

循序主題（二）：基本程式設計



by 田弘華 Hung-Hua Tien

在電腦中的主記憶體並不是無窮大，因此，如何有效的利用主記憶體呢？那就必須要學會各種資料型態及其在主記憶體中所佔用的大小。

【定義】依照不同性質的資料，給予不同的記憶體空間。

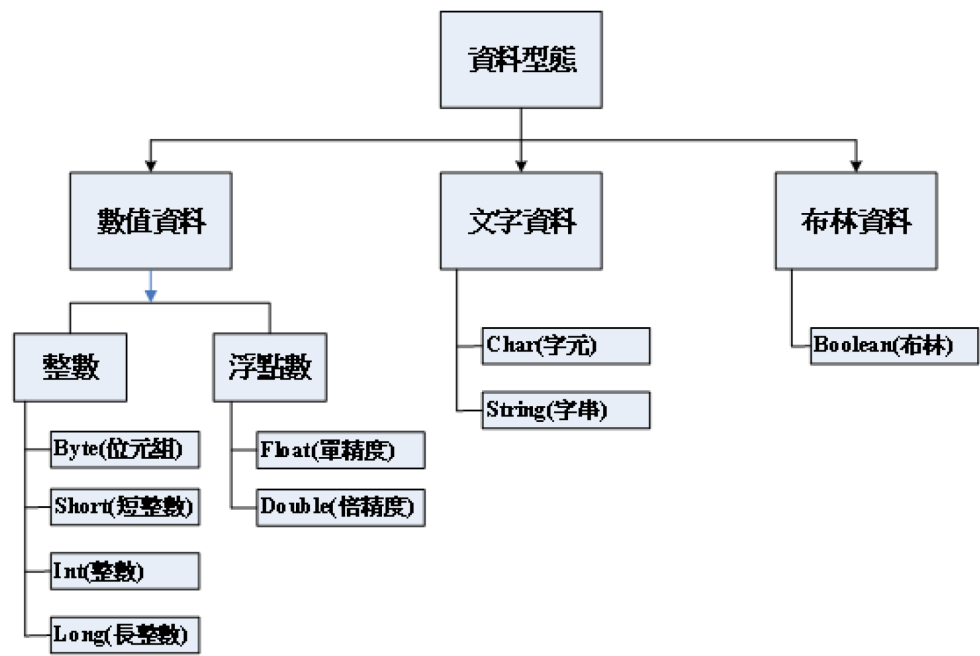
【概念】小東西用小盒子裝，大東西用大箱子裝。

【示意圖】

小東西用小盒子裝	大東西用大箱子裝
	

【目的】1.有效地利用主記憶體空間。 2.提高程式的可讀性。

【種類】




數字類資料型態

【定義】是指用來存放「數值資料」的空間。

【例如】每一位學生的座號、成績...等都是數值資料。

【示意圖】



Int(整數型態)	Float(單精準度型態)
	

文字類資料型態

【定義】是指用來存放「文字資料」的空間。

【例如】每一位學生的學號、姓名...等都是字串資料。

【示意圖】

Int(整數型態)	String(字串型態)
	

