

PYTHON 簡介與程式 編輯器介紹

Chih-Chung Hsu (許志仲)
Institute of Data Science
National Cheng Kung University
<https://cchsu.info>



```
),  
lt=1.0,  
  
setting = FloatProperty(  
    "Scale",  
    0.01, max=1000.0,  
    1.0,  
    1.0,  
  
self, context):  
  
folder  
h = (os.path.dirname(self.filepath))  
  
cts selected in the viewport  
selection = bpy.context.selected_objects  
  
rt objects  
list = viewport_selection  
e_selection_setting == False:  
port_list = [i for i in bpy.context.scene.objects]  
  
all objects  
ect.select_all(action='DESELECT')  
  
obj_export_list:  
lect = True  
.type == 'MESH':  
e_path = os.path.join(folder_path, "{}.obj".format(item.name))  
aps.export_scene.obj(filepath=file_path, use_selection=True,  
    axis_forward=self.axis_forward_setting,  
    axis_up=self.axis_up_setting,  
    use_animation=self.use_animation_setting,  
    use_mesh_modifiers=self.use_mesh_modifiers_setting,  
    use_edges=self.use_edges_setting,  
    use_smooth_groups=self.use_smooth_groups_setting,  
    use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,  
    use_normals=self.use_normals_setting,  
    use_uv=self.use_uv_setting,  
    use_materials=self.use_materials_setting,
```

Grading

- 期中考 10%
- 期末大作業 20 %
- 平時作業 80%
 - 課後程式作業
 - 每兩周一次的課堂程式測驗
- 出席率不算分
- 自願擔任小老師 + 5 分總分
- 資訊背景科系與其他背景之同學及格要求不同
 - 資訊背景：做出來就有分
 - 非資訊背景：沒做出來但有認真做，也有分數
 - 協助輔助補救教學一次總分 +2 分，以10分為上限

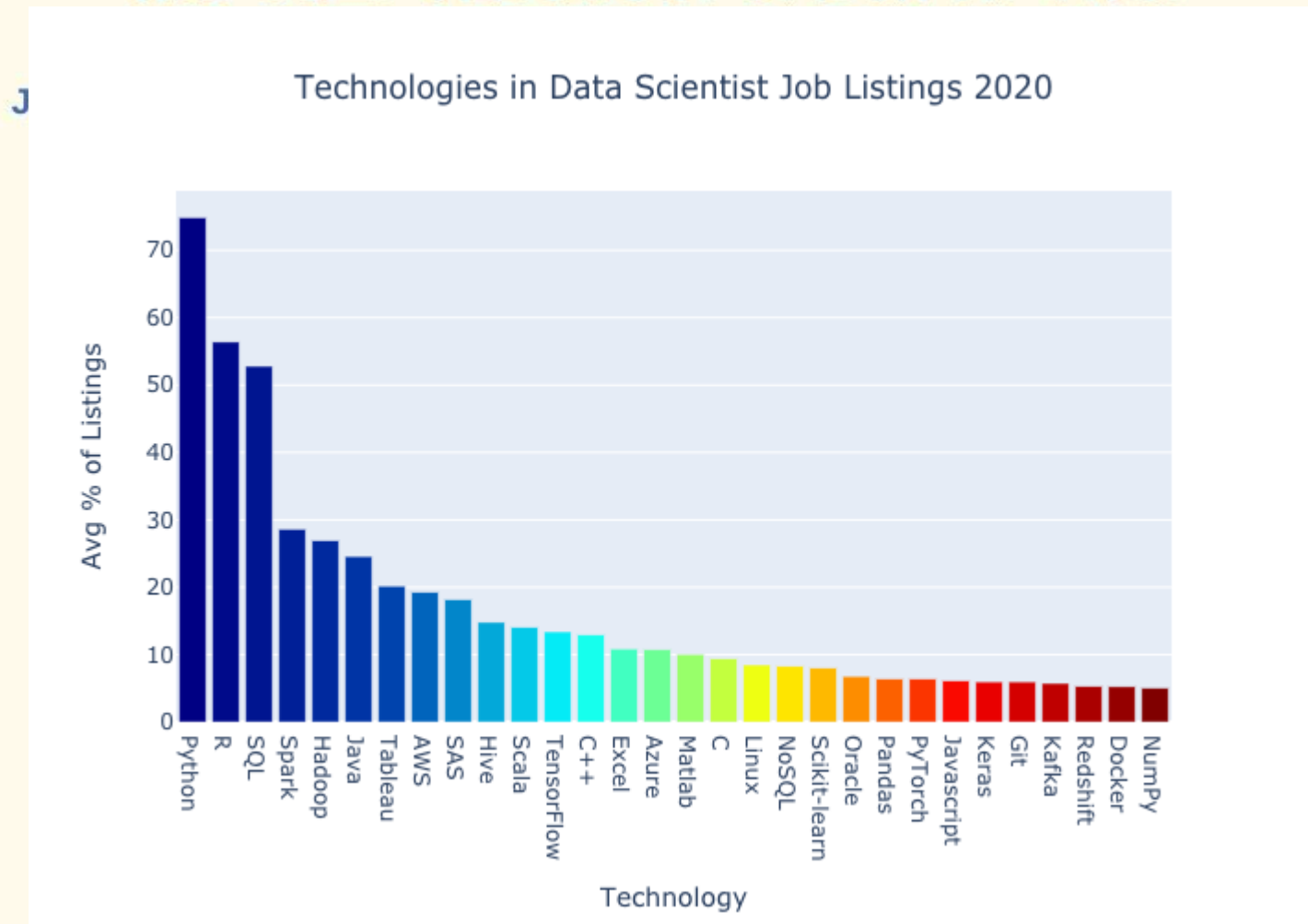
TA info

- 防疫須知：每周都需要透過 Zuvio 點名 (自下周起)，沒來真的不用特別起床開APP點名，不算分的
- 助教資訊 (爾後會放在Moodle上)
 - 李俊毅
 - 廖婉婷
 - 林郁芳
 - 龔修霆
 - 洪慶豪
 - (Office hour will ne announced later!)
- 注意，每2周測驗都是助教現場改分



WHY PYTHON?

