Chapter
03

資料型態與運算

學習大綱

- 3-1 資料型態
- 3-2 常數與變數
- 3-3 運算元與運算子
- 3-4 基本指令的操作
- 3-5 標準輸出輸入的操作
- 3-6 運算元與運算子的應用
- 3-7 程式的除錯
- 3-8 本章內容摘要

學習表現

1. 具備撰寫程式語言基本能力，運用資訊科技方法解決問題。
2. 具備程式設計之邏輯思考能力，展現系統思考、分析與探索之素養。
3. 能思辨勞動法令規章與相關議題，省思自我的社會責任。

3-1 資料型態

我們平常處理的資料依其型態可分爲整數、浮點實數、字元、字串與布林值，分別說明如下：

3-1-1 整數（Integer）

2,-3 等稱爲整數。大部分的程式語言必須依照整數的長度（例如，1000 或 100000000 等長度就不同）選用適當的資料型態，但 Python 則不用。例如 : 請鍵入以下程式，觀察執行結果。(本書強調由作中學，所以刻意沒有寫出執行結果，請同學務必自己動手鍵入程式，在下面底線寫出執行結果，這樣印象才會深刻)

```
a=1
b=12345678901234567890123456789O
c=-123456789012345678901234567890123456789O
print(a)
print(b)
print(c)
```

Python 可以處理的整數有兩種進位方式，分別是十進位 (Decimal) 與十六進位 (Hexadecimal)。其中十進位則以我們平常書寫數字的方式即可。十六進位應以 0X 或 0x 開頭，且以 0,1,2,3,4,5,7,8,9,a,b,c,d,e,f 代表 0 到 15。例如 0x1a 或 0XB 均爲十六進制，分別代表十進位的 26 與 11。請鍵入以下程式，寫出執行結果。

```
a=0x1a
print(a)#26
```

代表 $1*16^1+10=26$(十六進位的 a,b,c,d,e,f 等字元，大小寫都可以)

3-1-2 浮點實數（Float）

數字中含有小數點或指數的稱爲浮點數、實數或浮點實數。以指數爲例，E 或 e 表示 10 的次方，例如 0.0023、2.3E-3 及 2.3e-3 都是表示相同的浮點數；又例如 2.3E+2 則代表 230。浮點數可使用標準寫法或科學符號法表示，例如 321.123 即爲標準寫法，1.23e+4 即爲科學符號表示法。大部分的程

式語言必須依照浮點數的精密程度選用不同的資料型態，但 Python 則不用。例如：

```
a=5.0 #強調這個5,已經精確到小數點1位。
b=0.0023
c=2.3E2
d=2.3e-2
e=-2.3e2
f=-2.3e-2
print(a)                                    _____
print(b)                                    _____
print(c)                                    _____
print(d)                                    _____
print(e)                                    _____
print(f)                                    _____
```

Python 已經簡化資料型態的學習，如此就能降低學習門檻，減輕使用者負擔，就如同現代使用手機拍照，只要按下快門就好，所有光圈、快門、焦距等，電腦都能自動處理。

3-1-3　字元（Char）與字串（String）

Python 不分字元或字串，通通使用單引號或雙引號表示字元與字串。例如：

```
a="a"
b='ab'
c='Mary'
d="國勝"
```

但以下就不行

```
d='aa"
```

3-1-4　布林值（Boolean Value）

人類對於事情的正確與否的表達方式有『對』與『錯』、『正確』與『錯誤』、『是』與『非』。電腦爲了讓此事件有一致性，就特別有一個資料型態，稱爲布林型態，且只有兩個值，分別是『True』與『False』。例如：

```
print(3>2)
print(5<0)
```

3-2 常數與變數

所謂「常數」(Constant)是指程式執行的過程都不會也不需要改變的值，所以稱爲常數；「變數」（Variable）則有可能變動。因爲程式經常帶入不同的數值或必須經過多次反覆運算才能得到結果，這些值就必須用變數儲存，這樣才能跟隨反覆的計算而跟著變動，所以稱爲「變數」。但是 Python 並不用特別區分此兩個名詞，也就是常數不用特別使用 const 標注，都是直接用變數名稱代表以上整數、實數或布林值就可以，但是常數通常全大寫，這樣比較好區分。例如：

```
PI=3.14
E=2.61
RATE=0.062
```

3-3 運算子與運算元

前面有談到關於面積的計算 c=a*b，以上「a,b」稱爲運算元，「*，=」稱爲運算子，「a*b」稱爲運算式，c=a*b 稱爲「敘述」。所謂運算子 (Operator)，指的是可以對運算元 (Operand) 執行特定功能的特殊符號。Python 的運算子分爲四大類，分別是算術 (Arithmetic) 運算子、複合指派運算子、關係運算子、邏輯運算子。除此之外，我們還會討論運算子的優先順序 (Precedence) 與結合律 (Associativity)。優先順序用來決定同一式子擁有多個運算子時，每一個運算子進行運算的優先順序；而結合律則決定了在同一敘述中，相鄰的運算子有相同優先順序的運算子執行的順序。

算術運算子（Arithmetic operators）

小型計算機有加、減、乘、除等鍵，而程式語言要能計算加減乘除，也要有這些算術運算子。表 3-1 是 Python 的算術運算子列表：

表3-1　Python算術運算子

運算子	定義	優先順序	結合律
**	次方	1	由左至右
+/-	正負號	2	由右至左
*	乘法運算	3	由左至右
/	除法，得實數商	3	由左至右
//	除法，得整數商	3	由左至右
%	求餘數 (Modulus)	3	由左至右
+/-	加法 / 減法運算	4	由左至右
=	指派	14	由右至左

以上運算子的運算結果、運算子優先順序、運算子結合律等，分別說明如下：

指派運算子（Assignment operators）

指派運算子的符號爲「=」，又稱爲「賦值」運算子，其作用爲將運算符號右邊運算式的值指派給運算符號左邊的運算元。所以，以下敘述 s=a+b 是將 a+b 的值先計算，然後指派給 s，或稱賦值給 s。

```
s=0 ;a=3 ;b=5
s= a +b
print(s)
```

上式與數學的「等號」意義是不同的，所以不要一直糾結爲什麼 0 會等於 a+b，而是 a+b 先運算，然後指派或稱賦值給 s。其次，不能將常數放在指派運算子的左邊，例如：

```
8=a
```

此爲一個錯誤的敘述。但以下敘述，將常數 8 指派給變數 a 是正確的。

```
a=8
```

▶ 算術運算

以下是簡單的算術運算，請自行鍵入，並寫出執行結果。

```
a=5;b=4 #指派或稱賦值
print(a+b)                                    _____
print(a-b)                                    _____
print(a*b)                                    _____
print(a/b) #實數除法，得到實數商                  _____
print(a//b) #整數除法，得到整數商                 _____
print(a%b)                                    _____
print(a**2)                                   _____
print(9**(1/2)) #開根號                       _____
print(27**(1/3))#開立方                       _____
print(10**2)                                  _____
print(10**-2)                                 _____
print(1/10**2)                                _____
```

▶ 複合運算

程式設計常需要計算

```
s=s+a;s=s-a;s=s*a;s=s//a;s=s/a;s=s%a
```

所以以上可以寫成如下：

```
s+=a;s-=a;s*=a;s//=a;s/=a;s%=a
```

此稱爲複合運算子。請鍵入以下程式，並寫出執行結果。

```
s=0;a=3;b=4
s+=a
print(s)                                      _____
s-=b
print(s)                                      _____
```

