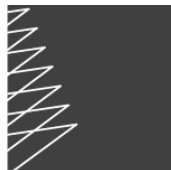
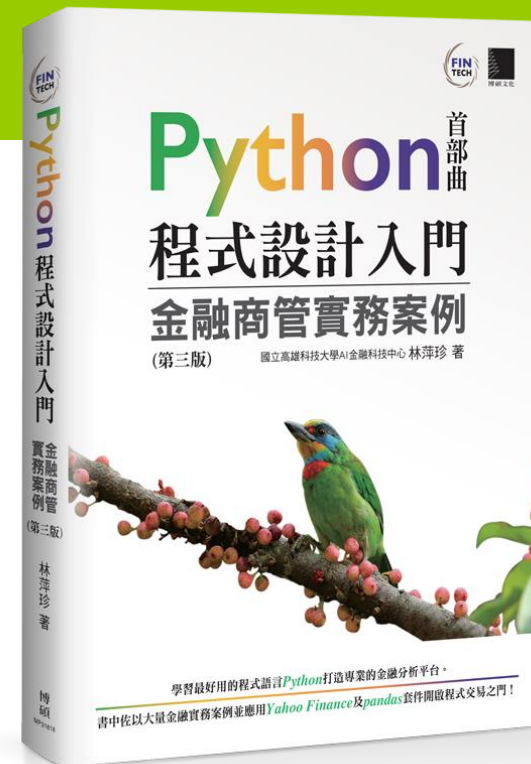


# 使用教材：Python 程式設計入門 金融商管實務 案例 第三版

作者：林萍珍



DrMaster

精選簡報・教師專用  
博碩文化・版權所有

[www.drmaster.com.tw](http://www.drmaster.com.tw)

# 4

## 資料運算

- 4-1 運算式
- 4-2 內建函數
- 4-3 模組簡介與應用

## 4-1 運算式

指令執行的優先順序中，算術運算最高，關係運算式次之、最後是邏輯運算式。

### 4-1-1 算術運算

Python 可藉由數值或變數的算術運算，以直譯方式將計算結果即時呈現。算術運算子（見表 4-1）。

表 4-1 算術運算

運算子	功能說明	運算符號與變數
+	加	$a + b$
-	減	$a - b$
*	乘	$a * b$
/	除	$a / b$
//	整數除法	$a // b$
**	次方	$a ** b$
%	取餘數	$a \% b$

## 4-1-2 關係運算

關係運算子（見表 4-2）主要是比較兩個變數值之間的大小或相等關係，例如：判斷兩個變數是否相等用雙等號（==）。

表 4-2 關係運算

運算子	功能說明	運算符號與變數
>	大於	a>0 為真。
<	小於	a<0 是假。
>=	大於等於	a>=0 為真。
<=	小於等於	a<=0 為假。
!=	不等於	A!=0 為真。
==	等於	a==3 為真。

關係運算的結果回傳 True 或 False，延續上例，a=3；b=4（見第 1 到 2 列）。a 大於等於 b，回傳結果為 False（見第 3 到 4 列）；a 等於 b，回傳結果為 False（見第 5 到 6 列）；a 不等於 b，回傳結果為 True（見第 7 到 8 列）；



### 4-1-3 邏輯運算

邏輯運算是在計算變數之間的邏輯關係，適用多個條件情境下，可判斷滿足某些條件下該採取什麼動作。Python 提供三種邏輯運算子，包括：或、且、非（見表 4-3）。

表 4-3 邏輯運算

運算子	功能說明	運算符號與變數
or	或（or）的運算，其中一個運算元為真，即為真。	s or t
and	且（and）的運算，所有運算元皆為真，才為真。	s and t
not	非（not）的運算，把為真轉為假；為假轉為真。	not t

