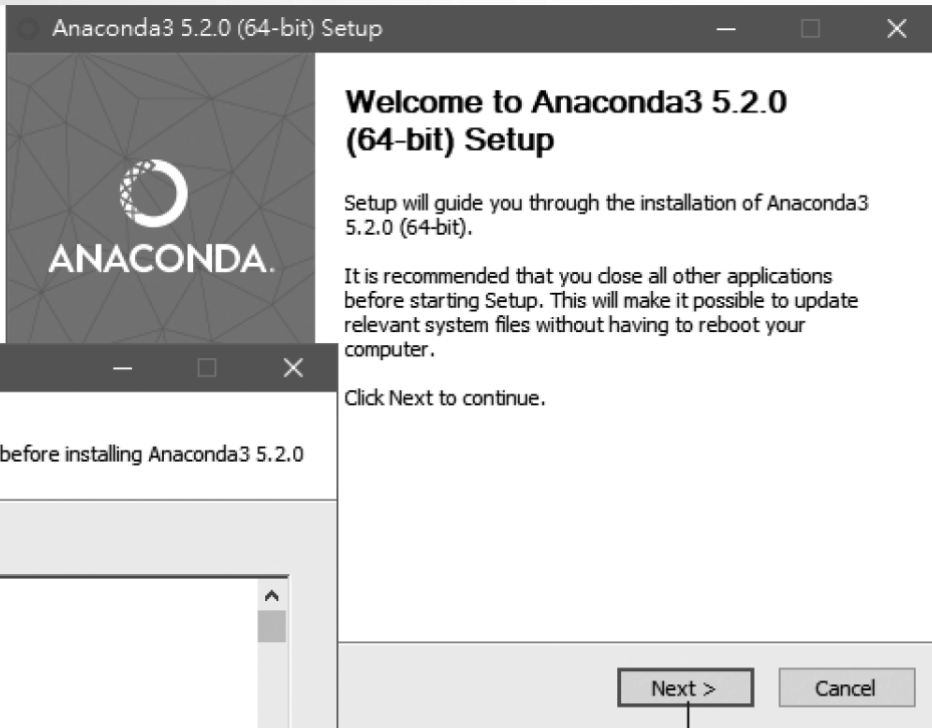
↓ Download

[64-Bit Graphical Installer \(564 MB\)](#) ⓘ
[32-Bit Graphical Installer \(443 MB\)](#)

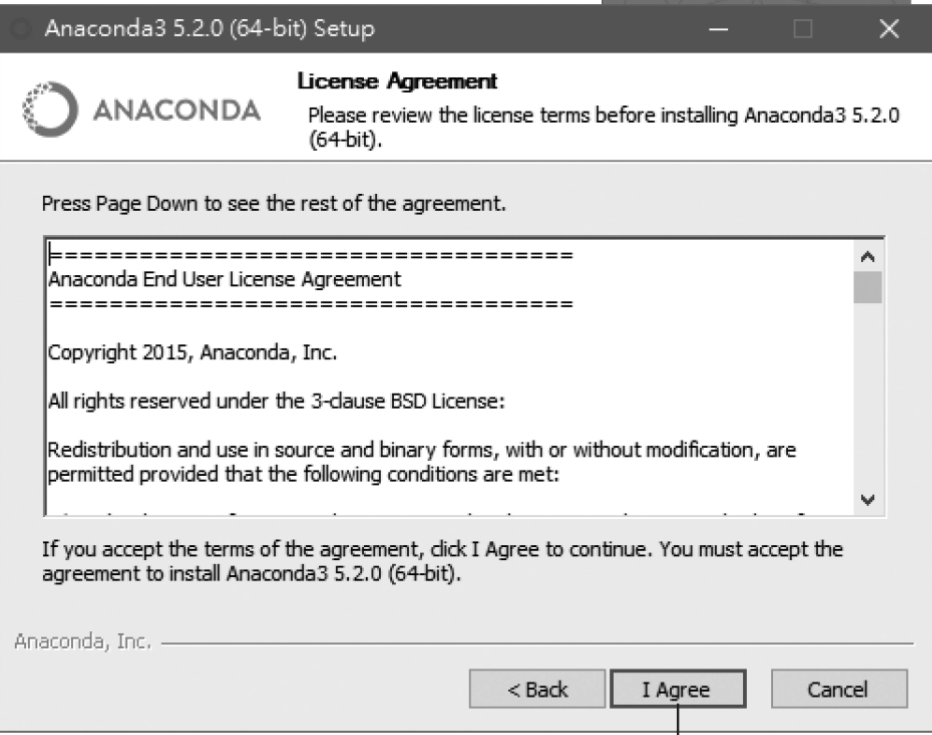
1 請按此下載安裝檔

Python

2 執行下載的安裝檔
(筆者安裝時為
Anaconda3-5.2.0-
Windows-x86_64.exe)