▶ 整數除法與取餘的應用

平常的算術運算對於整數除法與取餘並無特別著重。但在程式設計的領域，這兩個運算子有其獨到用途。因爲整數除法與取餘可以將一個整數分解爲單一個數字，且很多應用都會用到此「分解」運算。例如，若要在顯

示器顯示 152，那就要將此數字先分解爲 1、5、2 三個數字，請自行鍵入以下程式，並觀察結果。

```
a=152
a3=a%10 #2 個位數
a=a//10
a2=a%10 #5 十位數
a=a//10
a1=a    #1 百位數
print(a1,a2,a3)
```

▶ 程式語言與數學的差異

數學有

```
19/10=1…9
```

的書寫習慣，但程式語言並沒有此表示方式，因此以上運算方式，一定要先取餘數，再求整數商如下：

```
a=19;b=10
c=a%b;#餘數
a=a//b; #整數商
print(a,c)
```

順序錯也不行，以下程式，先除再取餘數，結果就不對。

```
a=19;b=10
a=a//b;
c=a%b;
print(a,c)
```

自我練習

1. 請指派一個四位數，並將其反向輸出此四個數字且求其數字和。例如，指派 1234(a=1234)，那可以輸出 4 3 2 1 與 10。(數字之間留空白，且先不要用迴圈)
2. 請寫一個程式，可以指派 4 個數字，然後將其兜成一個 4 位數輸出。例如，指派 a=3,b=4,c=5,d=6，那可以輸出「3456」。

3. 請寫一個程式，可以指派 1 個 0 到 86399 的整數，然後分解爲「時:分:秒」的輸出格式。
4. 請寫一個程式，可以指派 1 個 0 到 999999 的整數，然後從右邊兩兩分解與輸出。例如，指派輸入 123456 則輸出 56,34,12。

關係運算子(Relational Operators)

關係運算子又稱爲比較 (Comparison) 運算子，用於資料之間的大小比較，比較的結果可得到布林值的 True 或 False，表 3-2 是 Python 中的關係運算子符號。

表3-2 Python 關係運算子

運算子	定義	優先順序	結合律
<	小於	9	由左至右
>	大於	9	由左至右
<=	小於等於	9	由左至右
>=	大於等於	9	由左至右
==	等於	9	由左至右
!=	不等於	9	由左至右

例如，請鍵入以下程式，寫出執行結果。

```
x=3;y=4
print(x==y) #False
print(x!=y)#True
print(x<y)#True
print(x=y)#錯，此爲指派運算子，不是關係運算子
```

邏輯運算子(Logical Operators)

邏輯運算子 (Logical Operators) 又稱爲布林 (Boolean) 運算子。當同一個運算式要同時存在兩個以上的關係運算子時，每兩個關係運算子之間必須使用邏輯運算子連結，例如，您要找『男生』且『年齡大於 40』，此一選擇就同時含有兩個關係運算式，此時就要使用邏輯運算子。下表是 Python 的邏輯運算子，C/C++ 用符號『！ ,&&,||』表示，Python 直接用單字表示，如表 3-3 所示，這樣反而比較好記。

表3-3　Python 邏輯運算子

運算子	定義	優先順序	結合律
not	邏輯否定運算	10	由右至左
and	邏輯 and 運算	11	由左至右
or	邏輯 or 運算	12	由左至右

▶ not

邏輯not稱爲否定運算，可將『True』轉爲『False』，『False』轉爲『True』。例如：

```
a=False
print(not a)
```

▶ and

邏輯 and 是兩件事都 True，結果才是 True，其餘都是 False，其眞值表如表 3-4。

表3-4　邏輯 and 真值表

事件1	事件2	結果
True	True	True
True	False	False
False	True	False
False	False	False

例如：

```
x=3;y=4
print(x>0 and y>0) #True
print(x>0 and x==y) #False
```

▶ or

邏輯 or 是兩件事只要有一件爲 True，那就爲 True，只有兩件事全爲 False，才爲 False，其眞值表如表 3-5。

表3-5 邏輯 or 真值表

事件1	事件2	結果
True	True	True
True	False	True
False	True	True
False	False	False

例如：

```
x=3;y=4
print(x>0 or x==y) #True
print(x<0 or x==y) #False
```

又例如：若有一數學式，判斷 x 是否滿足 1<x<=6 , 請轉換爲 Python 語言敘述。因爲 1<x<=6 是數學語言，Python 是

```
x>1 and x<=6
```

請鍵入以下程式，並觀察執行結果。

```
x=3
print(x>1 and x<=6)                    _____
x=8
print(x>1 and x<=6)                    _____
```

但太多初學者習慣將以上數學語言直接用。例如：

```
x=8
print(1<x<=6)                          _____
```

以上寫法，Python 竟然也可以接受，其它語言都不行，這也是物件導向抽象化的特性。

自我練習

1. 表決器。假設有一項評審工作，有 3 位評審，當其中兩人或以上同意，則表示過關，請設計此程式。
2. 請指派 1 個整數 x，判斷 x 是否滿足 x>3 or x<-2 ？

3. 請指派 3 個線段長度，判斷這 3 個線段，是否可圍一個三角形？
   (提示：任兩邊之和大於第三邊的數學語言是 :a+b > c and a+c > b and b+c > a，請將以上數學語言轉爲 Python。測試資料 3,4,5 可以；1,1,3 不可以)
4. 若有一數學式，如何同時判斷六個數字是否滿足 $\frac{a_1}{a_2}=\frac{b_1}{b_2}=\frac{c_1}{c_2}$ ，請轉換爲 Python 敘述。例如，a1,b1,c1=1,2,3 與 a2,b2,c2=2,4,6 爲 True；1,2,3 與 1,2,5 爲 False。
5. 若有一數學式，如何同時判斷六個數字是否滿足 $\frac{a_1}{a_2}=\frac{b_1}{b_2}\neq\frac{c_1}{c_2}$ ？

運算子的優先順序（Precedence）

同一敘述，若同時含有多個運算子，即須定義運算子的優先順序。例如：

```
x=a+b*c
```

在數學裡，是定義先乘除再加減，程式語言也是相同，也必須定義這些運算子的優先順序，茲將使用手冊的運算子優先順序摘錄如表 3-6，有些還沒介紹，那就先瀏覽。

表3-6　運算子優先順序表

Operator	Description	優先順序（1最優先）
`lambda`	Lambda expression	17
`if – else`	Conditional expression	16
`or`	Boolean OR	15
`and`	Boolean AND	14
`not x`	Boolean NOT	13
`in, not in, is, is not, <, <=, >, >=, !=, ==`	Comparisons, including membership tests and identity tests	12
`|`	Bitwise OR	11
`^`	Bitwise XOR	10
`&`	Bitwise AND	9

表3-6 運算子優先順序表（續）

Operator	Description	優先順序（1最優先）
<<, >>	Shifts	8
+, -	Addition and subtraction	7
*, @, /, //, %	Multiplication, matrix multiplication, division, floor division, remainder 5	6
+x, -x, ~x	Positive, negative, bitwise NOT	5
**	Exponentiation 6	4
await x	Await expression	3
x[index], x[index:index], x(arguments...), x.attribute	Subscription, slicing, call, attribute reference	2
(expressions...), [expressions...], {key: value...}, {expressions...}	Binding or tuple display, list display, dictionary display, set display	1

