序

Python 是一個功能強大、語法簡潔且優雅的程式語言，目前常應用於爬蟲資料擷取、數據分析、人工智慧應用，其優點是容易學習，容易閱讀，沒有複雜的結構，而且容易維護，也因此 Python 成為目前各科系學習程式設計的首選。為使學生樂於學習且有效學習，奠定學生實作技能增進就業競爭力，以及方便教師教學，本書將學習內容分為十八章，十七和十八章為專題應用可視時間是否教學，章節一至十六章符合一學期教學時間。本書由淺入深說明程式語言、演算法、變數宣告、資料型態、運算式、流程及迴圈指令、串列、函式，以及資料庫與視窗程式的專題應用。同時訓練 Python 初學者打下前進大數據、機器學習與人工智慧的基礎。也期望學生能具備程式設計技術與運算思維能力，將來能積極面對與解決職場各種問題。

本書除對學習內容作詳細文字說明外，並加入大量示意圖和表格，強化學生學習效果。本書以免費且常用的「Spyder」為程式開發環境，在教材中加入適切的簡例，使學生觀念更加清楚。「上機實習」為程式應用實例，透過學中做、做中學和詳細的程式解析，培養學生運用演算法解決問題的能力，激發學生潛能及創造力。經由章末「練習題」的練習，能確實理解學習內容，奠定解題能力打好程式設計的基礎。

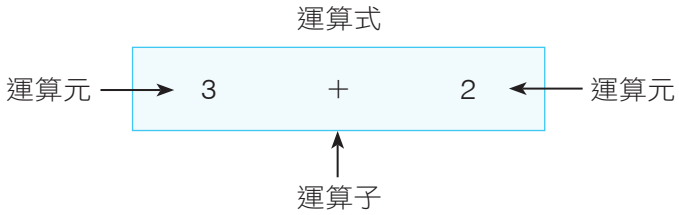
除此之外，本書不同於其他書籍介紹資料庫或視窗程式都是程式片段或是小範例。而是將全書所學到的知識整合在一起變成專題，例如 SQLite 資料庫與視窗應用程式，先介紹資料庫新增、修改、刪除與查詢的操作，接著整合視窗應用程式設計產品管理系統專題，讓初學者能學以致用，也方便任課教師教學。

6 運算子與運算元的運用

電腦擅長於資料運算及資料分析，為了要發揮電腦的這項專長，Python 內建了許多運算子，方便使用者針對資料進行運算及比對分析。本章將學習如何以運算子和運算元組合成運算式，以及運算時運算子的優先順序。

6.1 運算子、運算元與運算式

在動手撰寫程式之前，首先要認識「運算子」(operator)、「運算元」(operand) 及「運算式」(expression)。所謂**「運算式」，就是結合「運算元」與「運算子」所形成的計算式**。運算元是指運算的對象，運算元可以是變數、字面值或是另一個運算式。至於**運算子是指可以對運算元做特定運算的符號** (＋、－、⋯)。舉例來說：



▲ 圖 6-1　運算式構成示意圖

其中「3」和「2」是運算元，「＋」是運算子，運算元和運算子組合起來「3＋2」稱為運算式。

運算子若依照運算時，所需要的運算元數目來分類，可以分為：

1. 一元運算子 (unary operator)：以一個運算元和運算子組成一個運算式。

例

```
-6   # 數字 6 運算成負值
```

2. 二元運算子 (binary operator)：在運算子前後皆有運算元，大部分的運算子皆屬於此類。

例

```
8 - 2   # 數字 8 減數字 2
```

6.2 指定運算子

瞭解了「運算子」、「運算元」及「運算式」的關係之後，接下來將依運算子的功能來學習常用的運算子。在前面的章節已經介紹過指定運算子。在此不再贅述，在此要學習的是 Python 在指定運算子有一個和其他語言很大的差別，就是在其他程式語言中要交換兩個變數值時，都要使用額外的變數，但是在 Python 中只要將兩個變數互換就可以達成，這是因為 Python 直接將兩變數的記憶體位址交換，所以程式語法可以很精簡。