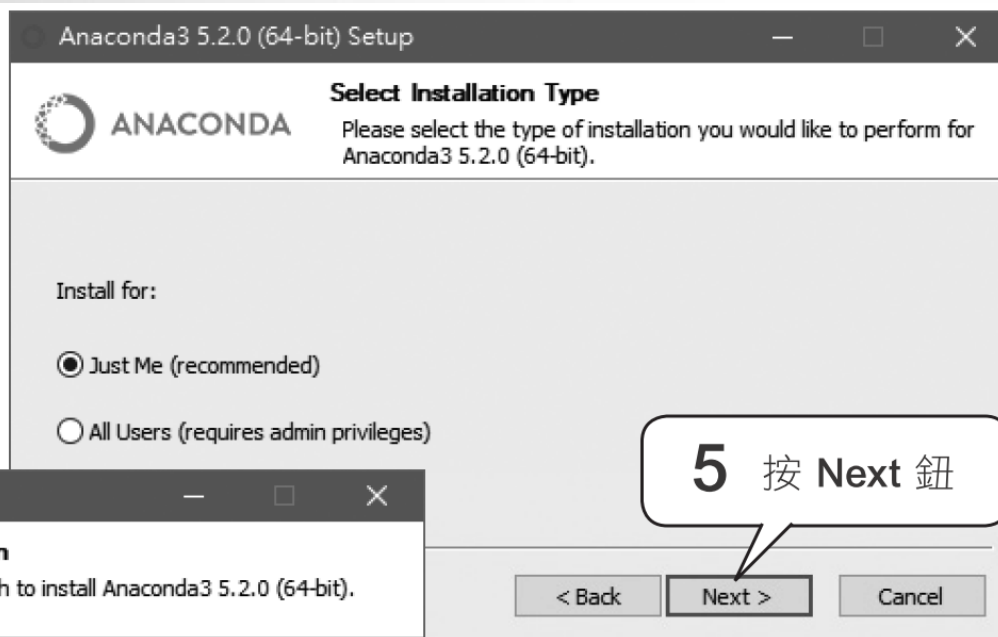


3 按 Next 鈕

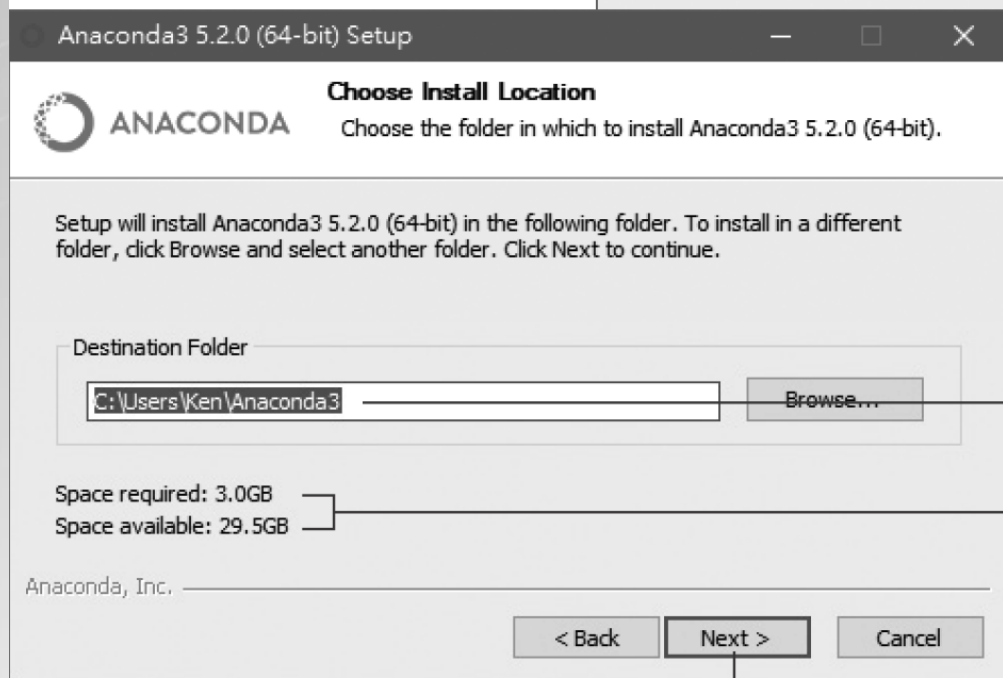


4 按 I Agree 鈕同意授權條款





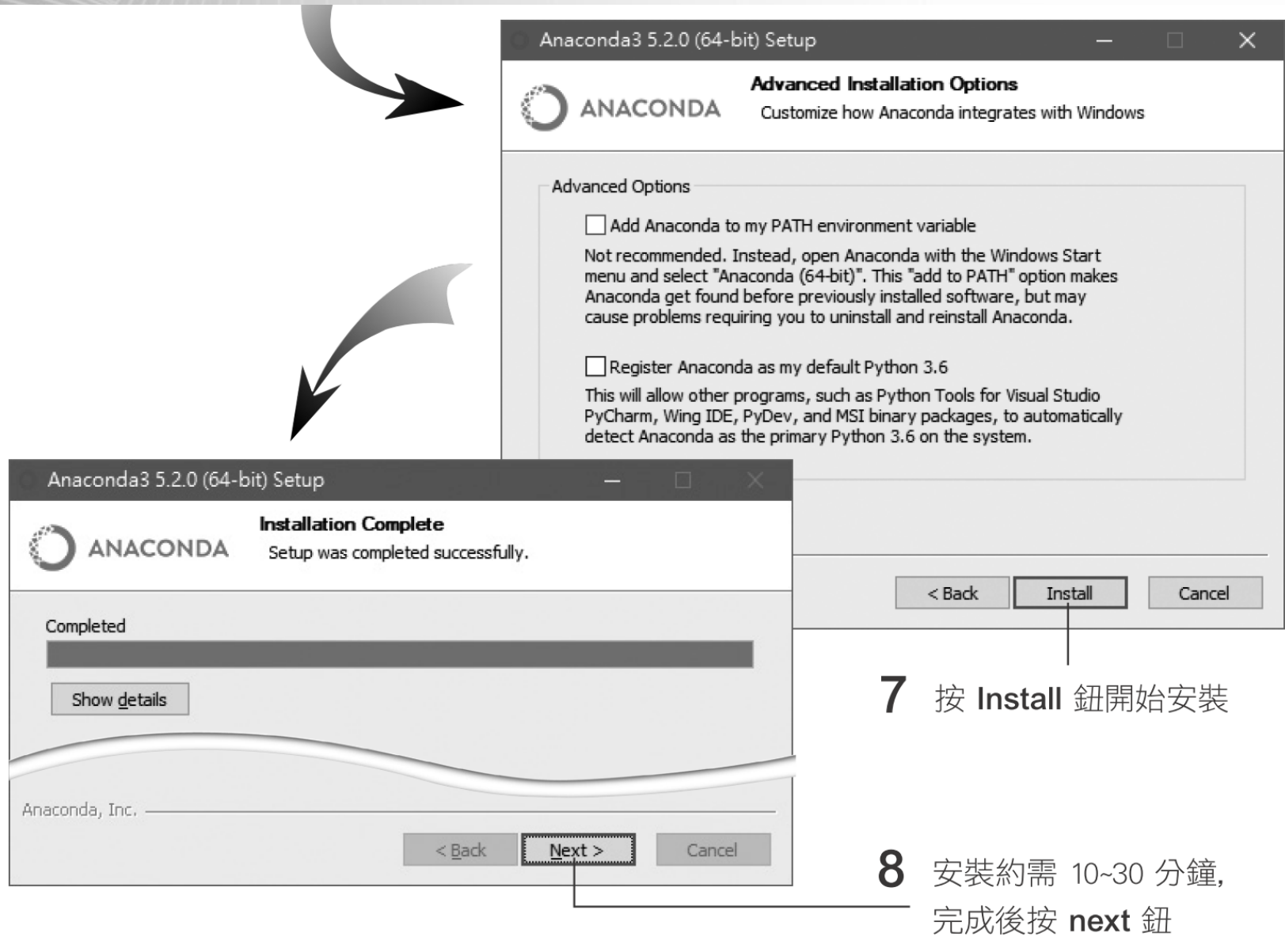
5 按 Next 鈕

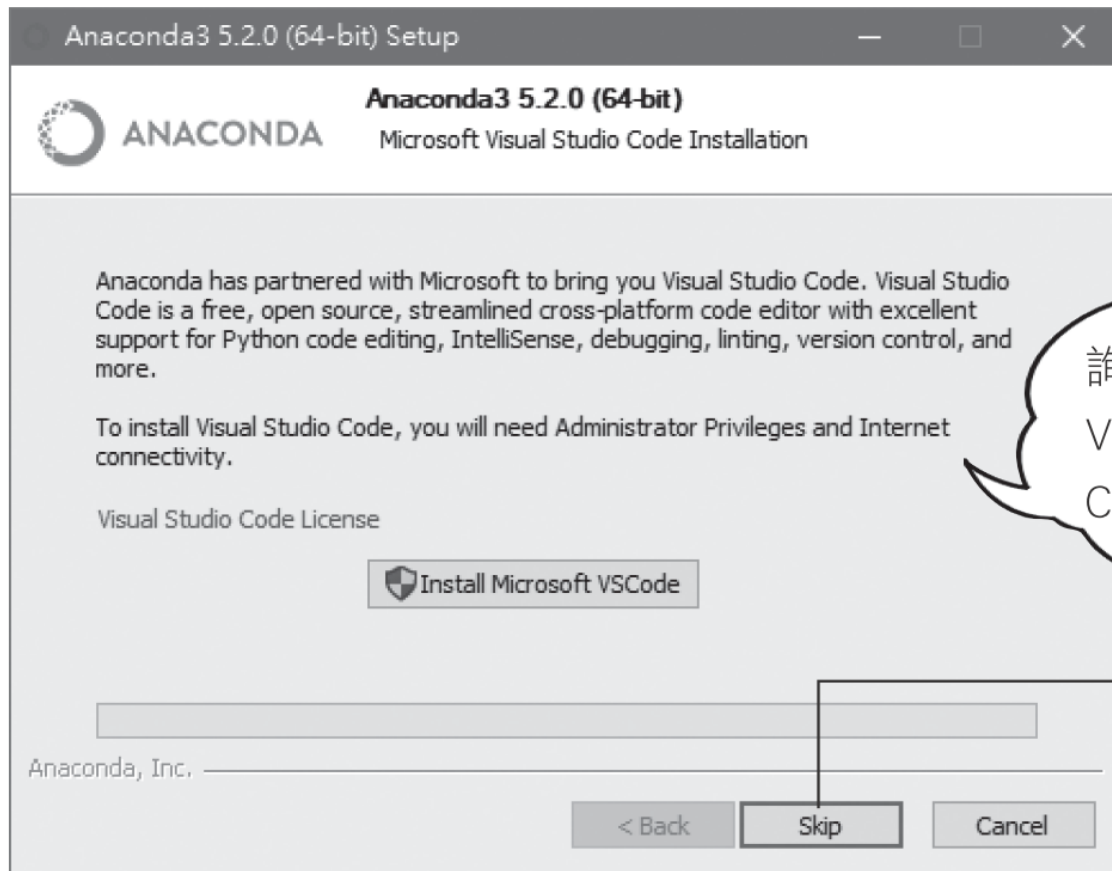


6 接著請按 Next 鈕

預設會安裝在 C:\Users\
使用者名稱\Anaconda3
安裝需要的及目前可用的
磁碟空間, 若空間不夠
可改安裝到其他硬碟中







詢問是否要安裝微軟的
VSCode (Visual Studio
Code) 整合開發編輯器

9 可先按此鈕略過, 以後
需要時再從 Anaconda
Navigator 中安裝

Python



Anaconda3 5.2.0 (64-bit) Setup



Thanks for installing Anaconda3!

Anaconda is the most popular Python data science platform.

Share your notebooks, packages, projects and environments on Anaconda Cloud!

- Learn more about Anaconda Cloud
