

在設計程式時，可將經常重複使用到的程式定義為函數，藉此重複利用提升開發效率。而函數可區分為內建函數與自訂函數。

內建函數，是 Python 事先寫好程式，並包裝為函數，提供給開發者使用，如同前面的章節，我們所學到的 `input()`、`print()`、`int()`...等皆是。

自訂函數，為開發者可依照需求，將自己寫好的程式包裝為函數，以便日後重複利用、除錯與維護。

**【特別注意】** 函數只有當被呼叫時，才會執行！

## 一、自訂函數

1. **定義**函數，其語法為：

```
def 函數名稱(形式參數 1, 形式參數 2, 形式參數 3, ...):
    程式區塊
    return 回傳值    # 不回傳值時，可省略
```

2. **呼叫**函數，其語法為：

```
函數名稱(實際參數 1, 實際參數 2, 實際參數 3, ...)
```

**例如 1：(不回傳值)**

```
1 def slhs(a,b):
2     avg = ( a + b ) / 2
3     print(avg)    # 不回傳值時，必須在函數內 print()，否則不會顯示結果，avg = 65.0
4 slhs(40,90)      # 程式首先執行第 4 行，呼叫函數後，再執行第 1 行 ~ 第 3 行
```

**例如 2：(回傳值)**

```
1 def slhs(a,b):
2     avg = ( a + b ) / 2
3     return avg    # 代表回傳 avg 變數
4 ans = slhs(40,90) # 程式首先執行第 4 行，呼叫函數後，再執行第 1 行 ~ 第 3 行
5 print(ans)       # ans = 65.0
```

## 二、不定個數的參數

1. **定義**函數，資料型態為**序對(Tuple)**，其語法為：

```
def 函數名稱(*形式參數):
    程式區塊
    return 回傳值    # 不回傳值時，可省略
```

2. **定義**函數，資料型態為**字典(Dict)**，其語法為：

```
def 函數名稱(**形式參數):
    程式區塊
    return 回傳值    # 不回傳值時，可省略
```

3. **呼叫**函數，其語法為：

```
函數名稱(實際參數 1, 實際參數 2, 實際參數 3, ...)
```

### 例如 1：(傳入不定個數的參數：序對)

```
1 def slhs(*score):    # score = (40, 90, 30)
2     sum = 0
3     for i in score:
4         sum += i
5     return sum
6 ans = slhs(40,90,30)
7 print(ans)          # ans = 160
```

### 例如 2：(傳入不定個數的參數：字典)

```
1 def slhs(**score):    # score = {"國文":40, "英文": 90, "數學":30}
2     sum = 0
3     for i in score:
4         sum += score[i]
5     return sum
6 ans = slhs(國文=40, 英文=90, 數學=30)
7 print(ans)          # ans = 160
```

### 三、遞迴函數

若函數中，含有呼叫自己的程式，則稱為遞迴函數。遞迴函數的優點便是較容易理解，缺點則是執行效率比較差。

例如 1: 數學上的階層函數，如:  $0! = 1$  ;  $4! = 4 * 3 * 2 * 1$ 。

#### 【思考方向】

該程式為 2 個條件：

- (1) 條件 1: 若  $n == 0$ ，則答案為 1。
- (2) 條件 2: 若  $n > 0$ ，則答案為  $n * (n-1) * (n-2) * (n-3) * \dots$ 。

```

1  def fac(n):
2      if n == 0 :
3          return 1
4      else:
5          return n * fac(n-1)
6  ans = fac(4)
7  print(ans)          # ans = 24

```

#### 【程式解析】

1. 程式首先執行第 6 行。
2.  $fac(4)$  執行後，會發現  $return 4 * fac(3)$ ，該行程式帶有呼叫函數  $fac(3)$ 。
3.  $fac(3)$  執行後，會發現  $return 3 * fac(2)$ ，該行程式帶有呼叫函數  $fac(2)$ 。
4.  $fac(2)$  執行後，會發現  $return 2 * fac(1)$ ，該行程式帶有呼叫函數  $fac(1)$ 。
5.  $fac(1)$  執行後，會發現  $return 1 * fac(0)$ ，該行程式帶有呼叫函數  $fac(0)$ 。
6.  $fac(0)$  執行後，回傳 1。
7. 最後回推，計算出所有答案：

$$fac(4) = 4 * \underline{fac(3)} = 24$$

$$\underline{fac(3)} = 3 * \underline{fac(2)}$$

$$\underline{fac(2)} = 2 * \underline{fac(1)}$$

$$\underline{fac(1)} = 1 * \underline{fac(0)}$$

$$\underline{fac(0)} = 1$$

## 例如 2: 費氏級數

## 【思考方向】

該程式為 3 個條件：

- (1) 條件 1:  $f_0$  答案為 0，
- (2) 條件 2:  $f_1$  答案為 1，
- (3) 條件 3:  $f_n$  為前兩個數字的加總，答案為  $f_{(n-1)} + f_{(n-2)}$

```

1 def fib(n):
2     if n == 0 :
3         return 0
4     elif n == 1 :
5         return 1
6     else:
7         return fib(n-1) + fib(n-2)
8 ans = fib(5)
9 print(ans)           # ans = 5

```