例　指定兩個變數並互換內容值。(檔名：assignment.py)

```
01  a, b = 2, 3
02  t = a   # 以變數 t 保存變數 a 的值
03  a = b   # 變數 b 的變數值指定給變數 a
04  b = t   # 保存在變數 t 的變數值指定給變數 b
05  print('變數 a 的內容值 %d 變數 b 的內容值 %d' % (a, b))
06  print('變數 a 的 id %x 變數 b 的 id %x' % (id(a), id(b)))
```

```
07  a, b = b, a    # 變數 a 及變數 b 變數數值交換
08  print('變數 a 的內容值 %d 變數 b 的內容值 %d' % (a, b))
09  print('變數 a 的 id %x 變數 b 的 id %x' % (id(a), id(b)))
```

說明

1. 第 2 ～ 4 行：傳統寫法。
2. 第 6 行：函式 id() 可以取得物件在記憶體中的位址，本行敘述會顯示變數 a 及變數 b 的位址以供我們觀察。
3. 第 7 行：變數值互換。
4. 第 8 ～ 9 行：顯示 a 和 b 兩變數的內容值及記憶體位址時，會發現 a 和 b 兩變數的記憶體位址會交換，就能達到變數數值交換的效果。

6.3 算術運算子

算術運算子（arithmetic operator）是用來執行一般數學運算，語法如下：

運算元 1　算術運算子　運算元 2

Python 所提供的算術運算子。如表 6-1：

▼ 表 6-1　算術運算子

運算子	說明	簡例	結果
+	加法	2 + 6	8
-	減法	9 - 3	6
*	乘法	7 * 2	14
/	浮點數除法	10 / 4	2.5
//	取除法的商數	10 // 3	3
%	取除法的餘數	5 % 2	1
**	次方（指數）	4 ** 2	16

例 撰寫計算 BMI 運算式 (英制單位 BMI 的公式 ＝ 英磅 / 英吋 ^{2} ╳ 703)。(檔名：bmi.py)

```
w, h = 150, 66                  # 體重 = 150 英磅 身高 = 66 英吋
h = h ** 2                      # 計算出身高的平方
print ('BMI =', w / h * 703)     # BMI = 24.207988980716255
```

6.4 ▶ 複合設定運算子

指定運算子若是結合算術運算子，就成為**複合設定運算子**（combination-assignment operator）或者稱為複合指定運算子。在程式中若需要將某個變數值運算後，再將運算結果指定給該變數時，可以利用複合設定運算子來簡化敘述。使用複合設定運算子時，變數必須先宣告並設定初始值否則會產生錯誤。語法如下：

```
變數　複合設定運算子　運算元；
```

例如將 x 變數值加 5，再指定給 x 變數寫法為：

```
x = x + 5
```

若是使用複合設定運算子來簡化敘述，程式寫法改為：

```
x += 5
```

常用的 Python 複合設定運算子。如表 6-2：

▼ 表 6-2　複合設定運算子

運算子	說明	指定運算式	複合設定運算式
+=	相加後再指定	x = x + y	x += y
-=	相減後再指定	x = x - y	x -= y
*=	相乘後再指定	x = x * y	x *= y
/=	浮點數相除後再指定	x = x / y	x /= y

運算子	說明	指定運算式	複合設定運算式
//=	相除取商數後再指定	x = x // y	x //= y
%=	相除取餘數後再指定	x = x % y	x %= y
**=	次方運算後再指定	x = x ** y	x **= y

6.5 位元運算子

位元運算子（bitwise operator）有「&」(And)、「|」(Or)、「^」(Xor) 及「~」(Not)。運算程序是**先將運算元轉換成二進位，接著再做指定的二進位布林運算**。位元運算子的運算方式如表 6-3 所示：