請自行鍵入以下程式，並寫出執行結果。

```
print(3**2)                                    _____
print(-3**2)                                   _____
print((-3)**2)                                 _____
print(9**(1/2))                                _____
print(9**1/2)                                  _____
print((3/10)**2)                               _____
print(3/10**2)                                 _____
print(4+3**2%2+1)                              _____
```

運算子的結合律（Associativity）

當同一敘述，相鄰的運算子擁有相同優先順序的運算子，即須定義運算子是「由左至右」的左結合或「由右至左」的右結合。例如：

```
x=a-b-c
```

同樣是減號「-」，優先順序相同，此時就要靠定義結合律，減法結合律是由左至右，所以以上同義於

```
x=((a-b)-c))
```

而

```
a=b=c=2
```

指派運算子的結合律是由右至左，所以以上式子同義於：

```
(a=(b=(c=2)))
```

請鍵入以下程式，寫出執行結果。

```
a,b,c=2,3,4
x=a+b-c
print(x)                                    _____
a=b=c=3
print(a,b,c)                                _____
```

函式

所有的編譯器爲了節省有限的記憶體，都是將最常用的運算寫成運算子，如以上的算術、關係、邏輯運算子。比較不常用的運算，例如，取絕對值就以函式 abs() 表示。函式將會在第七章詳細介紹，以下先將常用函式介紹如下：

▶ abs(x)

傳回數值 x 所對應的絕對值。例如：

```
a=-3
print(abs(a))#3
```

▶ round(a)

將數值 a 四捨五入到整數。例如：

```
print(round(255.5))#256
```

▶ str(a)

將數值 a 轉爲字串。例如：

```
a=123
b=456
print(a+b)
```

此爲數值相加，結果是 579，若要將這些數字進行字串串接，則要先轉爲字串，再串接，程式如下：

```
print(str(a)+str(b))  #字串串接123456
int(x)
```

將字串 x 轉爲數值。例如：

```
a="123"
b="456"
print(a+b)  #此爲字串串接123456
print(int(a)+int(b))  #先轉數值，則爲數值相加
```

▶ ord(x)

傳回字元 x 的 ASCII 編碼。例如：

```
print(ord("A"))  #65
```

補充說明

什麼是 ASCII 編碼呢？因爲資料要由電腦運算，所以每一個數字、符號、大小寫英文字元，都要數位化，才能用電腦儲存，再進行運算。這些常用字元沒有超過 127 個，所以使用 7 位元編碼，就可以儲存常用字元，此稱爲 ASCII 編碼。例如：字元 A 編碼爲「01000001」=65，叫到 65 號就是字元「A」;字元「B」的編碼是「01000010」=66，叫到 66 號，就是「B」;字元「1」的編碼是「00110001」=49，叫到 49 號就是「1」。就如同每個同學都有座號，所以就有對應位置。

▶ chr(x)

傳回數值 x 所對應的字元。例如：

```
a=65
print(chr(a))#A
```

▶ len(a)

傳回字串 a 的長度。例如：

```
a="abcd"
print(len(a))#4
```

▶ max(x)、min(x)

傳回參數 x 的極大或極小值。例如：

```
b=6;c=3;d=9
print(max(b,c))#6
print(min(b,c,d))#3
```

▶ eval(x)

傳回字串 x 算術運算式的運算結果。例如：

```
print(eval("3+2*4"))#11
```

這也很神，例如：用來製作計算器就很方便，只要讓使用者按完一個運算式，Python 就幫您自動計算結果。

3-4 基本指令的操作

以上已經介紹一些程式語法規則，以下介紹一些 Python 基本指令的操作，例如：變數的宣告、指令的排列、註解等。更多的指令介紹，將在後續章節陸續介紹。

變數的宣告

大部分的程式語言在變數宣告的同時要指派其型態是整數、實數或字串，但是 Python 只要指派其初值就好，編譯器會依照實際情況指派記憶體的多寡。請鍵入以下程式，並觀察執行結果。

```
a=2  # 整數
print ( 2 ** 100 ) # 2 的100 次方
b = 3.4   # 浮點實數
print ( b ** 100 ) # b 的100 次方
```

程式指令的排列

以上我們都將是同一列僅放置一個敘述，若要將兩個敘述放在同一列，請用分號「;」，例如：

```
a=1;b=2;print(a+b)
```

而以下程式，雖然沒有錯，但卻是多此一舉，因爲這不是 C 或 Java，每一列不用以分號「;」做結束。

```
a=1; b=2;
print(a+b);
```

▶ 變數不用宣告的困境

變數只要放在指派運算子「=」左邊就等於宣告，這樣就可在指派運算子右邊使用，但也會有其缺點。請鍵入以下程式，並觀察執行結果

```
student=0
student=stutent+1  # 本例表示s t u d e n t 變數要執行累加的動作
print(student)
```

以上第二列程式指派運算子「=」右邊的 stutent 拚字錯誤（因爲此變數未曾在左邊出現過），Python 解譯器可以馬上幫忙找出錯誤，就可以修正如下：

```
student=0
student=student+1   # 本例表示s t u d e n t 變數要執行累加的動作
print(student)
```

得到正確結果。但以下程式可就沒那麼順利了。

```
student=0
stutent=student+1
print(student)
```

請問錯在哪了？剛剛是錯在指派運算子「=」的右邊，Python 找出了錯誤，這次則是錯在指派運算子左邊。因爲指派運算子「=」右邊的變數要先宣告，這樣可預防打字錯誤，左邊則是你說了算，電腦不會幫忙除錯，所以指派運算子左邊的變數要特別留意。又例如：

```
Score=1
score=Score+1 （本例表示同一變數要執行累加的動作）
print(Score)
```

沒有出現錯誤訊息，但結果也是錯誤的，因爲大小寫不同，表示不同的變數名稱，請特別留意運算子左邊的變數名稱不能錯誤。以下程式也不行，不小心重複使用同一變數，造成嚴重錯誤。

```
a=[3,4,5];print(a)#宣告a爲串列
a=6;print(a)#又宣告a爲單一變數
if a>3:
    print("OK")#OK
if a[2]>2:#錯誤，因a已經是單一變數了
    b=3
```

▶ 變數同時交換內容

變數同時交換內容是程式設計常需使用的運算，例如，變數 a,b 要交換內容，大部分的程式語言都要先找一個臨時變數 t，程式如下：

```
a=1;b=2
t=a
a=b
b=t
print(a,b)#2 1
```

Python 就允許使用者同時交換變數內容。請鍵入以下程式，並觀察與寫出執行結果。

```
a,b=1,2
a,b=b,a
print(a,b)#2,1
a,b,c=1,2,3
a,b,c=b,c,a
print(a,b,c) #2,3,1
```

註解 (Comments)

適當的程式註解才能增加程式的可讀性，註解是僅給人看，電腦不會理會。Python 的註解有兩種方式。第一，使用『"""』當註解開頭，直到遇到『"""』，兩個『"""』中間的文字通通是註解。

```
"""我是註解
我也是註解"""
```

上式『""" 』符號以後的字串視爲註解，編譯程式不予處理，直到遇到『 """ 』爲止。其次，同一列中，井號『#』後面的也視爲註解，編譯器均不予處理。例如：

```
# 我是註解
```

