指令

7

CHAPTER

莎士比亞用了 26 個英文字母寫出了曠世鉅作，電腦程式語言的關鍵字彙更少，嚴格說來電腦的邏輯架構中存在的運算只有三種：設定、判斷和迴圈。所有電腦語言應該都只有這三種運算指令，能徹底了解和活用這三個指令，就已經學會電腦程式語言的一半，其他還要學習的就是：數（常數和變數）、運算（數值、字串、關係、邏輯）、函數（內建函數和自訂函數）、容器（串列、元組、集合、字典）。函數類似數學中的對應函數，例如有一個函數 y 的值是由參數 x 的數值來決定 y=f(x)，「容器」是有次序性的變數，就是其他語言所稱的陣列（Array）。

7-1 設定 / 運算

7.1.1 設定數值

對初學者來講，學習電腦程式語言的第一個指令就是設定（set），其實設定這一行就是運算式，例如：

```
a=6
b=5
c=a/b
```

上面這 3 行就是設定指令，意思是把等號右邊的值傳到左邊的變數，設定也可以是

```
a=a+1
```

這一行對學過數學的人會覺得不可思議，因為沒有一個數在加 1 之後，還會等於原來的數，所以它的真正意義是把 a 的數值加 1 之後，再存入變數 a 裡面。如果 a 的值原來是 6，加 1 之後就變成 7，所以印出來的變數 a 數值便是 7。

```
print(a)
```

上面這 5 行可以在進入 Python,org 的解譯器之後一行一行輸入就可以得到結果。

在寫電腦運算式的時候，其實和學習數學的規則相同，括號裡面的數值要優先運算，還有先乘除後加減的優先順序。

```
n=(10/2+(3*5+3))-3
```

這一行運算式會印出 n 等於 20，等號的左邊一定是變數，等號的右邊可以是變數或是運算式。

下面還有 4 組簡單的算術運算，各位可在 Python Shell 練習並觀察列印的結果。

```
-------------
a=3
b=2
print(a-b)
print(a*b)
--------------
a=7
b=4
print(a % b )
print(a // b)
---------------
a = 2
b = 3
print(a ** b)
a = 9
b = 0.5
print(a ** b)
---------------
a = 100
b = 9
print(a // b)
a = 24.5
b = 7.2
print(a // b)
---------------
```

這裡僅用算術運算提供各位做簡單的練習，單元練習列出各種不同運算（例如：算術運算、位元運算、比較運算、指派運算、位元指派運算等），請各位分別操作練習。

7.1.2 交換數值

一般電腦在交換數值時，例如 a=6，b=5 兩個數值要交換一般指令是用：

```
t=a ; a=b ;b=t
```

Python 很特殊，可以用這種方法交換：

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

【指令練習】

```
>>>a=5
>>>b=10
>>>a,b=b,a
>>>a,b
(印出)10 5
```

7-2 判斷 / 決策

7.2.1 if 判斷

這個語法就是當條件式成立時（真），則執行陳述句一，要不然就執行陳述句二；如果條件式不成立，並且不想做任何事，則 else 可以省略。

【指令格式】

if(判別式)：

```
    程式區塊
```

elif(判別式)：

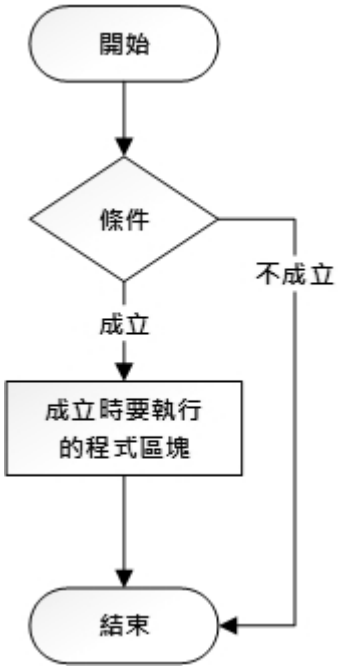
```
    程式區塊
```

else：

```
    程式區塊
```

if 判斷語句是透過一條或多條語句的執行結果（真（True）或者假（False））來決定執行的路徑。可以透過下圖來了解判斷語句的執行過程：

語法	流程圖

if(條件判斷)：
　成立時執行的程式區塊
else:
　不成立時執行的程式區塊
	

範例： n = int(input('輸入一整數:')) if n % 2 ==0 : 　print('偶數') else: 　print('奇數')	