▼ 表 6-3 位元運算子

A	B	A & B	A | B	A ^ B	~A
1	1	1	1	0	0
1	0	0	1	1	0
0	1	0	1	1	1
0	0	0	0	0	1
說明		& (And，且) 運算時，兩者皆為 1 才會是 1。	| (Or，或) 運算時，要一個為 1 就會是 1。	^ (Xor，互斥或) 運算時，只要兩者不同就會是 1。	~ (Not，非) 運算時，1 變成 0；0 變成 1。

表 6-3 中 ^ (Xor) 為互斥或邏輯運算，表示兩個二進制的位元作互斥或運算，若 A 或 B 值相同時結果為 0，即兩者都為 0 或都為 1 時；若 A 和 B 值不相同時結果為 1，即一個為 0 另一個為 1。

例 位元運算子運算的範例。(檔名：bitwise.py)

```
print('2 & 5 =', 2 & 5)        # 結果  2 & 5 = 0
print('6 | 3 =', 6 | 3)        # 結果  6 | 3 = 7
print('7 ^ 1 =', 7 ^ 1)        # 結果  7 ^ 1 = 6
print('~3 =', ~3)              # 結果  ~3 = -4
```

說明

	第 1 行：2 & 5	第 2 行：6 | 3
運算	0 0 1 0 ⇦ 2 的二進位 & 0 1 0 1 ⇦ 5 的二進位	0 1 1 0 ⇦ 6 的二進位 | 0 0 1 1 ⇦ 3 的二進位
結果	0 0 0 0 ⇦ 0 的二進位	0 1 1 1 ⇦ 7 的二進位
	第 3 行：7 ^ 1	**第 4 行：~3**
運算	0 1 1 1 ⇦ 7 的二進位 ^ 0 0 0 1 ⇦ 1 的二進位	~ 0 0 1 1 ⇦ 3 的二進位
結果	0 1 1 0 ⇦ 6 的二進位	1 1 0 0 ⇦ -4 的二進位

▲ 圖 6-2 位元運算子範例運算結果圖示

6.6 移位運算子

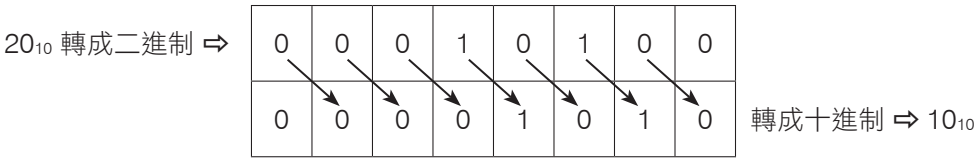
移位運算子（shift operator）可用來做數值運算，做法是先將指定的運算元轉成二進制，接著再使用「>>」右移運算子指定該運算元往右移幾個位元（bit），或是使用「<<」左移運算子指定該運算元往左移幾個位元（bit）。>>1 表右移一個位元，運算元等於除以 $2(2^1)$，>>2 表右移兩個位元，運算元等於除以 $4(2^2)$，其餘可類推。<<1 表左移一個位元，運算元等於乘以 $2(2^1)$，<<2 表左移兩個位元，運算元等於乘以 $4(2^2)$，其餘類推。

例 移位運算子運算的範例。(檔名：shift.py)

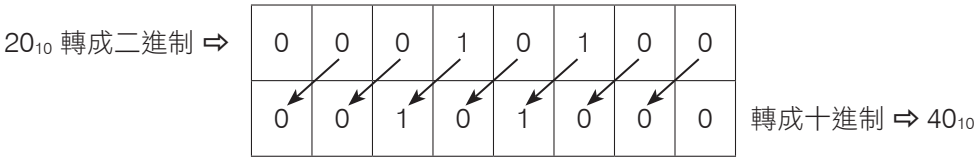
```
print(20 >> 1)  # 結果 10
print(20 << 1)  # 結果 40
```

