

Function

ฉัตรชัย เกษมทวีโชค

Chatchai.kase@ku.th

https://www.w3schools.com/python/python_functions.asp

Python Functions

- Function เป็นชุดคำสั่งที่ใช้รันเมื่อถูกเรียกใช้งาน โดยผู้เรียกใช้สามารถส่งข้อมูลไปยัง function ที่เรียกว่า "Parameter"
- บางครั้ง function จะมีการส่งคืนค่าผลลัพธ์กลับออกมาด้วย

```
def function_name():  
    command1  
    command2  
    .....  
    return result_value
```

การเรียกใช้ฟังก์ชัน (calling function)

- การเรียกใช้งานฟังก์ชัน ด้วยการเรียกชื่อฟังก์ชันแล้วตามด้วยวงเล็บ ()

```
def my_hello():  
    print("Hello from a function")
```

```
my_hello()
```

Arguments

- Arguments เป็นข้อมูลที่ถูกส่งไปยังฟังก์ชัน
- การสร้าง arguments ในฟังก์ชันสามารถทำได้โดยระบุชื่อในวงเล็บด้านหลังชื่อฟังก์ชัน โดยมีตัวคั่น argument แต่ละตัวด้วย , (comma)

```
def print_name(fname, lname):  
    print("Hello " + fname + " " + lname)  
  
print_name("John", "Doe")  
print_name("Mary", "Jones")
```

การส่งคืนค่าจากฟังก์ชัน (return value)

- การส่งคืนค่าจากฟังก์ชัน สามารถทำได้โดยใช้คำสั่ง return
- สามารถส่งคืนมากกว่าหนึ่งค่า

```
def cal_grade(score):  
    if score >= 80:  
        grade = "A"  
    elif score >= 65:  
        grade = "B"  
    elif score >= 50:  
        grade = "C"  
    else:  
        grade = "F"  
    return grade
```

```
grade1 = cal_grade(75)  
grade2 = cal_grade(85)  
grade3 = cal_grade(55)  
print (grade1, grade2, grade3)
```

```
def cal_grade(score):  
    if score >= 80:  
        grade = "A"  
        grade_num = 4.0  
    elif score >= 65:  
        grade = "B"  
        grade_num = 3.0  
    elif score >= 50:  
        grade = "C"  
        grade_num = 2.0  
    else:  
        grade = "F"  
        grade_num = 1.0  
  
    return grade, grade_num
```

```
grade1, grade_num1 = cal_grade(75)  
grade2, grade_num2 = cal_grade(85)  
grade3, grade_num3 = cal_grade(55)  
  
print (grade1, grade2, grade3)  
print (grade_num1, grade_num2, grade_num3)
```

Default Parameter Value

- เรากำหนดค่าเริ่มต้นของพารามิเตอร์แต่ละตัวได้โดยการให้ตัวแปรพารามิเตอร์เท่ากับค่าเริ่มต้น (Default Value) ที่ต้องการ
- ในกรณีที่เราไม่ส่งค่าพารามิเตอร์นั้น ค่าพารามิเตอร์จะเป็นค่าเริ่มต้นทันที

```
def my_country (country = "Norway"):  
    print("I am from " + country)
```

```
my_country("Sweden")  
my_country("India")  
my_country("Brazil")  
my_country()
```

Passing a List as an Argument

- เราสามารถส่งค่าพารามิเตอร์ให้กับอาร์กิวเมนต์ด้วยชนิดรูปแบบใด ๆ ก็ได้ เช่น String, int, float, list, dictionary, array โดยค่าพารามิเตอร์จะเป็นค่าชนิดข้อมูลแบบเดียวกัน

```
def my_foods(food):  
    for x in food:  
        print(x)  
  
fruits = ["apple", "banana", "cherry"]  
my_foods(fruits)
```

Arbitrary Arguments, *args

- ถ้าเราไม่ทราบจำนวนอาร์กิวเมนต์ที่จะส่งให้กับฟังก์ชัน เราสามารถใช้ * ก่อนกำหนดชื่อพารามิเตอร์
- การรับค่าข้อมูลเข้าจะใช้เป็น **tuple** ของอาร์กิวเมนต์ และสามารถเข้าถึงรายการข้อมูลได้โดยการระบุหมายเลขลำดับ

```
def my_kids(*kids):  
    print("The youngest child is " + kids[len(kids)-1])  
  
my_kids("Ton", "Krung", "Lek", "Jew")
```


คีย์เวิร์ดอาร์กิวเมนต์ (Keyword Arguments)

- เราสามารถส่งค่าอาร์กิวเมนต์ด้วยการระบุชื่อตัวแปรพารามิเตอร์ในรูปแบบ **key = value**
- การส่งอาร์กิวเมนต์ลักษณะนี้จะทำให้ลำดับของอาร์กิวเมนต์สามารถสลับตำแหน่งได้

```
def my_children(child3, child2, child1):  
    print("The youngest child :" + child3)  
  
my_children(child1 = "Ton", child2 = "Krung", child3 = "Lek")
```

Arbitrary Keyword Arguments, **kwargs

- ถ้าเราไม่ทราบจำนวนอาร์กิวเมนต์แบบคีย์เวิร์ดที่จะส่งให้กับฟังก์ชัน เราสามารถใช้ ****** ก่อนกำหนดชื่อพารามิเตอร์
- การรับค่าข้อมูลเข้าจะใช้เป็น dictionary ของอาร์กิวเมนต์ และสามารถเข้าถึงรายการข้อมูลได้โดยการระบุหมายเลขลำดับ

```
def my_name(**kid) :  
    print("His first name is " + kid["fname"])  
    print("His last name is " + kid["lname"])  
  
my_name(fname = "John", lname = "Doe")
```