

匿名函數與變數可見度

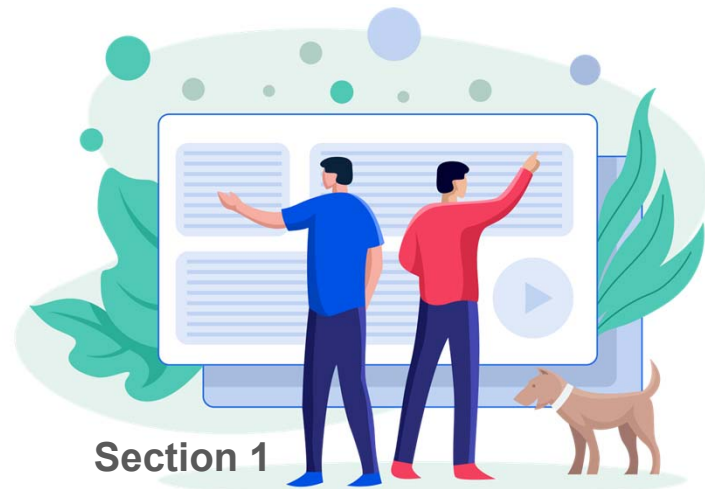
學習目標

- 使用匿名函數(Function)
- 全域變數(Global variables) 用法
- 區域變數(Local variables) 用法



Lambda運算

■介紹 Lambda 運算的語法與使用



Lambda運算 -1

- 又稱為 Lambda 函數或是匿名(Anonymous) 函數
- 使用 “lambda” 關鍵字來建立匿名的函數
- Lambda 運算可以接收多個引數，但只能執行一個運算，傳回一份資料
- 語意(Semantically) 上來說，Lambda 運算只是正規函數定義的語法糖(syntactic sugar)
- 語法糖(Syntactic sugar)，也譯為糖衣語法，是由英國電腦科學家彼得·蘭丁發明的一個術語，指電腦語言中添加的某種語法，這種語法對語言的功能沒有影響，但是更方便程式設計師使用。語法糖讓程式更加簡潔，有更高的可讀性
- Lambda函數不需要像正規函數定義那麼複雜，所以算是正規函數定義的語法糖
- 語法
 - lambda [arg1 [,arg2,.....argn]]: expression

Lambda運算 -2

- 範例 (使用變數參考 lambda 函數)

- # Lambda function

```
sum_data = lambda arg1, arg2: arg1 + arg2
```

```
# Use lambda function
```

```
print('Total :', sum_data(10, 20)) # 30
```

```
print('Total :', sum_data(20, 20)) # 40
```

Lambda運算 -3

■範例 (直接將 Lambda 運算當做引數)

■# function: return sum of two values

```
def sum_data(a, b):  
    return a+b
```

function: using function as parameter

```
def operate_on(x, y, func):  
    return func(x, y)
```

```
r = operate_on(16, 20, sum_data) # call function  
print('operate_on(16, 20, sum_data) = ', r)
```

```
r = operate_on(10, 40, lambda a,b:a+b) # with lambda function  
print('operate_on(10, 40, anonymous_function) = ', r)
```

Lambda運算(Demo)

■ 如何使用 Lambda 運算



全域與區域變數

■介紹全域(Global) 變數與區域(Local) 變數



命名空間(Namespace)

- 變數(Variables) 是對應到物件的名稱(Identifiers)
- 命名空間(Namespace) 是存放變數名稱與其對應物件的資料結構
- Python所管理的記憶體中，有一部分用來存放變數名稱與其對應物件的資料結構，這個記憶體的區域稱之為命名空間(Namespace)
- Python 的敘述可以存取區域命名空間與全域命名空間的變數
- Python可以存取區域變數或全域變數，實際在Python操作時，會先搜尋區域變數，如果找不到該變數名稱，再尋找全域變數。如果區域變數與全域變數同名，將會以區域變數為優先
- 假如區域與全域變數同名，區域變數會蓋掉全域變數
- 每一個函數有自己的區域命名空間，類別的方法(Methods) 與一般函數相同
- Python作者Guido van Rossum說：Method跟Function是一樣的

變數的可見度(Scope)

- Python 程式內所有的變數不一定都可以被存取
 - 這與變數在何處宣告有關係
- 變數的可見度決定程式可以在何處存取該變數
- Python語言有兩種變數的可見度
 - 全域變數(Global variables)
 - 區域變數(Local variables)

全域與區域變數 -1

- 在函數內的變數稱為區域變數
- 在函數之外的變數稱為全域變數
- 區域變數只能在該函數內存取
- 全域變數可以在整個程式內使用，包含所有的函數
- 函數可以直接參考全域變數
- Python的世界中，並沒有宣告變數這件事，並沒有「`int x = 5`」這種寫法
- 如果函數內有指定一個值給變數，該變數則自動視為區域變數，除非明確使用 “`global`” 關鍵字宣告該變數為全域變數

全域與區域變數 -2

■範例

```
■total = 0 # This is global variable
# Function definition
def sum_data(arg1, arg2):
    total = arg1 + arg2; # total is local variable
    print('In function local total:', total) # print 30

# Call function
sum_data(10, 20)
print('Out function global total:', total) # print 0
```

全域與區域變數 -3

■ 範例

```
■ def cake():  
    global fruits  
    fruits = 'blueberry' # this is the global
```

```
def icecream():  
    print(fruits) # this is the global
```

```
fruits = 20 # this is the global  
icecream()  
cake()  
print(fruits) # blueberry
```

全域與區域變數(Demo)

■ 如何使用全域與區域變數



本章重點精華回顧

- 如何使用 Lambda 運算
- 如何使用全域(Global) 與區域(Local) 變數



Lab: 匿名函數與變數可見度

- Lab01: 使用 Lambda 運算(Expressions)
- Lab02: 使用全域(Global) 與區域(Local) 變數

Lab01: 使用 Lambda 運算(Expressions)

1. 啟動Python IDLE環境，做以下練習。
2. 使用 “File/Open...” 開啟 “anonymous.py” 程式，了解Lambda 運算的用法。
3. 關閉anonymous.py程式視窗

Lab02: 使用全域(Global) 與區域(Local) 變數

1. 啟動Python IDLE環境，做以下練習
2. 使用 “File/Open...” 開啟 “local_var.py” 程式，了解全域與區域變數的用法
3. 關閉local_var.py程式視窗
4. 使用 “File/Open...” 開啟 “global_var.py” 程式，了解全域與區域變數的用法
5. 關閉global_var.py程式視窗