

# 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

```
def Select(parameter1, parameter2, parameter3):
    statement 1
    .
    statement n
    return result
```

ชื่อของฟังก์ชัน

# การสร้าง Function ในภาษา Python

สามารถประกาศ Function ได้โดยใช้ keyword def ตามด้วยชื่อ Function และ ปิดท้ายด้วย ( )

```
a def Introduce():

nารประกาศ 4 print("Hello, I'm Function")
    print("I'll do nothing at all")
    print("Call me if you need help")
```

คำสั่งภายในฟังก์ชัน

#### Function จะไม่ทำงานจนกว่าจะมีการเรียกใช้

## <u>ตัวอย่าง</u>

```
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()
```

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

# แบบฝึกหัดที่ 4.1

# <u>คำชี้แจง</u> ให้นักเรียนเขียนโปรแกรมร้านขายของโดยมีการสร้างฟังก์ชั่น ดังต่อไปนี้

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

โดยโปรแกรมจะทำงานวนซ้ำจนกว่าจะเรียกใช้ฟังชั่นปิดโปรแกรม

# โปรแกรมร้านค้าออนไลน์ 1. แสดงรายการสินค้า 2. หยิบสินค้าเข้าตะกร้า 3. แสดงรายจำนวนและราคาของสินค้าที่หยิบ 4. หยิบสินค้าออกจากตะกร้า 5. ปิดโปรแกรม กรุณเลือกทำรายการ

กรุณาเลือกทำรายการ 1 รายการสินค้า
1.ยาดม ราคา 15 บาท 2.น้าเปล่า ราคา 10 บาท
3.มาม่า ราคา 20 บาท 4.สบู่ ราคา 30 บาท 5.แปรงสีฟีน ราคา 25 บาท

กรุณาเลือกทำรายการ 2
1.ยาดม
2.น้าเปล่า
3.มาม่า
4 . สบู่
5 . แปรงสีฟัน
6.ออกจากฟังก์ชัน
เ ลือกหยิบสินค้าหมายเ ลข :

กรุณเลือกทำรายการ 3							
สินค้าทีคุณได้หยิบไปมีดังนี							
สินค้าราคาจำนวน							
ยาดม460 น้าเปล่า440							
มาม่า6120							
สบู่390 นปรงสีฟัน375							
รวม20385							

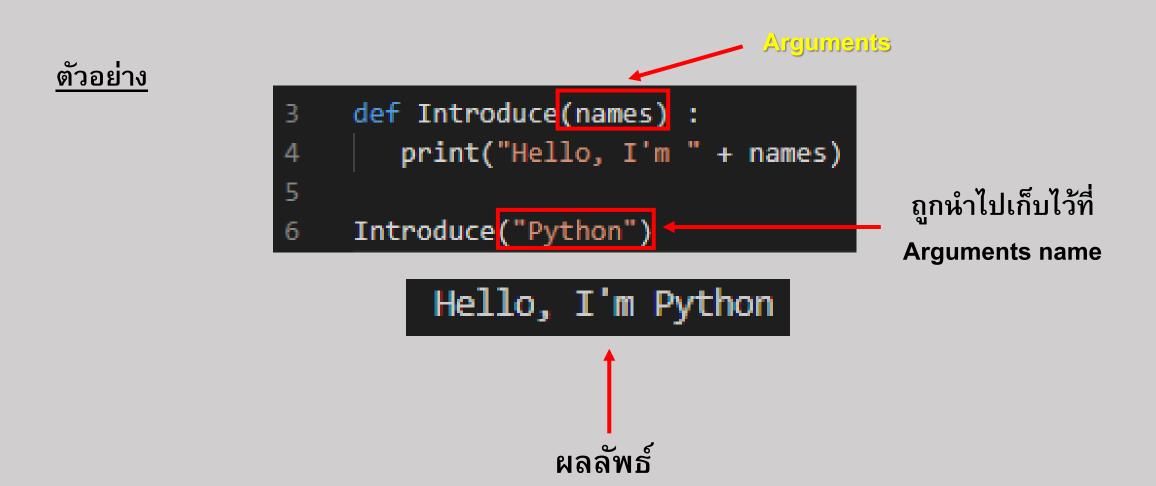
กรุณ เ ลือกทำรายการ 4 สินค้าในตะกร้ามีดังนี้ 1.ยาตม 2.น้ำเปล่า 3.มาม่า 4.สบู่ 5.แปรงสีฟัน เ ลือกลำดับสินค้าที่จะหยิบออก หรือพิมพ์ -1 เ พื่อออก:

- 1. แสดงรายการสิ้นค้า
- 2. หยิบสินค้าเข้าตะกร้า
- 3. แสดงรายจำนวนและราคาของสินค้า*ที่ห*ยิบ
- 4. หยิบสินค้าออกจากตะกร้า
- 5. ปิดโปรแกรม

กรุณาเลือกทำรายการ 5 ต้องการออกจากโปรแกรมใช่หรือไม่ y/n: y

# Arguments

Arguments เป็นตัวแปรที่กำหนดขึ้นเพื่อรองรับข้อมูลที่ส่งมาที่ Function โดย สามารถกำหนด Arguments ในวงเล็บหลังชื่อ Function ในตอนประกาศ Function



เมื่อประกาศ Function แล้ว สามารถเรียกใช้งาน Function ได้ตลอดเวลา ไม่จำกัด จำนวนครั้ง

#### <u>ตัวอย่าง</u>

```
def Introduce(names):
    print("Hello, I'm " + names)
    Introduce("Python")
    Introduce("Natakorn")
    Introduce("Pramayan")
```

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



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

# ัตัวอย่าง def Introduce(province, nation) : print("Hello, I come from " + province + ", " + nation) Introduce("Khon Kaen", "Thailand") ถูกนำไปเก็บไว้ที่ Hello, I come from Khon Kaen, Thailand **Argument** nation ถูกนำไปเก็บไว้ที่ **Argument** province

## ข้อควรระวังในการประกาศ Function

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

#### <u>ตัวอย่าง</u>

```
def Introduce (province, nation):

print("Hello, I come from " + province + ", " + nation)

Introduce ("Khon Kaen")

argument
```

```
Introduce("Khon Kaen")

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

# 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 น้อยกว่าจำนวนที่ประกาศไว้ตอนสร้าง Function

```
ตัวอย่าง
```

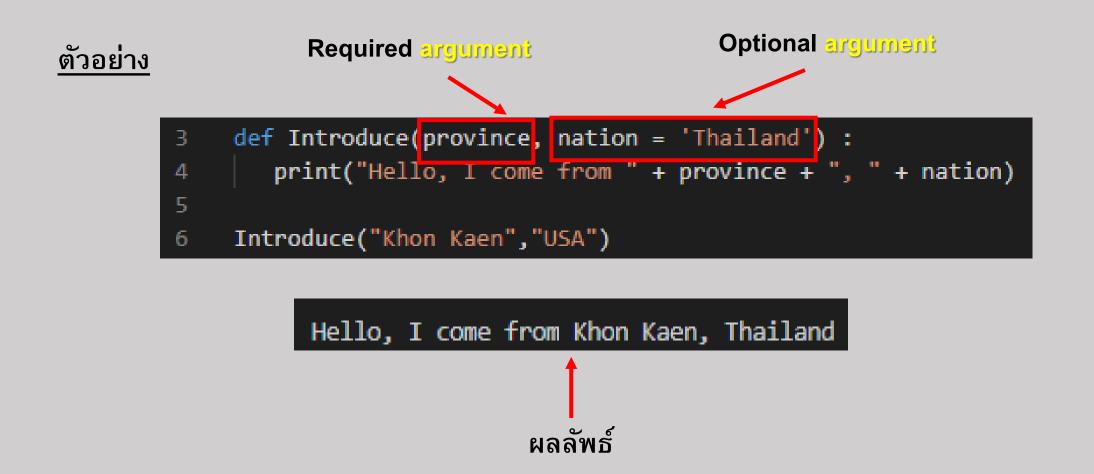
```
def Introduce(province, nation = 'Thailand') :
   print("Hello, I come from " + province + ", " + nation)
Introduce("Khon Kaen")
```

Hello, I come from Khon Kaen, Thailand

**Default argument** 

#### ในการเขียนโปรแกรม

เรียก Default argument ว่า Optional argument (มอบค่าหรือใช้ค่า Default) ส่วน Argument ที่ไม่ใช่ Default argument เรียกว่า Required argument (จำเป็นต้องมอบค่า)



## 2. Keyword Arguments (kwarg)

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

## Function ทั่วไป

เมื่อเรียกใช้งาน ต้องมอบค่า Argument เรียงลำดับตามที่ได้ประกาศไว้เมื่อตอน

```
      def Introduce(province, hation)

      4
      print("Hello, I come from " + province + ", " + nation)

      5
      Introduce("Khon Kaen", "Thailand")

      ลูกนำไปเก็บไว้ที่
      ถูกนำไปเก็บไว้ที่

      argument province
      argument nation
```

#### **Keyword Arguments Function**

#### ตัวอย่าง

