

# 主題觀念二：基本程式設計（二） - 資料

---

by 田弘華 Hung-Hua Tien

## 1. 寫程式是怎麼一回事

### 1-1. 舉例說明

例如：我們想記錄電影的資訊，這可以動筆也可以動電腦，動電腦可以用 EXCEL 軟體，也可以自己寫程式。

<https://www.imdb.com/title/tt4154796/>

將上述電影的名稱(movie\_title)、上映時間(release\_date)、片長(movie\_time)、評分(movie\_rating)等資訊印出來？

- 以 = 將資訊儲存至一個物件名稱中

```
1 movie_title = "Avengers: Endgame"
2 release_date = "26 April 2019(USA)"
3
4 print(movie_title)
5 print(release_date)
```

- 以 # 對程式功能做註解

```
1  # 儲存資訊 <--- 註解
2  movie_title = "Avengers: Endgame"
3  release_date = "26 April 2019(USA)"
4
5  # 印出資訊 <--- 註解
6  print(movie_title)
7  print(release_date)
```

在這個例子中，我們用寫程式的方式記錄了電影的相關資訊。同學可能會覺得寫這個程式有什麼用？可是，如果同學要處理的是一萬筆電影資料、一百萬筆電影資料，請問用電腦Excel軟體按菜單操作，你要做多久？一天、兩天，還是一星期才能做完，而且因為手動容易錯誤百出！若是寫程式很可能是一兩個小時就搞定了！換句話說，如果你要處理量大且重複性的工作，寫程式是一個好的選擇。

## 1-2 程式註解

～程式的好壞在於是否每一個人都看得懂你寫的程式。

～在**Python**加入註解的方法是用井字「#」符號，並在後面寫註解。**Python**編譯器會忽略#字和後面的任何東西，因為它不是**Python**程式碼，而是給人類讀的筆記。



註解敘述(**comment statements**)很重要，註解可以幫助看懂程式碼。程式碼越多，越需要加入相關註解，將來除錯才會比較容易。

- 註解是幫助自己和他人看懂程式。
- 註解不是寫給電腦看的，是寫給未來的自己和他人看的。

- 註解越清楚詳細越好。

## Python 的註解有二種：單行註解與多行註解

- 單行註解以 井字號# 開頭，在同一行後面的文字即為註解敘述；
- 多行註解以 三個單引號或三個雙引號開頭與結尾，之間的文字即為註，適用於補充內容較多時。

```
1 # 字後內容都是註解，註解內容一行
2
3 """
4 這個區域之內的內容都是註解
5 這個區域之內的內容都是註解
6 """
7 '''
8 這個區域之內的內容都是註解
9 這個區域之內的內容都是註解
10 '''
```

個人建議：

若你打算以.py檔為主儲存程式，那可以在程式開頭加上註解，說明此程式的用意（即加上虛擬碼，顯示自己運算思維與程式設計的思路）。在一行程式碼結束的後面加上註解，說明Python程式語言的語法。

如果你打算以.ipynb為主，要同時儲存筆記、程式和結果，那建議要學習markdown語法，並將所有的註解都放到markdown筆記中，Python程式本身儘量乾乾淨淨什麼都不放。

## 1-3 程式寫作規範與風格

（待補）

## 2. 變數

『三種資料類型』：數字，文字，布林

『二種操作方法』：運算子，函式

『一個特殊符號』： $=$ （名字 = 資料）

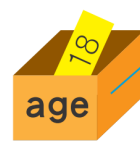
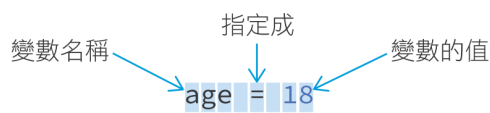
## 2-1 變數的設定

『一個特殊符號』： $=$ （名字 = 資料）

～以變數儲存資料

電腦把資訊儲存在記憶體，供程式使用。這種資訊稱做**資料(data)**。有時候這種資料由使用者提供，有時候在程式執行時產生。

資料儲存在「**變數(variable)**」裡。我們可以把變數想成電腦記憶體裡的一個盒子。盒子的名稱就是變數的名稱，而盒子的內容物就是儲存的資料，稱為變數的「**值(value)**」。（數學課學過的變數，同樣也代表資料（未知數），但不可以直接對應儲存位置。）