7.2.2 if 範例

程式區塊縮排：上面的語法「if (判斷式): 」下一行必須內縮四個空白，也就是在 print 的前面要有四個空白。

其間也可以加入多層 elif，下面是有二個 elif 的例子：

【程式範例】

```
x=int(input('x=') )    # 使用者輸入
if x<0 :
    print('成績小於零')
elif  x>100 :
```

```
    print('成績大於 100')
elif x<60:
    print('成績不及格')
else:
    print('成績及格')
```

【執行結果】

```
程式執行時輸入 80 結果:
x=80
成績及格
```

7-3 迴圈 / 重複

7.3.1 for 迴圈

【指令格式】

for 變數 in range(起始值, 結束值, 增值)：

程式區塊

格式中 range 是一個數列，可以用循環方式印出級數數列。

【程式範例】

```
r=range(1,11,2)
print(r)
print(type(r))

for i in r:
    print(i,end='')
```

【執行結果】

```
range(1, 11, 2)
<class 'range'>
1 3 5 7 9
```

for 指令的最基本練習：

【程式範例】

```
for i in range(1,10):
```

```
    print(i, end='')
print()

for i in range(1,10): print(i,end=' ')    # 這二行可以寫成一行
```

【執行結果】

```
123456789
```

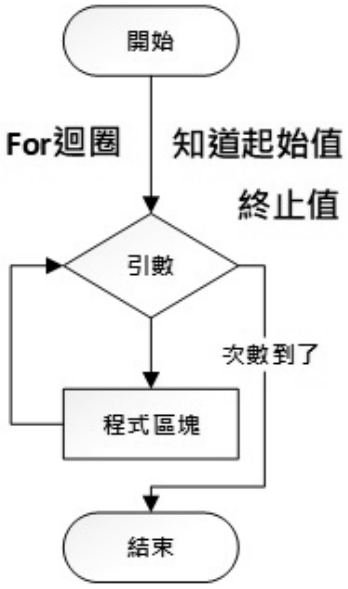
1 2 3 4 5 6 7 8 9 【程式範例】

```
>>> for i in range( 10,1,-3):
        print(i,end=' ')
```

【執行結果】

```
10 7 4
```

for 迴圈格式說明：

for 語法	流程圖
for 引數 in range(起值, 結值, 增值) (程式區塊)	

範例一、使用 range 數列
for i in range(1,11,2):
print(i)
範例二、使用 list 串列
w = ['Sun' , 'Mon' , 'Tue' , 'Wed' , \
'Thu' , 'Fri' , 'Sat']
for days in w:
print (days)
範例三、使用 string 字串
for letter in 'Python':
print (letter)	

Python 的 for 迴圈還有一個異於其他語言的特殊用法，那就是可以使用關鍵字「else」。下例是「找質數」：

【程式範例】

```
# 找 1 ~ 100 所有的質數
for num in range(2,100):
    for i in range(2, num):
        if num % i == 0:
            break
    else:
        print ( num, end=' ')
```

【執行結果】

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

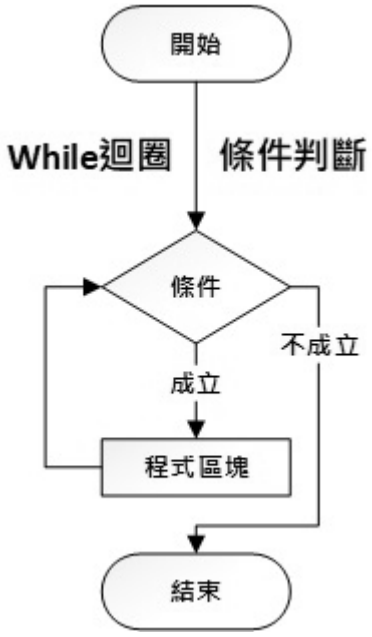
7.3.2 while 迴圈指令

不知道會執行幾次的迴圈程式就適合用 while 指令。

【指令格式】

while(判斷式)：

程式區塊

while 語法	流程圖
引數=起始值 While (引數條件): 　(程式區塊)	
範例： c = 1 while (c < 11): 　print (c) 　c = c + 2	

【程式範例】sum-100.py

```
sum = 0
i = 1
while i <= 100:
     sum += I
     i += 1

print("1 + 2 + ... + 99 + 100 =", sum)
```

