Chapter 1

安裝程式工具、基本語法介紹

安裝Python (1/2)

1. 到 Python 官網,下載安裝檔。https://www.python.org/





【註】在開始之前,如果曾經有在電腦安裝過 Python 的話,可以先到

「C:\Users\使用者名稱\AppData\Local\Programs」

將 Python 資料夾刪掉,不然有可能在安裝完後,遇到裡面沒有 pip.exe 的狀況。

安裝Python (2/2)

開啟命令提示字元(或終端機)

- 1. 輸入「python -V」,可查看目前安裝的 Python 版本。
- 2. 輸入「pip list」,可察看目前電腦已安裝的套件。

```
Microsoft Windows [版本 10.0.22621.2215]
(c) Microsoft Corporation. 著作權所有,並保留一切權利。
C:\Users\User>python -V
Python 3.11.4
```

C:\Users\User>pip	list
Package	Version
attrs	23.1.0
beautifulsoup4	4.12.2
bs4	0.0.1
certifi	2023.7.22
cffi	1.15.1
charset-normalizer	3.2.0
contourpy	1.1.0
cycler	0.11.0
exceptiongroup	1.1.3
fonttools	4.42.1
h11	0.14.0
idna	3.4
kiwisolver	1.4.5
lxml	4.9.3
matplotlib	3.7.2
numpy	1.25.2
outcome	1.2.0
packaging	23.1
Pillow	10.0.0
pip	23.1.2

安裝Spyder

到 Spyder 的官網,下載安裝檔並執行。

https://www.spyder-ide.org/

安裝完畢後就可以開啟使用啦~!

HOME OVERVIEW COMPONENTS PLUGINS DOWNLOAD

Ready to give Spyder a try? Let's get started!

Want to join the community of scientists, engineers and analysts all around the world using Spyder? Click the button below to download the suggested installer for your platform; we offer standalone installers on Windows and macOS. For Linux, we recommend the cross-platform Anaconda distribution, which includes Spyder and many other useful packages for scientific Python. You can also try out Spyder right in your web browser by launching it on Binder.

For a detailed guide on the many different methods of obtaining Spyder, please refer to our full installation instructions, and check out our release page for links to all our installers. These approaches are generally intended for experienced users and those with specific needs, so we recommend sticking with the recommended installer unless you have a specific reason to go with another. Happy Spydering!

macOS Big Sur users: Full support for macOS 11 Big Sur will be included in Spyder 4.2.1, scheduled for release on December 18, 2020. However, see our FAQ question on Big Sur for how to get it working right now.



2. 再點這裡下載

撰寫您的第一行Python程式語言

Python的輸出非常簡單,即為print()函數。將您想輸出的東西放在括號裡。

```
untitled0.py* X
       # -*- coding: utf-8 -*-
       Created on Thu Sep 14 17:22:37 2023
       @author: User
       print("嗨~我叫陳彥碩", "英文名字叫Daniel<mark></mark>")
  8
   Console 1/A 🗙
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 8.12.2 -- An enhanced Interactive Python.
In [1]: runfile('C:/Users/User/untitled0.py', wdir='C:/Users/User')
嗨~我叫陳彥碩 英文名字叫Daniel
```

認識變數 (1/2)

- 1. 變數在程式中,扮演中儲存資料的角色。將資料儲存下來,就可在需要的時候拿出來用。
- 2. 在程式中,每個變數會有自己的型別。那型別是什麼呢? 簡單來說,型別決定了這個變數可用來儲存什麼樣格式的資料,並且定義了相對應的操作。
- 3. Python中常見的基本型別有: int (整數)、float (浮點數)、str (字串)。
- 4. 在Python中,宣告變數並不用事先給定型別,他會依照你給他的初始值來決定變數的型別。

認識變數 (2/2)

範例

```
      1
      # -*- coding: utf-8 -*-

      2
      """

      3
      Created on Thu Sep 14 17:22:37 2023

      4
      5

      6
      """

      7
      x = 120
      # 令x代表120這個數字

      9
      print(x*3)
      # 請輸出3x這個數字
```

【註】在程式碼後面加上#為寫入註解。

In [3]: runfile('C:/Users/User/untitled0.py', wdir='C:/Users/User')
360

數學運算子

運算子的符號及功能如右:

【範例】

複利公式:

本利和 = 本金(1+年利率) ^ 年

若本金為100000元,年利率3%,則

7年後可拿回多少錢?