變數像一個有標籤的盒子，你能將資料放入盒子中，再利用盒子上的標籤找到你的資料

圖 4-2.1 變數的定義

### (1) 變數用等號進行指派

「變數名稱 = 值」（名字 = 資料）。

～等號「 $=$ 」是指派、是賦值(**assignment**)，給名字給資料的意思，不用事先宣告。

- 名字放在 $=$ 的左邊，而且要符合命名的規定；
- 資料放在 $=$ 右邊，可以指定變數一個特定值，也可以是一個運算式，而運算式可以運算成為一個值。
- 等號前後是否加入空白字元均可正常執行，前後都加空白字元，閱讀程式碼時會較清楚，除錯時也較容易。

- 變數第一次指定就是要設定初始值，沒先指定就直接使用會出現錯誤

```
1 #光有名字不行
2 print(a)
```

```
1 # 要有等號
2 # 要有內容（值）
3 a = 1
4 a
```

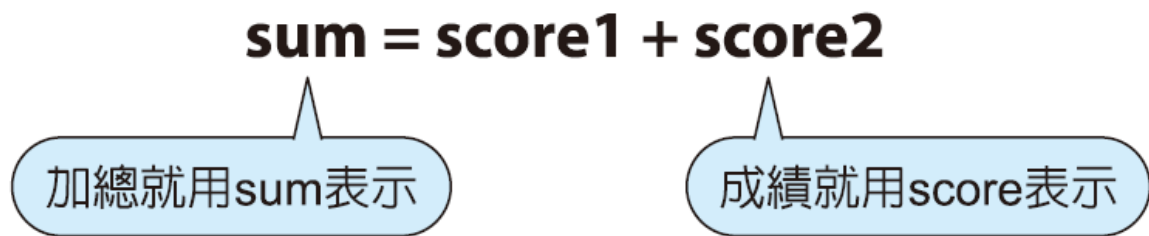
```
1 # 請記得名字在等號左邊，內容在等號右邊
2 a = "Python"
3 a
```

```
1 ## 內容有三種：數字（整數、浮點數）、布林與文字（字串）。
2 a = 123
3 b = 123.456
4 c = "Python"
5 d = False
```

## (2)變數的命名，選個好名字

變數名稱有基本規則，不能任意命名，要使用有意義的變數名稱。

- 變數的第一個字母是英文大小寫字母，其後可以接英文大小寫字母、底線或數字。
- 不能以數字開頭，不建議使用中文命名變數。
- 大小寫英文字母視為不同變數，A 與a 視為不同的變數。
- 不能以Python關鍵字為變數名稱，例如：if、else、elif、for 等。
- 變數名稱可用多個有意義的小寫單字組合而成，單字之間使用底線（\_）串接。比如表示數學成績的變數可以使用math\_score來表示，這個規定並沒有強制性。



**圖 2-2 變數命名規則示意圖**

表 2-6 變數的命名範例

正確	不正確	不正確原因
SCORE_1	1_SCORE	無法使用數字開頭
成績	成績?	包含「?」，「?」不是英文字母

～補充：常見的變數命名方式

- 駝峰式命名法，camelCase
- 大駝峰命名法，Pascal Case
- Snake\_Case，底線連結

## 2-2 三種資料類型

『三種資料類型』：數字，文字，布林

- Python資料可以分成數字（整數、浮點數）、文字（字串）與布林三種。
- 使用type()函數判斷到底資料是哪一種型別。
- 使用函數bool(), int(), float()與str()建立或轉換資料類型。

### Python的資料型別（一）：數字、文字與布林

(1)數字：分成整數(integer)與浮點數(float)兩種。

- 整數就是不含小數點的整位數。
- 浮點數類似於小數，但不等同於小數。
  - 許多計算結果，採用二進位與十進位的答案相近，但不完全一樣。

- 電腦是使用二進位，浮點數表達的是小數的近似值，和我們習慣十進位有時不完全相等。
- 電腦因為記憶體的原因，無法表示無窮位數的小數點，如 `pi` 只是取有限位數的近似值。

```
1 # 數字
2 a = 100
3 b = 100.001
4 print(a)
5 print(b)
```

```
1 #浮點數用二進位表示小數，和我們習慣的十進位表示不同。
2 print(0.1+0.1+0.1)
```

```
1 # 記憶容量有限，電腦無法表示無窮位數的小數，只能取有限位數的近似值。
2 import math          #引進math套件
3 print(math.pi)
```

