



**Week 4 :**

# **Python Function**

**Natakorn Pramayan, Ph.D.**



# Content

- **Python Functions**
- **Arguments**
- **Python Lambda**

# Python Functions

- Function เป็นกลุ่มของคำสั่ง ที่จะทำงานเมื่อถูกเรียกใช้งานเท่านั้น
- สามารถส่งข้อมูลให้กับ function ได้ โดยเรียกข้อมูลนั้นว่า parameter และ function สามารถ return ค่าผลลัพธ์กลับมาได้
- การประกาศ Function ทำได้โดยพิมพ์ function ตามด้วยชื่อ function

# ตัวอย่าง การประกาศ Function ในภาษา Python

การประกาศ Function

```
3 def Select(parameter1, parameter2, parameter3):  
4     statement 1  
5     .  
6     statement n  
7     return result
```

ชื่อของฟังก์ชัน

# การสร้าง Function ในภาษา Python

สามารถประกาศ Function ได้โดยใช้ keyword `def` ตามด้วยชื่อ Function และปิดท้ายด้วย `()`

ชื่อของฟังก์ชัน

การประกาศ  
Function

```
3 def Introduce() :  
4     print("Hello, I'm Function")  
5     print("I'll do nothing at all")  
6     print("Call me if you need help")  
7
```


คำสั่งภายในฟังก์ชัน

# Function จะไม่ทำงานจนกว่าจะมีการเรียกใช้

## ตัวอย่าง

```
3  def Introduce() :  
4      print("Hello, I'm Function")  
5      print("I'll do nothing at all")  
6      print("Call me if you need help")  
7  Introduce()
```

การเรียกใช้ฟังก์ชัน



```
Hello, I'm Function  
I'll do nothing at all  
Call me if you need help
```

ผลลัพธ์



## แบบฝึกหัดที่ 4.1

คำชี้แจง ให้นักเรียนเขียนโปรแกรมร้านขายของโดยมีการสร้างฟังก์ชัน ดังต่อไปนี้

1. แสดงรายการสินค้า (ต้องมีสินค้าอย่างน้อย 5 ชนิด)
2. หยิบสินค้าเข้าตะกร้า
3. แสดงรายการจำนวนและราคาของสินค้าที่หยิบ
4. ปิดโปรแกรม

โดยโปรแกรมจะทำงานวนซ้ำจนกว่าจะเรียกใช้ฟังก์ชันปิดโปรแกรม



## โปรแกรมร้านค้าออนไลน์

1. แสดงรายการสินค้า
2. หยิบสินค้าเข้าตะกร้า
3. แสดงรายการจำนวนและราคาของสินค้าที่หยิบ
4. หยิบสินค้าออกจากตะกร้า
5. ปิดโปรแกรม

กรุณเลือกทำรายการ

## กรุณเลือกทำรายการ 2

1. ยาดม
2. น้ำเปล่า
3. มาม่า
4. สบู่
5. แปรงสีฟัน
6. ออกจากฟังก์ชัน

เลือกหยิบสินค้าหมายเลข:

## กรุณเลือกทำรายการ 1 รายการสินค้า

1. ยาดม ราคา 15 บาท
2. น้ำเปล่า ราคา 10 บาท
3. มาม่า ราคา 20 บาท
4. สบู่ ราคา 30 บาท
5. แปรงสีฟัน ราคา 25 บาท

## กรุณเลือกทำรายการ 3

สินค้าที่ลูกค้าหยิบไปมีดังนี้		
.....สินค้า.....	.....ราคา.....	.....จำนวน.....
.....ยาดม.....	4.....	60.....
.....น้ำเปล่า.....	4.....	40.....
.....มาม่า.....	6.....	120.....
.....สบู่.....	3.....	90.....
.....แปรงสีฟัน.....	3.....	75.....
.....รวม.....	20.....	385.....