- Learn how to get started with Anaconda

< Back

Finish

Cancel

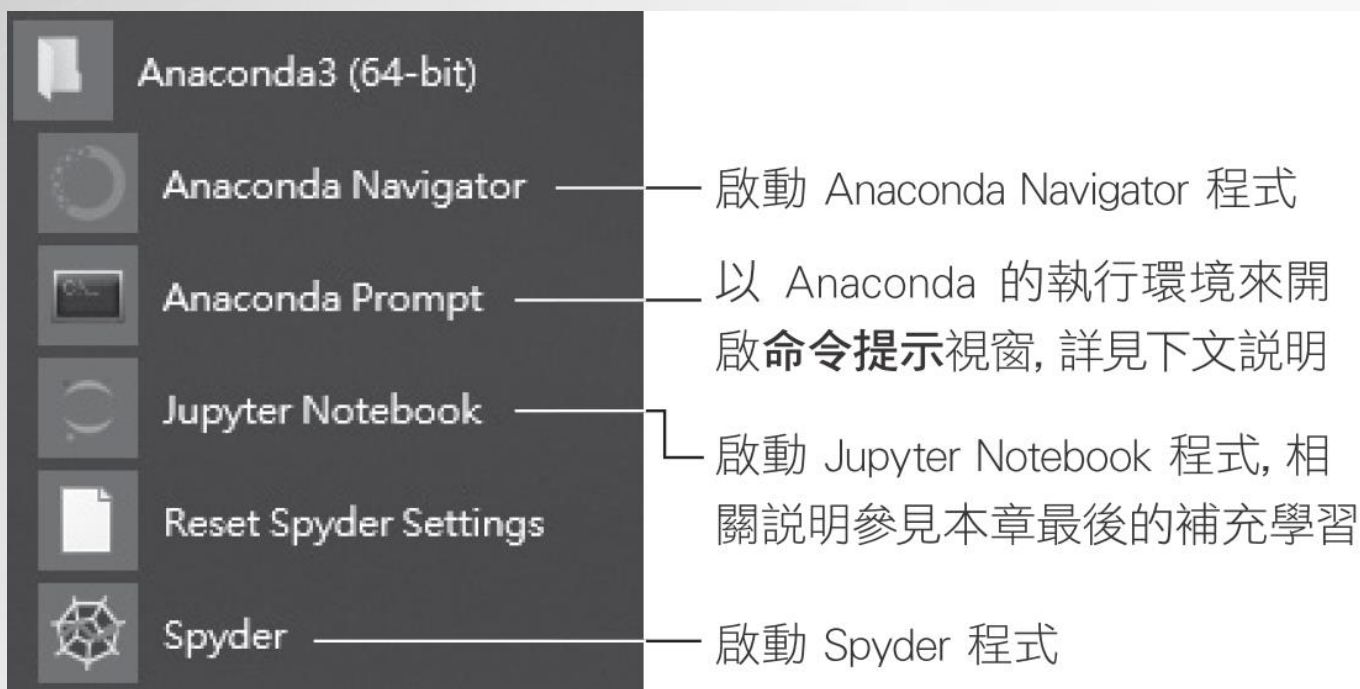
若不想看說明
可取消勾選

10 按 **Finish** 鈕
完成安裝

Python



- 安裝好之後，可在 Windows 的開始功能表中看到 Anaconda 的選單命令



Anaconda Prompt (命令提示) 視窗



```
(base) C:\Users\Ken>
```

The screenshot shows a standard Windows command prompt window titled "Anaconda Prompt". The prompt text is "(base) C:\Users\Ken>". A vertical line from the title points to the window title bar. A bracket under "C:\Users\Ken" points to the label "目前所在的路徑". A bracket under "(base)" points to the label "執行環境的名稱". A bracket under the prompt character ">" points to the label "可在此輸入命令 然後按 Enter 執行".