前者，因爲可超過兩列，較適合編寫較長的註解，後者，則僅能寫在同一列。

亂數

電腦常常需要模擬一些執行結果，例如，樂透開獎、紙牌遊戲或擲骰子，此時就需要產生亂數，Python 產生亂數的方法是使用 random 模組。使用這些模組前，請先載入此模組，程式如下：

```
import random
a=random.randint(1,6)#產生1～6整數亂數
print(a)
```

每次用『類別名稱 . 方法』有點麻煩，可以取一個簡單的物件名。例如：

```
import random as a # a是物件名
b=a.randint(1,6) #產生1～6整數亂數
print(b)
```

▶ randint(min,max)

傳回 min(含) 與 max(含) 之間的亂數。例如，以下程式可模擬擲骰子的結果。

```
import random
a=random.randint(1,6)
print(a)
```

（補充說明：以上 randint 在物件導向的程式設計稱爲方法。）

小寫字元的 ASCII 是 97～122，所以以下程式，可產生一個小寫字元。

```
import random
a=random.randint(97,122) #產生97～122整數亂數
print(a)
print(chr(a))#傳回對應字元
print("%c" %a)#將整數以字元形式輸出
```

ord() 函數可取得傳入字元的 ASCII 值，所以以下程式也可以傳回一個小寫字元。

```
import random
a=random.randint(ord('a'),ord('z'))#ord傳回對應整數
print(a)
print(chr(a))
print("%c" %a)
```

自我練習

1. 請寫一個程式，可以產生大寫字元。
2. 請寫一個程式，可以產生二位數的整數。
3. 請寫一個程式，可以產生一個介於 0~1 且包含 0，但不含 1 的浮點實數，小數點取 2 位。

字串運算子

Python 是一個物件導向程式語言。物件導向的抽象化，就是希望程式語法能越來越接近人類語言，所以，Python 開發『+, *, in』等運算子 , 希望字串的運算也與人類的運算習慣相同。請鍵入以下程式：

```
a="aa"
b='Mary'
print (a+b)#合併 aaMary
c=a*3
print(c)#三次  aaaaaa
print ('a' in b) #True
print ( 'a'  not in b)#False
```

補充說明

以上「+」、「*」都是物件導向「多型」的表現，使用者都是使用相同的運算子，但編譯器就能依照實際情形，完成整數、浮點數與字串等運算。

字串與數值互轉

a=12 是數值，b="34" 是字串，但是有時候輸入時是字串，但要改爲數值來運算；或某些數值希望轉爲字串處理，此時就要先用 str() 將數值轉爲字串，用 int() 將字串轉爲數值，如此才能進行相關運算。例如：

```
a=12;b=34
print(a+b)#46
print(str(a)+str(b))#1234
c='12';d='34'
print(c+d)#1234
print(int(c)+int(d))#46
```

但是，不是數字的字串，例如：強制將 "Mary" 轉爲數值，也是沒意義，且發生錯誤訊息。

```
b="Mary"
c=int(b)#錯
print (c)
```

字串的長度

字串的長度也是依使用者習慣，英文字元與中文字元都能順利按照其習慣，計算其長度。(以前的程式語言，一個中文字長度是 2，這樣雖然對，但是反而不好資料處理，這也是物件導向「多型」的表現。)

```
a='aa'
b='泉勝'
print(len(a))#2
print(len(b))#2
```

字串的字元處理

字串是由字元組成，若我們要取個別字元，也可直接用串列 (list) 處理，串列請看第六章，例如：

```
a="a"
b='ab'
c='Mary'
d="國勝"
print(a[0])# a
print(b[0],b[1])# a b
print(d[0],d[1])#國 勝
```

字串的分割

Python 並沒有時間與日期等資料型態，關於日期與字串都是以字串表示，例如：

```
a="112/3/2"
b="18:30:22"
```

此時若要進一步處理日期與時間相關運算問題，則需要分解。Python 提供 split() 函式分解字串。例如：請鍵入以下程式，寫出執行結果。

```
a="112/3/2"
y,m,d=a.split('/')#分解後資料型態爲字串
print(y,m,d)                                    _____
print(y+m+d)                                    _____
```

```
y,m,d=int(y),int(m),int(d)#轉爲數值
print(y,m,d)                                    _____
print(y+m+d)                                    _____
```

以上分解 (split)、轉數值 (int) 兩個動作，亦可使用 map() 函式。例如：請鍵入以下程式。

```
y,m,d=map(int,a.split('/'))
print(y+m+d)                                    _____
```

自我練習

1. 請計算以下兩個時間相差的時間間隔。（提示：通通轉爲秒數）

```
b1="18:30:22"
b2="20:10:42"
```

2. 請判斷以下兩個時間何者在前。（提示：通通轉爲秒數）

```
b1="18:30:22"
b2="20:10:42"
```

3. 請判斷以下時間 b 有沒有在時間 b1、b2 的中間。（提示：通通轉爲秒數）

```
b="17:40:52"
b1="18:30:22"
b2="20:10:42"
```

發聲

要讓電腦發聲，要先載入 ctypes 類別，且取 ctypes.windll.kernel32 物件。例如：

```
import ctypes
p = ctypes.windll.kernel32
```

電子琴鍵盤頻率對照表如表 3-7。

表3-7　電子琴鍵盤頻率對照表

	n	1	2	3	4	5	6	7	8	9	10	11	12
音階	音符	C (Do)	C# (Do#)	D (Re)	D# (Re#)	E (Mi)	F (Fa)	F# (Fa#)	G (So)	G# (So#)	A (La)	A# (La#)	B (Si)
低音	頻率 (Hz)	262	277	294	311	330	349	370	392	415	440	466	494
中音	頻率 (Hz)	523	554	587	622	659	698	740	784	831	880	932	988
高音	頻率 (Hz)	1046	1109	1175	1245	1318	1397	1480	1568	1661	1760	1865	1976

以下程式可讓電腦發出 Do、Re、Mi、Fa…等 8 個音，523、587 就是震盪的頻率。

```
import ctypes
p = ctypes.windll.kernel32
p.Beep(523,200)
p.Beep(587,200)
p.Beep(659,200)
p.Beep(698,200)
p.Beep(784,200)
p.Beep(880,200)
p.Beep(998,200)
p.Beep(1046,200)
```

以上的數值 200，表示發聲的延遲的時間，單位是 ms。

演奏音樂

要讓電腦演奏音樂，那就要先下載一個簡譜，如圖 3-1。

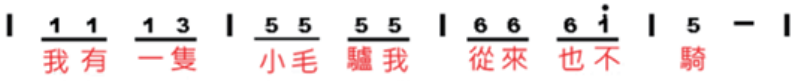


圖3-1　小毛驢簡譜

本例一分鐘 80 拍，所以 1 拍的時間是，60000ms/80，以一個四分音符爲 1 拍，所以上圖一個八分音符所佔的時間是 60000ms/(2*80)，每小節是 2 拍。其次，上圖，所有音符的最小延遲時間是八分音符，所以就取八分音