熱門程式語言台灣月薪職缺數



Ruby on Rails



資訊素養的重要性

- 沒錯，現在學程式比以前簡單多了
 - 但千萬就不要以為我隨便找人來幫我自己專業做結合，這樣就可以了
 - 沒有有效的溝通，產品=無用XD
- 什麼叫有效的溝通？
 - 你跟Programmer說話時，你要聽的懂關鍵術語
 - 那你就要會基礎的Coding

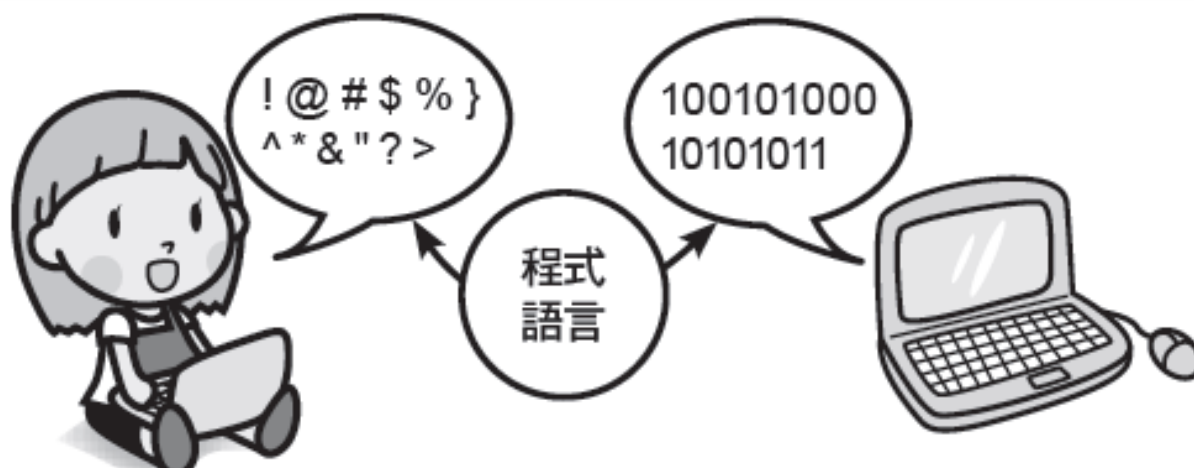
Outline

- 認識程式設計與Python
- 1.1 認識程式語言與程式設計
- 1.2 Python 程式語言簡介
- 1.3 建置 Anaconda 開發環境
- 1.4 Spyder 編輯器
- 重點整理

1.1 認識程式語言與程式設計

■ 程式語言的重要性

- 電腦的基本構成是一大群極小的電子開關，只要將這些開關設定成不同的組合，就能讓電腦執行作業。但是電腦並不懂人類的語言，它只認識 0 與 1 來代表開關的啟停，雖然效率極高，但對於人類來說實在是太難以理解了！所以如果要進行溝通，人類就必須利用程式語言進行轉譯的動作，將程式由自然語言轉為電腦看得懂的指令進行工作。



程式設計的運作模式

- 程式設計的運作模式

- 簡單來說：程式設計(programming)，是針對電腦要解決的問題提供處理方式的過程，也是軟體開發時的重要步驟。程式設計必須以某種程式設計語言為工具，依照語法的規定與功能安排執行工作的順序與流程，控制電腦進行相關的工作。程式設計過程應包括分析問題、設計流程、程式編碼、功能測試、程式除錯等不同階段，設計時必須根據問題的需求，依照上述的階段進行，一直到問題解決為止。



1.2 Python 程式語言簡介

- 1.2.1 Python 程式語言發展史

- 1989 年 12 月，吉多范羅蘇姆於荷蘭國家數學及計算機科學研究所開發出 Python 程式語言，Python 擁有 C 語言的強大功能，能夠全面調用電腦的各種功能接口，同時容易學習及使用，又具備良好的擴展性。1991 年推出第一個 Python 編譯器後，受到廣大程式設計師的喜愛。

■ 1.2.2 Python 程式語言的特色

- 簡單易學
- 免費且開源
- 高階程式語言
- 可移植性
- 直譯式程式語言
- 可嵌入性
- 豐富且多元的套件

1-1 Python Versions

- 2000年10月16日發布Python 2.0
- 2008年12月 日發布Python 3.0
 - 此版不完全相容於Python 2.0
 - Python2.0只到Python 2.7
- 所有新功能都加入到Python 3.0以後的版本

1-1 Python 簡介

- Python是支援程序導向、物件導向的動態語言
 - 動態語言不需事先宣告變數的資料型別，變數的資料型別可以在執行時再指定
- Python使用直譯器執行程式
 - 直譯器從頭到尾一行接著一行執行程式碼，又稱作腳本語言(scripting language)，不需要編譯就可以執行

1-1 Python 簡介

- Python提供許多標準函式庫，又稱內建電池(batteries included)，並有許多第3方模組(third-party module)可以使用
- Python使用縮排方式表達程式區塊，語法直覺而簡單
- Python常用於字串處理、數學運算、科學計算、系統管理、網頁框架、大數據分析與網頁分析等

1-1 Python 簡介

- Python具備垃圾回收(garbage collection) 功能，會自動管理記憶體，回收沒有使用的記憶體
- Python中變數、數字、字串、函式、模組都是物件，完全支援物件導向的程式設計
- Python能夠結合C與C++語言所撰寫的擴充程式
 - 使用Python將C與C++語言所撰寫的程式整合起來，因此Python又稱為膠水語言(glue language)