執行環境的名稱

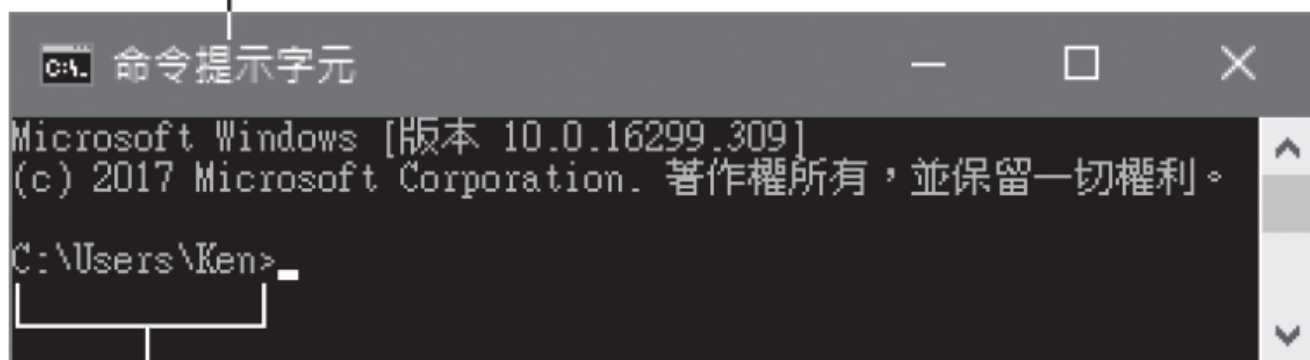
目前所在的路徑

可在此輸入命令
然後按 **Enter** 執行

Python



Windows 的**命令提示字元**視窗, 視窗標題和 Anaconda Prompt 不一樣



```
C:\> 命令提示字元
Microsoft Windows [版本 10.0.16299.309]
(c) 2017 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\Ken>
```

命令提示文字只有目前路徑



0-2 在 Anaconda Prompt 中撰寫 及執行程式

- Python 是直譯式的語言, 因此需要使用直譯器 (Interpreter) 來執行我們的程式, 例如 Python 內建的 CPython, 或是強調互動性的 IPython 直譯器

Python



IPython 直譯器

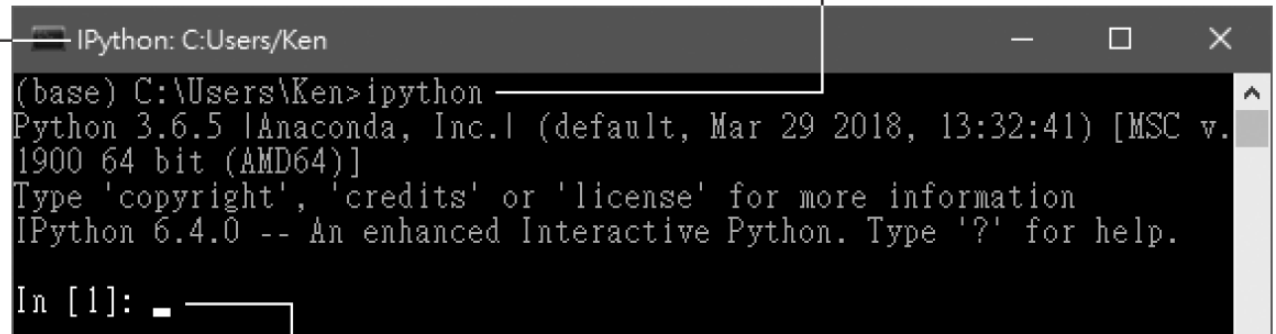
- IPython 是 CPython 的擴充版本，它們的基本操作都相同，但 `ipython` 增加了許多額外的輔助功能，並強化了交談模式的易用性，因此像 `Spyder`、`Jupyter Notebook` 都是以 IPython 做為其直譯器
- IPython 有 2 種使用方式：第一種是交談模式

Python



1 請先開啟在 Anaconda Prompt 視窗, 然後輸入 "ipython" 再按 **Enter** 鍵即可進入交談模式

2 進入交談模式後, 視窗的標題會改為「IPython: 目前所在路徑」



```
IPython: C:\Users\Ken
(base) C:\Users\Ken>ipython
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.4.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

3 會出現 In [序號]: 表示可以在此輸入程式

Python



IPython 是用 In [序號]: 做為輸入提示符號, 並以 Out[序號]: 做為輸出的提示號

每輸入完一行程式後都要按 **Enter** 鍵來執行

```
Anaconda Prompt
IPython 6.4.0 -- An enhanced Interac
In [1]: a = 1
In [2]: a
Out[2]: 1
In [3]: a + 2
Out[3]: 3
In [4]: print('總和為', a + 2)
總和為 3
In [5]: exit()
(base) C:\Users\Ken>_
```

4 輸入 a = 1 建立一個名為 a 的變數

5 輸入 a

6 IPython 會顯示 a 的內容

7 計算 a + 2

8 IPython 會計算並顯示結果

9 也可以用 print() 來輸出計算的結果

10 IPython 顯示 print() 的結果

11 若要結束 IPython 可輸入 exit() 然後按 **Enter** 鍵, 請注意! exit() 之後, 視窗的標題現在已變回 Anaconda Prompt 了



IPython 直譯器

- 第二種方式是直接用 IPython 來執行程式檔

Anaconda Prompt

```
(base) C:\Users\Ken>ipython D:\pybook\ch00\print.py
總和為 3
(base) C:\Users\Ken>_
```

請將路徑改為實際儲存範例程式的路徑

2 只會輸出 print() 的內容

1 用「ipython 程式檔路徑」來執行程式檔

print.py 的內容