【執行結果】

```
1 + 2 + 3 + ... + 98 + 99 + 100 = 5050
```

【程式範例】guess-num-1.py

練習猜數字遊戲

```
c=0
guess=0
n=38
while (guess != n):
     guess=int(input('請輸入 (1~100)？'))
     c=c+1
     print( '你已經猜了:',c,'次')
```

【執行結果】

```
請輸入 (1~100)？12
你已經猜了: 1 次
請輸入 (1~100)？23
你已經猜了: 2 次
請輸入 (1~100)？38
你已經猜了: 3 次
```

在 while 迴圈中可以利用 break 或 continue 來控制迴圈執行的指令流向！

7.3.3 break 敘述使用時機

在迴圈（不論是 for 迴圈或 while 迴圈）執行時，通常要把「要重複的敘述群」執行完一遍之後，再去檢視迴圈「條件式」是否成立。如果需要臨時強迫離開迴圈，也就是中止還沒執行完的敘述，可以使用 break 敘述直接離開迴圈（不管條件式成立或不成立）。

```
while (判別式) :
    if  (判別式) :
        Break
    其他程式區塊
其他程式區塊
```

【程式範例】Prime-break.py

```
for n in range(2, 10):
    for x in range(2, n):
        if n % x == 0:
            print(n, '=', x, '*', n//x)
            break
    else:   # 如果 for 迴圈都沒有執行，然後就跳 else 執行
        print(n, '是質數')
```

【執行結果】

```
2 是質數
3 是質數
4 = 2 * 2
5 是質數
6 = 2 * 3
7 是質數
8 = 2 * 4
9 = 3 * 3
```

7.3.4 continue 敘述使用時機

在某些時候如果需要暫停「本次」迴圈，也就是中止還沒執行完的敘述，要重新檢視迴圈的「條件式」並重新執行迴圈時，就可以使用 continue 敘述，此敘述要搭配 if 敘述使用。

```
while (判別式) :
    if  (判別式) :
        continue
    其他程式區塊
其他程式區塊
```

【程式範例】Even-Odd-continue.py

```
for num in range(2, 10):
    if num % 2 == 0:
        print(num,"是偶數")
        continue
    print(num , "...是奇數")
```

【執行結果】

```
2 是偶數
3 ...是奇數
4 是偶數
5 ...是奇數
6 是偶數
7 ...是奇數
8 是偶數
9 ...是奇數
```

7-4 習題

一、選擇題

(　) 1. 若要印出九九乘法表，則較適合使用何種結構，能使程式碼精簡且正確？

(A) 單一 for 迴圈　(B) if 條件分支

(C) while 迴圈　(D) 巢狀 for 迴圈

(　) 2. 執行下列程式後，total 變數之輸出為？

```
total=0
for i in range(11):
    if (i%2):
        continue
    total=total + i

print("total=",total)
```

(A) 0　(B) 10　(C) 30　(D) 55

(　) 3. 需精確控制執行次數時，用下列何者迴圈較為適當？

(A) for　(B) while　(C) break　(D) continue

(　　) 4. 執行下列程式，其輸出為？

```
i=0
while (i<5):
    print("%d" %i)
    i += 2
```

(A) 0　　(B) 0 2 4　　(C) 1 3 5　　(D) 0 1 2 3 4 5

(　　) 5. 若不小心寫出了無窮迴圈，則可以按下列何者強迫程式停止執行？

(A) Ctrl+C　　(B) Ctrl+V　　(C) Shift+C　　(D) Shift+C

二、實作題

1. 任意輸入 10 個數（數字需控制在 0 ~ 100 之間），(a) 依照輸入順序依序印出、(b) 依照輸入順序反序印出、(c) 列出比平均數高的所有數值、(d) 請將此數由大到小依序列出（用選擇或氣泡排序法）、(e) 數列中第三大的數值是第幾個輸入的數呢？

2. 凱撒密文：「凱撒密文」產生的方法，是將「明文」內的每一個英文字母，以其在英文字母排列順序中向後移動 n 個位置的字母取代之，若向後移動 n 個位置之後已超出 Z 的位置，則繞到最前面 A 的位置繼續往下對應。例如 n=6 時，字母取代的方式為 A 用 G 取代、B 用 H 取代、 Z 用 F 取代。請寫一程式由鍵盤輸入一字串（均為大寫英文字母）及一整數 n (0 <= n < 10)，然後以 n 位位移的凱撒加密法將明文加密後輸出密文。