說明

1. 第 1 行：20 右移 1 個 bit (相當於除以 2)。



2. 第 2 行：20 左移 1 個 bit (相當於乘以 2)。



▲ 圖 6-3 位移運算子範例運算結果圖示

6.7 in、is 運算子

in 和 not in 稱為**隸屬運算子**（membership operator）或是成員運算子，**in 用來判斷第一個運算元是否為第二個運算元的元素**，若是就回傳 True；否則回傳 False。**not in 運算子用來判斷第一個運算元是否不屬於第二個運算元的元素**。第二個運算元為字串、串列…等物件，這些物件會在後面再做詳細介紹。

例 in 運算子運算的範例。(檔名：in.py)

```
01 print('T' in 'Python')        # 結果 False
02 print('t' not in 'Python')    # 結果 False
```

說明

1. 第 1 行：字元 T 是否為字串 Python 的元素。
2. 第 2 行：字元 t 是否不為字串 Python 的元素。

例 is 運算子運算的範例。(檔名：is.py)

```
01  x = 50
02  y = 60
03  print(" x位址 = %d , y位址= %d " % (id(x), id(y)))
04  print(" x is y = " , (x is y))  # False
05  x = 60
06  print(" x位址 = %d , y位址= %d " % (id(x), id(y)))
07  print(" x is y = " , (x is y))  # True
```

說明

1. 第 1、2 行：變數 x、y 分別設定為 50、60。
2. 第 3、6 行：函式 id() 可以回傳物件在記憶體中的位址，所以本行敘述會顯示變數 x、y 的記憶體位址。
3. 第 4 行：以運算子 is 來判斷變數 x、y 是否為同一物件。
4. 第 5 行：變數 x 重新設為 60。
5. 第 7 行：以運算子 is 來判斷，設定新值的變數 x 和變數 y 為同一物件。

6.8 運算子的優先順序

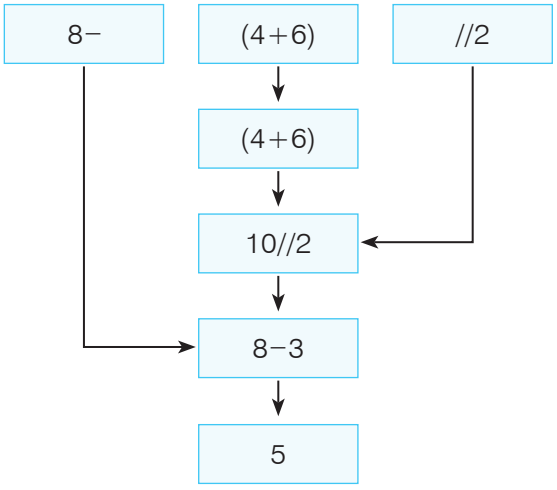
程式中的運算式可能非常複雜，如果同時有多個運算子時，Python 語言就必須根據一套規則，才能計算出正確的結果。**優先順序基本上以由左至右依序運算為原則**，但有些運算子優先權較高必須要優先處理。表 6-4 為常用運算子的優先執行順序：

▼ 表 6-4 運算子的運算優先等級

優先順序	運算子 (operator)
1	() (括弧)
2	** (次方)
3	+ (正號)、- (負號)、~ (非)

優先順序	運算子 (operator)
4	% (取餘數)、// (整數除法)、/ (浮點數除法)、* (乘)
5	+ (加)、- (減)
6	<< (左移)、>> (右移)
7	& (且) 、 | (或)、 ^ (互斥或)
8	< (小於)、<= (小於等於)、> (大於)、>= (大於等於)、!= (不等於)、== (等於)
9	= 、 += 、 -= 、 *= 、 /= 、 %=、<<=、>>=、&=、^=、|= (指定、複合設定運算子)
10	in (包含於)、not in (不包含於)、is (是否是)、is not (是否不是)
11	and (且)、or (或)、not (非)