其他類資料型態

【定義】是指用來存放「布林資料」的空間。

【適用時機】以布林資料型態為例：1. 選擇結構的「條件式」。2. 重複結構中的「條件迴圈」。

【示意圖】



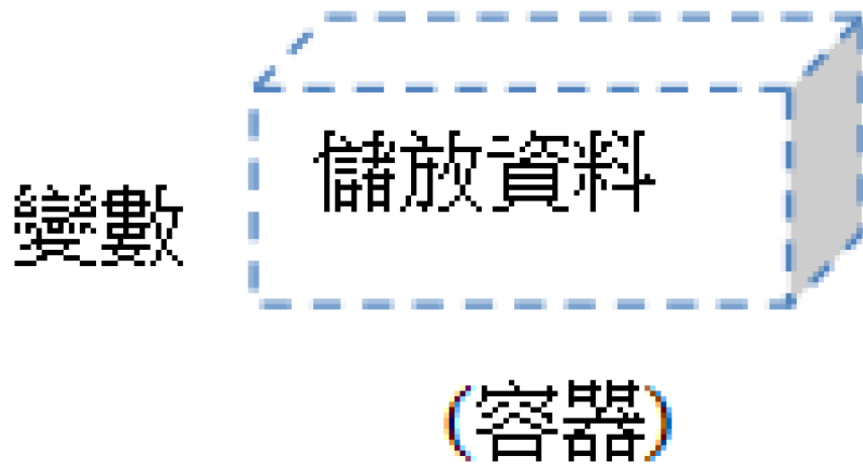
布林資料(Boolean)通常是用來表示「條件式」的真假，當條件式成立時布林資料就會變成 True(真)，不成立時布林資料就會變成 False(假)。

變數(Variable)

【定義】是指程式在執行的過程中，其「內容」會隨著程式的執行而改變。

【概念】1.將「變數」想像成一個「容器」，它是專門用來「儲放資料」的地方。2.「容器的尺寸」是由宣告時的「資料型態」來決定。

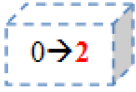
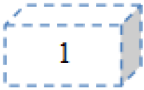
【示意圖】



【例如】 $A=B+1$

其中A、B則是變數，其內容是可以改變的。

【圖解說明】

執行的過程	變數的內容變化
$A=0 : B=1$ $A=B+1$ ↓	A  B 

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）

1-1 變數的設定

『一個特殊符號』：等號（名字 = 資料）

～以變數儲存資料。

電腦將資料儲透「變數」(variable)儲存在記憶體，供程式使用。資料有時候由使用者提供，有時候在程式執行時產生。



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。）

變數用等號進行指派：「變數名稱 = 值」（名字 = 資料）。

～等號「=」是指派、賦值(assignment)，給名字給資料的意思。

- 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

```
#光有名字不行  
print(a)
```

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

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

```
## 內容有三種：數字（整數、浮點數）、布林與文字（字串）。  
a = 123  
b = 123.456  
c = "Python"  
d = False  
  
# 選個好名字，變數的命名
```

1-2 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。
- 不要有空格。
- 不能以Python關鍵字（保留字、變數名稱）為變數名稱，例如：if、else、for等。
- 建議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。
- 不建議使用中文命名變數。
- 注意大小寫英文字母不同，即A與a被視為不同的變數。

