



# 函數宣告

朱克剛

# 函數宣告

- 語法為 `def function_name():`
- 使用 `return` 傳回值，沒有傳回值就不需要 `return`，不論是否有傳回值，函數宣告方式都一樣

```
def plus(x, y):  
    return x + y  
  
print (plus(5, 3))  
print (plus(y = 7, x = 6))
```

# 參數預設值

```
def hello(name = 'everyone'):
    print('hello ' + name)
```

```
hello()          # hello everyone
hello('John')    # hello John
```

# 不固定參數

## ■ 函數參數不固定的宣告方式

```
def sum(*values):  
    total = 0  
    for i in values:  
        total += i  
    return total  
  
print(sum(1, 2, 3, 4))
```

# 函數說明

- 好的函數設計應該要在函數中加上說明文件
- 呼叫函數的\_\_doc\_\_即可印出該文件

```
def f():  
    """This is a simple function."""  
  
print (f.__doc__)
```

# 參數傳遞方式

- 函數參數傳遞一律傳值呼叫
- 若函數中需要修改原值，必須以陣列或是類別封裝資料後傳遞
- 答案是？

```
def f1(n):  
    n = n + 1  
  
n = 10  
f1(n)  
print(n)
```

# 生命週期與整體變數

- 函數中第一次使用的變數均為區域變數，函數結束後變數即消失
- 若函數中要使用整體變數，必須先宣告該變數為 `global`

```
a = 10

def f1():
    global a
    a = 20
```