APCS 試題分析

14

CHAPTER

題目來源：APCS 大學程式設計先修檢測官網

題目網址：https://apcs.csie.ntnu.edu.tw/index.php/samplequestions/previousexam

下面為 105 年 3 月 APCS 的考題及參考解答（解題程式放在資料夾：APCS-10503-檢測題解）。

14-1 概念題

APCS
大學程式設計先修檢測

105 年 3 月 5 日
程式設計觀念題

1. 右側程式正確的輸出應該如下：

```
    *
   ***
  *****
 *******
*********
```

在不修改右側程式之第 4 行及第 7 行程式碼的前提下，最少需修改幾行程式碼以得到正確輸出？

```
1  int k = 4;
2  int m = 1;
3  for (int i=1; i<=5; i=i+1) {
4     for (int j=1; j<=k; j=j+1) {
5        printf (" ");
6     }
7     for (int j=1; j<=m; j=j+1) {
8        printf ("*");
9     }
10    printf ("\n");
11    k = k - 1;
12    m = m + 1;
13 }
```

(A) 1
(B) 2
(C) 3
(D) 4

1【解題說明】	【Python 解題程式】(c-1.py)

程式外部迴圈，控制要輸出幾列文字（i=1~5 共 5 列），第一個內迴圈控制每列前面要輸出幾個空格。
第二個內迴圈控制每列要輸出 1~m 個星號，原程式最末列 m 隨外圈遞增 1，表示每次只遞增 1 顆星與題意不符，應遞增 2 才對，故只要將最末列 m=m+1 改為 m=m+2 即可。故答案為 (A)。
	`k=4`
	`m=1`
	`for i in range(1,6):`
	`  for j in range(1,k+1):`
	`  print(" ",end="")`
	
	`  for j in range(1,m+1):`
	`  print("*",end="")`
	
	`  print("\n")`
	`  k=k-1`
	`  m=m+2`

2. 給定一陣列 **a[10]={ 1, 3, 9, 2, 5, 8, 4, 9, 6, 7 }**，i.e., a[0]=1,a[1]=3, …, a[8]=6, a[9]=7，以 **f(a, 10)** 呼叫執行右側函式後，回傳值為何？

(A) 1
(B) 2
(C) 7
(D) 9

```
int f (int a[], int n) {
   int index = 0;
   for (int i=1; i<=n-1; i=i+1) {
      if (a[i] >= a[index]) {
         index = i;
      }
   }
   return index;
}
```

2【解題說明】	【Python 解題程式】(c-2.py)

此程式為找出陣列內最大值的索引值，該迴圈將全部執行過一次，因此所找出的最大值應為 a[7] 而不是 a[2]，故回傳值 7。故答案為 (C)。	`def f(a,n):`
	`  index=0`
	`  for i in range(1,n):`
	`  if (a[i] >= a[index]): index=i`
	`  return index`
	
	`#main`
	`a=[1,3,9,2,5,8,4,9,6,7]`
	`print("回傳值為",f(a,10))`

3. 給定一整數陣列 a[0]、a[1]、…、a[99]且 a[k]=3k+1，以 value=100 呼叫以下兩函式，假設函式 **f1** 及 **f2** 之 **while** 迴圈主體分別執行 n1 與 n2 次 (i.e, 計算 **if** 敘述執行次數，不包含 **else if** 敘述)，請問 n1 與 n2 之值為何?　註： (low + high)/2 只取整數部分。

```
int f1(int a[], int value) {
   int r_value = -1;
   int i = 0;
   while (i < 100) {
      if (a[i] == value) {
         r_value = i;
         break;
      }
      i = i + 1;
   }
   return r_value;
}
```

```
int f2(int a[], int value) {
   int r_value = -1;
   int low = 0, high = 99;
   int mid;
   while (low <= high) {
      mid = (low + high)/2;
      if (a[mid] == value) {
         r_value = mid;
         break;
      }
      else if (a[mid] < value) {
         low = mid + 1;
      }
      else {
         high = mid - 1;
      }
   }
   return r_value;
}
```

(A) n1=33, n2=4
(B) n1=33, n2=5
(C) n1=34, n2=4
(D) n1=34, n2=5

3【解題說明】	【Python 解題程式】(c-3.py)