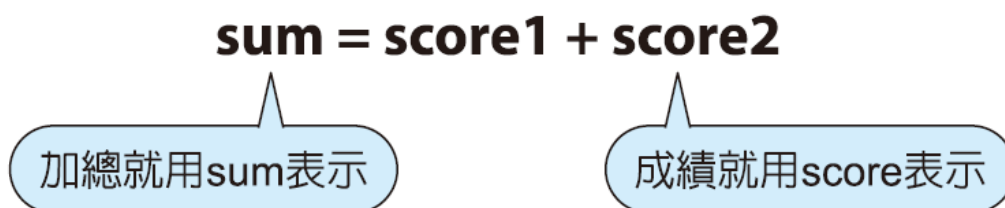


圖 2-2 變數命名規則示意圖



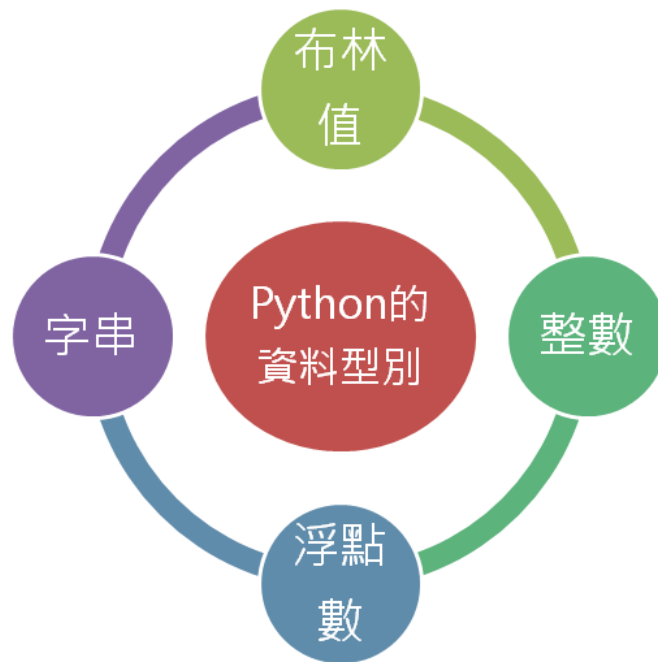
～補充：常見的變數命名方式

- 駝峰式命名法，camelCase
- 大駝峰命名法，Pascal Case
- snake_case，小寫英文字＋底線連結

1-3. 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。



(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。

```
# 數字
a = 100
b = 100.001
print(a)
print(b)
```

(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。

```
s1 = '春眠不覺曉，處處聞啼鳥。'
print(s1)
```

```

s2 = "夜來風雨聲，花落知多少。"
print(s2)

s3 = '作者"孟浩然" 詩名"春曉"'
print(s3)

s4 = "作者'孟浩然' 詩名'春曉'"
print(s4)

s5 = '''
    春眠不覺曉，處處聞啼鳥。
    夜來風雨聲，花落知多少。
    作者"孟浩然" 詩名"春曉"
'''
print(s5)

```

```

s1 = "I'm a lady."
print(s1)

s2 = 'I/'m a lady.'
print(s2)

s3 = "" I'm a lady.""
print(s3)

s4 = '''I'm a lady.'
print(s4)

```

(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。

```

# Python的資料型別(一)：布林
print(False)
print(True)
print(bool(0)) # False為0
print(bool(1)) # True為1

```

（二）判斷到底是哪一種資料類型

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

```
# 方法（二）：type()判斷到底變數資料是哪一種型別。
my_int = 87
my_float = 8.7
my_str = "Hello Python"
bool_true = True
bool_false = False
none_type = None

print(type(my_int))
print(type(my_float))
print(type(my_str))
print(type(bool_true))
print(type(bool_false))
print(type(none_type))
```

（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

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

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



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

```
# 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。
print(bool(0))
print(bool(0.0))
print(bool(1))
print(bool(1.0))
print(bool(8.7))
print(bool(-8.7))
```

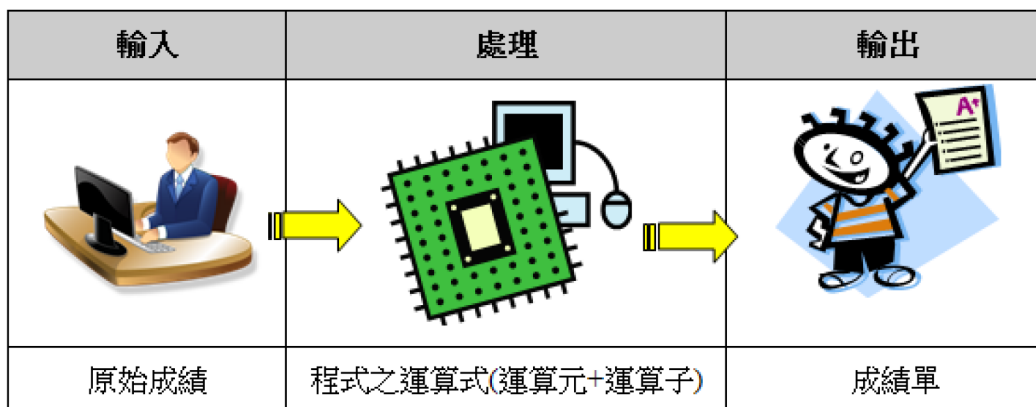
```
# 在 bool() 函數中若輸入文字，無論輸入文字內容為何都一律轉換成 True。
print(bool("True"))
print(bool("TRUE"))
print(bool("true"))
print(bool("False"))
print(bool("FALSE"))
print(bool("false"))
```