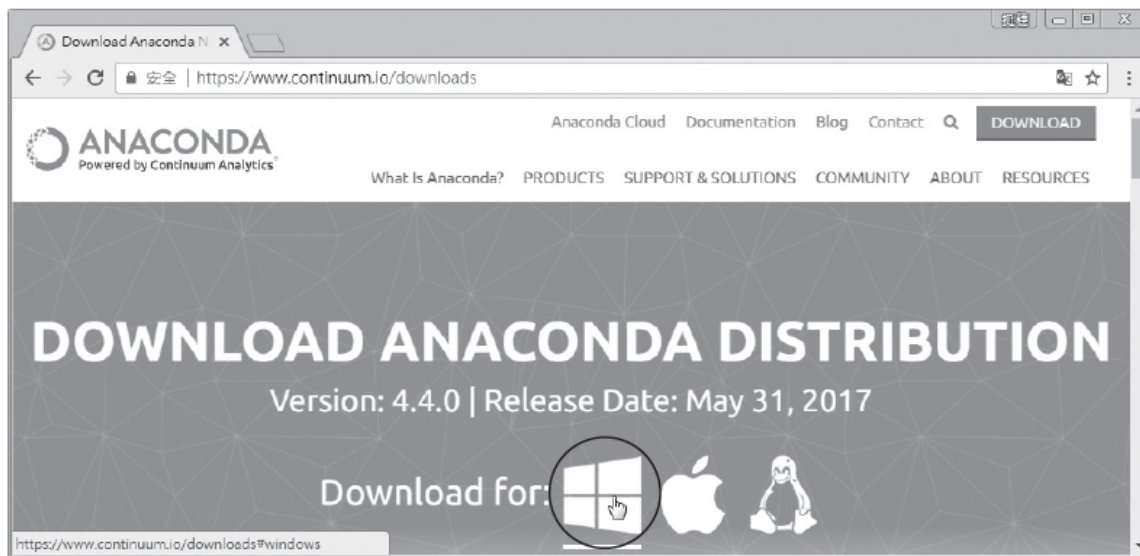
1.3 建置 Anaconda 開發環境

■ 1.3.1 安裝 Anaconda 套件組

- Anaconda 套件組擁有下列特點，使其成為初學者最適當的 Python 開發環境：
 - 內建眾多流行的科學、工程、數據分析的 Python 套件。
 - 完全免費及開源。
 - 支援 Linux、Windows 及 Mac 平台。
 - 支援 Python 2.x 及 3.x，且可自由切換。
 - 內建 Spyder 編譯器。
 - 包含 jupyter notebook 環境。

安裝 Anaconda 的步驟

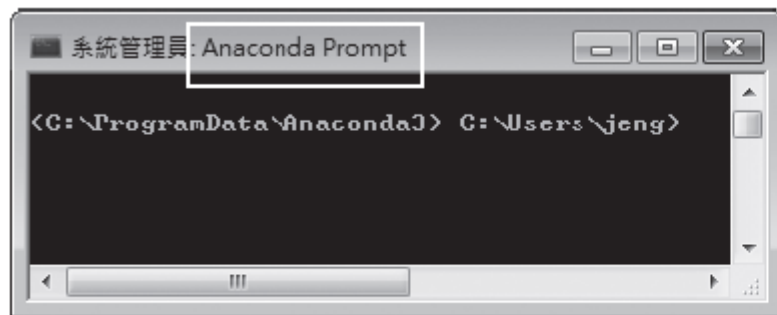
- 在瀏覽器開啟 Anaconda 官網
「<https://www.continuum.io/downloads>」下載頁面，於
DOWNLOAD ANACONDA DISTRIBUTION 下方點選 Windows 系統
圖示。



查看套件

■ 1.3.2 Anaconda Prompt 查看套件

- Anaconda Prompt 命令視窗類似 Windows 系統「命令提示字元」，可以輸入命令，按 Enter 鍵就會執行。Anaconda Prompt 命令視窗會在標題列顯示「Anaconda Prompt」，做為與 Windows 系統「命令提示字元」的區別。Anaconda Prompt 命令視窗預設路徑為 <C:\Users\ 電腦名稱>。
- 執行 開始 / 所有程式 / Anaconda3 (64-bit) / Anaconda Prompt 即可開啟 Anaconda Prompt 命令視窗。