## (2)字串：由字元所組成的一種序列(sequence)

～我們使用成雙的單引號`"`或成對的雙引號`"`來建立文字類型（`str`），多數的時候使用單引號或者雙引號不會有任何分別。

- 字串內文字可以儲存Unicode編碼的文字，支援中文。
- 單引號內使用雙引號，可以正確顯示雙引號。
- 雙引號內使用單引號，也可以正確顯示單引號。
- 使用三個單引號可以用於顯示多行文字，且每行前面的空白也會正常顯示，連換行字元也會被保留。

```
1 s1 = '春眠不覺曉，處處聞啼鳥。'
2 print(s1)
3
4 s2 = "夜來風雨聲，花落知多少。"
5 print(s2)
6
```

```

7  s3 = '作者"孟浩然" 詩名"春曉"'
8  print(s3)
9
10 s4 = "作者'孟浩然' 詩名'春曉'"
11 print(s4)
12
13 s5 = '''
14     春眠不覺曉，處處聞啼鳥。
15     夜來風雨聲，花落知多少。
16     作者"孟浩然" 詩名"春曉"
17 '''
18 print(s5)

```

## 動動腦練習

Python的字串可以前後利用「雙引號"」或「單引號'」包圍起來，為何要用兩種？理由是當內容本身要輸出單引號，就可以用雙引號將單引號包圍，例如請用print 輸出I'm a boy.

**(3)布林：**布林只有 **True** 與 **False** 這兩個值。

- 當我們進行判斷條件或者資料篩選時，會需要仰賴布林（bool）。
- 觀念提醒：Python（或者絕大多數的程式語言）對於英文的大小寫是敏感的（case-sensitive），像是 True 會被識別為布林，但是 TRUE 或者 true 則會被視作物件名稱。
- 在 Python 中，True 跟數值 1 相等；False 跟數值 0 相等。
- 如果在數值運算中納入了布林，不會產生任何問題。

```

1  # Python的資料型別(一)：布林
2  print(False)
3  print(True)
4  print(False + 3) # False為0
5  print(True + 3) # True為1
6  print(False == 0)
7  print(True == 1)

```



## Python的資料型別（二）：判斷到底是哪一種資料類型

```
1 # 方法（一）：type()判斷到底資料是哪一種型別。
2 print(type(87))
3 print(type(8.7))
4 print(type("Hello Python"))
5 print(type(True))
6 print(type(False))
7 print(type(None))
```

```
1 # 方法（二）：type()判斷到底變數資料是哪一種型別。
2 my_int = 87
3 my_float = 8.7
4 my_str = "Hello Python"
5 bool_true = True
6 bool_false = False
7 none_type = None
8
9 print(type(my_int))
10 print(type(my_float))
11 print(type(my_str))
12 print(type(bool_true))
13 print(type(bool_false))
14 print(type(none_type))
```

## Python的資料型別（三）：資料類型的轉換

用函數 **int()**, **float()**, **str()**與**bool()**指定或轉換資料類型。

～使用與目標轉換類型同名的函數轉換純量類型。

- **int()**：轉換純量為整數類型
- **float()**：轉換純量為浮點數類型
- **str()**：轉換純量為文字類型
- **bool()**：轉換純量為布林類型

```
1 # 使用 int() 函數可以輸入浮點數、布林與文字讓 Python 轉換成  
   整數。  
2 print(int(8.7))  
3 print(int(True))  
4 print(int(False))  
5 print(int("87"))
```

```
1 # 使用 float() 函數可以輸入整數、布林與文字讓 Python 轉換成  
   浮點數。  
2 print(float(87))  
3 print(float(True))  
4 print(float(False))  
5 print(float("87"))
```

```
1 # 使用 str() 函數可以輸入整數、浮點數與布林讓 Python 轉換成  
   文字。  
2 print(str(87))  
3 print(str(87.0))  
4 print(str(True))  
5 print(str(False))
```