1-4 資料的操作

資料的運算

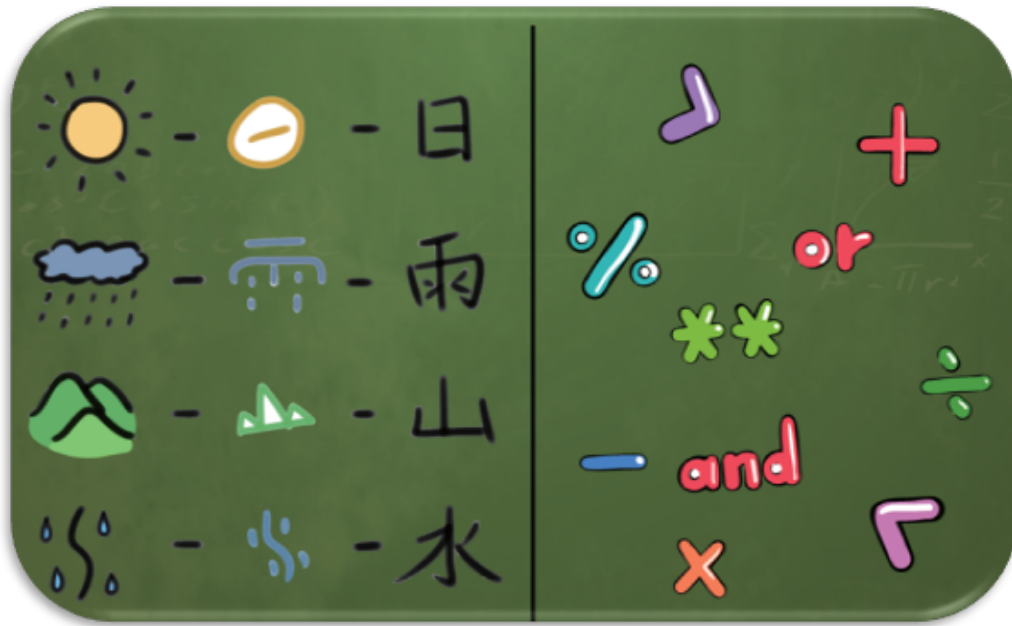
【引言】我們都知道電腦處理資料的過程為：輸入---處理---輸出，其中「處理」程序通常是藉由運算式(Expression)來完成。

