第5章 自訂函式 (Functions)

函式 (Function) 的應用非常廣泛，前面章節中我們已經學習過多個函式，比如：input() 、range()、len() 等函式，這些都是 Python 的內建函式，可以直接使用。

除了可以直接使用的內建函式外，Python 還讓我們自訂函式，也就是說讓我們自己把一段可重複使用的程式定義成函式，以便一次編寫，而後可以多次調用，也可以供他人使用。自利又利他，很棒吧！

下面就來學習如何自訂函式。

「定義函式」指令 (def statement)

語法

```
def <函式名稱> (<參數列>):

        <程式區塊>

        return <回值>
```

其中:

1. def 意思是 define（定義）宣告一個函式的開始
2. 函式名稱，命名規則與變數命名規則相同
3. 參數列中每一個參數都是變數，兩個參數間須以逗號 ‘，’ 分開
4. 參數種類又分：位置參數和關鍵字參數兩種
5. 位置參數只有變數名稱，依參數列的位置順序排列
6. 關鍵字 (keyword) 參數就是 '<變數>=<預設值>' 的型式定義
7. 程式區塊必須至少一個縮格開始定義
8. 函式結束前，Python 讓我們用 return 返回指令，返回到呼叫程式
9. 返回時，也可以視需求帶回一個回值物件（returned object)

自訂函式例子如下：

In [1]:
```
def bigger(a, b):
    # 例子1：兩數比較大小，返回較大的數字
    if a >= b:
        return a
    else:
        return b

def smaller(a, b):
    # 例子2：兩數比較大小，返回較小的數字
    if a >= b:
        return b
    else:
        return a

print(f'bigger (3, 5): {bigger(3, 5)}')
print(f'smaller (7, 9): {smaller(7, 9)}')

```

```
bigger (3, 5): 5
smaller (7, 9): 7
```

注意

1. 若參數列含有位置參數和關鍵字參數兩種時，所有位置參數必須放在關鍵字參數之前
2. 注意保留字 def 那一行最後的字元必須是「：」

In [3]:
```
def hello_func(greeting, name = 'You'):
    # 位置參數必須放在關鍵字參數之前
    return f'{greeting}, {name}'

# location of function
print(f'hello_func() 物件類型是: {type(hello_func)}')
print(hello_func) # 沒有小括號，Python 只印函式定義，不會執行函式

# 呼叫並執行函式，省掉一個有預設值的參數
print(hello_func('===>哈囉'))

# 呼叫並執行函式，二個參數
print(hello_func('===>哈囉', name='尤勇'))

```

```
hello_func() 物件類型是: <class 'function'>
<function hello_func at 0x7fbd8eeb5430>
===>哈囉, You
===>哈囉, 尤勇
```

In [54]:
```
# 例子：自訂函式列印學員資料

def student_info(*args, **kwargs):
    print('Student info listed below:')
    print('\t', args)
    print('\t', kwargs)

# Notice that "=" b/t keyword and value when calling the function
#    but the function returns key-value paors with ":" as a dictio
student_info('Math', 'Art', name='John', age=22)

courses = ['Math', 'Art']
info = {'name': 'John', 'age': 22}

student_info(*courses, **info)

```

```
Student info listed below:
	 ('Math', 'Art')
	 {'name': 'John', 'age': 22}
Student info listed below:
	 ('Math', 'Art')
	 {'name': 'John', 'age': 22}
```

In [55]:
```

```

```python
# 例子：自訂函式判斷閏年月與否

# 每個月天數
month_days = [0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30 ,31]
#                 1   2   3   4   5   6   7   8   9  10  11  12

def is_leap(year):
    """ 閏年回覆為真，否則為偽
    Return True for leap years,
    False for non-leap year."""

    return year % 4 == 0 and (year % 100 != 0 or year % 400 == 0)

def days_in_month(year, month):
    """由年月份判斷該月份有幾天
    Return number of days, given a year and a month."""

    # checking valid month
    if not 1 <= month <= 12:
        return '非正常月份'

    # checking if it is a leap year
    if month == 2 and is_leap(year):
        return 29

    return month_days[month]

year = 2017
month = 8
print(f'問：Year {year} 閏年嗎?', is_leap(year))
print(f'答：{year} 年 {month} 月有: {days_in_month(year, month)} day

year = 2000
month = 2
print(f'問：Year {year} 閏年嗎?', is_leap(year))
print(f'答：{year} 年 {month} 月有: {days_in_month(year, month)} day

year = 2022
month = 2
print(f'問：Year {year} 閏年嗎?', is_leap(year))
print(f'答：{year} 年 {month} 月有: {days_in_month(year, month)} day

```

```
問：Year 2017 閏年嗎? False
答：2017 年 8 月有: 31 days

問：Year 2000 閏年嗎? True
答：2000 年 2 月有: 29 days

問：Year 2022 閏年嗎? False
```

答：2022 年 2 月有: 28 days