กรุณเลือกทำรายการ 4

สินค้าในตะกร้ามีดังนี้

1. ยาดม

2. น้ำเปล่า

3. มะม่วง

4. สบู่

5. แปรงสีฟัน

เลือกลำดับสินค้าที่จะหยิบออก หรือพิมพ์ -1 เพื่อออก:

1. แสดงรายการสินค้า

2. หยิบสินค้าเข้าตะกร้า

3. แสดงรายจำนวนและราคาของสินค้าที่หยิบ

4. หยิบสินค้าออกจากตะกร้า

5. ปิดโปรแกรม

กรุณเลือกทำรายการ 5

ต้องการออกจากโปรแกรมใช่หรือไม่ y/n: y

# Arguments

Arguments เป็นตัวแปรที่กำหนดขึ้นเพื่อรองรับข้อมูลที่ส่งมาที่ Function โดยสามารถกำหนด Arguments ในวงเล็บหลังชื่อ Function ในตอนประกาศ Function

ตัวอย่าง

```
3 def Introduce(names) :  
4     print("Hello, I'm " + names)  
5  
6 Introduce("Python")
```

Arguments

ถูกนำไปเก็บไว้ที่  
Arguments name

Hello, I'm Python

ผลลัพธ์

เมื่อประกาศ Function แล้ว สามารถเรียกใช้งาน Function ได้ตลอดเวลา ไม่จำกัดจำนวนครั้ง

### ตัวอย่าง

```
3 def Introduce(names) :  
4     print("Hello, I'm " + names)  
5 Introduce("Python")  
6 Introduce("Natakorn")  
7 Introduce("Pramayan")
```

```
Hello, I'm Python  
Hello, I'm Natakorn  
Hello, I'm Pramayan
```

ผลลัพธ์

การประกาศ Function สามารถมี arguments ได้ไม่จำกัดจำนวน  
โดยใช้เครื่องหมาย Comma ( , ) คั่นระหว่าง argument

## ตัวอย่าง

```
2  
3 def Introduce(province, nation) :  
4     print("Hello, I come from " + province + ", " + nation)  
5  
6 Introduce("Khon Kaen", "Thailand")  
7
```

ถูกนำไปเก็บไว้ที่  
**Argument** province

Hello, I come from Khon Kaen, Thailand

ผลลัพธ์

ถูกนำไปเก็บไว้ที่  
**Argument** nation

## ข้อควรระวังในการประกาศ Function

ที่มี arguments จำนวน argument ที่ Function รับ (ตอนสร้าง Function) และจำนวน argument ที่ส่งให้ Function (ตอนเรียกใช้งาน) ต้องมีจำนวนเท่ากัน

### ตัวอย่าง

```
3 def Introduce(province, nation):  
4     print("Hello, I come from " + province + " " + nation)  
5  
6 Introduce("Khon Kaen")
```

← 2 argument

← 1 argument

```
Introduce("Khon Kaen")  
TypeError: Introduce() missing 1 required positional argument: 'nation'
```

ผลลัพธ์เกิดการ Error

# รูปแบบการประกาศ Function เพิ่มเติม

สามารถประกาศ Function ที่รับค่า argument ไม่เท่ากับตอนสร้างได้ และไม่ทำให้เกิด Error ในภาษา Python มีรูปแบบการประกาศ Function แบ่งเป็น 3 แบบใหญ่ ๆ ดังนี้

## 1. Default Argument

เป็นรูปแบบมีประโยชน์สูง เมื่อต้องการเรียกใช้ Function โดยมอบค่า argument **น้อยกว่าจำนวนที่ประกาศ**ไว้ตอนสร้าง Function

ตัวอย่าง

```
3 def Introduce(province, nation = 'Thailand') :  
4     print("Hello, I come from " + province + ", " + nation)  
5  
6 Introduce("Khon Kaen")
```

Default argument

```
Hello, I come from Khon Kaen, Thailand
```

ผลลัพธ์



## ในการเขียนโปรแกรม

เรียก Default argument ว่า Optional argument (มอบค่าหรือใช้ค่า Default)

ส่วน Argument ที่ไม่ใช่ Default argument เรียกว่า Required argument (จำเป็นต้องมอบค่า)

ตัวอย่าง

Required argument

Optional argument

```
3 def Introduce(province, nation = 'Thailand') :  
4     print("Hello, I come from " + province + ", " + nation)  
5  
6 Introduce("Khon Kaen", "USA")
```