此 100 個元素的陣列 a 其值為 a[k]=3k+1，分布情形是：
a[0]=1a[1]=4a[2]=7a[3]=10 . . . . a[96]=289a[97]=292a[98]=295a[99]=298
f1 為循序搜尋，一直搜尋到 a[33]=100，所以 n1=34
f2 為二分搜尋，
第 1 次，a[mid]=a[49]=148　>100
	`def f1(a, value,c):`
	`    i=0 ;  r_value = -1`
	`    while ( i < 100 ):`
	`        c += 1`
	`        if (a[i] == value):`
	`            r_value = i`
	`            break`
	`        i += 1`
	`    return  c`
	
	`def  f2(a,value,c):`
	`    low=0 ; high=99`
	`    while (low <= high):`
	`        c += 1`
	`        mid = (low + high)//2`

3【解題說明】	【Python 解題程式】(c-3.py)

第 2 次，a[mid]=a[24]= 73 <100
第 3 次，a[mid]=a[36]=109 >100
第 4 次，a[mid]=a[30]= 91 <100
第 5 次，a[mid]=a[33]=100 =100
所以 n1=34，n2=5，故答案為 (D)。
	`if (a[mid] == value):`
	`r_value = mid`
	`break`
	`elif (a[mid] < value):`
	`low = mid + 1`
	`else:`
	`high = mid -1`
	`return c`
	
	`#main`
	`a=[] ; value=100 ;n1=0 ; n2=0`
	`for k in range(100):`
	`a.append(3*k+1)`
	
	`n1=f1(a,value,n1)`
	`n2=f2(a,value,n2)`
	`print("n1=%d, n2=%d" %(n1,n2)))`

4. 經過運算後，右側程式的輸出為何？

(A) 1275
(B) 20
(C) 1000
(D) 810

```
for (i=1; i<=100; i=i+1) {
    b[i] = i;
}
a[0] = 0;
for (i=1; i<=100; i=i+1) {
    a[i] = b[i] + a[i-1];
}
printf ("%d\n", a[50]-a[30]);
```

4【解題說明】	【Python 解題程式】(c-4.py)

第一個迴圈設定 b 陣列內容
b[1]=1、b[2]=2、…b[100]=100
第二個迴圈設定 a 陣列內容
a[i] = b[i] + a[i-1]
a[0]=0
a[1]=b[1]+a[0]=1+0=1
a[2]=b[2]+a[1]=2+1=3
….
a[30]=465
	`a=[];b=[]`
	`b.append(0)`
	`for i in range(1,101):`
	`b.append(i)`
	
	`a.append(0)`
	`for i in range(1,101):`
	`a.append(i)`
	`a[i]=b[i]+a[i-1]`
	
	`print("%d\n" %(a[50]-a[30]))`

4【解題說明】	【Python 解題程式】(c-4.py)
a[50]=1275 … a[50]-a[30] = 810，故答案為 (D)。	

5. 函數 **f** 定義如下，如果呼叫 **f(1000)**，指令 **sum=sum+i** 被執行的次數最接近下列何者？

(A) 1000
(B) 3000
(C) 5000
(D) 10000

```
int f (int n) {
   int sum=0;
   if (n<2) {
      return 0;
   }
   for (int i=1; i<=n; i=i+1) {
      sum = sum + i;
   }
   sum = sum + f(2*n/3);
   return sum;
}
```

5【解題說明】	【Python 解題程式】(c-5.py)

程式用到遞迴，n<2 是遞迴的終止條件，開始 n=1000 判斷 n<2 不成立，因此執行 for 迴圈進行 sum 的累加 1000 次。
接著在 sum = sum + f(2*n/3) 中呼叫 f(666)，由於 f 函數之參數 n 為整數，在此時的 f(666) 函數中會累加 666 次，其餘依此類推。
n 值多少，sum=sum+i 就執行多少次。
可知共累加次數：
1000+666+444+296+197+131+87+58+38+25+16+10+6+4+2=2980
故答案為 (B)。
	`def f(n,c):`
	`  sum=0`
	`  if n<2 : return 0`
	
	`  print("n=",n);print("c=",c)`
	`  for i in range(1,n+1):`
	`  sum =sum + i`
	`  c += 1`
	`  n=int(2*n/3)`
	
	`  sum += f(n,c)`
	`  return sum`
	
	`#main`
	`f(1000,0)`