符的時間爲 1 個單位。第 1 音『1 我』半拍，取 1 個單位;『5- 騎』是 2 拍，取 4 個單位。

資料的數位化

圖 3-1 的小毛驢簡譜，所有音符的最小延遲時間是八分音符，所以就取八分之一音符的時間爲 1 個單位，以下程式可演奏『我有一隻小毛驢，我從來也不騎』。

```
import ctypes
p = ctypes.windll.kernel32
t=60000//160
p.Beep(523,t)
p.Beep(523,t)
p.Beep(523,t)
p.Beep(659,t)
p.Beep(784,t)
p.Beep(784,t)
p.Beep(784,t)
p.Beep(784,t)
p.Beep(880,t)
p.Beep(880,t)
p.Beep(880,t)
p.Beep(1046,t)
p.Beep(784,4*t)
p.Beep(784,t)
```

若加上歌詞，就有點像 KTV 了，程式如下：

```
import ctypes
p = ctypes.windll.kernel32
t=60000//160
print('我')
p.Beep(523,t)
print('有')
p.Beep(523,t)
print('一')
p.Beep(523,t)
print('隻')
p.Beep(659,t)
```

```
print('小')
p.Beep(784,t)
print('毛')
p.Beep(784,t)
print('驢')
p.Beep(784,t)
print('我')
p.Beep(784,t)
print('從')
p.Beep(880,t)
print('來')
p.Beep(880,t)
print('也')
p.Beep(880,t)
print('不')
p.Beep(1046,t)
print('騎')
p.Beep(784,4*t)
p.Beep(784,t)
```

但是以上程式有點冗長，等到介紹串列與迴圈，程式就會簡化。

自我練習

1. 請自行下載一個簡譜，撰寫程式演奏音樂（請含歌詞）。

3-5 標準輸出輸入的操作

Python 使用 print() 函式輸出，使用 input() 函式輸入，分別說明如下：

print()

前面我們已經使用 print 函式輸出結果，每一個 print() 預設輸出後跳列。例如：

```
print('aa')
print('bb')
```

輸出結果是：

```
aa
bb
```

若您不想讓它跳列，請自行指派 end 值，程式如下：

```
print('aa',end='')
print('bb')
```

輸出結果是：

```
aabb
```

又例如：

```
print('aa',end=',')
print('bb')
```

輸出結果是：

```
aa,bb
```

print() 內若是接變數，就輸出變數的內容，請鍵入以下程式，並觀察與寫出輸出結果。

```
a,b,c=1,2,3
print(a)                                    _________
print(b)                                    _________
print(c)                                    _________
```

若您不想讓它跳列，也是請自行指派 end 值。請鍵入以下程式，並觀察與寫出輸出結果。

```
a,b,c=1,2,3
print(a,end=' ')
print(b,end=',')
print(c)                                    _________
```

若要在輸出結果套用文字說明，如「a=2」，則可使用『%』當作控制字元。使用「%」控制資料輸出時，還要依資料的型態使用對應的符號，整數請用「d」，浮點數使用「f」，字元用「c」，字串請用「s」，如表 3-8。

表3-8　資料型態列印格式對照表

資料型態	列印格式
整數	d
浮點數	f
字元	c
字串	s

請鍵入以下程式，並觀察與寫出輸出結果。

```
a=2
print("a=%d" %(a))
```

又例如：

```
a,b,c=1,2,3
print("a=%d,b=%d,c=%d" %(a,b,c))
```

以上「a=」是字串直接輸出，控制字元「%」會到後面變數區 (a,b,c) 依照順序找對應的變數。所以，輸出會是 :

```
 a=1,b=2,c=3
```

又例如：

```
a="永松";b=56
print("帥哥%s 年齡是%d" %(a,b))
```

輸出結果是：

```
帥哥永松 年齡是56
```

浮點數輸出是 %f。請鍵入以下程式，並觀察與寫出輸出結果。

```
a=3.4
print("%f" % a)
print("%d" % a)#以整數輸出
```

若是控制字元 % 接數字，則表示此欄位的寬度且靠右排列，剩的以空白表示。例如：

```
a="Mary"
b,c=1,2
print("123456789012345678901234")
print("a=%6s,b=%3d,c=%4d" %(a,b,c))
```

輸出結果如下圖：

```
123456789012345678901234
a=  Mary,b=  1,c=   2
```

input()

前面我們一直都用指派的方式指派變數值，input() 可以讓使用者於程式執行後輸入資料。例如，以下程式可以讓我們輸入資料。

```
a=input("input a:")
print(a)
```

其次，Python 將輸入的資料一律視爲字串，所以，若您要進行數值運算，那請先用 int() 函數轉爲數值。請鍵入以下程式、輸入數值，並觀察結果。

```
a=input("input a:")
b=input("input b:")
print(a+b)
a=int(input("input a:"))
b=int(input("input b:"))
print(a+b)
```

eval()

前面使用 input() 每次僅能輸入一筆資料，善用 eval() 函式可以讓我們同時輸入多筆資料，資料的分隔採用逗號「,」。請鍵入以下程式，輸入『12,3,4』，並觀察輸出結果。

```
a,b,c=eval(input("input a,b,c:")) #數字請用逗號『,』隔開，例如 12,3,4
print(a,b,c)                                                    _____
print(a+b+c)                                                    _____
```

請留意其資料型態是數值。eval 還可輸入一個數學運算式，例如：

```
a=eval("3+4*2" )
print(a)                                                        _____
```

結果是 11。請鍵入以下程式，並輸入『3+4*2』，並觀察輸出結果。

```
a=eval(input("input a 數學運算式:" ))
print(a)                                                        _____
```

所以利用 eval() 函式，待學完視窗輸出入元件，很快就可以自行作出具有按鍵功能的小型計算機功能。

範例3-5a

請寫一程式，可以輸入長方形的長與寬，並計算周長與面積，且輸出結果。

程式列印

```
a=int(input("input a:"))
b=int(input("input b:"))
c=a*b
d=2*(a+b)
print("面積=%d ,周長=%d" %(c,d))
```

執行結果

```
input a:3

input b:4
面積=12 ,周長=14
```

補充說明

1. 本例若用

```
a,b=eval(input("input a,b:"))
```

則 a,b 的資料型態是數值，所以程式如下：

```
a,b=eval(input("input a,b:"))
c=a*b
d=2*(a+b)
print("面積=%d ,周長=%d" %(c,d))
```

但使用 input 所輸入的資料型態卻是字串，此時要先轉數值再運算，請特別留意。

自我練習

1. 請分四次輸入 1 個 0 到 9 的整數，並將它合併爲 1 個整數。例如，輸入 1，輸入 2，輸入 3，輸入 4，則輸出 1234。
2. 請先輸入 3 個 0 到 9 的整數，再輸入兩個 0 到 9 的整數，並將其合併爲 1 個浮點數。例如，輸入 1，輸入 2，輸入 3，輸入 4，輸入 5，則輸出 123.45。

3-6 運算子與運算元的應用

以上已經介紹算術運算子與運算元，有了運算子與運算元，我們就可以先將國小與國中的數學問題以程式語言爲工具，撰寫程式，而由電腦求出解答，請看以下範例。

範例3-6a

請寫一程式，可以任意輸入兩個點，計算其距離。

程式設計步驟

1. 將資料數位化與變數設計。本例指派兩個座標如下：

```
x1=3;y1=0
x2=0;y2=4
```

2. 寫出演算法。本例已知兩點座標分別是 (x1,y1),(x2,y2)，則其距離的公式的數學語言是：

$$d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

3. 以程式語言完成演算法。以上的距離公式以 Python 語言可表示如下：

```
d=((x1-x2)**2+(y1-y2)**2)**(1/2)#此爲Python語言表示法
```

