



用 Python

學程式設計
運算思維

Using Python to
Learn Programming
Thinking

函式

7-1 函式的定義與呼叫

- 函式(Function)可以視為是一種獨立的模組，一段程式敘述的集合，透過函式的呼叫，可以使用該段程式碼的功能。正確的使用函式，可以使程式的可讀性增加，也可以使程式在偵錯及修改上更為容易。
- 函式是結構化語言的一個重要元素，使用函式可以將一個複雜的程式難題，分解為數個較小的問題，分別用函式表現出來，將大程式切割後，可分由多人撰寫，縮短程式開發時間。另外，可以將某項功能的程式碼寫成函式，當其他的程式需要使用該功能時，呼叫函式即可。

- Python語言提供了功能強大的標準函式庫，另外還有許多第三方公司所開發的函式，善用這些函式有利於程式問題的解決。另外，還可以使用**import**敘述來引入特定功能的套件（多個函式的組合），來解決程式設計問題。
- 除了Python內建的函式與標準函式庫之外，程式設計師可以利用**def**關鍵字，自行定義函式來解決問題。解決問題的過程中，通常我們為了程式碼的可讀性以及程式專案的規劃，通常會將程式切割成一個個功能明確的函式。

Python定義函式的宣告語法

def 函式名稱 (參數串列):

 程式區塊

return 值

函式名稱：由程式設計者依Python識別字名稱的命名規範來自訂。

參數串列：參數串列部分可以省略也可以包含多個參數，多個參數之間以逗號來分隔。

程式區塊：此為函式主體，可以包含單行或多行敘述，依程式的設計而定。

return指令：函式可以有傳回值或沒有傳回值，當有傳回值時，需搭配**return**指令來回傳，多個傳回值之間以逗號來分隔。

- 函式建立後並不會被呼叫執行，必須在程式中呼叫函式名稱，才會執行該函式，呼叫函式的語法如下：

[變數] = 函式名稱（引數串列）

變數：用於接收函式運算後的傳回值，如沒有傳回值，不需加入變數資料。

引數串列：當函式宣告時有宣告參數串列，則呼叫函式時需加入相對應的引數串列，多個引數之間以逗號來分隔。

7-2多個參數的函式呼叫

- 若函式宣告了2個參數，則進行函式呼叫時，必須傳遞2個引數，當傳遞的引數不足或超過宣告的參數個數，都會產生錯誤。另外，函式呼叫所傳遞的引數是具有順序性的。
- 定義函式時，我們可以為參數設定初值，當呼叫函式時，如果沒有引數傳入該參數，就會使用其初值，參數設定初值的方法為使用指定運算子「=」來設定即可。

7-3 函式回傳值

- 函式可以有傳入值，也可以有回傳的值，**return**關鍵字可以將變數傳回呼叫它的函式內。部分的函式設計會進行相關運算，當函式運算完畢後，使用**return**關鍵字，回傳函式之計算結果。
- 函式的回傳值可以有多筆資料。

7-4 引數的傳遞

- 每一個函式都是獨立的，一般來說，函式只瞭解自己程式區塊的資料，並不認識函式外的任何變數，因此當我們需要外部的資料或是會在函式內修改到外部的資料時，就必須將資料以引數的方法傳遞進函式。

- 在傳統的程式語言開發上，函式的引數傳遞分為兩種方式：傳值呼叫（**Call-by-Value**）與傳址呼叫（**Call-by-Reference**）。
 - 傳值呼叫：將所傳的變數的資料複製一份，儲存在函式所宣告的自訂變數中。
 - 傳址呼叫：將所傳的變數之記憶體位址傳遞進函式，可以直接修改變數資料。

小提示：變數有效範圍

- 每一個變數都有自己的生命週期（**scope**），當一個變數被宣告時，也決定了這個變數存在的範圍。變數依有效範圍可以分為區域變數與全域變數兩類，說明如下：
 - 區域變數：在函式內定義的變數，其作用有效範圍限於該函式範圍內。
 - 全域變數：在函式外定義的變數，其作用有效範圍為整個Python檔案。

7-5 模組與套件

- 在Python中，模組(Module)是指包含相關定義好的函式，使用import指令可以將模組匯入檔案中，然後程式開發者可以使用模組中的函式，加快程式的開發。
- 套件（Package）是指包含了許多相關的模組，以目錄的架構來組成，變成一個模組庫或函式庫，一樣也是使用import指令將套件匯入到檔案中。

- 7-5-1 匯入單一套件

Python內建許多功能強大的標準函式庫，內建套件需要使用import命令來匯入才能使用，其語法如下：

import 套件名稱

- random套件可隨機產生亂數，如要匯入random套件，其程式碼如下：

```
import random
```

- 匯入套件後，就可以使用套件中的函式，使用套件函式的語法為：

套件名稱.函式名稱

- 此處以random套件的randint()函式為例，該函式會亂數產生兩數之間的任一整數，以下以亂數產生1到100之間的整數為例，其參考程式碼為：

```
import random  
random.randint (1, 100)
```

- 有時我們想要省略套件名稱的輸入，加快程式的輸入效率，我們會使用下列語法，一次將套件內的函式全部匯入。

`from 套件名稱 import *`

- 以下同樣以亂數產生1到100之間的整數為例，其程式碼如下：

```
from random import * #匯入random套件的所有函式  
randint(1,100) #省略套件名稱random
```

- 雖然一次匯入該套件的所有函式寫法相當方便，但在多個套件中具有相同名稱函式時，容易造成函式引用的錯誤。因此，為了避免錯誤的發生，有以下兩種不同寫法。

方法一：直接指定要引用的套件函式名稱，其語法如下：

`from 套件名稱 import 函式1, 函式2, 函式3...`

方法二：將套件名稱另外命名一個較簡短的別名，其語法如下：

`import 套件名稱 as 別名`

- `randrange()` 函式是另一個產生隨機整數的方法，其格式為（最小值,最大值,間隔值），該函式可以設定亂數產生的間隔，例如：

`randrange(5,100,5)` #產生數字是5的倍數，最小值5，最大值小於100

如果要產生最小值為0.0，最大值是1.0的隨機浮點數，會使用`random()` 函式，該函式沒有參數，其程式碼如下：

```
import random as rd
```

```
rd.random() #產生最小值為0.0，最大值是1.0的隨機浮點數
```

- 7-5-2 匯入多個套件

如果想要一次匯入多個套件，則我們可以使用逗號來分隔各個套件名稱，其import語法如下：

import 套件名稱1, 套件名稱2, 套件名稱3...

- 7-5-3 第三方套件的安裝

Python最讓程式開發者欣賞的優點，是擁有大量第三方公司所開發的套件可以匯入使用，要匯入之前需先確認該套件已安裝。

- 7-5-4 常用內建函式

Python有許多常用的內建函式，程式開發者可以呼叫內建函式來撰寫程式，先前有用過的`int()`函式或是`range()`函式，都是Python的內建函式，請參考書本所介紹的一些常用內建函式。

小提示：*sorted()* 函式的 *reverse* 參數

- *sorted()* 函式加上「*reverse=True*」參數，則可以變成由大到小的排序規則，請參考下列程式碼範例：

```
sorted([1,3,5,2,4], reverse=True)
```

其執行結果如下：

```
[5, 4, 3, 2, 1]
```

7-6遞迴函式

- 遞迴函式(recursive function)的定義是：一個函式直接（在程式敘述內直接呼叫函式本身）或間接（程式敘述內呼叫其他函式，在該函式內又呼叫了原先的函式）的呼叫函式本身，稱為遞迴函式。

7-7 程式練習

7-8 習題