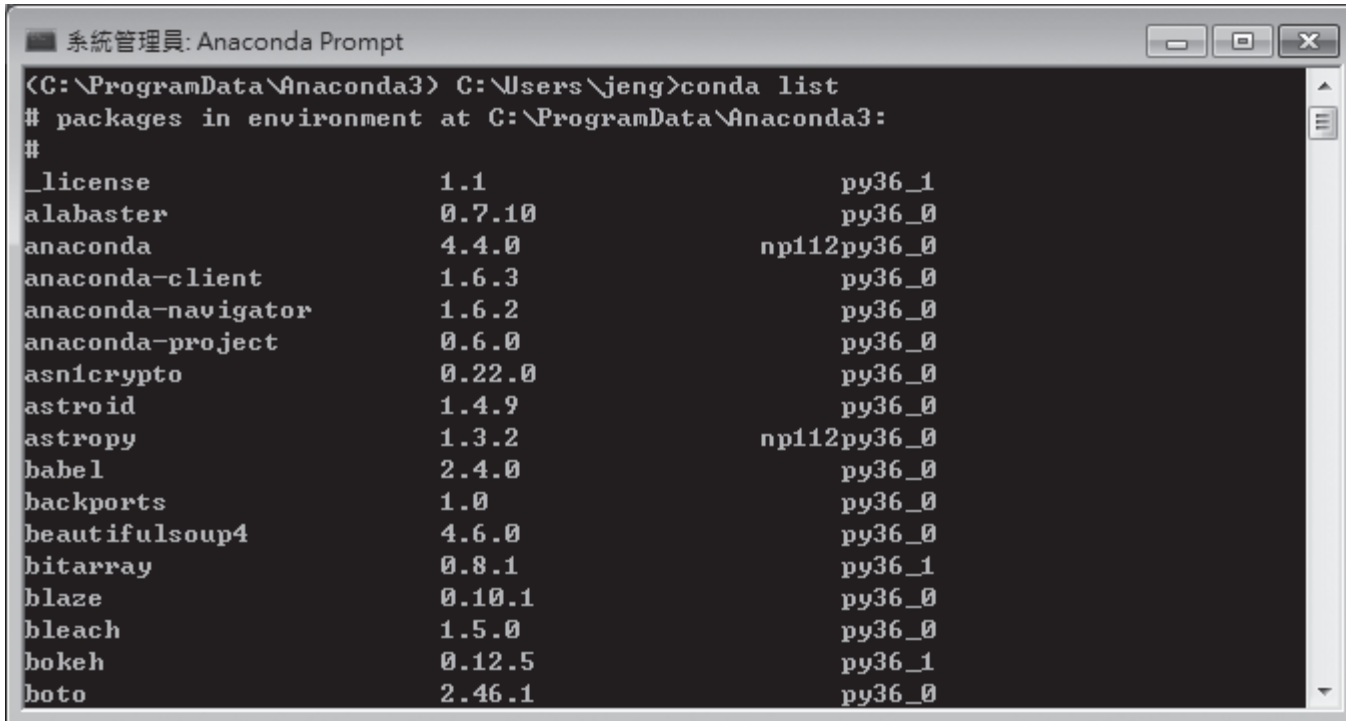
▲ Anaconda Prompt



▲ 命令提示字元

查看已安裝的套件

- 顯示 Anaconda 已安裝套件的命令為：`conda list`
- 命令視窗會按照字母順序顯示已安裝套件的名稱及版本：



The screenshot shows a Windows command prompt window titled "系統管理員: Anaconda Prompt". The command prompt shows the command `conda list` being executed in the directory `C:\ProgramData\Anaconda3`. The output lists installed packages in alphabetical order, including their names, versions, and build identifiers.

```
<C:\ProgramData\Anaconda3> C:\Users\jeng>conda list
# packages in environment at C:\ProgramData\Anaconda3:
#
_license                  1.1                      py36_1
alabaster                 0.7.10                  py36_0
anaconda                  4.4.0                   np112py36_0
anaconda-client           1.6.3                   py36_0
anaconda-navigator        1.6.2                   py36_0
anaconda-project          0.6.0                   py36_0
asn1crypto                0.22.0                  py36_0
astroid                   1.4.9                   py36_0
astropy                   1.3.2                   np112py36_0
babel                     2.4.0                   py36_0
backports                 1.0                      py36_0
beautifulsoup4            4.6.0                   py36_0
bitarray                  0.8.1                   py36_1
blaze                     0.10.1                  py36_0
bleach                    1.5.0                   py36_0
bokeh                     0.12.5                  py36_1
boto                       2.46.1                  py36_0
```

交談式命令視窗

■ 1.3.3 IPython 交談式命令視窗

- 執行 開始 / 所有程式 / Anaconda3 (64-bit) / IPython 即可開啟 IPython 命令視窗，在 IPython 命令視窗中輸入 Python 程式碼，按 Enter 鍵後會立刻執行並顯示執行結果。
- 每一列程式碼皆有延續性，例如下圖為設定兩個變數，再列印兩數總和：



```
IPython: C:\jeng\Documents
In [2]: a = 12
In [3]: b = 34
In [4]: print("總和：" + str(a + b))
總和：46
```

其他特色

- 重覆使用程式碼

- 如果要輸入的程式碼與曾經輸入過的程式碼雷同，可以修改曾經輸入過的程式碼。按「↑」鍵可顯示上一列程式碼，按「↓」鍵可顯示下一列程式碼，找到程式碼後加以修改，按 Enter 鍵就會執行。

- 觀看全部程式碼

- 當程式碼數量較多時，常會忘記前面曾經輸入過的程式碼，此時可執行「history」命令觀看全部程式碼。

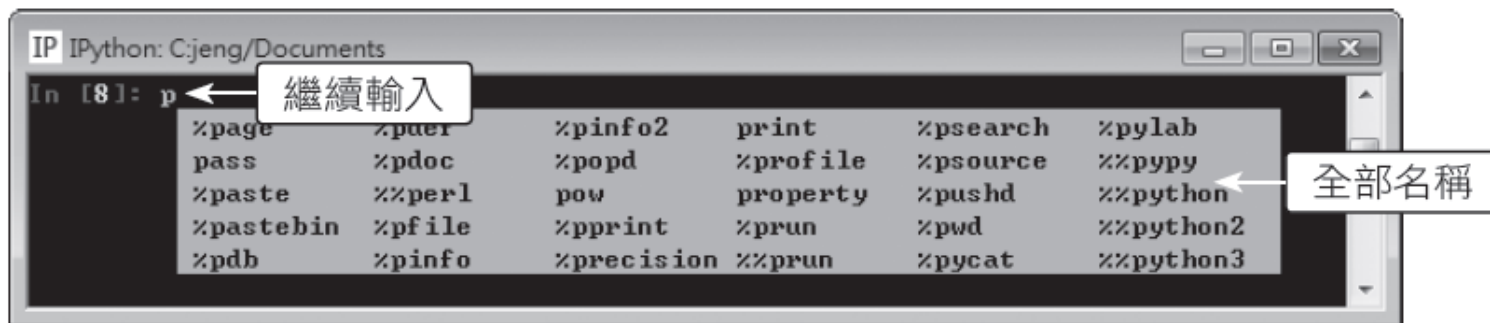
- 查詢使用說明

- IPython 命令視窗提供非常強大的查詢使用說明功能，只在變數、命令、函式、套件等名稱後面加上「？」，就會顯示該項目的使用說明。

執行程式

- 簡易智慧輸入

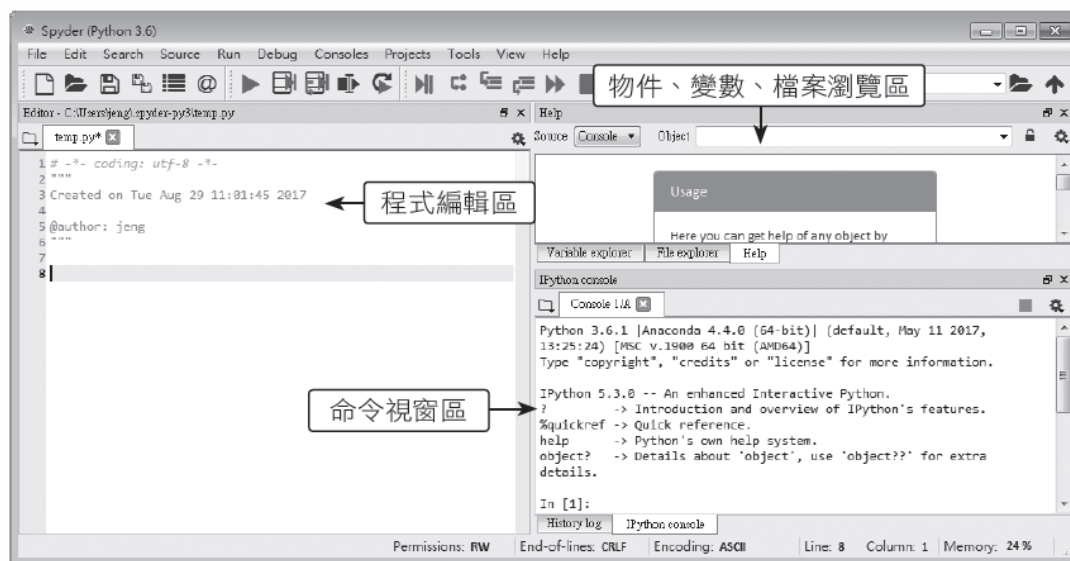
- 使用者在 IPython 命令視窗輸入部分文字後按 Tab 鍵，如果包含輸入文字的名稱超過一個，系統會列出所有名稱讓使用者參考，使用者可以繼續輸入。例如輸入「p」後按 Tab 鍵：