4. 輸出此兩點距離。

```
print(d)#5.0
```

5. 完整程式列印如下：

```
x1=3;y1=0
x2=0;y2=4
d=((x1-x2)**2+(y1-y2)**2)**(1/2)#此爲Python語言表示法
print(d)#5.0
```

執行結果

```
5.0
```

補充說明

1. 本例所有變數內容先使用『指派』，請練習改爲輸入。
2. 因爲有運算子優先順序問題，所以請留意括號不能省略。

自我練習

1. 輸入三角形三邊長 a、b、c，求其面積。（提示：先計算 d=(a+b+c)/2，則三角形面積＝$\sqrt{d(d-a)(d-b)(d-c)}$，本例假設所輸入的三角形三邊長可圍成三角形，例如指派 a=3;b=4;c=5 則得三角形面積 6。其次，並不是任意三條線都可圍成三角形，若要判斷是否可圍成三角形，請繼續研讀下一章）
2. 請寫一程式，可以指派三點座標，輸出其面積。提示：三角形面積公式如下。

$$=\frac{1}{2}\begin{vmatrix} x_1 & x_2 & x_3 & x_1 \\ y_1 & y_2 & y_3 & y_1 \end{vmatrix}$$ 其中↘乘爲正，↙乘為負

$$=\frac{1}{2}|(x_1y_2+x_2y_3+x_3y_1)-(x_2y_1+x_3y_2+x_1y_3)|$$

例如：x1=0;y1=0;x2=3,y2=0;x3=0;y3=4，執行結果是 6。

資料的數位化與變數設計

前面都是指定數值，完成某些功能運算，但有時候，資料比較複雜，這時就需要將資料抽象出來，此資料抽象的過程，又稱爲資料的數位化或資料的抽象。例如：若有一直線方程式是 ax+by+c=0，點 p(m,n) 到直線的距離是

$$d=\frac{|am+bn+c|}{\sqrt{a^2+b^2}}$$ (提示：絕對值函式是 abs())

這件事要由電腦作，就要將資料從人類約定的書寫習慣 ax+by+c=0 獨立抽象出來。例如，直線方程式是 3x+4y+5=0，求點 p(1,2) 到直線的距離，首先，以變數

```
a=3;b=4;c=5
```

約定此爲直線 3x+4y+5=0。然後以變數

```
m=1;n=2
```

約定此爲點 p(1,2)，以上過程稱爲資料的抽象化與變數的設計。所以全部程式如下：

```
a=3;b=4;c=5
m=1;n=2
d=abs(a*m+b*n+c)/((a*a+b*b)**(1/2))
print(d)
```

又例如，您要判斷 q(1,-2) 是否在直線上，也是將資料抽象出來以變數來表示，程式如下：（if 請看下一章）

```
a=3;b=4;c=5
m=1;n=-2
if (a*m+b*n+c)==0:
    print("yes")
else:
    print("no")
```

以上 3x+4y+5=0 是人類在數學上書寫的約定，電腦只要給 a,b,c，我們就約定此爲 ax+by+c=0，此稱爲資料的抽象化與變數的設計。

範例 3-6b

寫一個程式，可以輸入一個一元二次方程式，並求其解（本例假設所輸入的方程式恰有二解）。

程式設計步驟

1. 資料的數位化與變數設計。設有一元二次方程式如下：

```
ax² + bx + c = 0
```

本例指派 a,b,c 三個整數如下：

```
a=2;b=-7;c=3
```

即表示此爲方程式 $2x^2-7x+3=0$。

2. 寫出演算法。

國中解一元二次方程式是先用配方法，配方法需要一點判斷，可減少計算。但是本範例使用公式法，公式法雖計算較多，但完全不用判斷，最適合由電腦完成。這種強調多計算少判斷，就是電腦與人類不同的運算思維，人類計算能力較差，所以希望使用一些技巧來簡化計算，但是電腦則是計算能力強，判斷能力差，所以強調用計算來簡化程式。解一元二次方程式的公式法，演算步驟如下：

(1) 令 d= $\sqrt{b^2-4ac}$ 。提示：此爲數學語言，以 Python 語言表示則是 d=(b*b-4*a*c)**(1/2) ，括號、乘號均不能省。)

(2) 則其二解分別爲 $x_1=\frac{-b+d}{2a}$ ， $x_2=\frac{-b-d}{2a}$ 。

（提示：此爲數學語言，以 Python 語言表示則是 x1=(-b+d)/(2*a)，括號、乘號均不能省。)

(3) 例如，$2x^2-7x+3=0$。其解爲 $x_1=3.0$,$x_2=0.5$。

3. 以程式語言完成演算法。本例程式參考程式如下：

```
a=2;b=-7;c=3
d=(b*b-4*a*c)**(1/2)
x1=(-b+d)/(2*a)
x2=(-b-d)/(2*a)
print(x1)#3.0
print(x2)#0.5
```

執行結果

```
3.0
0.5
```

補充說明

1. 本例先假設所輸入的係數一定有實數解，待下一章再判斷所輸入係數有沒有實數解。
2. 電腦運算思維：人有人的特性與優勢，電腦有電腦的特性與優勢，所謂電腦運算思維，就是要學習電腦有哪些特性與優勢，然後使用電腦的運算思維寫程式。以本範例爲例，人類可能用配方法較快，但是電腦是使用公式法較快，此即爲電腦的運算思維，往後各章還有更多電腦運算思維，學習這些運算思維，就可增加程式設計功力，請繼續研讀以下各章，即可瞭解。

自我練習

1. 寫一個程式，可以輸入一個二元一次方程式，並求其解（本例假設所輸入的方程式恰有一解）。

運算思維

解二元一次方程式，國中會先教代入消去法與加減消去法，這都需要一點判斷，但可以簡化計算。本題使用公式法，可完全不用判斷，直接計算而得，這樣最適合計算機了，此也是電腦的運算思維。解二元一次方程式的克拉馬公式演算過程如下：

(1) 假設二元一次方程式如下：

$a_1x + b_1y = c_1$

$a_2x + b_2y = c_2$

(2) 指派或輸入 a1,b1,c1,a2,b2,c2 六個整數。（本例假設整係數方程式）

(3) 令 $d = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1b_2 - a_2b_1$。

(4) 則其解分別是 $x = \dfrac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{d} = (c_1b_2 - c_2b_1)/d$　　$y = \dfrac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{d} = (a_1c_2 - a_2c_1)/d$

(5) 例如：

3x+y=5

x-2y=-3

則其解爲 x=1 y=2

計數器

日常生活常需要計數，例如，景點通常有一個計數器，累計今天進園人數。程式設計也不例外，常常需要統計某些敘述的執行幾次，此件事由電腦做的程式如下：首先，先宣布與清除計數值變數。

```
q=0;
```

計數值加 1 的程式如下：

```
q=q+1;
print(q)                                          _____
```

請鍵入以下程式，並觀察與寫出執行結果：

```
q=0
a=8
b=3
a=a-b
q=q+1
print(a,q)                                        _____
a=a-b
q=q+1
print(a,q)                                        _____
```