```
def Introduce(arg1, arg2 = 'com' , arg3 = 'ed' , arg4 = 'kku' ) :
    print("Hello, I am "+arg1+", "+arg2+" "+arg3+" "+arg4)

Introduce("Python")
Introduce(arg1 = "Python")
Introduce(arg1 = "Python" , arg3 = "Sci" )
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
```

## ข้อควรระวังในการเรียกใช้งานแบบ Keyword Arguments

#### ตัวอย่าง

```
def Introduce(arg1, arg2 = 'com', arg3 = 'ed', arg4 = 'kku'):
    print("Hello, I am "+arg1+", "+arg2+" "+arg3+" "+arg4)

Introduce() # missing 1 required argument
    Introduce(arg1 = "Python", "CMU") # non-kwarg after kwarg
    Introduce("Python 2", arg1 = "Python 3") # same argument
    Introduce(arg99 = "CMU") # unknown kwarg

Introduce(arg99 = "CMU") # unknown kwarg
```

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 keyword argument Arbitrary argument** def Introduce (name, \*hobby, \*\*address) print ("Hello, I am " +name+".") print ("My address : ") for kw in address: print (kw + ":" + address [kw]) print ("My hobby : ") for arg in hobby: print(arg) Introduce("P", "Sport", "Music", "game", province = "Khon Kaen", nation = "Thailand" Hello, I am P. My address : การรับข้อมูลแบบ Tuple province:Khon Kaen nation:Thailand การรับข้อมูลแบบ Dictionary My hobby: Sport ผลลัพธ์ Music game

# แบบฝึกหัดที่ 4.2

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

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

```
พลนานุกรม
1) เพิ่มคำศัพท์
2) แสดงคำศัพท์
3) ลบคำศัพท์
4) ออกจากโปรแกรม

Input Choice: 1
เพิ่มคำศัพท์ anger
ชนิดคำศัพท์ (n., v., adj., adv., ) n.
ความหมาย ความโกรธ
เพิ่มคำศัพท์เรียบร้อยแล้ว
```

#### 

```
พลนานกรม
1) เพิ่มคำศัพท์
2) แสดงคำศัพท์
3) ลบคำศัพท์
4) ออกจากโปรแกรม
Input Choice: 2
        คำศัพท์มีทั้งหมด 6 คำ
     คำศัพท์ ประเภท
                         ความหมาย
     dog
            n. คำนาม
                         สนัข
     bat n. คำนาม ค้างคาว
     eat v. คำกริยา รับประทาน
     kick
            v. คำกริยา
                        เตร
             v. คำกริยา อ่าน
     read
                       ความโกรธ
              ท. คำนาม
     anger
```

#### พฉนานุกรม

- 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)
```

Function position\_only รับค่าได้เฉพาะแบบตาม Position เท่านั้น

```
1
def position_only(arg, /) :

2
print (arg)

3
position_only(1)

4
position_only(arg=1)

5
เพราะฟังก์ชันเป็น

การรับค่าแบบ Position

ไม่ใช่แบบ Keyword
```

# Function keyword\_only เป็นการประกาศให้ Parameter arg รับค่าได้เฉพาะแบบ Keyword เท่านั้น

```
ผลลัพธ์
def keyword_only(*, arg) :
    print (arg)
keyword_only(arg=1)
keyword only(1)
                                            ผลลัพธ์เกิดการ Error
                                             เพราะฟังก์ชันเป็น
                                          การรับค่าแบบ Keyword
                                             ไม่ใช่แบบ Position
```

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

```
def combined(pos_only , / , standard , * , kwd_only) :
          print(pos only, standard , kwd only)
      combined(1,2,3)
      combined(1,2,kwd_only=3)
      combined(1,standard=2,kwd only=3)
เกิดการ Error เพราะฟังก์ชัน
กำหนดให้รับค่าตามตำแหน่ง
   ได้เพียง 2 ค่าเท่านั้น
```