%run Python 程式檔案路徑

1.4 Spyder 編輯器

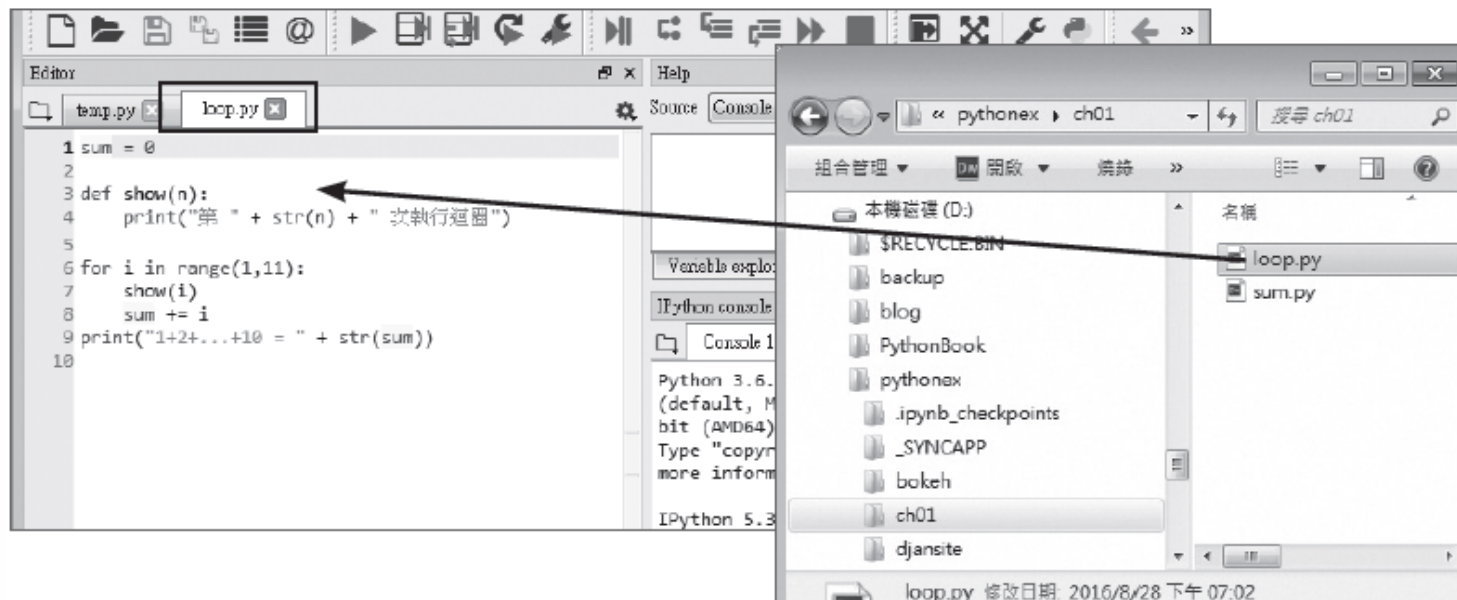
- Anaconda 內建 Spyder 做為開發 Python 程式的編輯器。在 Spyder 中可以撰寫及執行 Python 程式，Spyder 還提供簡單智慧輸入及強悍的程式除錯功能。另外，Spyder 也內建了 IPython 命令視窗。本書所有範例皆使用 Spyder 編輯及執行。
- 1.4.1 啟動 Spyder 編輯器及檔案管理
 - 執行 開始 \ 所有程式 \ Anaconda3 (64-bit) \ Spyder 即可開啟 Spyder 編輯器。



如何編輯檔案?

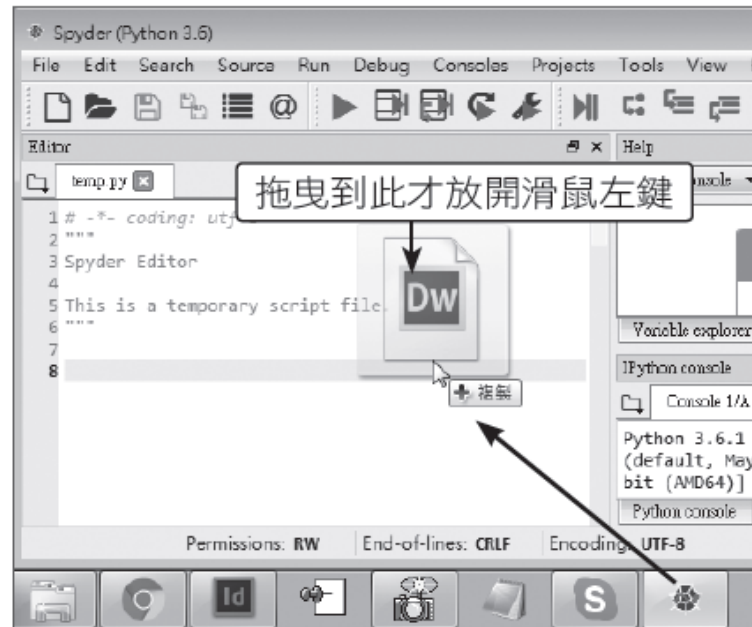
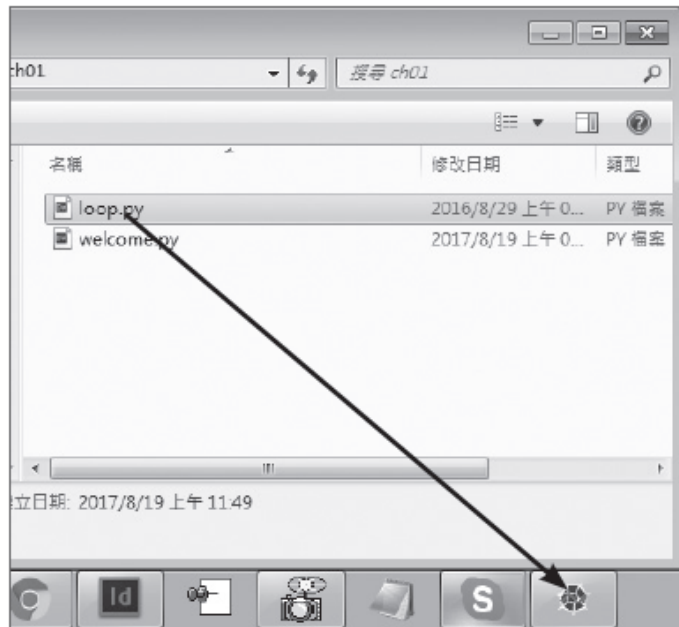
■ 檔案開啟