【示意圖】



資料的操作有兩種：運算式和函式。

資料的操作（一）：運算 式



【運算式的組成】

運算元(Operand)與運算子(Operator)。

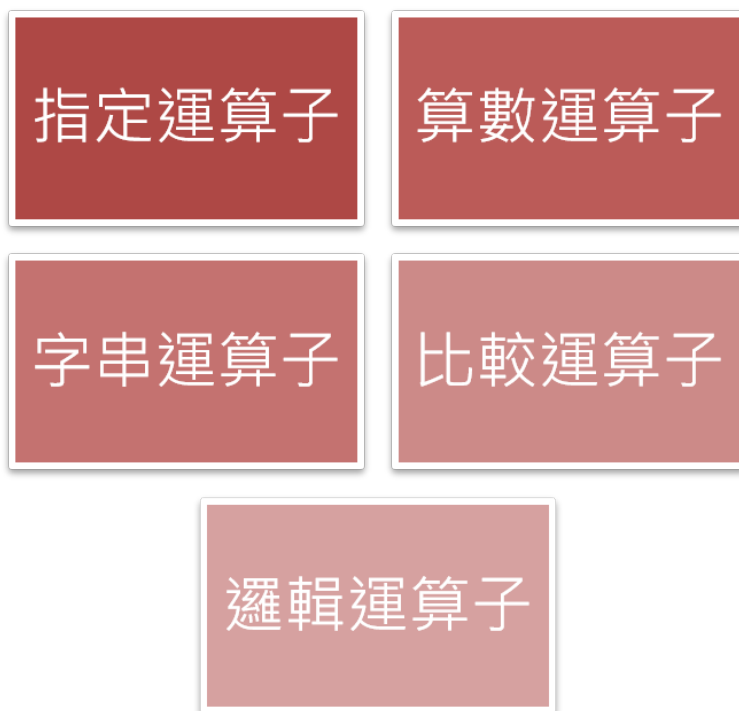
【例如】 $A=B+1$

其中「A,B,1」稱為運算元

「=,+」則稱為運算子。

一般而言，「運算元」(operand)不是變數就是常數，而運算子(operator)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。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 $a = 1 + 2$ 。

- 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。



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。

2. 數字的操作：運算式與函式

【初學者的疑問】

在撰寫程式，遇到數學上的等號「`=`」時，都會有一些疑問，那就是何時才是真正的「等號」，何時才能當作「指定運算子」來使用，基本上，在B4A中的等號「`=`」大部份都是當作「指定運算子」來使用。

指定運算子

【定義】將「右邊」運算式的結果指定給「左邊」的運算元。

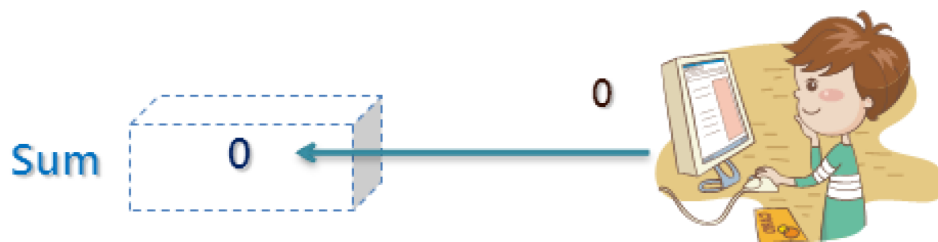
【方法】從「`=`」指定運算子的右邊開始看

【例子】`Sum=0`



運算元	指定運算子	運算式的結果 
Sum	=	0

【圖解說明】



範例：將變數A與B的值相加以後，指定給Sum變數

其程式如下：

```
Sum = 0
A = 1
B = 2
Sum = A + B
```

如果初學者以「數學上的觀點」來看，則一定會覺得Sum變數的值不是為0嗎？但是，依照「程式設計的觀點」其結果為3。

【隨堂練習】請判斷下列四行程式是否正確？如果有誤，請說明其原因。

```
Sum = 0
A = 1
2 = B # 「=」的左邊不能有常數。
A + B = Sum # 「=」的左邊不能是運算式或兩個變數。
```

【注意】指定運算子「左邊的運算元」，不可以常數或二個及二個以上變數。

2-1 數字運算子（一）：算術運算子

算術運算子，使用整數與浮點數進行數字運算，常用的數學運算符號有七種：加、減、乘、除、次方、餘數與商數。

表 2-7 算術運算子

運算子	說明	舉例	結果
+	加	A=5+2	A=7
-	減	A=5-2	A=3
*	乘	A=5*2	A=10
/	浮點除法	A=5/2	A=2.5
//	整數除法（去除小數點）	A=5//2	A=2
%	相除後求餘數	A=5%2	A=1
**	次方	A=5**2	A=25

～ (1)Python可以成為強大的計算機。

```
# 數字（一）算術運算子 - 直接運算
print(5 + 2) #加
print(5 - 2) #減
print(5 * 2) #乘
print(5 / 2) #除

print(5 ** 2) #次方
print(5 // 2) #商
print(5 % 2)  #餘數

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

～(2) 我們可以用變數來儲存一些數，用它們來算數。