例如運算式 8 – (4 + 6) // 2，若依優先順序來分析，可以用下列流程表示：



▲ 圖 6-4 運算子的優先順序範例運算流程圖示

為避免因為運算子的優先順序，造成運算結果錯誤，編寫運算式時應該多使用 ( ) 括號來區隔。

上機實習（一）

已知有一個邊長是 8 的立方體，請計算並顯示出立方體的表面積及體積。

結果

```
立方體的邊長：8
立方體的表面積：384
立方體的體積：512
```

程式碼

檔名：\ex06\cube.py

```
01  x = 8
02  print('立方體的邊長：', x)
03  print('立方體的表面積：', x ** 2 * 6)
04  print('立方體的體積：', x ** 3)
```

解析

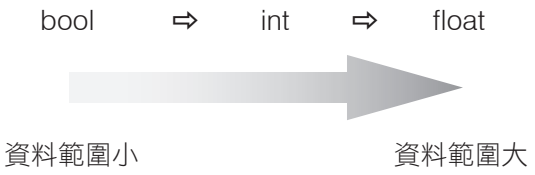
1. 第 3、4 行：x ** 2 和 x ** 3 分別代表邊長的平方和三次方。

6.9 運算式的資料型態轉換

Python 語言在宣告變數時，並不需要考慮資料的範圍，系統會自動根據變數值採用適當的資料型態。Python 使用這種方式宣告變數，對程式設計者而言是非常友善。但是在運算這些變數時，因為資料型態不同，就需要做一些特殊的處理，才不會造成運算的結果不如預期，甚至造成執行時產生錯誤。為避免上述的問題，可以利用系統的「自動型態轉換」（automatic type conversion）來轉型，或是使用強制型態轉換（cast）來自行轉型。前者是屬於隱含（implicit）方式，而後者則是屬於外顯（explicit）方式轉型。

6.9.1 自動型態轉換

當運算式中如果有資料型態不同的數值要做運算時，除非主動使用強制型態轉換外，否則系統會做自動型態轉換，將資料型態轉成一致後才進行運算。自動型態轉換是屬於隱含方式，也就是由系統自動處理。如果兩不同資料型態的資料需要做運算時，是**將型態長度較小的資料先轉成型別長度較大者**，兩者調整為相同的資料型態才做運算。其轉型規則如下：



▲ 圖 6-5 自動型態轉換規則示意圖

例 自動型態轉換的範例：(檔名：automatic.py)

```
01 b, i, f = True, 2, 6.6
02 print(b + i, b + f, i + f) # 結果  3 7.6 8.6
03 print(b - i, b - f, i - f) # 結果  -1 -5.6 -4.6
04 print(b * i, b * f, i * f) # 結果  2 6.6 13.2
05 print(b / i, b / f, i / f) # 結果 0.5 0.15151515151515152 0.30303030303030304
```

說明

1. 第 1 行：宣告 b、i 和 f 三個變數，變數值分別為 True、2 和 6.6。所以 b 變數為 bool 布林資料，i 變數為 int 整數資料，而 f 變數為 float 浮點數資料型態。
2. 第 2 ～ 4 行：bool 資料型態和 int 資料型態變數做運算時，bool 變數會自動轉型為 int。布林值 True 會轉成整數 1，False 則是 0。
3. 第 5 行：/ 運算子為浮點數除法，所以運算子會先轉成浮點數再做運算。

6.9.2 強制型態轉換

在程式當中，必要時可以將變數的資料型態做轉換，例如將整數轉型為浮點數，或將浮點數變數轉型為整數。強制型別轉換是使用函式，以外顯方式轉換型態。Python 常用的型別轉換函式在前面已經介紹，現在彙整如下：

```
整數轉浮點數：float（整數資料）
浮點數轉整數：int（浮點數資料）
浮點數轉整數：round（浮點數資料）
數值轉字串：str（數值資料）
轉布林值：bool（資料）
```

小範圍資料型態轉型為較大範圍型別時，變數值沒有問題。但是**大範圍轉型為較小範圍資料型態時，變數值就會失真**。例如浮點數用 int() 函式強制轉型為整數時，會將小數部分直接捨棄，不會做四捨五入的運算。