- 要開啟已存在的 Python 程式檔，可執行 File \ Open 或點選工具列鈕，於 Open file 對話方塊點選檔案即可開啟。
- Spyder 另外提供兩種快速開啟檔案的方法：第一種是從檔案總管將檔案拖曳到 Spyder 程式編輯區就會開啟該檔案，例如拖曳 <d:\pythonex\ch01\loop.py> 到 Spyder 程式編輯區：



如何編輯檔案？

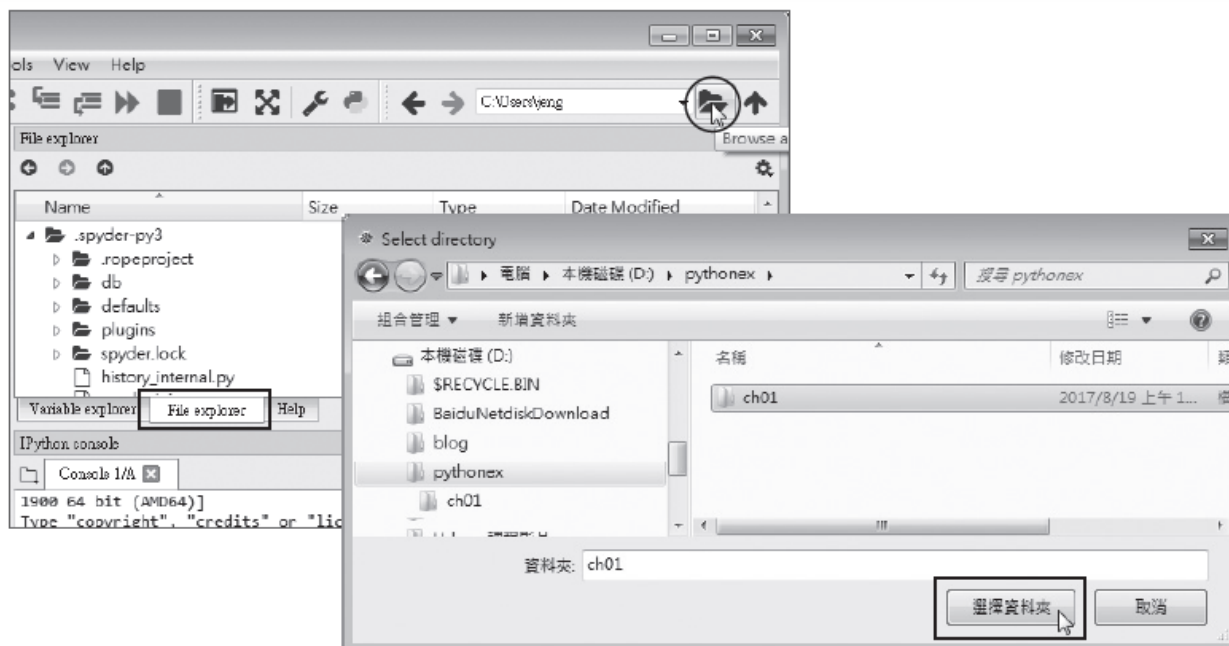
- 第二種方法更方便，從檔案總管將檔案拖曳到快速啟動列 Spyder 圖示就會切換到 Spyder 應用程式，再拖曳到 Spyder 程式編輯區即會開啟該檔案，例如拖曳 <d:\pythonex\ch01\loop.py>：



如何編輯檔案?



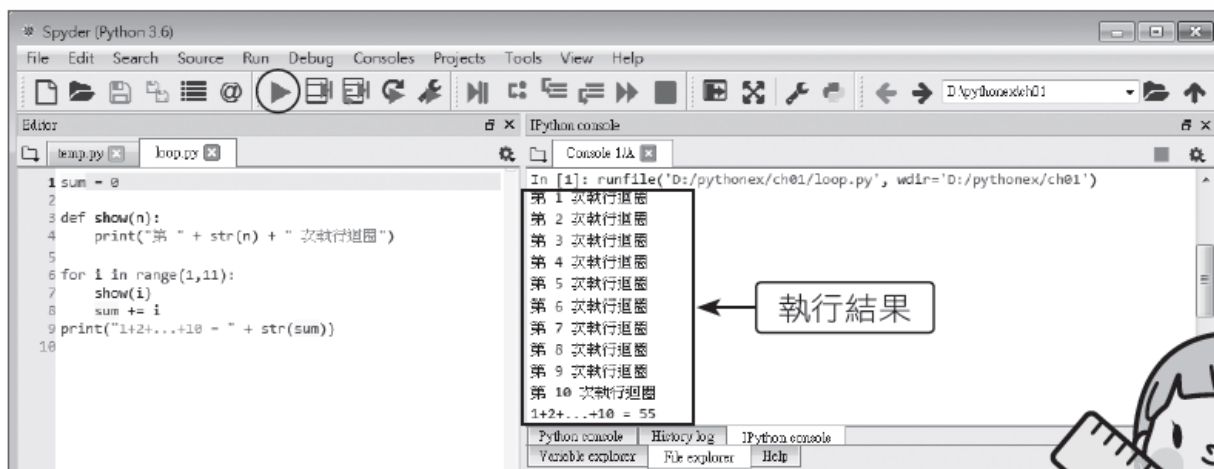
- File Explorer (檔案瀏覽器) 面板
 - 在右上方面板區點選 File Explorer 頁籤切換到 檔案瀏覽器 面板，按 右上角鈕 開啟 Select directory 對話方塊，選取資料夾 (此處為 <d:\pythonex\ch01>) 後按 選擇資料夾 鈕。



要跑程式了!!