```
a = 1
a
a + 2
print('總和為', a + 2)
```

這 2 行程式都不會顯示內容

此程式的內容和前面輸入的程式完全一樣



```
Anaconda Prompt
(base) C:\Users\Ken>D:
(base) D:\>cd \pybook\ch00
(base) D:\pybook\ch00>python print.py
總和為 3
(base) D:\pybook\ch00>
```

目前的資料夾

- 1 先切換到 D: 磁碟
- 2 用 cd 命令來切換資料夾
- 3 python 會在目前資料夾中尋找程式檔



0-3 用 Spyder 撰寫與執行程式

- **Spyder** 是一個整合式的程式開發編輯器, 它整合了 **IPython** 直譯器, 不只可以撰寫及執行程式, 更提供了輔助輸入、查詢說明、以及好用的偵錯功能

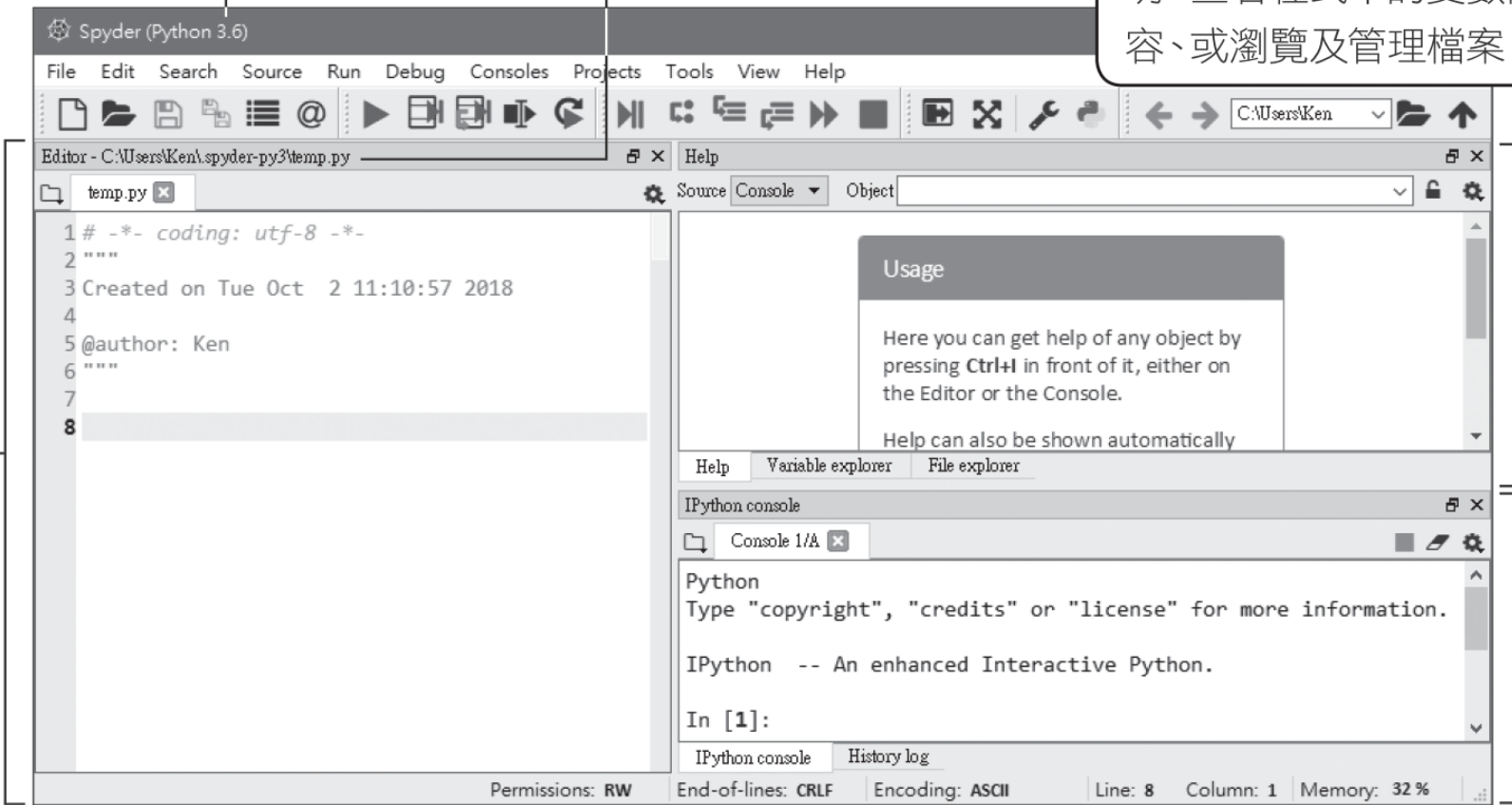
Python



目前使用的 Python 版本

目前編輯中程式檔的路徑

輔助窗格：可在此查詢說明、查看程式中的變數內容、或瀏覽及管理檔案



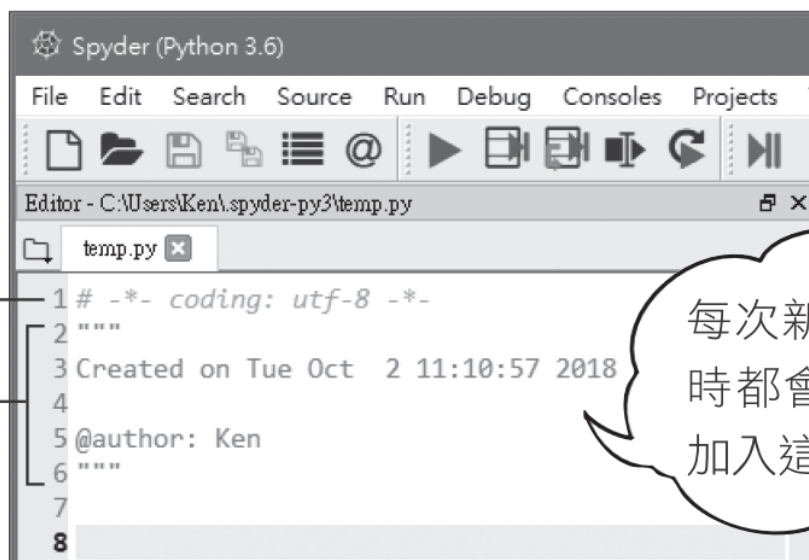
程式編輯窗格：可在此撰寫程式並儲存為程式檔

IPython 窗格：此為 IPython 的交談模式，我們可在此輸入程式來立即測試執行結果



- 新增、管理、與執行程式檔

這行是用來註明檔案的編碼方式, 不過 Python 3.x 預設就是 utf-8 編碼, 因此這行也可省略



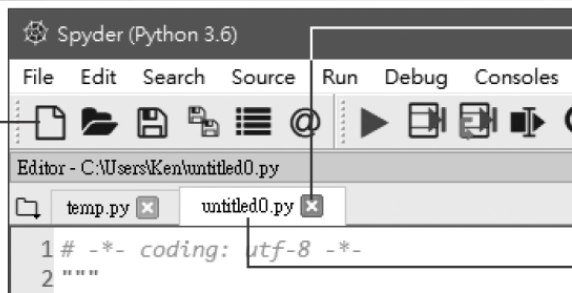
```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Oct 2 11:10:57 2018
4
5 @author: Ken
6 """
7
8
```

每次新增程式檔時都會自動幫你加入這些註解

可在 3 個引號標記 "" 的區間內加入程式的說明, 若不需要也可刪除



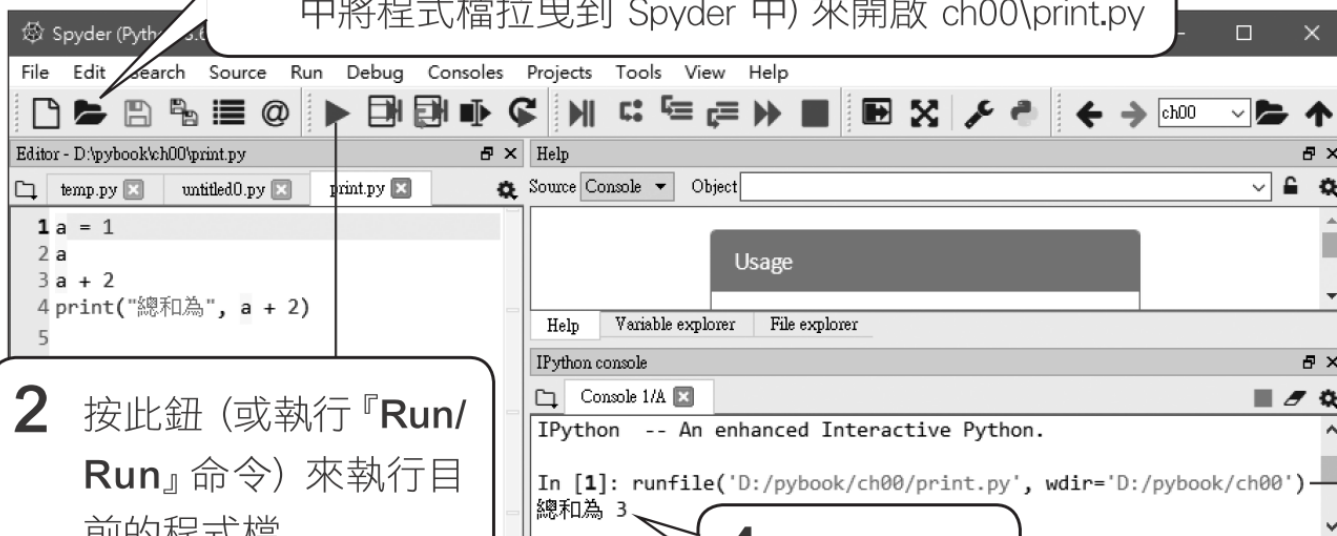
1 按此鈕 (或執行『File/New file』命令) 新增程式檔



2 按此圖示可關閉程式檔
新增的檔名預設為 untitled 加序號 (由 0 開始), 要等到存檔時才會要求指定存檔路徑及檔名

接著再開啟本書的範例程式 ch00\print.py 來執行看看：

1 按此鈕 (或執行『File/Open』命令, 或在檔案總管中將程式檔拉曳到 Spyder 中) 來開啟 ch00\print.py



2 按此鈕 (或執行『Run/Run』命令) 來執行目前的程式檔

4 執行的結果

3 程式會在 IPython 窗格中執行

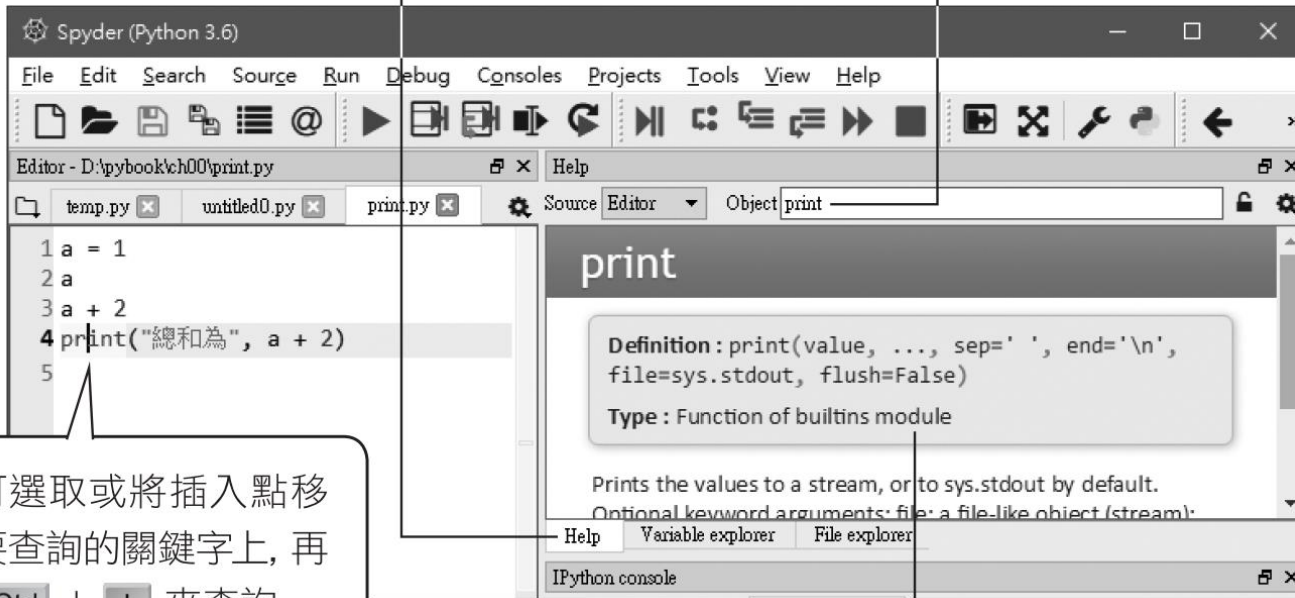


Help 輔助說明窗格

- 輔助說明窗格可以方便我們查詢各種物件的說明

1 切到 Help 頁次

2 輸入要查詢的物件, 再按 **Enter** 鍵



也可選取或將插入點移
要查詢的關鍵字上, 再
按 **Ctrl + I** 來查詢

3 即會顯示相關說明