運算子	功能
x + y	X力□Y
x - y	X減Y
x * y	X乘Y
x / y	X除以Y
x // y	X除以Y,只取整數解
x % y	求X除以Y的餘數
x ** y	X的Y次方

流程控制 if – else 敘述

達成什麼條件(if)要輸出什麼東西

未達成(else)要輸出什麼東西

運算子	效果
x < y	X是否小於Y
x <= y	X是否小於等於Y
x > y	X是否大於Y
x >= y	X是否大於等於Y
x == y	X是否等於Y
x != y	X是否不等於Y

In [7]: runfile('C:/Users/User/untitled1.py', wdir='C:/Users/User')
False!!

流程控制 if-elif-else 敘述(1/2)

多個判斷條件時,可以使用 elif(else if 的縮寫) 敘述多個條件。

```
untitled0.py* X
                untitled1.py* X
        # -*- coding: utf-8 -*-
        Created on Thu Sep 14 17:22:37 2023
        @author: User
        grade = 85
  10
        if grade >= 80:
  11
            print("Great!!")
  12
        elif grade >= 60:
  13
            print("Pass!!")
  14
        else:
            print("False!!")
  15
```

```
In [10]: runfile('C:/Users/User/untitled0.py', wdir='C:/Users/User')
False!!
In [11]: runfile('C:/Users/User/untitled0.py', wdir='C:/Users/User')
Pass!!
In [12]: runfile('C:/Users/User/untitled0.py', wdir='C:/Users/User')
Great!!
```

流程控制 if-elif-else 敘述(2/2)

多個判斷條件時,可以使用 elif(else if 的縮寫) 敘述多個條件。

```
C:\Users\User\untitled0.py
    untitled0.py* X
                untitled1.py* X
        # -*- coding: utf-8 -*-
       Created on Thu Sep 14 17:22:37 2023
       @author: User
       x = input("請輸入成績:")
                                 # 變數x代表輸入的數字
       grade = int(x)
                                  # 變數grade 代表變數x的型態為整數
        if grade >= 80:
  11
  12
            print("Great!!")
  13
        elif grade >= 60:
  14
           print("Pass!!")
       else:
  15
            print("False!!")
```

```
In [18]: runfile('C:/Users/User/untitled0.py', wdir='C:/Users/User') 請輸入成績:90 Great!!
In [19]: runfile('C:/Users/User/untitled0.py', wdir='C:/Users/User') 請輸入成績:72 Pass!!
```

布林運算

邏輯判斷時使用布林運算:

運算子	效果
a or b	A或B其中一個條件成立就回傳True
a and b	A或B兩個條件都成立才回傳True
not A	如果A為True,則回傳False,反之則回傳True

```
untitled0.py* X
             untitled1.py* X
     # -*- coding: utf-8 -*-
     Created on Thu Sep 14 17:22:37 2023
     Qauthor: User
     x = input("請輸入程式語言成績:")
                                     # 變數x代表輸入的數字
     y = input("請輸入品質管理成績:")
     a = int(x)
                          # 變數grade 代表變數x的型態為整數
11
     b = int(y)
12
13
     if a >= 80 or b >= 80:
         print("Great!!")
15
     elif a >=60 or b >=60:
         print("nice!!")
     else:
         print("False!!")
```

```
In [20]: runfile('C:/Users/User/untitled0.py', wdir='C:/Users/User') 請輸入程式語言成績:83 請輸入品質管理成績:62 Great!!

In [21]: runfile('C:/Users/User/untitled0.py', wdir='C:/Users/User') 請輸入程式語言成績:40 請輸入品質管理成績:70 nice!!
```

練習題

1. 您有一筆5萬元的金額,若年利率 3.5%且使用複利計算的話,8年後可以拿回多少錢?

- 2. 一遊樂園的門票售價為100元,但滿足下列條件時會有折扣:
 - (1) 大於等於80歲或小於等於6歲,門票售價打5折
 - (2) 大於等於60歲或小於等於12歲,門票售價打8折

請撰寫一程式輸入年齡後可算出門票價格。