累加器

生活上也常面對累加問題，程式設計的累加程式如下：

```
a=2
s=0
s=s+a
print(s)
s=s+a
print(s)
```

自我練習

1. 請鍵入以下程式，寫出執行結果。

題號	程式	執行結果
1	a=21 b=9 r=a%b print(r) a=b b=r r=a%b print(a) print(b) print(r)	
2	a=6 n=2 r=0 d='' r=a % n d=str(r)+d print(d) a=a//n print(a) r=a%n d=str(r)+d print(d) a=a//n print(a) r=a%n d=str(r)+d print(d)	

3	a=542 s=0 n=0 r=a%10 n=n+1 s=s+r a=a//10 r=a%10 n=n+1 s=s+r a=a//10 r=a%10 n=n+1 s=s+r print(s) print(n)	
4	a,b="12","34" d="" d=d+a print(d) d=d+b print(d) d=b+d print(d) d=a+d print(d)	

3-7 程式的除錯（Debug）

程式除錯是所有程式設計者的惡夢。程式設計常見的錯誤是指令拚字錯誤、演算法錯誤等，分別說明如下：

拚字錯誤

拚字錯誤是初學者最容易發生的錯誤，但是現在很多整合開發環境已經將程式解析，給予提示。圖 3-1 是 Spyder 的解譯指令畫面，圖 3-2 是 Python 官版解譯指令畫面，兩者都能將指令、函式、常數解析給予不同的顏色。所以當鍵入程式並打完關鍵字時，若沒有變色，此時就要馬上更正，直到指令變色爲止。 Spyder 更強的是，還會給指令提示，幫忙補冒號、將

對稱的括號換色。例如：圖 3-2「as」前已經出現錯誤訊息，Spyder 爲其標示顏色，表示「as」是保留字，不能拿來當識別字。

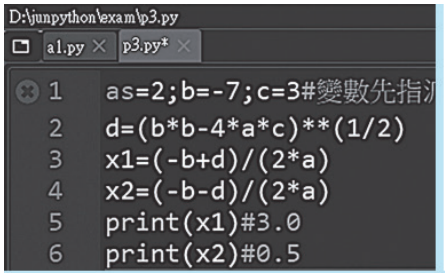


圖3-1 Spyder解譯指令畫面

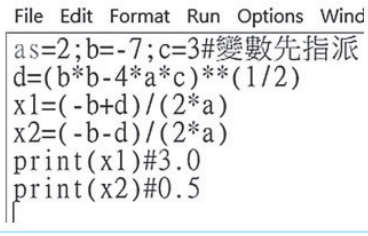


圖3-2 Python IDLE解譯指令畫面

演算法錯誤

若程式沒有錯誤訊息，結果卻錯誤，此時就要一步一步執行與觀察變數的變化。觀察的方法有兩種，分別是自行使用 print() 方法與使用整合編輯環境的 Debug 功能，分別說明如下：

▶ print()方法

在程式中，我們可以自行使用 print() 輸出變數的內容。例如，程式原本如下：

```
a,b,c=eval(input("input a,b,c:"))
d=(b*b-4*a*c)**(1/2)
x1=(-b+d)/(2*a)
x2=(-b-d)/(2*a)
print(x1)#3.0
print(x2)#0.5
```

我們可以中途先輸出變數內容 a,b,c,d，這樣可以縮小範圍，找出程式哪裡錯。

```
a,b,c=eval(input("input a,b,c:"))
print(a);print(b);print(c)
d=(b*b-4*a*c)**(1/2)
print(d)
x1=(-b+d)/(2*a)
```

```
x2=(-b-d)/(2*a)
print(x1)#3.0
print(x2)#0.5
```

▶ Debug

現今大部分整合開發環境都有 Debug 功能，Spyder 也不例外，點選 Spyder 功能表的「Debug/Debug」，畫面如圖 3-3，請留意第 1 行敘述出現箭號。

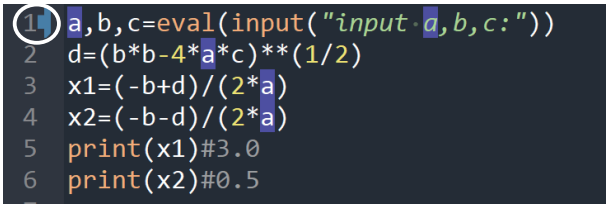


圖3-3　Spyder Debug畫面

逐一點選工具列的「Run current line」，即可逐一執行箭號所在敘述。圖 3-4 是執行 input() 且輸入資料後，於輔助說明區點選「Variable explorer」，此窗格會自動出現變數內容，如圖 3-5，此時即可觀察每個變數的內容。

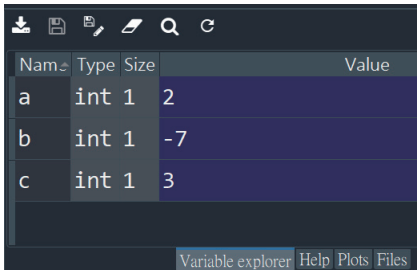


圖3-4　變數值窗格（一）

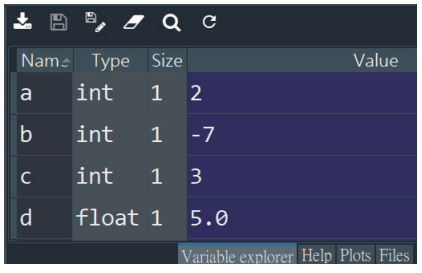


圖3-5　變數值窗格（二）

自我練習

1. 請將上一節的自我練習，以 Debug 追蹤每一個變數的結果。
2. 請同學們兩個人一組，每個人將自己的程式設定 2 個錯誤，然後交換電腦，由對方更正這些錯誤。

3-8 本章內容摘要

1. 我們平常處理的資料依其型態可分爲整數、浮點實數、字元、字串、與布林值。
2. Python 的運算子分爲四大類，分別是算術運算子、複合指派運算子、關係運算子、邏輯運算子。
3. 優先順序用來決定同一式子擁有多個運算子時，每一個運算子進行運算的優先順序。
4. 結合律則決定了在同一敘述中，相鄰的運算子有相同優先順序的運算子執行的順序。
5. Python 算術運算子有次方 (**)、乘法 (*)、除法 (/)、整數除法 (//)、取餘 (%)、加法 (+)、減法 (-)、指派 (=) 等。
6. Python 關係運算子有 <、>、<=、>=、==、!= 。
7. Python 邏輯運算子有 not、and、or。
8. abs() 是絕對值函式，int() 可將字串轉數值，str() 可將數值轉字串，round() 可將數值取四捨五入。
9. Python 變數的宣告語法只要宣告其初值如下：

```
a=0
```

10. Python 變數可同時交換。例如：

```
a,b=b,a
```

11. Python 的註解有兩種方式。第一，使用『"""』當註解開頭，直到遇到『"""』，兩個『"""』中間的文字通通是註解。第二，同一列中，井號『#』後面的也視爲註解。
12. Python 產生亂數的方法如下：

```
import random
a=random.randint(1,6)#產生1～6整數亂數
```

13. Python 數值相加與字串串接分別如下：

```
a=3;b=4
print(a+b)  #數值相加_________
print(str(a)+str(b))  #字串串接_________
```

14 字串可用串列直接取子字串，串列將於第 6 章介紹。例如：

```
a='0123'
print(a[0])  #_______
```

15. map() 與 split() 函式可協助字串分割。例如：

```
a="112/3/2"
y,m,d=a.split('/')            #得到字串型態
y,m,d=map(int,a.split('/'))    #得到數值型態
```