## การ Return ค่า

วิธีการสร้าง Function ให้สามารถส่งค่ากลับมาได้ ทำได้ด้วยการใช้คำสั่ง return

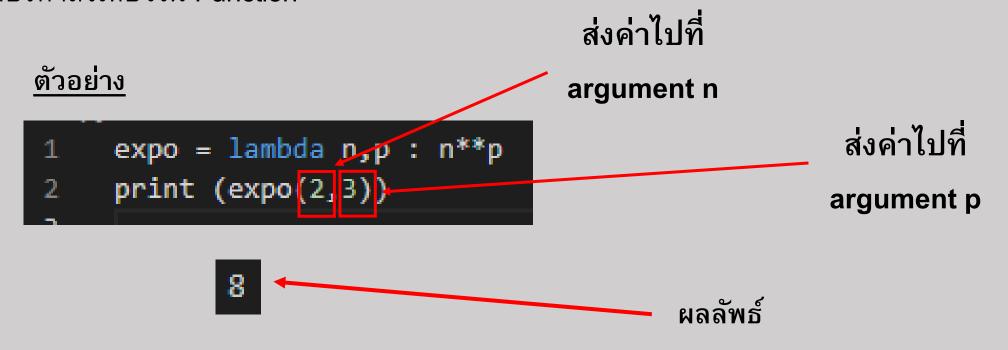
#### <u>ตัวอย่าง</u>

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

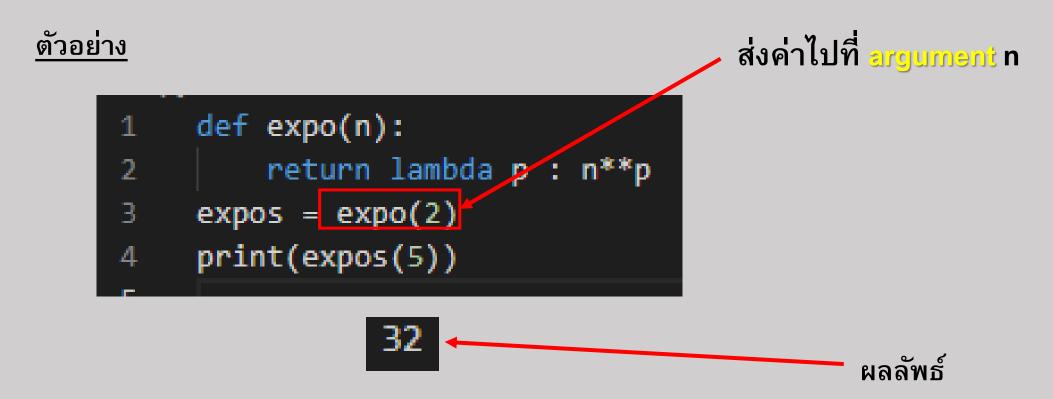
ุ ผลลัพธ์

# Python Lambula

Lambda เป็น Function ขนาดเล็ก ไม่มีชื่อ สร้างโดยใช้คำสั่ง lambda โดยฟังก์ชันนี้จะ return ค่า ผลลัพธ์ของ argument ที่มอบให้ Lambda สามารถมี argument ไม่จำกัด แต่สามารถมีคำสั่งได้ เพียงคำสั่งเดียวใน Function



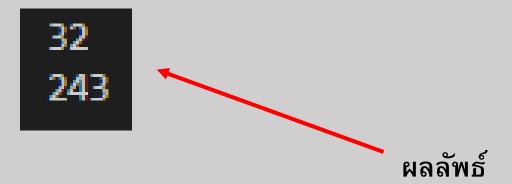
จากตัวอย่าง เป็นการเขียน Function หาผลลัพธ์ของเลขยกกำลังโดยใช้ lambda รับค่า argument 2 ค่า คือ n และ p จากนั้น return ค่า n ยกกำลัง p Lambda จะมีประสิทธิภาพมากขึ้น เมื่อใช้งาน Lambda ใน Function อื่น เช่น สร้าง Function ที่รับ argument 1 ค่า และ argument นั้นจะถูกกระทำ ด้วยจำนวนที่ไม่รู้จัก



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))
```



# แบบฝึกหัดที่ 4.3

คำชี้แจง ให้นักศึกษาเขียนโปรแกรมคำนวณผลการซ้อมยิงปืนลูกซอง โดยให้มีตัวแปรรับค่าจำนวนผู้เข้าร่วม และข้อมูลการซ้อม(ชื่อผู้ซ้อม,คะแนน,ระยะเวลาที่ใช้) โดยให้โปรแกรมวนซ้ำเท่ากับจำนวนผู้เข้าร่วม และให้มีฟังชั่นในการจัดเรียงอันดับ แล้วให้แสดงผลออกทางหน้าจอ ดังนี้

Con	dtion	_	raining 20 :27	21		
No.	PTS-	TIME	COMPET	ITOR#NameHIT FACTOR	STATE POINTS	-STATE PERCENT
1	48	10	В	4.8000	48.0000	100.00
2	50	12	Α	4.1667	43.4028	90.42
3	50	20	С	2.5000	26.0417	54.25

# ลิ้งค์คลิปโปรแกรมและไฟล์ข้อมูลผู้เข้าร่วม

https://drive.google.com/drive/folders/1Qtw5XFM7OwPYXy\_EoK\_8MIZcc2x8 lmNo?usp=sharing



Q & A