

# python基本語法與方法

---

## 綱要/目標

- Github Repo 建立, 並知曉如何使用codespace進行python作業練習, 並能將作業成果上傳到Github

## 預計流程

1. 用python完成 數字與字串兩類變數的宣告、
2. 知道List, 添加、查詢、更新、刪除List內的元素
3. 知道dict, 添加、查詢、更新、刪除dict內的元素
4. 用for loop 操作list
5. 用for loop 搭配 if else, 對list內的元素作判斷
6. 編寫一個沒有帶參數的python def
7. 編寫一個有帶參數的python def
8. 編寫一個沒有帶參數, 但有帶return的python def
9. 編寫一個有帶參數, 有帶return的python def

## LAB

0527python-1:

```
#數字與字串兩類變數的宣告 x、y為數字, a、b為字串
x=11
y=22
a="13"
b="14"
print("x+y=",x+y)
print("a+b=",a+b)
```

執行結果:

```
PS C:\Users\T14 Gen 3\Desktop\tt> python 527homework.py
x+y= 33
a+b= 1314
```

```
#List添加、查詢元素
a_list=[1,2,3,4]
print(a_list[:])
a_list.append(5)
```

```
print(a_list[:])
#List更新、刪除元素
print("first num is",a_list[0])
a_list[0]=8
print("first num becomes",a_list[0])
del a_list[0]
print(a_list[:])
```

執行結果:

```
[1, 2, 3, 4]
[1, 2, 3, 4, 5]
first num is 1
first num becomes 8
[2, 3, 4, 5]
```

0527python-2:

```
# Dict的操作
my_dict = {'a': 1, 'b': 2, 'c': 3}

# 添加元素到Dict
my_dict['d'] = 4

# 查詢Dict的元素
print(my_dict['a'])

# 更新Dict的元素
my_dict['b'] = 5

# 刪除Dict的元素
del my_dict['c']

print("OUTCOME=",my_dict)
```

執行結果:

```
1
OUTCOME= {'a': 1, 'b': 5, 'd': 4}
```

0527python-3:

```
my_list=[1,2,3,4,5]

# 用for loop操作List
```

```

for item in my_list:
    print(item)

# 用for loop搭配if else, 對list內的元素作判斷
for item in my_list:
    if item > 3:
        print(item)
    else:
        print("Less than or equal to 3")

# 沒有帶參數的python def
def my_function_no_params():
    print("This is a function without parameters")

# 有帶參數的python def
def my_function_with_params(param1, param2):
    print("Parameter 1:", param1)
    print("Parameter 2:", param2)

# 沒有帶參數, 但有帶return的python def
def my_function_no_params_with_return():
    return "This is a function without parameters but with return"

# 有帶參數, 有帶return的python def
def my_function_with_params_and_return(param1, param2):
    return param1 + param2

# 測試以上定義的函數
my_function_no_params()
my_function_with_params("Hello", "World")
print(my_function_no_params_with_return())
print(my_function_with_params_and_return(3, 5))

```

執行結果:

```

1
2
3
4
5
Less than or equal to 3
Less than or equal to 3
Less than or equal to 3
4
5
This is a function without parameters
Parameter 1: Hello

```

Parameter 2: World

This is a function without parameters but with return

8