

第七章、函式

在撰寫程式時，常會遇到一些具有特定功能而又重複出現的程式片段，將這樣的程式片段獨立出來，出要使用時給予參數就能呼叫執行使用，像這樣獨立出來的程式片段，可以成立一個單獨的程式區塊，我們將這樣的陳述結構稱為「函式(function)」或稱為「方法(method)」。

函式分為三類：內建函式、套件函式、自定函式。內建函式是編譯系統設計好的，可直接呼叫使用，例如：`print()`、`input()`、`int()`。套件函式是需要透過 `import` 語法引入函式庫，才可以呼叫使用，例如：`time` 套件、`datetime` 套件、`math` 套件。自定函式是程式設計者在撰寫程式時，應程式自己需求而定義的。

內建函式、套件函式我們只要會使用，知道給予什麼參數就可傳回所要的結果(要注意參數的數量與型別)，不必了解函式內部設計的行情形，也無法做更改。而自定函式是程式設計者竹中生有產生的，隨時可以調整更改。設計好的自定函式有下列好處：

1. 函式可以重複使用，大程式只需要注重在系統架構的規劃，功能性或主題性的工作交給函式處理，程式碼可以較精簡。
2. 若較大程式軟體，可依功能切割成多個單元，再交由多人共同設計。如此不但可以縮短程式開發時間，也可以達到程式模組化的目的。
3. 將相同功能的程式敘述片段寫成函式，只需要做一次，有助於提高程式的可讀性。也讓程式的除錯及維護更加容易。

內建函式

數值函式	使用說明
<code>abs(int)</code>	取得 <code>n</code> 的絕對值。
<code>round(float)</code>	取得 <code>n</code> 的四捨五入後的整數值。當小數第一位數字是 5 時，若前一位數字是嘔數則捨去；若前一位數字是奇數則進位。
<code>int(str float)</code>	將 <code>n</code> 轉換成整數(小數部分直接捨去)。
<code>float(int)</code>	將 <code>n</code> 轉換成浮點數。
<code>pow(n, m)</code>	取得 <code>n</code> 的 <code>m</code> 次方。

套件函式

除了內建的函式以外，我們也可以引入套件，來執行套件中的函式。

引入的語法如下：

```
import 套件名稱
from 套件名稱 import 函式
from 套件名稱 import 函式 as 重新命名
from 套件名稱 import * (全函式引入)
```

math 套件函式	使用說明
pi	圓周率常數 π 。
ceil(n)	取得大於 n 的最小整數。
floor(n)	取得小於 n 的最大整數。
fabs(n)	取得浮點數 n 的絕對值。
sqrt(n)	取得 n 的平方根。

練功坊

給予半徑 100，計算出圓面積及圓周長。

```
import math as M
r = 100
a = r*r*M.pi
l = 2*r*M.pi
print(f“半徑為 {r} 的圓， 面積為 {a:.2f}， 圓周長為 {l:.2f}”)
```

random 套件函式	使用說明
randint(n1, n2)	從 n1 到 n2 之間隨機產生一個整數(包含 n1, n2)
randrange(n1, n2, n3)	從 n1 到(n2-1)之間每隔 n3 的數，隨機產生一個整數。
random()	從 0 到 1 之間隨機產生一個浮點數。
choice(s)	從 s 字串或串列中隨機取得一個元素。
sample(s, n)	從 s 字串或串列中隨機取得 n 個不重複的元素。
shuffle(list)	使串列中的元素重新隨機排列。

datetime 套件函式應用

```
import datetime as DT
nowTime = DT.datetime.now()
print('{:%Y/%b/%d %A}'.format(nowTime))
print(f'{nowTime:%Y/%b/%d %A}')
print(nowTime.strftime("%Y/%b/%d %A"))
```

以上輸出都是 2021/Oct/01 Friday

datetime 套件的格式化%字元對照如下：

%字元參數	轉換成字串的使用說明
%Y	西元年份，如：'2021'
%y	西元年份不含世紀，如：'21' 代表西元 2021 年
%B	月份使用完整英文，如：'October'
%b	月份使用縮寫英文，如：'Oct' 代表十月
%m	月份使用數字('00' ~ '12')，如：'10' 代表十月
%d	所在月份的第幾天('00' ~ '31')
%A	星期幾使用完整英文，如：Friday
%a	星期幾使用完整英文，如：'Fri' 代表星期五
%w	星期幾使用數字，如：'5' 代表星期五，'0' 代表星期日
%H	時 使用 24 小時制('00' ~ '23')
%I	時 使用 12 小時制('00' ~ '12')
%M	分 ('00' ~ '59')
%S	秒 ('00' ~ '59')
%p	呈現 'AM' 或 'PM')

自定函式

自定函式是程式設計者依需求自行定義開發設計。自行定義的函式是一段獨立的程式碼敘述集合區塊，這個區塊稱為 function(函式)，建立時要賦予函式名稱(跟變數名稱的命名規則一樣)。在其它程式敘述只要透過呼叫該函式名稱，就可以使用該函式敘述的功能。

函式的建立語法如下：

```
def 函式名稱(引數 1, 引數 2, .....):
    程式區塊
    return 傳回值 1, 傳回值 2, .....
```

1. 函式名稱：函式名稱的命名方式比照識別字的命名規則。在同一個程式中，自定函式的名稱不可以重複定義使用，也不可以和內建函式的名稱相同。
2. 引數：引數又稱為參數，可以傳入一個、二個或多個，也可以不傳入引數。在定義端的引數稱為虛引數，用途是接收來自呼叫端的程式敘述傳遞過來的資料。呼叫端程式敘述傳入的資料稱為實引數，是真正參與函式執行的資料。若傳入之引數是二個以上，引數間要用「，」隔開。若不傳入引數，這時引數列只留有一個空的()。
3. 傳回值：執行函式的敘述區段從事某項功能或任務時，有時會傳回一個或多個結果，這就是傳回值。而傳回值可以是資料、可以是變數、也可以是運算式，但回傳值最終會式資料常值得結果。使用 return 敘述就可以傳回資料結果，並跳出該函式的敘述區塊。若函式的任務沒有傳回值時，則 return 敘述可省略。

練功坊

定義一個函式主體，函式名稱為 average，傳入兩個引數 n1, n2，傳回值是兩個引數的平均值，傳回值使用 a 浮點數。

```
def average(n1, n2):  
    a = (n1 + n2) / 2  
    return a
```

自定函式完成定義後，並不會執行敘述區塊，必須另外呼叫函式才可以執行(注意引數的數量與型別)。類似我們知道如何打掃環境，但只有在需要打掃環境時，才會執行打掃工作。呼叫函式的語法如下：

變數名稱 = 函式名稱(引數 1, 引數 2,)
函式名稱(引數 1, 引數 2,)

有時自訂函式在定義時會給引數預設值，使得在呼叫函數時，可以不用傳入引數。當定義函式時，使用了預設引數，其右側其它的引數也必須被當成預設引數定義。請參考以下範例：

```
def triangle (B = 6, H = 6):  
    print()  
    print(f'三角形的底為{B}, 高為{H}')  
    A = B * H / 2  
    retrun A  
  
base = 10  
high = 5  
areal = triangle(base, high)  
print(f'三角形的面積為{areal}')  
base = 10  
area2 = triangle(base)  
print(f'三角形的面積為{area2}')  
area3 = triangle()  
print(f'三角形的面積為{area3}')
```