```
# 數字（一）算術運算子：變數運算
a = 11
b = 3.8

print(a + b)
print(a - b)
print(b * 2)
print(b / 2)

print(a % 3)
print(a // 3)
print(a ** 2)
```

～(3) 如果要轉換資料型別，可以用int(), float()等。

```

number = input("Enter a number: ")
number + number
number

number = input("Enter a number: ")
number = int(number)
number + number

```

範例中的變數`number`看起來是整數2，但它其實是文字：字串2。若再輸入`number`可得到'22'，因為文字2+文字2是串連字串得'22'。所以，要做數學計算，要先將文字轉成數字，使用`int()`函式進行轉換之後，才能做數學計算。

2-2 數字運算子（二）：算術指定運算子

算術指定運算字，為指定運算子和算術運算子的綜合。

表 2-10 算術運算子範例

運算子	說明	範例	將範例進行縮寫
<code>+=</code>	加	<code>A=A+2</code>	<code>A += 2</code>
<code>-=</code>	減	<code>A=A-2</code>	<code>A -= 2</code>
<code>*=</code>	乘	<code>A=A*2</code>	<code>A *= 2</code>
<code>/=</code>	浮點除法	<code>A=A/2</code>	<code>A /= 2</code>
<code>//=</code>	整數除法（去除小數點）	<code>A=A//2</code>	<code>A //= 2</code>
<code>%=</code>	相除後求餘數	<code>A=A%2</code>	<code>A %= 2</code>
<code>**=</code>	次方	<code>A=A**2</code>	<code>A **= 2</code>

```

# 數字（二）算術指定運算字
a = 5
a = a + 2
print(a)

a+=2 #a = a + 2
print(a)

a-=2 #a = a - 2
print(a)

a*=2 #a = a * 2
print(a)

a/=2 # a = a / 2
print(a)

a//=2 # a = a // 2
print(a)

a%=2 # a = a % 2
print(a)

```

```

a**=2 #a = a**2
print(a)

## 請同學猜猜看列印結果
a = 5
a = a + 2
print(a)

a+=2 # a = a + 2
print(a) # 注意不能使用 print(a+=2)

print(a+2) # 沒有等號，可以列印

```

2-3 數字運算子（三）：運算的優先順序

運算的優先順序，和數學習慣相同。

數值運算中 Python 遵循常見的次方優先、先乘除後加減等慣例。如果在算式中希望調整優先順序，可以使用小括號 **()** 括著要先完成運算的部分，但不能用中括號大括號。

動動腦

請利用Python計算 $(23+32) \times 4 = ?$ ，以及9開三次方？

～在數學運算中，括號優先權最高，其次是指數 \rightarrow %餘數 \rightarrow //商 \rightarrow /除法 \rightarrow *乘法 \rightarrow + 加法 \rightarrow -減法。

```

print(23 + 32 * 4)
print((23 + 32) * 4)
print([23 + 32] * 4) # 因為中括號，23+32被視為文字，做文字的運算
print({23 + 32} * 4) # 錯誤訊息

print(9**1/3)
print(9**(1/3))
print(9**[1/3]) # 錯誤訊息
print(9**{1/3}) # 錯誤訊息

```

範例一	乘除先運算	接著加減運算
$F=2+3*5-14/7$	$F=2+15-2$	$F=15$
範例二	括號先運算	接著求餘數運算
$F=(2+3)\%4$	$F=5\%4$	$F=1$

隨堂練習：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。BMI = 公斤 / 公尺平方

```

# 數字（三）運算的優先順序，不用背。
height = 216
weight = 147
bmi = weight/(height/100)**2 # 將身高轉換為以公尺為單位
# bmi1 = weight/height/100**2

print(bmi)