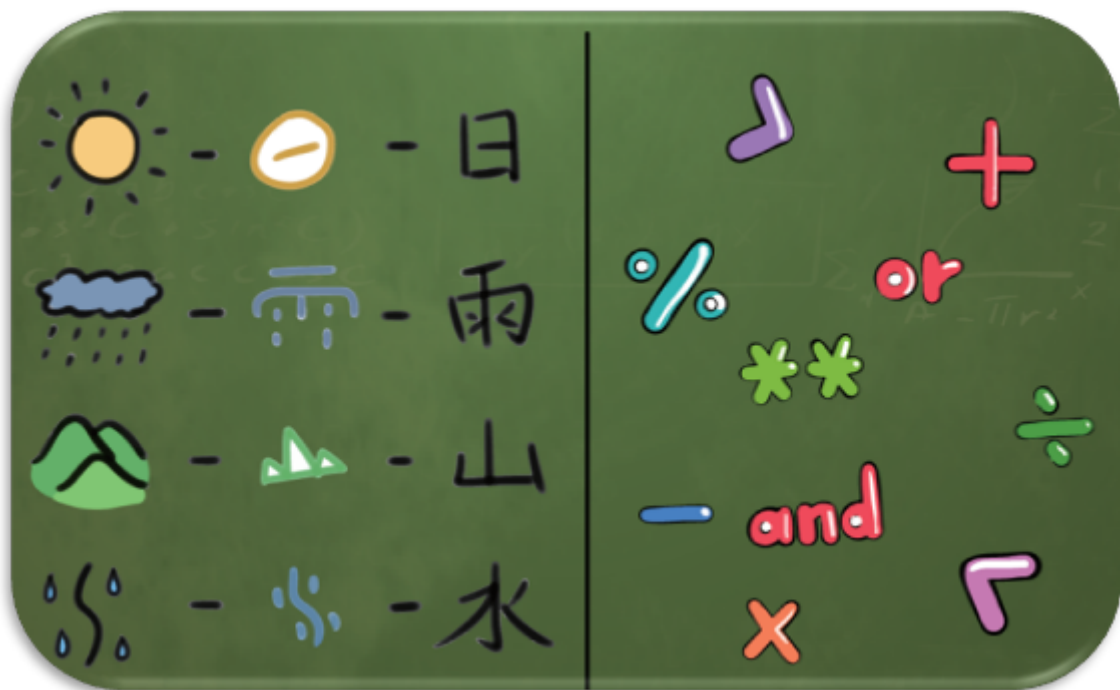
```
1 # 使用 bool() 函數可以輸入整數、浮點數與文字讓 Python 轉換  
   成布林，輸入浮點數或整數類型的 0 會轉換成為 False，其他數字  
   則一律轉換為 True。  
2 print(bool(0))  
3 print(bool(0.0))  
4 print(bool(1))  
5 print(bool(1.0))  
6 print(bool(8.7))  
7 print(bool(-8.7))
```

```
1 # 在 bool() 函數中若輸入文字，無論輸入文字內容為何都一律轉換  
    成 True 。  
2 print(bool("True"))  
3 print(bool("TRUE"))  
4 print(bool("true"))  
5 print(bool("False"))  
6 print(bool("FALSE"))  
7 print(bool("false"))
```