■ 執行程式

- 執行 Run \ Run 或點選工具列  鈕就會執行程式，執行結果會在命令視窗區顯示，例如下圖為 <loop.py> 的執行結果。



```
1 sum = 0
2
3 def show(n):
4     print("第 " + str(n) + " 次執行迴圈")
5
6 for i in range(1,11):
7     show(i)
8     sum += i
9 print("1+2+...+10 = " + str(sum))
10
```

In [1]: runfile('D:/pythonex/ch01/loop.py', wdir='D:/pythonex/ch01')

第 1 次執行迴圈
第 2 次執行迴圈
第 3 次執行迴圈
第 4 次執行迴圈
第 5 次執行迴圈
第 6 次執行迴圈
第 7 次執行迴圈
第 8 次執行迴圈
第 9 次執行迴圈
第 10 次執行迴圈
1+2+...+10 = 55

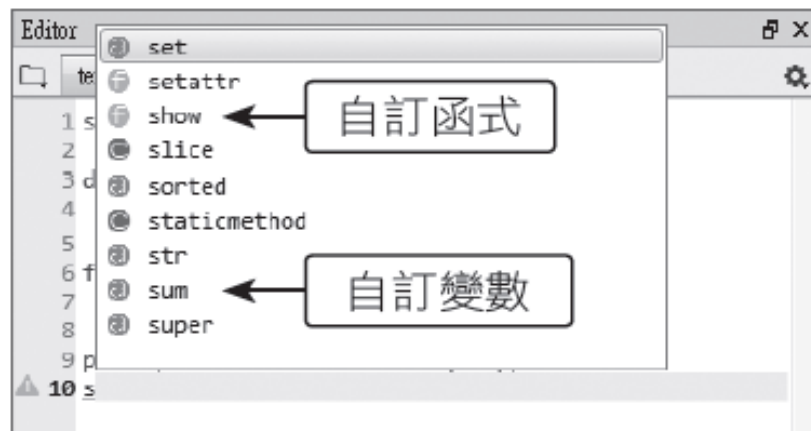
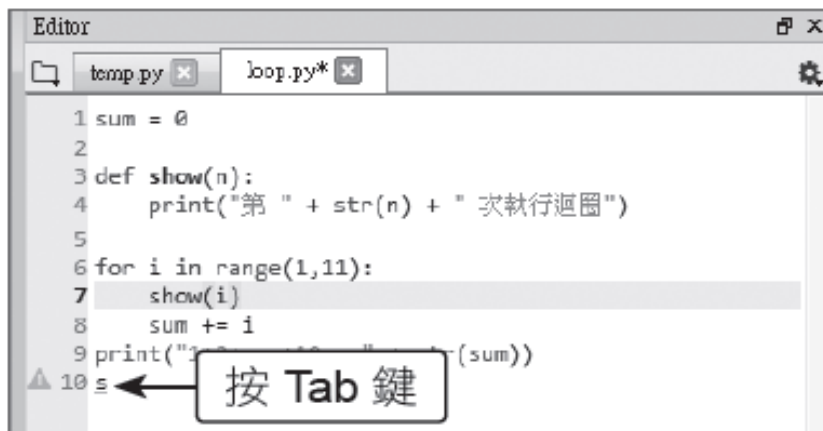
執行結果



Spyder特色


■ 1.4.2 Spyder 簡易智慧輸入

- Spyder 簡易智慧輸入功能與 IPython 命令視窗雷同，但操作方式比 IPython 命令視窗方便。使用者在 Spyder 程式編輯區輸入部分文字後按 Tab 鍵，系統會列出所有可用的項目讓使用者選取，列出項目除了內建的命令外，還包括自行定義的變數、函式、物件等。例如在 <loop.py> 輸入「s」後按 Tab 鍵：



檢查程式碼的錯誤

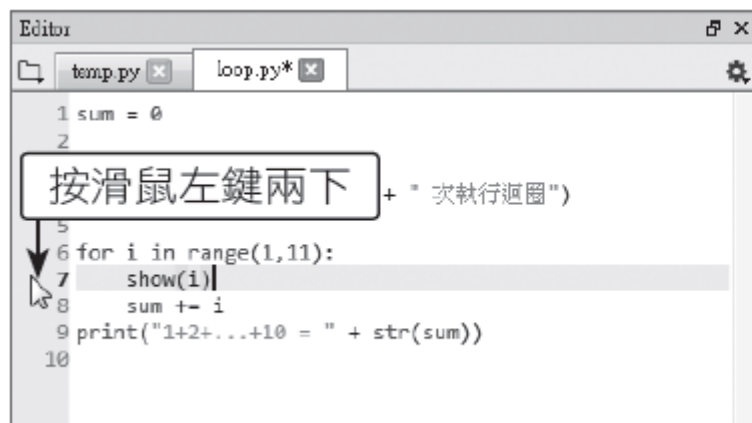
■ 1.4.3 程式除錯

- 於 Spyder 輸入 Python 程式碼時，系統會隨時檢查語法是否正確，若有錯誤會在該列程式左方標示  圖示；將滑鼠移到  圖示片刻，會提示錯誤訊息。



除錯的好用工具







- Spyder 的除錯工具相當強大。
- 首先為程式設定中斷點：設定的方式為點選要設定中斷點的程式列，按 F12 鍵；或在要設定中斷點的程式列左方快速按滑鼠左鍵兩下，程式列左方會顯示紅點，表示該列為中斷點。程式中可設定多個中斷點。