# 參考答案
player_name = input("請輸入球員姓名：")
player_height = input("請輸入球員身高（cm）：")
player_weight = input("請輸入球員體重（kg）：")

player_height = float(player_height)
player_weight = float(player_weight)
player_bmi = player_weight/(player_height*0.01)**2

print("{}的身體質量指數為：{:.2f}".format(player_name, player_bmi))
print(player_name, '的身體質量指數為', player_bmi )

```

2-4. 常用的內建函數

- abs(), max(), min(), pow(a,b), round(x), round(x, n)

 表4-1 一些常用的內建函式

函式	功能說明	範例
abs(x)	回傳x的絕對值	abs(-10) 是10
max(x1, x2, ...)	回傳x1, x2, ...的最大值	max(1, 8, 6) 是8
min(x1, x2, ...)	回傳x1, x2, ...的最小值	min(1, 8, 6) 是1
pow(a, b)	回傳 a^b	pow(2, 4) 是16
round(x)	回傳最接近x的整數。若與兩整數接近，則回傳偶數的整數	round(5.4) 是5 round(5.5) 是6 round(4.5) 是4
round(x, n)	回傳捨位到小數點後n位的浮點數	round(6.667, 2) 是6.67 round(6.663, 2) 是6.66
int(x)	取x的整數值	int(12.34) 是12

```

# 數字：內建函數
a = -2
print(abs(a)) #絕對值
print(abs(-10))

print(max(1,2,3,4,5)) #極大值
print(min(1,2,3,4,5)) #極小值

print(pow(2,3)) ##次方 pow(a,b)為a的b次方

print(round(5.8)) #四捨五入
print(round(3.14159,2)) #取小數點兩位死捨五入

```


2-5. math 數學模組

~import math，將數學math模組載入

- 圓周率 π 、指數 e
- 指數函數 $\exp(x)$ 、對數函數 $\log(x)$ 、開根號 \sqrt{x}
- 三角函數 $\sin(x)$ 、 $\cos(x)$ 、 $\tan(x)$ 、 $\cot(x)$

 表4-2 一些常用的數學函式

函式	功能說明	範例
<code>fabs(x)</code>	以浮點數回傳 x 的絕對值	<code>fabs(-2.3)</code> 是2.3
<code>ceil(x)</code>	回傳大於 x 的最小整數	<code>ceil(2.6)</code> 是3 <code>ceil(-2.6)</code> 是 -2
<code>floor(x)</code>	回傳小於 x 的最大整數	<code>floor(2.6)</code> 是2 <code>floor(-2.6)</code> 是 -3
<code>exp(x)</code>	回傳 e^x	<code>exp(1)</code> 是2.71828
<code>log(x)</code>	回傳 $\log_e(x)$	<code>log(2.71828)</code> 是1.0
<code>log(x, base)</code>	回傳以指定基底的對數	<code>log(100, 10)</code> 是2.0

函式	功能說明	範例
<code>sqrt(x)</code>	回傳 \sqrt{x}	<code>sqrt(100)</code> 是10.0
<code>sin(x)</code>	回傳以弧度為單位的sine三角函式	<code>sin(π / 2)</code> 是1 <code>sin(π)</code> 是0
<code>cos(x)</code>	回傳以弧度為單位的cosine三角函式	<code>cos(π / 2)</code> 是0 <code>cos(π)</code> 是 -1
<code>tan(x)</code>	回傳以弧度為單位的tangent三角函式	<code>tan(π / 4)</code> 是1 <code>tan(0.0)</code> 是0
<code>degrees(x)</code>	將 x 角度從弧度(radian)轉換為度數(degree)	<code>degrees(1/2 * π)</code> 是90
<code>radians(x)</code>	將 x 角度從度數轉換為弧度	<code>radians(180)</code> 是3.14159

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。

```
import math          #引進math套件
print(math.pi)
print(math.e)

# 計算圓面積
r = eval(input("輸入圓的半徑："))
area1 = 3.14159 * (r**2)
area2 = math.pi * (r**2)
print('圓面積1:', area1)
print('圓面積2:', area2)
```