邏輯運算子可藉由真值表（見表 4-4）來判斷運算的結果。「或」的運算符號為「or」，表示邏輯變數中，其中一個為真（True）即傳回真，其值為 1，反之為假（False），傳回 0。「且」的運算符號為 and，表示所有邏輯變數中均為真即傳回真，反之為假。非的運算符號為「not」，表示邏輯變數不成立，意即為假時才會傳回真，反之為假。



表 4-4 真值表

s	t	s or t	s and t	not s	not t
True	True	True	True	False	False
True	False	True	False	False	True
False	True	True	False	True	False
False	False	False	False	True	True

## 4-2 內建函數

Python 提供一些基礎功能，如對不同領域的使用者都很實用的內建函數（built-in functions），給使用者呼叫使用。本節篩選出 Python 最常用的內建函數，共分七類：

1. 資料型別轉換；
2. 資料運算與處理；
3. 格式化與檔案；
4. 容器；
5. 系統；
6. 迭代；
7. 其他。

### 4-2-1 資料型別轉換

Python 提供數值與字串轉換的功能（見表 4-5）。可以將參數在二進位、八進位與十六進位之間做資料型別的轉換；數值型別如整數、浮點數等轉換成字串，或將字串轉換成數值型別。

表 4-5 資料型別轉換

函數名稱	功能說明
bin()	傳入的參數轉成二進位資料型別。
hex()	傳入的參數轉成十六進位資料型別
oct()	傳入的參數轉成八進位資料型別。



函數名稱	功能說明
chr()	傳入的參數轉成 ASCII 編碼。
ord()	傳入的參數轉成 Unicode 編碼值。
float()	傳入的參數轉成浮點數。
int()	傳入的參數轉成整數資料型別。
str()	傳入的參數轉成字串型別。

1. bin 函數可以將傳入的資料轉成二進位型別（見第 1 到 2 列）。
2. hex 函數可以將傳入參數轉成十六進位型別（見第 3 到 4 列）。
3. oct 函數可以將傳入的資料轉成八進位型別（見第 5 到 6 列）。
4. chr 函數則用以查詢字元的 ASCII 碼（見第 7 到 8 列）。
5. ord 函數可以將傳入的參數轉成 Unicode 編碼的資料型別（見第 9 到 10 列）。
6. float 函數可以將傳入的參數轉成浮點數，此原傳入的參數型別可以是整數、浮點數或字串，如第 11 列的 a 是字串，float 函數可以將字串轉成浮點數（見第 12 列）再做算術運算加 10（見第 12 到 13 列）。





7. `int` 函數可將傳入的參數轉成整數資料型別，先決條件是數值只有字串型別時，可以藉由 `int` 函數轉成整數資料型別，若是非數值或浮點數要轉成整數，會發生數值錯誤（`ValueError`）（見第 14 到 16 列）。若是字串型別的整數資料可以轉換成整數，就可以做算術運算（見第 17 到 19 列）。
8. `str` 函數可以將傳入參數轉換成字串，轉換前是浮點數 `<class 'float'>`（見第 20 到 22 列），經由 `str` 函數轉換後為字串型別 `<class 'str'>`（見第 23 到 24 列）。

In	1	<code>print(bin(8))</code>
Out	2	<code>0b1000</code>
In	3	<code>print(hex(8))</code>
Out	4	<code>0x8</code>
In	5	<code>print(oct(8))</code>
Out	6	<code>0o10</code>
In	7	<code>chr(67)</code>
Out	8	<code>'C'</code>
In	9	<code>ord('C')</code>



```
Out      10      67
In       11      a='3.1416'
          12      print(float(a)+10)
Out      13      13.1416
In       14      int('a')
Out      15      ...
          16      ValueError: invalid literal for int() with base 10: 'a'
In       17      a='90'
          18      print(int(a)+10)
Out      19      100
In       20      a=9.15
          21      print(type(a))
Out      22      <class 'float'>
In       23      print(type(str(a)))
Out      24      <class 'str'>
```