Hello, I come from Khon Kaen, Thailand

ผลลัพธ์

## 2. Keyword Arguments (kwarg)

Function ใน Python สามารถเขียนในรูปแบบ Keyword Argument ได้ (kwarg = value)

### Function ทั่วไป

เมื่อเรียกใช้งาน ต้องมอบค่า Argument เรียงลำดับตามที่ได้ประกาศไว้เมื่อตอน

สร้าง Function

```
2
3 def Introduce(province, nation):
4     print("Hello, I come from " + province + ", " + nation)
5
6 Introduce("Khon Kaen", "Thailand")
7
```

ถูกนำไปเก็บไว้ที่  
argument province

ถูกนำไปเก็บไว้ที่  
argument nation

# Keyword Arguments Function

## ตัวอย่าง

```
3 def Introduce(arg1, arg2 = 'com' , arg3 = 'ed' , arg4 = 'kku' ) :  
4     print("Hello, I am "+arg1+", "+arg2+" "+arg3+" "+arg4)  
5  
6 Introduce("Python")  
7 Introduce(arg1 = "Python")  
8 Introduce(arg1 = "Python" , arg3 = "Sci" )  
9 Introduce("Python", arg4 = "CMU")
```

Keyword Arguments

```
Hello, I am Python, com ed kku  
Hello, I am Python, com ed kku  
Hello, I am Python, com Sci kku  
Hello, I am Python, com ed CMU
```

ผลลัพธ์

## ข้อควรระวังในการเรียกใช้งานแบบ Keyword Arguments

### ตัวอย่าง

```
2
3 def Introduce(arg1, arg2 = 'com' , arg3 = 'ed' , arg4 = 'kku' ) :
4     print("Hello, I am "+arg1+", "+arg2+" "+arg3+" "+arg4)
5
6 Introduce() # missing 1 required argument
7 Introduce(arg1 = "Python" , "CMU") # non-kwarg after kwarg
8 Introduce("Python 2" , arg1 = "Python 3" ) # same argument
9 Introduce(arg99 = "CMU") # unknown kwarg
10
```

หากไม่รู้จำนวนของ Argument ทั้งหมดที่ต้องการใช้งานในตอนประกาศ Function สามารถใช้เครื่องหมาย \* หรือ \*\* นำหน้าชื่อของ argument ได้

- นำ \* นำหน้าชื่อ argument จะเป็นการรับข้อมูลแบบ Tuple เรียกวิธีการนี้ว่า Arbitrary argument
- นำ \*\* นำหน้าชื่อ argument จะเป็นการรับข้อมูลแบบ Dictionary เรียกวิธีการนี้ว่า Arbitrary keyword argument (หากใช้ร่วมกัน \*name ต้องมาก่อน \*\*name เสมอ)

## ตัวอย่าง

Arbitrary argument

Arbitrary keyword argument

```
1 def Introduce (name, *hobby, **address) :  
2     print ("Hello, I am " + name + ".")  
3     print ("My address : ")  
4     for kw in address :  
5         print (kw + ":" + address [kw])  
6     print ("My hobby : ")  
7     for arg in hobby :  
8         print(arg)  
9  
10 Introduce("P", "Sport", "Music", "game", province = "Khon Kaen", nation = "Thailand")
```

การรับข้อมูลแบบ Tuple

```
Hello, I am P.  
My address :  
province:Khon Kaen  
nation:Thailand  
My hobby :  
Sport  
Music  
game
```

การรับข้อมูลแบบ Dictionary

ผลลัพธ์

## แบบฝึกหัดที่ 4.2

คำชี้แจง ให้นักศึกษาเขียนโปรแกรมพจนานุกรม โดยให้สร้างตัวแปร Dictionary มาเพื่อเก็บคำศัพท์ กำหนดให้ในมีคำศัพท์อย่างต่ำ 5 คำ โดยประกอบด้วย คำศัพท์ ชนิดของคำ ความหมาย โดยมีฟังก์ชันคำสั่ง ดังนี้

- 1) เพิ่มคำศัพท์
- 2) ลบคำศัพท์
- 3) แสดงคำศัพท์ทั้งหมด
- 4) ออกจากโปรแกรม

### พจนานุกรม

- 1) เพิ่มคำศัพท์
- 2) แสดงคำศัพท์
- 3) ลบคำศัพท์
- 4) ออกจากโปรแกรม

Input Choice: 1

เพิ่มคำศัพท์                      anger

ชนิดคำศัพท์ (n. , v. , adj. , adv. , ) n.