## 2-3 資料的操作

『二種操作方法』：運算子，函式

資料有兩種操作：運算子和函式。



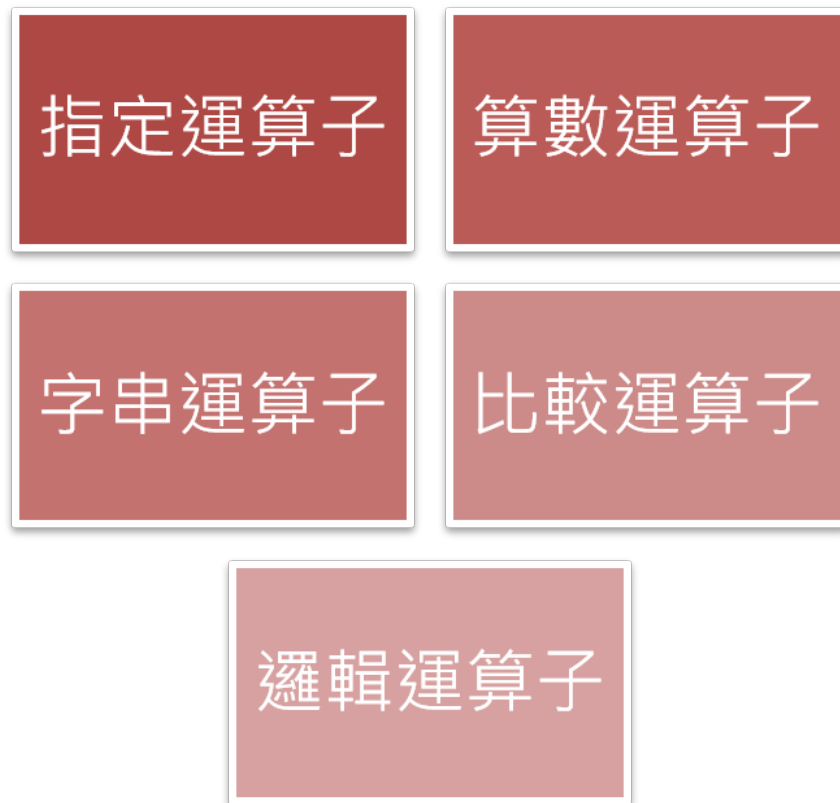
### 資料的操作（一）：運算子

～ 運算子是一種符號，它具有特定的功能，用符號來表示某種意義。

～ 運算式（**expression**）就是運算元與運算子的結合。

- 瞭解數字的操作：算數運算子、算數指定運算子、優先順序。
- 瞭解文字的操作：字串運算子，索引與切片。

- 瞭解布林的操作：比較運算子、邏輯運算子、in與is。



例如：指定運算子

用等號(=) 表示。等號右邊先運算，再將運算結果指定給左邊變數，如`a = 1 + 2`。

## 資料的操作（二）：函式

把常用的功能模組化，以函數的方式撰寫程式，方便日後呼叫使用。

- 內建函式：Python內建了許多必要的函式，像是 `print()`、`type()` 與 `help()` 等，可以直接使用。
- 標準函式庫：有一些常用的Python函式，不可以直接使用，需要 `import`後方可使用，像是數學模組`math`、或隨機模組`random`等，稱為標準函式庫。
- 自訂函式：開發者自己也可以撰寫函式使用。

## 3. 輸入與輸出

## 3-1 輸出函數

輸出指令**print**是列印的意思，是指將運算結果顯示於螢幕上，要注**print**的英文字母都是小寫。



▲ 圖 4-3.1 print 語法說明

- 由於 Python 3.x 是將 **print** 當作函數呼叫來使用，所以 **print** 後面要緊跟著 **()** 小括號，若沒有括號便會發生錯誤。
  - 括號裡面所謂的物件可能是字串、數字、變數名稱、運算式等。
  - **sep** 是分隔(separated)的意思。
    - 當同時輸出兩個以上的物件內容時，則物件中間的預設值是用一個空白字元區隔之。
  - **end** 是結束的意思。
    - 預設的「**\n**」相當於按下**Enter** 鍵，程式輸出會跳到下一行的第一個位置。

表2-1 轉義序列

轉義序列	功能說明
<code>\n</code>	換行
<code>\t</code>	跳八格
<code>\\</code>	輸出反斜線
<code>\"</code>	輸出雙引號
<code>\'</code>	輸出單引號

```

1  # 螢幕列印（一）：print函式
2  a = 33          #整數型態
3  b = 6.8         #浮點數
4  c = "abc"       #文字串型態，前後要加雙引號
5  d = '123'       #文字串型態，加單引號亦可
6
7  print("Hello")  #直接印出文字
8  print(a)        #直接印出數字
9  print(a + b)
10 print("b = ", b)
11
12 print("Bigflower Francis")
13 print("Bigflower \n Francis") # 跳行
14 print("Bigflower \t Francis") # 8個空格tab

```

```

1  # 螢幕列印（二）：print函式
2  print("A", "B", "C", "D", sep = " ")
3  print("A", "B", "C", "D", sep = "\t")
4  print("A", "B", "C", "D", sep = "|")
5  print("A", "B", "C", "D", sep = "\\")
6
7  print("A", "B", "C", "D", end = "\n")
8  print("A", "B", "C", "D", end = "\t END++")