## 4-2-2 資料運算與處理

python 提供許多常用的資料運算與處理，包含最大值、最小值、總和、絕對值、次方、四捨五入等（見表 4-6）。

表 4-6 資料運算與處理

函數名稱	功能說明
max()	計算傳入序列型別參數的最大值。
min()	計算傳入序列型別參數的最小值。
sum()	計算傳入序列型別參數的總和。
sorted()	計算傳入序列型別參數，由小到大排序。
len()	計算傳入序列型別參數的元素個數。
abs()	計算傳入參數的絕對值。
pow()	計算傳入參數 x 的 y 次方。
round()	計算傳入參數的四捨五入。
divmod()	計算傳入參數 x 除以 y 的商及餘數。
slice()	計算傳入參數資料切段。



1. `len` 函數可以計算串列型別參數內的長度，即元素個數。
2. `sorted` 相關介紹，前章已見，可參考，已於前章說明。
3. 絕對值 `abs()` 只能對純量的數值（見第 1 到 2 列）。
4. `pow()` 是求 `x` 的 `y` 次方（見第 3 到 5 列），可以 `x**y` 的計算結果相同（見第 6 到 7 列）。
5. `round()` 可以計算四捨五入，並可在第 2 個參數指定到小數點以下幾位（見第 8 到 9 列）。
6. `divmod()` 可以同時計算 `x, y` 兩個參數的商以及餘數，回傳值是兩個 `a, b`（見 10 到 12 列）。
7. `slice()` 可以幫串列型別的元素做切片，可以指定某串列從第 0 個元素開始（第 1 個參數）至結束（第 2 個參數）`n` 減 1 個元素做切片，如 `slice(0,5)` 即第 0 至 4 個元素做切片（見第 13 到 14 列），也可以在第 3 個參數設定切片時的順序增減量，如 `slice(1,10,2)` 即第 1 元素開始到第 9 個元素做切片，每次增加 2（見第 15 到 17 列）。



In	1	<code>print(abs(-2.33))</code>
Out	2	2.33
In	3	<code>print(pow(a,3))</code>
In	4	8
Out	5	<code>print(a**3)</code>
In	6	8
Out	7	<code>print(round(2.1416,2))</code>
In	8	2.14
Out	9	<code>[a,b]=divmod(10,3)</code>
In	10	<code>print(a,b)</code>
	11	3 1
Out	12	<code>li2=[0,1,2,3,4,5,6,7,8,9,10]</code>
In	13	<code>li2[slice(0,5)]</code>



	14	li2[slice(0,5)]
Out	15	[0, 1, 2, 3, 4]
In	16	li2[slice(1,10,2)]
Out	17	[1, 3, 5, 7, 9]

### 4-2-3 格式化輸出

任何一種程式語言，其格式化輸出是最常用的功能，主要目的是依格式顯示資料與使用者溝通。Python 有關格式化的部分，分為格式化函數、格式化輸出的形式、旗標、特殊字元四個部分說明。

#### ☑ 分格式化函數

最常用的格式化函數包含 `input` 與 `print` 兩個函數，可以處理由鍵盤輸入的字串以及格式化輸出資料（見表 4-7）。



表 4-7 格式化函數

函數名稱	功能說明
input()	藉由鍵盤接受使用者輸入資料。
print()	格式化輸出到螢幕。

`input` 函數以提示字串等待使用者藉由鍵盤於命令視窗輸入資料。`input` 函數內的字串是顯示於命令列的提示字串，游標隨即會在此字串後等待使用者輸入資料。當使用者輸入完畢時按下「Enter」，資料會回傳並儲存於 `age` 變數中（見第 1 到 4 列）。`input` 函數藉由鍵盤接受到使用者輸入的資料，其型別預設為字串（見第 5 到 6 列），若要進一步做算術運算，必須把字串轉成數值型別，如 `float` 或 `int`（見第表 4-5），指令（見第 7 到 8 列）。

In	1	<code>age=input('How old are you? ')</code>
	2	<code>print(age)</code>
Out	3	How old are you? 25
	4	25
In	5	<code>print(type(age))</code>