16. 要讓電腦發聲，要先載入 ctypes 類別，且取 ctypes.windll.kernel32 物件。例如：

```
import ctypes
p = ctypes.windll.kernel32
p.Beep(523,200)
```

17. Python 使用 print() 函式輸出，使用 input() 函式輸入。
18. 計數器的程式如下：

```
p=0；  p=p+1
```

19. 累加器的程式如下：

```
a=2；    s=0
s=s+a
```

課後評量

一、填充題（請寫出以下程式執行結果）

題號	題目	輸出結果
1	a=0x3b; print(a)	
2	a=3.9E3 print(a)	
3	a=8;b=3;c=2 print(a**b) print(a//b) print(a/b) print(a%b) print(a**(1/3)) print((1/2)**2) print(5**-2)	
4	a=3;s=0 s+=a print(a) a=a-1 print(a)	
5	a=162 b=a%10 print(b) a=a//10 print(a)	
6	print(3==2) print(4!=2)	
7	print(3>=2 or 4==5)	
8	print(not(3>=2 and 4==5))	
9	print(9**(1/2)) print(9**1/2) print(1/3**2) print(1/2**2/4)	
10	print(3-2-2) print(2**2**3)	

11	money=3 momey=money+3 print(money)	
12	a=3;b=4 a,b=b,a print(a)	

13	a=3
	#a=a+1
	print(a)	

14	a=123;b=456 print(a+b) print(str(a)+str(b))	
15	a='123' print(a+a) print(a*3) print(int(a)+2)	
16	a='1234' print(len(a)) print(a[0]+a[1]) print(int(a[0])+int(a[1]))	
17	a='東西南北' print(len(a)) print(a[0]+a[1])	
18	a="112/11/2" y,m,d=a.split('/') print(y+m+d) y,m,d=map(int,a.split('/')) print(y+m+d)	
19	print('aaa',end='') print('bbb')	
20	a,b,c=1,2,3 print("123456789012345678 90") print("a=%2d,b=%3d,c=%4d" %(a,b,c))	
21	a=65 print("%d"%a) print("%c"%a)	
22	a="老師";b=56 print("帥哥%s,年齡是%d" %(a,b))	

23	a=eval("3+4**2//4") print(a)	
24	s=0 s=10*s+1 s=10*s+2 s=10*s+3 print(s)	
25	x1=3;y1=0 x2=0;y2=-4 d=((x1-x2)**2+(y1-y2)**2)**(1/2) print(d)	
26	a,b,c=3,4,5 d=(a+b+c)/2 s=(d*(d-a)*(d-b)*(d-c))**(1/2) print(s)	
27	x1=0;y1=0;x2=3;y2=0;x3=0;y3=4 r=abs((x1*y2+x2*y3+x3*y1)- (x2*y1+x3*y2+x1*y3))/2 print(r)	
28	a=3;b=4;c=5 m=4;n=1 d=abs(a*m+b*n+c)/((a*a+b*b)**(1/2)) print(d)	
29	a=1;b=5;c=-14 d=(b*b-4*a*c)**(1/2) x1=(-b+d)/(2*a) x2=(-b-d)/(2*a) print(x1) print(x2)	
30	a1,b1,c1=3,1,5 a2,b2,c2=1,-2,-3 d=a1*b2-a2*b1 x=(c1*b2-c2*b1)/d y=(a1*c2-a2*c1)/d print(x,y)	

本章習題

1. 本章已經介紹任意 3 點所圍的面積公式如下：

$$=\frac{1}{2}\begin{vmatrix} x_1 & x_2 & x_3 & x_1 \\ y_1 & y_2 & y_3 & y_1 \end{vmatrix}$$ 其中↘乘為正，↙乘為負

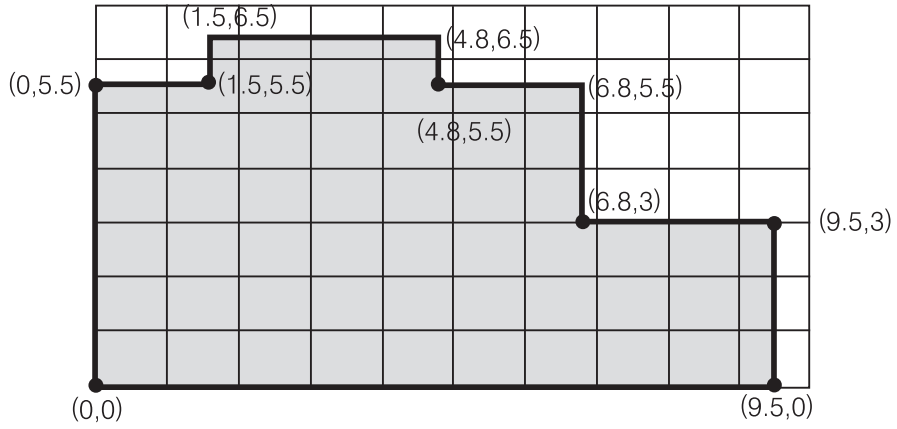
$$=\frac{1}{2}[(x_1y_2+x_2y_3+x_3y_1)-(x_2y_1+x_3y_2+x_1y_3)]$$

任意凸多邊形面積可推廣如下：

$$\text{多邊形面積}=\frac{1}{2}\begin{vmatrix} x_1 & x_2 & \cdots & x_n & x_1 \\ y_1 & y_2 & \cdots & y_n & y_1 \end{vmatrix}$$ 其中↘乘為正，↙乘為負

請依此公式求 (0,0),(5,0),(3,4),(2,4) 的面積。

2. 實例探討。請於 Goole 地圖下載自己學校的地圖，將學校的角落都座標化，然後計算學校面積。例如，以下是某學校的地圖，於左下角指派為座標 (0,0)，將學校等距離畫水平、垂直方格直線，估計每個點的座標，如下圖：



蒐集以上座標 (0,0),(9.5),(9.5,3),(6.8,3),(6.8,5.5),(4.8,5.6),(4.8,6.5),(1.5,6.5),(1.5,5.5),(0,5.5)，代入以上公式，再乘以比例尺比例的平方，即為所求。

3. 日幣的買入。日幣的匯率每天都在變動，請寫一個程式，可以輸入一個匯率、輸入一個台幣金額，得到一筆日圓金額。

4. 同上第 3 題，但是日鈔最小金額是千元，請將日圓千元以下捨棄，再換算回來台幣金額。例如，匯率當日若是每 1 元新台幣可換 4.443 元日圓，表示 10000 元新台幣可換 44430，但日鈔最小的面額是 1000 元，表示僅可換 44000 元日幣，然後請再換算回來台幣多少錢。

5. 同上第 4 題。銀行日鈔僅提供萬元、5 千元與千元紙鈔，請寫一個程式，可以將以上日幣換算各種面額鈔票的張數。
6. 銀行利息計算。假設銀行年利率是百分之 1.2，請寫一個程式，可以輸入定存金額，計算每年利息。
7. 營業稅的計算。營業稅若外加，則營業稅 = 定價 *0.05，若內含，則營業稅 = 定價 /21，且都四捨五入到整數。請寫一程式，可以輸入定價，並分別計算稅外加、與稅內含的營業稅。