例 強制型態轉換的範例：(檔名：cast.py)

```
i1=10
f1=float(i1)              # 使用 float() 函式將整數轉型為浮點數
print(i1,f1,type(f1))     # 結果 10 10.0 <class 'float'>
f2=1234.5678
i2=int(f2)                # 使用 int() 函式將浮點數轉型為整數（捨棄小數）
print(f2,i2,type(i2))     # 結果 1234.5678 1234 <class 'int'>
i3=round(f2)              # 使用 round() 函式將浮點數轉型為整數（四捨五入）
print(f2,i3,type(i3))     # 結果 1234.5678 1235 <class 'int'>
s=str(i2)                 # 使用 str() 函式將整數轉型為字串
print(s,type(s)           # 結果 1234 <class 'str'>
b=bool(f1)                # 使用 bool() 函式將浮點數轉型為布林值
print(b,type(b))          # 結果 True <class 'bool'>
```

上機實習（二）

撰寫一個貸款試算程式。計算方式，將貸款期間內全部貸款本金與利息平均分配於每一期中償還。假設以利率不變的前提下，利用下面公式計算出每月本息的平均攤還率。再將貸款總金額乘以平均攤還率，即得到每月應攤還的本金及利息，顯示之金額採整數值顯示（小數第一位請四捨五入）。

提示：每月本息的平均攤還率 = {[(1+ 月利率)^{月數}]× 月利率 }÷{[(1+ 月利率)^{月數}]-1}

結果

```
請輸入貸款金額：500000 Enter
輸入分幾年攤還：20 Enter
輸入貸款年利率：4 Enter
每月應還本息：3030 Enter
```

程式碼

檔名：\ex06\loan.py

```
01 x = int(input('請輸入貸款金額：'))
02 y = int(input('輸入分幾年攤還：'))
03 r = int(input('輸入貸款年利率：'))
04 r /= 100          # 求年利率
05 r /= 12           # 月利率 = 年利率 / 12
06 m = y * 12        # 月數 = 年數 × 12
07 rate = (((1 + r) ** m) * r) / ((1 + r) ** m - 1)      # 求每月平均攤還率
08 print('每月應還本息：', round(x * rate))                # 顯示本息攤還金額
```

解析

1. 第 4 行：輸入之年利率除以 100 可得年利率的百分比。
2. 第 7 行：以括弧確保運算順序如預期。
3. 第 8 行：以 round() 函式將運算所得浮點數，小數第一位四捨五入。

練習題

一．選擇題

(　　) **1.** x = x + y 敘述可以改寫成下列哪一個敘述？

① x = +y　② x += y　③ x ++ y　④ y = x + y

(　　) **2.** 8 / 3 和 8 // 3 這兩個運算式運算的結果，分別是屬於下列哪種資料型別？

① int、int　② float、int　③ int、float　④ float、float

(　　) **3.** 執行以下 Python 程式片斷，其結果為何？

```
01 x, y = 60, 50
02 print (x ^ y)
```

① 14　② 18　③ 48　④ 62

(　　) **4.** 執行以下 Python 程式片斷，哪一行的運算結果最大？

```
01 2 * 3 * 3 / 2
02 2 * 3 ** 2 / 2
03 5 / 2 ** 3 + 9
04 5 + 4 * 2 - 2
```

① 01 行　② 02 行　③ 03 行　④ 04 行

二．程式設計

1. 已知圓的半徑是 10，請算出圓的面積、圓周長和圓球的體積。

```
圓的半徑：10
圓的面積：314.159
圓周長：62.8318
球體的體積：4188.786666666666
```

2. 撰寫一個程式，使用者輸入攝氏溫度 (資料型態為 int) 後，會顯示換算後的華氏溫度。(提示：華氏溫度 = 攝氏溫度 ×9 / 5 + 32)

```
請輸入攝氏溫度（整數）：100 Enter↵
華氏溫度：212.0
```