ความหมาย                      ความโกรธ

เพิ่มคำศัพท์เรียบร้อยแล้ว

### พจนานุกรม

- 1) เพิ่มคำศัพท์
- 2) แสดงคำศัพท์
- 3) ลบคำศัพท์
- 4) ออกจากโปรแกรม

Input Choice: 2

-----  
คำศัพท์ทั้งหมด 6 คำ  
-----

คำศัพท์	ประเภท	ความหมาย
dog	n. คำนาม	สุนัข
bat	n. คำนาม	ค้างคาว
eat	v. คำกริยา	รับประทาน
kick	v. คำกริยา	เตะ
read	v. คำกริยา	อ่าน
anger	n. คำนาม	ความโกรธ

### พจนานุกรม

- 1) เพิ่มคำศัพท์
- 2) แสดงคำศัพท์
- 3) ลบคำศัพท์
- 4) ออกจากโปรแกรม

Input Choice: 3

พิมพ์คำศัพท์ที่ต้องการลบ:                      dog

ต้องการลบ dog ใช่หรือไม่ (y/n): y

ลบ dog เรียบร้อยแล้ว

### พจนานุกรม

- 1) เพิ่มคำศัพท์
- 2) แสดงคำศัพท์
- 3) ลบคำศัพท์
- 4) ออกจากโปรแกรม

Input Choice: 4

ต้องการออกจากโปรแกรมใช่หรือไม่ (y/n):     y

ออกจากโปรแกรมเรียบร้อยแล้ว

### 3. Special parameters

ภาษา Python สามารถกำหนดได้ว่าต้องการให้ argument รับค่าแบบใด  
การกำหนดวิธีการรับค่า

- ใช้เครื่องหมาย / เพื่อกำหนดให้ argument รับค่าตาม Position เท่านั้น
- ใช้เครื่องหมาย \* เพื่อกำหนดให้ argument รับค่าด้วย Keyword เท่านั้น
- ไม่กำหนดเครื่องหมาย เพื่อให้ argument รับค่าได้ทั้ง Position และ Keyword



## ตัวอย่าง Function ที่มีการใช้ Special Parameter

Function `standard_arg` รับค่าได้ทั้งแบบตาม Position หรือแบบ Keyword

```
1 def standard_arg(arg) :  
2     print (arg)  
3 standard_arg(1)
```

ผลลัพธ์

1

Function `position_only` รับค่าได้เฉพาะแบบตาม Position เท่านั้น

```
1 def position_only(arg, /) :  
2     print (arg)  
3 position_only(1)  
4 position_only(arg=1)  
5
```

1

ผลลัพธ์เกิดการ Error  
เพราะฟังก์ชันเป็น  
การรับค่าแบบ Position  
ไม่ใช่แบบ Keyword

**Function keyword\_only** เป็นการประกาศให้ Parameter arg  
รับค่าได้เฉพาะแบบ Keyword เท่านั้น

```
1 def keyword_only(*, arg) :  
2     print (arg)  
3 keyword_only(arg=1)  
4 keyword_only(1)  
5
```

ผลลัพธ์

1

ผลลัพธ์เกิดการ Error  
เพราะฟังก์ชันเป็น  
การรับค่าแบบ Keyword  
ไม่ใช่แบบ Position