Out	6	<class 'str'>
In	7	print(int(age)+10)
Out	8	35

## ☑ 格式化輸出的形式

`print` 函數依參數內容格式化輸出，最常用的格式化輸出的形式（見表 4-8）。使用格式說明如下：

```
%<width>.<precision><specifier>
```

`width` 是欄位寬度，含小數點「`.`」與小數（`precision`）個數，`specifier` 是格式化後的資料型別。（`%d`, `%f`, `%s` 等見表 4-8）。





表 4-8 常用格式化輸出形式

格式化符號	功能說明
%d	格式化輸出具有正負號的整數資料。
%f	格式化輸出浮點數資料。
%s	格式化輸出字串資料。

`print` 函數範例中，帶入的參數 `%s` 指定字串格式輸出；`%d` 指定整數格式化輸出；`%f` 指定浮點數格式化輸出。指派字串 `name`、整數 `id` 與浮點數 `price`（見第 1 到 3 列）。

格式化輸出分別要用 `%s`, `%d`, `%f` 符號。其中，`%5.2f` 指含整數位、小數位以及句號，合計 5 位，小數點以下限 2 位（見第 4 到 5 列）。



## 4-2-4 容器

Python 相關容器有：list, tuple, set, dict 等四種型別，使用方法已見前章，此處依內建函數的使用方法介紹此四類函數的用法，將傳入參數轉換成容器型別。另外，再介紹與容器搭配良好的內建函數 range 做說明，Python 最常用的容器內建函數，整理如表 4-11。

表 4-11 最常用的容器內建函數

函數	功能說明
list()	將傳入參數轉換成串列。
tuple()	將傳入參數轉換成元組。
set()	將傳入參數轉換成集合。
dict()	將傳入參數轉換成字典。



```
In      1      print(list('abc'))
Out     2      ['a', 'b', 'c']

In      3      print(list(4,5,6))
Out     4      ...
         5      TypeError: list() takes at most 1 argument (3 given)

In      6      print(list({4,5,6}))
Out     7      [4, 5, 6]

In      8      print(tuple('abc'))
Out     9      ('a', 'b', 'c')

In     10      print(tuple({1: 'a', 2:'b', 3:'c'}))
Out    11      (1, 2, 3)

In     12      print(set([1,3,5]))
Out    13      {1, 3, 5}
```

In	14	<code>print(dict([ ['one', 1],[two', 2]]))</code>
Out	15	<code>{'one': 1, 'two': 2}</code>
In	16	<code>print(range(0,5))</code>
Out	17	<code>range(0, 5)</code>
In	18	<code>print(list(range(0,5)))</code>
Out	19	<code>[0, 1, 2, 3, 4]</code>

## 4-2-5 系統

Python 與系統相關的內建函數包含查詢物件方法與編號、求助等，如 `callable`, `dir`, `help` 等（見表 4-12）。



## 4-2-6 迭代

迭代 (iteration) 是 Python 的特色之一，可以想像成一個序列，從容器中將元素一個接一個的取出處理。容器資料型別 (list, tuple, str) 因為有順序性，才有迭代功能。迭代與迴圈 (見第 5 章) 可以配合得很完美。迭代運作有兩個主角：迭代者 `iter()` 與迭代器 `next()`，呼叫 `iter()` 可以產生迭代器，呼叫 `next()` 會傳回迭代器內的下個元素。常用的迭代相關的內建函數 (見表 4-13)。

表 4-13 常用的迭代內建函數

函數名稱	功能說明
<code>iter()</code>	呼叫迭代器。
<code>next()</code>	呼叫迭代器內的下個元素。
<code>zip()</code>	將傳入參數的元素進行配對，回傳一組序列配對。
<code>enumerate()</code>	將傳入參數的元素進行連續整數配對，序列為整數，起始值預設為 0。