```

## 3-2 輸入函數

輸入函數**input** 是讓使用者由鍵盤輸入資料，要注意英文字母都是小寫。

例如老師要利用電腦計算學生成績，就要先用鍵盤輸入學生的成績。

變數名稱 = input( [輸入提示] )

選擇性使用，沒用  
就不會出現提示

圖 4-3.2 input 語法說明

- `input`函數後面要緊跟著()`小括號`，若沒有括號便會發生錯誤。
- 若要告知使用者如何輸入相關資訊，可以選擇使用輸入提示。
- 使用者輸入資料、按下 **Enter** 鍵後，`input`函數會回傳使用者輸入的資料給左邊的變數。

```
1 # 螢幕輸入（一）：input函數無提示
2 s = input()
3 s
```

```
1 # 螢幕輸入（二）：input函數有提示
2 s = input("Enter Your Name: ")
3 s
```

```
1 #螢幕輸入與輸出
2 s = input("Enter Your Name: ")
3 print("Hi,", s)
```

## 觀念（一）：Input-Process-Output

使用`input`函數可透過鍵盤從使用者取得資訊。

- `input`函式會取給予使用者的提示。
- 使用者輸入的資料永遠是一個文字字串。
- 會把來自使用者的資訊放入變數。
- 變數名稱裡面的儲存值就是剛才輸入的資訊。

## 觀念（二）：不同型態的資料不能做運算。

- `input()` 函數輸入的資料都會以文字型態儲存，像打字機一樣。
  - 輸入數字也是以文字資料型態儲存，所以無法直接做數學運算。
- `input()`函數輸入的數字，經過轉換變成數字類型後，就可以做數學運算。
  - 常用的轉換函數有`int()`、`float()`、`eval()`。

```
1 ## input() 函數輸入的資料類型都是文字，不可以做數學運算。
2 s = input ("Enter Your Number: ")
3 print(s + 100)
```

```
1 ## 純量的轉換
2 #文字轉數字的整數，請用int()函數
3 s = int(input("Enter Your Number: "))
4 print(100 + s)
5
6 #文字轉數字的浮點數，請用float()函數
7 s = float(input("Enter Your Number: "))
8 print(100 + s)
9
10 #文字轉數字，均可用eval()函數
11 s = eval(input("Enter Your Number: "))
12 print(100 + s)
```

### 3-3 熟能生巧

#### 範例3-3-1 基本資料調查的輸入與輸出

寫一個程式

- 螢幕輸出「請問貴姓大名？」，等待輸入姓名，顯示輸入姓名在螢幕上。
- 螢幕輸出「請問年紀？」，等待輸入年紀，顯示輸入年紀在螢幕上。
- 螢幕輸出「請問體重？」，等待輸入體重，顯示輸入體重在螢幕上。

```
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
```



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?')
2 print('你好', name)
3
4 y = int(input('請問年紀?'))
5 print('原來你', y, '歲')
6
7 w = float(input('請問體重?'))
8 print('體重為', w)
```

**Question:** 請告訴我**print()**之後，電腦跑出的答案是什麼？

```
1 #參考答案
2 a = input("Input a = ")
3 b = input("Input b = ")
4 print(a)
5 print(b)
6 print(a + b) #請注意這裡是文字運算
7 print(a + 1) #請注意這裡不能做數學運算
```

**隨堂練習3-3-2：** 請計算NBA 史上最偉大的中鋒之一，「柴油引擎」俠客歐尼爾（**Shaquille O’Neal**）巔峰時期的**BMI**。

請透過 **input()** 函數計算使用者輸入的球員身高(216)與體重(147)與其所對應的身體質量指數。

```
1  # 參考答案
2  player_name = input("請輸入球員姓名：")
3  player_height = input("請輸入球員身高（cm）：")
4  player_weight = input("請輸入球員體重（kg）：")
5
6  player_height = float(player_height)
7  player_weight = float(player_weight)
8  player_bmi = player_weight/(player_height*0.01)**2
9
10 print("{}的身體質量指數為：{:.2f}".format(player_name,
    player_bmi))
11 print(player_name,'的身體質量指數為', player_bmi )
```