**Function combined** เป็นการประกาศให้ Parameter pos\_only รับค่าได้เฉพาะแบบตาม Position, standard รับค่าได้ทั้ง 2 แบบ และ kwd\_only รับค่าได้เฉพาะแบบ Keyword เท่านั้น

```
1 def combined(pos_only , / , standard , * , kwd_only) :  
2     | print(pos_only, standard , kwd_only)  
3     combined(1,2,3)  
4     combined(1,2,kwd_only=3)  
5     combined(1,standard=2,kwd_only=3)  
6     |
```

1 2 3

เกิดการ Error เพราะฟังก์ชัน  
กำหนดให้รับค่าตามตำแหน่ง  
ได้เพียง 2 ค่าเท่านั้น

ผลลัพธ์

# การ Return ค่า

วิธีการสร้าง Function ให้สามารถส่งค่ากลับมาได้ ทำได้ด้วยการใช้คำสั่ง return

## ตัวอย่าง

```
1  def exponents (base,power) :  
2      return base**power ;  
3  print (exponents (2,3))  
4
```

8

ผลลัพธ์



**Python Lambda**

Lambda เป็น Function ขนาดเล็ก ไม่มีชื่อ สร้างโดยใช้คำสั่ง lambda โดยฟังก์ชันนี้จะ return ค่าผลลัพธ์ของ argument ที่มอบให้ Lambda สามารถมี argument ไม่จำกัด แต่สามารถมีคำสั่งได้เพียงคำสั่งเดียวใน Function

### ตัวอย่าง

```
1 expo = lambda n,p : n**p
2 print (expo(2,3))
```

ส่งค่าไปที่  
argument n

ส่งค่าไปที่  
argument p

8

ผลลัพธ์

จากตัวอย่าง เป็นการเขียน Function หาผลลัพธ์ของเลขยกกำลังโดยใช้ lambda รับค่า argument 2 ค่า คือ n และ p จากนั้น return ค่า n ยกกำลัง p

Lambda จะมีประสิทธิภาพมากขึ้น เมื่อใช้งาน Lambda ใน Function อื่น  
เช่น สร้าง Function ที่รับ argument 1 ค่า และ argument นั้นจะถูกกระทำ  
ด้วยจำนวนที่ไม่รู้จัก

### ตัวอย่าง

```
1 def expo(n):  
2     return lambda p : n**p  
3     expos = expo(2)  
4     print(expos(5))  
5
```

ส่งค่าไปที่ **argument n**

32

ผลลัพธ์

จากตัวอย่าง เป็นการสร้าง Function ที่จะยกกำลัง 2 เลขที่ส่งไปเสมอ

## ตัวอย่าง

```
def expo(n):  
    return lambda p : n**p  
power2 = expo(2)  
power3 = expo(3)  
print(power2(5))  
print(power3(5))
```

32

243

ผลลัพธ์





## แบบฝึกหัดที่ 4.3

คำชี้แจง ให้นักศึกษาเขียนโปรแกรมคำนวณผลการช้อมยิงปืนลูกซอง โดยให้มีตัวแปรรับค่าจำนวนผู้เข้าร่วม และข้อมูลการช้อม (ชื่อผู้ช้อม, คะแนน, ระยะเวลาที่ใช้) โดยให้โปรแกรมวนซ้ำเท่ากับจำนวนผู้เข้าร่วม และให้มีฟังก์ชันในการจัดเรียงอันดับ แล้วให้แสดงผลออกทางหน้าจอ ดังนี้

```
Shotgun Sunday Training 2021
```

```
Condition : 1
```

```
2021-01-28 16:37:27
```

```
-----  
No.--PTS---TIME---COMPETITOR#Name--HIT FACTOR---STATE POINTS---STATE PERCENT  
-----
```

1	48	10	B	4.8000	48.0000	100.00
2	50	12	A	4.1667	43.4028	90.42
3	50	20	C	2.5000	26.0417	54.25

# ลิงค์คลิปโปรแกรมและไฟล์ข้อมูลผู้เข้าร่วม

[https://drive.google.com/drive/folders/1Qtw5XFM7OwPYXy\\_EoK\\_8MlZcc2x8lmN0?usp=sharing](https://drive.google.com/drive/folders/1Qtw5XFM7OwPYXy_EoK_8MlZcc2x8lmN0?usp=sharing)



**Q & A**