除錯功能

■ Spyder除錯工具列



- ：以除錯方式執行程式。
- ：單步執行，不進入函式。
- ：單步執行，會進入函式。
- ：程式繼續執行，直到由函式返回或下一個中斷點才停止執行。
- ：程式繼續執行，直到下一個中斷點才停止執行。
- ：終止除錯模式回到正常模式。

重點整理

- 在本章中介紹了下面的重點：
 - Python 程式語言是由吉多范羅蘇姆 (Guido van Rossum) 所創建，是一種物件導向、直譯式的電腦程式語言。
 - 本書以 Anaconda 套件組做為開發環境，不但包含超過 300 種常用的科學及資料分析套件，還內建 Spyder (IDLE 編輯器加強版) 編輯器。
 - Anaconda Prompt 命令視窗類似 Windows 系統「命令提示字元」，可以輸入命令，按 Enter 鍵就會執行。
 - 顯示 Anaconda 已安裝套件的命令為：`conda list`
 - 在 IPython 命令視窗中輸入 Python 程式碼，按 Enter 鍵後會立刻執行並顯示執行結果。

小試身手：Python 的輸入與輸出

- 首先練習一下輸入輸出，測試安裝環境是否正常
 - 第一題：
 - 安裝完成後請按 Windows鈕 並鍵入 “anaconda”
 - 在裡面鍵入python 進入互動模式
 - 第二題：
 - 按 Windows鈕 並鍵入 “spyder”
 - 開啟Spyder
 - 第三題：
 - 開啟瀏覽器，搜尋Online python complier
 - 成功進入程式編輯畫面
- 測試目標
 - Python3語言中最常用輸入與輸出的函式為input與print

表 1-1 函式 print 使用方式

函式 print	說明	範例與執行結果
print(*objects)	將 objects 顯示在螢幕上，若沒有指定間隔字元與結束字元，預設間隔字元 (sep) 使用一個空白鍵，一行的結束字元 (end) 為「\n」表示換到下一行。	<p>name = input(' 請問貴姓大名？ ')</p> <p>說明 預設 sep 使用一個空白鍵，所以每一句話之間都有一個空白鍵。</p> <p>執行結果</p> <pre>學習 Python 真有趣</pre>
print(*objects, sep=' ', end='\n')	將 objects 顯示在螢幕上，預設間隔字元 (sep) 為一個空白鍵，一行的結束字元 (end) 為「\n」表示換到下一行，sep 與 end 都可以依照使用者需求而重新定義。	<p>print(' 學習 ', 'Python', ' 真有趣 ', sep='\t')</p> <p>說明 修改 sep 使用「\t」，所以每一句話之間使用一個 tab 鍵，間隔就改成 tab 鍵。</p> <p>執行結果</p> <pre>學習 Python 真有趣</pre>

表 1-2 函式 input 使用方式

函式 input	說明	範例與執行結果
input([prompt])	顯示 prompt 在螢幕上，等待使用者輸入資料，資料會以字串回傳。	<pre>name = input(' 請問貴姓大名？ ') print(' 你好，', name)</pre> <p>說明 因為函式 input 將輸入的資料使用字串回傳，姓名是字串型別所以不用修改，將結果指定給物件 name，使用函式 print 顯示「你好」與物件 name 在螢幕上。物件 name 與運算子「=」的概念將於下一章介紹。</p> <p>執行結果</p> <pre>請問貴姓大名？ John 你好， John</pre>

1-4 第一個Python 程式

- 範例1-1 輸入與輸出範例 — 基本資料調查

- ch1\1-4- 基本資料調查.py

- 問題敘述

- 寫一個程式，螢幕輸出「請問貴姓大名？」，等待使用者輸入姓名，顯示輸入的姓名在螢幕上。螢幕輸出「請問年紀？」，等待使用者輸入年紀，顯示輸入的年紀在螢幕上。螢幕輸出「請問體重？」，等待使用者輸入體重，顯示輸入的體重在螢幕上。

1-4 第一個Python 程式

- 解題想法

- 這個程式需要使用input 與print 兩個函式，函式input 用於輸入資料，函式print 用於顯示資料到螢幕

```
1 name = input(' 請問貴姓大名？ ')\n2 print(' 你好，', name)\n3 y = int(input(' 請問年紀？ '))\n4 print(' 原來你 ', y, ' 歲 ')\n5 w = float(input(' 請問體重？ '))\n6 print(' 體重為 ', w)
```

1-4 第一個Python 程式

- 程式解說

- **第1 行**：在螢幕上顯示「請問貴姓大名？」，等待使用者輸入，輸入的資料指定給物件name，物件name 與運算子「=」的概念將於下一章介紹
- **第2 行**：輸出「你好，」與物件name在螢幕上
- **第3 行**：在螢幕上顯示「請問年紀？」，等待使用者輸入，輸入的資料經由函式int將字串轉換成整數指定給物件y

1-4 第一個Python 程式

- 第4 行：輸出「原來你」、物件y 與「歲」在螢幕上
- 第5 行：在螢幕上顯示「請問體重？」，等待使用者輸入，輸入的資料經由函式float 將字串轉換成浮點數指定給物件w
- 第6 行：輸出「體重為」與物件w在螢幕上

1-4 第一個Python 程式

- 執行結果

請問貴姓大名？ John

你好， John

請問年紀？ 16

原來你 16 歲

請問體重？ 64.4

體重為 64.4

充電時間

- 若想要知道函式的功能可以使用函式[help](#)，以函式名稱為輸入，找出該函式的說明
 - 例如：`help(input)`，獲得函式 的解說

`input(prompt=None, /)`

Read a string from standard input. The trailing newline is stripped.

The prompt string, if given, is printed to standard output without a trailing newline before reading input.

If the user hits EOF (*nix: Ctrl-D, Windows: Ctrl-Z+Return), raise EOFError. On *nix systems, readline is used if available.

充電時間

■ 程式的註解

- 使用井字號「#」之後的文字都會被視為註解，使用註解讓後續維護程式的工程師，更容易了解程式
- 如下圖，其中「# 輸入姓名到name」為註解

```
name = input(' 請問貴姓大名? ') # 輸入姓名到 name  
print(' 你好，', name)
```