以 `range` 函數產生起始值為 2 到終止值為 6，每次遞增 2 的序列給 `li0` (見第 1 列)，顯示 `li0` 只會看到 `range` 的設定範圍 (見第 2 到 3 列)。`li0` 必須經過呼叫迭代器函數 `iter()`，再指派給 `li1` (見第 4 列)；呼叫 `li1`，會顯示已為一個迭代物件 (見第 5 到 6 列)。



```
In      1      li0=range(2,6,2)
          2      li0
Out     3      range(2, 6, 2)
In      4      li1=iter(li0)
          5      li1
Out     6      <range_iterator at 0xcca41c2b70>
In      7      next(li1)
Out     8      2
In      9      next(li1)
Out    10      4
In     11      next(li1)
Out    12      ...
          13      ----> next(li1)
          14      StopIteration:
In     15      li2=['愛','你']
          16      li4=range(2, 0, -1)
          17      tp1=zip(li4, li2)
          18      next(tp1)
Out    19      (2, '愛')
```

## 4-3 模組簡介與應用

程式要用到的功能愈來愈多，全部放在一支程式碼則過於複雜。寫好的程式若照某種規則或機制撰寫與儲存，並將檔案分別歸類在不同地方即為模組，再匯入使用程式，將會更加簡潔，還可資源共享，發揮程式撰寫的加乘效果。模組是 Python 在直譯時的最小單位，當使用者要引用到外部模組到主程式時，必須匯入模組，語法有以下幾種：

### 4-3-1 import 模組名稱

Python 本身常用的模組，供使用者引用，例如：`random`, `sys` 等。匯入 `random` 模組（見第 1 列），該模組內的函數 `randint()` 可以產生指定範圍（1 至 6）的整數亂數，每次呼叫此函數會產生 1 個亂數，放到 `num` 中（見第 2 列），顯示 `num` 的結果為 2（見第 3 到 4 列）。



In	1	<code>import random</code>
	2	<code>num=random.randint(1,6)</code>
	3	<code>print(num)</code>
Out	4	2

### 4-3-2 import 模組名稱 as 別名

模組匯入後可以取更簡短的別名，方便重覆輸入（見第 1 列），可以依別名使用 `dir` 函數進行查詢 `random` 模組內含之函數（見第 2 到 5 列）。可以使用 `help` 函數查詢 `rd.uniform` 的使用方法（見第 6 到 10 列）。`uniform` 可以隨機亂數產生指定範圍內（0 至 1）的浮點數（見第 11 到 13 列）。



### 4-3-3 from 模組名稱 import 函數名稱 as 別名

使用者可以依指定匯入模組內的函數，同時指定別名，模組名稱放於 `from` 後，函數名稱則置於 `import` 後，別名是依函數名稱取別名。匯入模組的函數取別名，並非只匯入此函數而已，而是整個模組都匯入。

所以，模組內的其他函數也可使用，但不取別名，而要完整寫出模組名稱、函數名稱。匯入 `random` 模組的 `randint` 取別名 `ri`（見第 1 列）。呼叫 `ri` 產生整數 1 到 10 之間的任一亂數，結果為 7（見第 2 到 3 列）。

請注意每次執行結果都會不同，因為每執行一次即會產生一次亂數。使用 `range` 函數產生 0 到 5 的數列後，依 `list` 函數轉換為串列，再顯示（見第 4 到 6 列）。

可以使用 `random` 模組內的其他函數 `choice` 抽 `num` 變數內的號碼（見第 7 到 8 列），此時不能用別名，因為是其他函數，需完整寫出模組與函數名稱。



In	1	from random import randint as ri
	2	print(ri(1,10))
Out	3	7
In	4	num=list(range(0,5))
	5	print(num)
Out	6	[0, 1, 2, 3, 4]
In	7	print(random.choice(num))
Out	8	1

# 本章講解完畢

現場同學們如有不懂的地方，  
請提出問題。



# 練習

輸出  $((11 / 23 + 38 \times 5) - 200)^2$  結果。