```
1 a = 1
2 a
3 a + 2
4 print("總和為", a + 2)
5
```

print

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

Type: Function of builtins module

Prints the values to a stream, or to sys.stdout by default.

Optional keyword arguments: file: a file-like object (stream):



File Explorer 檔案瀏覽窗格

- 檔案瀏覽窗格可以方便我們管理程式檔：

The screenshot shows the Spyder Python IDE interface. The File Explorer window is open, displaying a list of files in the current directory. A context menu is open over the 'print.py' file, showing options like 'Run', 'Edit', and 'Show in external file explorer'. Annotations with arrows point to specific parts of the interface:

- 1 選此頁次**: Points to the 'print.py' file in the File Explorer list.
- 雙按即可開啟檔案做編輯**: Points to the 'print.py' file in the File Explorer list.
- 可在此選擇或切換工作路徑**: Points to the address bar at the top of the File Explorer window.
- 2 在檔案上按右鈕**: Points to the right-click action on the 'print.py' file.
- 選此項可直接在 IPython 窗格中執行程式檔 (而不開啟檔案)**: Points to the 'Run' option in the context menu.

The File Explorer window shows the following table of files:

Name	Size	Type	Date Modified
hello.py	115 bytes	py File	2018/10/1 下午 06:47
print.py	49 bytes	py File	2018/10/22 下午 09:28

The IPython console shows the following output:

```
In [1]: runfile('D:/pybook/ch00/hell  
你好! Python  
  
In [2]:
```



Variable explorer 變數瀏覽窗格

- 變數瀏覽窗格可以檢視所有已建立的變數：

- 1 切到此頁次
- 2 執行程式
- 3 即可看到由程式建立的變數

The screenshot displays the Spyder Python IDE interface. The main editor window shows a Python script with the following code:

```
1 a = 1
2 a
3 a + 2
4 print("總和為", a + 2)
5
```

The Variable explorer window is open, showing a table of variables:

Name	Type	Size	Value
a	int	1	1

The IPython console at the bottom shows the output of the script:

```
wdir='D:/pybook/ch00')
總和為 3

In [2]:
```

Annotations in the image point to the following elements:

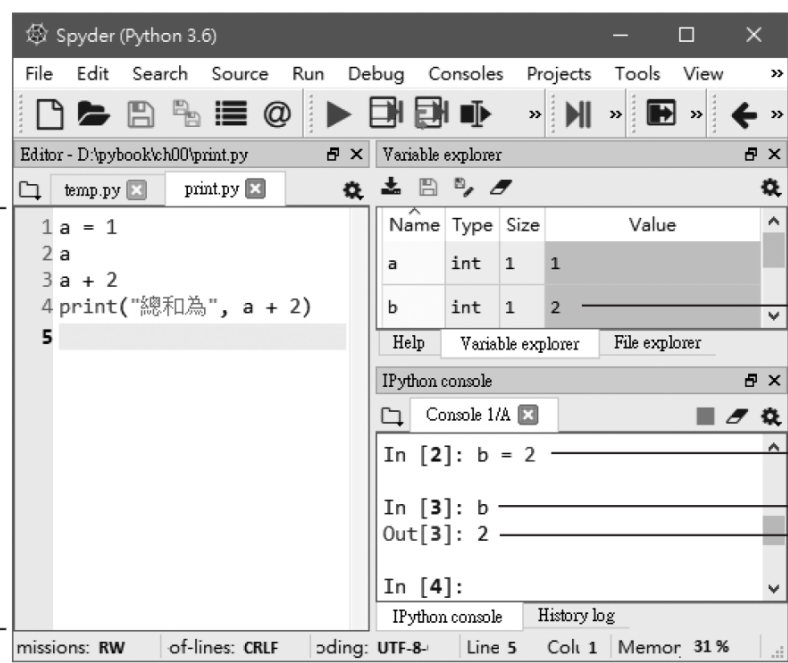
- 1 切到此頁次: Points to the code editor window.
- 2 執行程式: Points to the Run button in the toolbar.
- 3 即可看到由程式建立的變數: Points to the Variable explorer window.
- 變數的型別: Points to the 'Type' column in the Variable explorer table.
- 變數的值: Points to the 'Value' column in the Variable explorer table.

At the bottom of the IDE, the status bar shows: Permissions: RW | End-of-lines: CRLF | Encoding: UTF-8-GUESSED | Line: 5 | Column: 1 | Memory: 31%



IPython 窗格

- 如果只是臨時想測試一下某段程式的執行結果,則可在右下方的 IPython 窗格中進行,例如:



本書 1、2 章的程式大部份都會在 IPython 窗格中操作。

2 變數瀏覽窗格中立刻增加了變數 b

1 輸入 b = 2 並按 Enter 鍵執行

3 輸入變數名稱 b

4 會顯示 b 的內容

程式編輯窗格

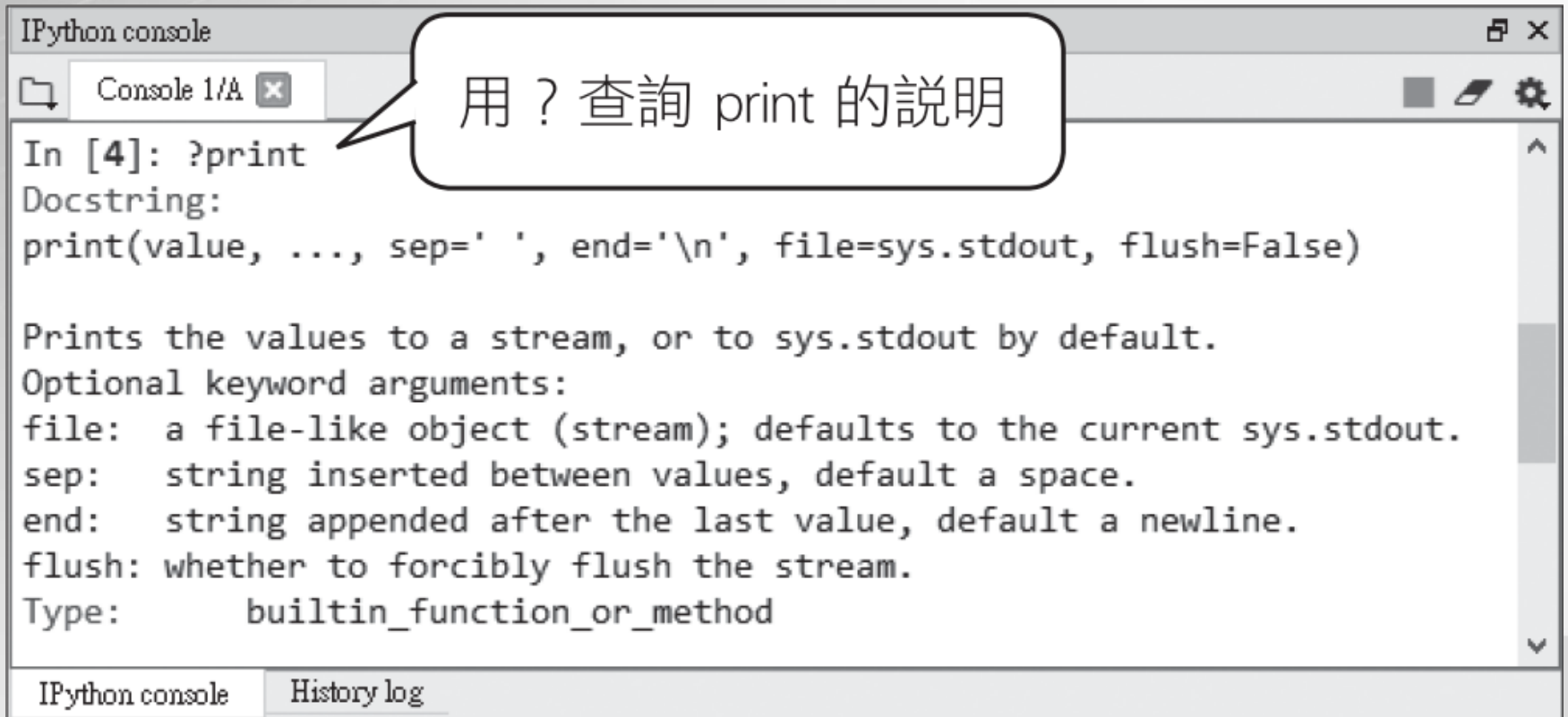
Name	Type	Size	Value
a	int	1	1
b	int	1	2

