

函數(Function)引數

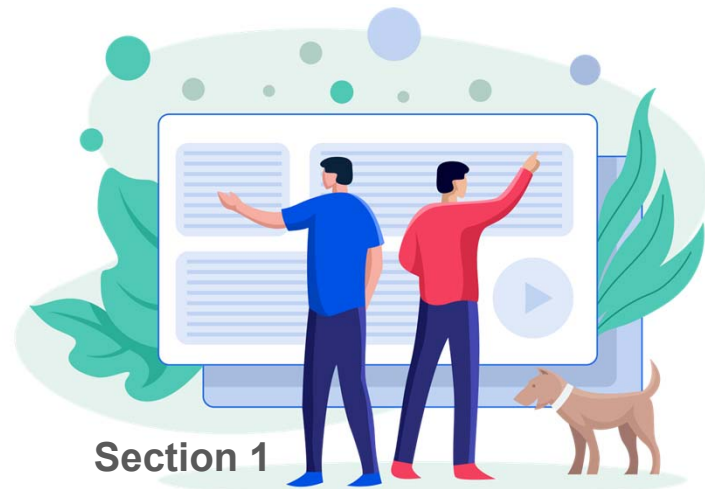
學習目標

- 關鍵字引數(Keyword arguments)
- 預設引數(Default arguments)
- 不固定個數引數(Arbitrary arguments)



函數的引數

- 熟悉 Python 函數傳引數(Arguments) 的方式



函數傳引數

■ Python 函數有以下四種傳引數(Arguments) 的方法

- 位置引數(Positional arguments)
- 關鍵字引數(Keyword arguments)
- 預設引數(Default arguments)
- 不固定個數引數(Arbitrary arguments)

位置引數

- 必要的引數必須按照正確的順序傳給函數
- 呼叫函數傳送的引數順序與個數必須與定義函數時的形式參數(Formal parameters) 一致
- 位置引數又稱為必要引數(Required arguments)
- 範例
 - # Function definition
def printme(pstr):
 print(pstr)

Call function
printme('test')

關鍵字引數 -1

- 關鍵字引數是相對於函數的呼叫
- 當呼叫函數時，使用引數名稱來傳遞引數值
- 如此可以不需依照順序傳引數，因為 Python 直譯器可以依照引數名稱正確的傳引數給函數
- 函數定義的方式與位置引數的定義方式一樣

關鍵字引數 -2

■ 範例

■ # Function definition

```
def printinfo(dbname, port):  
    print('DB Name:', dbname)  
    print('Port:', port)
```

Call function

```
printinfo('MSSQL', 1433)  
printinfo(port=3306, dbname='MySQL')
```

預設引數 -1

- 預設引數會先給一個預設值，如果該引數沒有提供資料，則直接用預設值處理
- 如此可以在呼叫函數時，傳入比函數定義時還少的引數，藉此簡化函數的使用方式
- 預設引數必須定義在位置引數之後
- # Function definition
def printinfo(tblname, dbname='MySQL', port=3306):
 print('Table Name:', tblname)
 print('DB Name:', dbname)
 print('Port:', port)

預設引數 -2

■ 正確的函數呼叫

- ▣ `printinfo('person')` # 1 positional argument
- ▣ `printinfo(tblname='person')` # 1 keyword argument
- ▣ `printinfo(port=3330, tblname='products')` # 2 keyword arguments
- ▣ `printinfo('person', 'MSSQL', 1433)` # 3 positional arguments
- ▣ `printinfo('products', port=3330)` # 1 positional, 1 keyword

預設引數 -3

■ 錯誤的函數呼叫

- # required argument missing
`printinfo()`
- # positional after keyword argument
`printinfo(port=8888, 'person')`
- # duplicate arguments
`printinfo('person', tblname='products')`

不固定個數引數 -1

- 函數可以定義成接收不同個數的引數
- 這些引數稱為不固定個數(Arbitrary) 的引數，用一個統一的參數(Parameter) 名稱來表達
- 不固定個數的引數必須放在函數參數定義的最後
- 在參數名稱前面加上一個星號(*) 代表不固定個數的引數
 - 以 tuple 資料型態存放，如果呼叫函數時沒有提供，則視為空的 tuple

不固定個數引數 -2

```
■ # Function definition
def printinfo(arg1, *vartuple):
    print('Output is: ')
    print(arg1)

    for var in vartuple:
        print(var)
    print()

# Call function
printinfo(10)
printinfo(70, 60, 50)
```

*args 和 **kwargs

- `str.format(*args, **kwargs)`

- `*args` 代表可以收多個引數，所收集的資料是位置引數

- `**kwargs` 代表可以收多個關鍵字引數，所收集的資料是關鍵字引數，使用dict來取得關鍵字名稱 (關鍵字名稱即為dict的key)

- `args` 是 `argument`的簡寫，是引數的意思，透過 `*` 收集的引數會被放到一個 `tuple` 中，所以我們可以使用 `for` 來對它進行拆解。

```
print('{2}, {1}, {0}'.format(*'abc'))  
# 'c, b, a'
```

- `kwargs` 則是 `Keyword Argument` 的簡寫，透過 `**` 可以拆解dict或是將資料收集至dict

```
data = {'name': 'LabVIEW360', 'est': 20}  
print('The name is {name}, aged {est}'.format(**data))  
# The name is LabVIEW360, aged 20
```

函數的引數(Demo)

■介紹函數引數(Arguments) 的用法



本章重點精華回顧

- 關鍵字引數(Keyword arguments)
- 預設引數(Default arguments)
- 不固定個數引數(Arbitrary arguments)



Lab: 函數(Function)引數

■ Lab01: 使用函數的引數(Arguments)

Lab01: 使用函數的引數(Arguments)

1. 啟動Python IDLE環境，做以下練習
2. 使用 “File/Open...” 開啟 “require_arg.py” 程式，了解位置引數的用法
3. 關閉require_arg.py程式視窗
4. 使用 “File/Open...” 開啟 “keyword_arg.py” 程式，了解關鍵字引數的用法
5. 關閉keyword_arg.py程式視窗
6. 使用 “File/Open...” 開啟 “default_arg.py” 程式，了解預設引數的用法
7. 關閉default_arg.py程式視窗
8. 使用 “File/Open...” 開啟 “varlen_arg.py” 程式，了解不固定個數引數的用法
9. 關閉varlen_arg.py程式視窗