```
1 a = 1
2 a
3 a + 2
4 print("總和為", a + 2)
5
```

```
In [2]: b = 2
In [3]: b
Out[3]: 2
In [4]:
```



另外, 在 IPython 中也可用 `?xxx` 來查詢 `xxx` 的相關說明



```
IPython console
Console 1/A x
In [4]: ?print
Docstring:
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

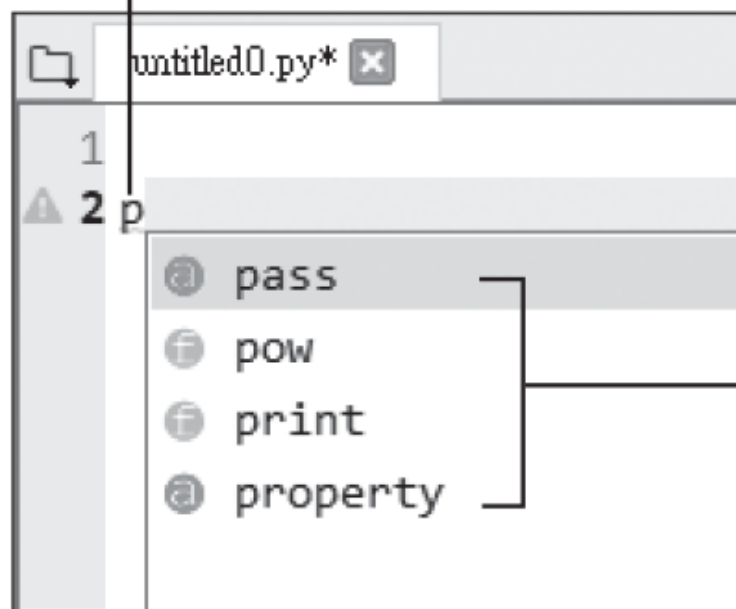
Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file:  a file-like object (stream); defaults to the current sys.stdout.
sep:   string inserted between values, default a space.
end:   string appended after the last value, default a newline.
flush: whether to forcibly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

用 ? 查詢 print 的說明



使用 Spyder 的智慧輔助輸入功能

1 輸入 p 再按 **Tab** 鍵



2 會列出所有可用的關鍵字供你選擇，此時可用 **↑** 或 **↓** 鍵選取要輸入的項目，或繼續輸入、讓 Spyder 幫你選取正確的項目

Python



3 繼續輸入 r

```
1  
2 pr
```

- pass
- pow
- print
- property

4 已選到 print 了, 直接按 Tab 或 Enter 鍵 即完成 print 的輸入

5 接著輸入左括號時, Spyder 又會幫你加上右括號, 並顯示 print() 函式的參數說明供您參考

```
1  
2 print()
```

Arguments

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

0-4 開始寫程式：用 `print()` 把資料顯示出來

ch0-4

- `print()` 是一個好玩又神奇的工具，只要把任何資料放在它的小括號中，就可以將內容顯示出來。

```
print('你好！ Python')
```



你好！ Python

請在 *Spyder* 左邊的程式編輯區中輸入程式並執行，本書會用  或  來標示執行後在 *IPython* 窗格的輸出結果

Python



0-4 開始寫程式：用 `print()` 把資料顯示出來

- 讀者在輸入程式時請特別注意，在程式中英文大、小寫是不同的！因此不可將 `print` 寫成 `Print`，否則執行時會發生「`Print` 未定義」的語法錯誤。我

一步一腳印

```
In [1]: Print(1) ← 輸入開頭大寫的 Print
```

```
Traceback (most recent call last):
```

啊！出錯了...

```
File "<ipython-input-6-66db82d39ad5>", line 1, in <module>
```

```
Print(1)
```

```
NameError: name 'Print' is not defined
```

↑
發生名稱錯誤

↑
IPython 不認得開頭大寫的 Print

學習程式語言最重要的是實際體驗，本書的“一步一腳印”就是希望你能一一的 key 入每一個字，體驗程式執行的每個過程。對了！很有成就感，錯了！印象深刻，學習效果更好。



- 接著我們再來玩點數值資料, 看看 Python 可以顯示多大的數值

Python



0-5 程式的組成單元：敘述

- Python 程式是由敘述 (Statement) 所組成的。敘述是程式中最小的執行單位, 例如前面的「`print('你好！ Python')`」就是一個敘述。

`print(2**3)` ← 未加空格, 感覺有點擠!

`print(2 ** 3)` ← 在 `**` 的前後加上空白, 是不是比較美觀了呢?

`print(1, 2, 3)` ← 若有多個參數(下文說明), 也建議在逗號之後加一個空白

Python



敘述的分行與併行

```
print \  
( '目前敘述被\  
分成' , 4 , \  
'行了' ) # 可以在符號的前後、或字串內部做分行
```



目前敘述被分成 4 行了



```
print(1, #此種方式在各分行  
      2, # 的後面都可加  
      3) # 註解
```



1 2 3

```
print(1); print(2); print(3) ← 以 ; 分隔的多個敘述
```



1
2
3

