Mahidol University
Wisdom of the Land

Lecture 08: Function I

Lecturers:

Aj. Jidapa Kraisangka

Aj. Akara Supratak

Aj. Tipajin Thaipisutikul

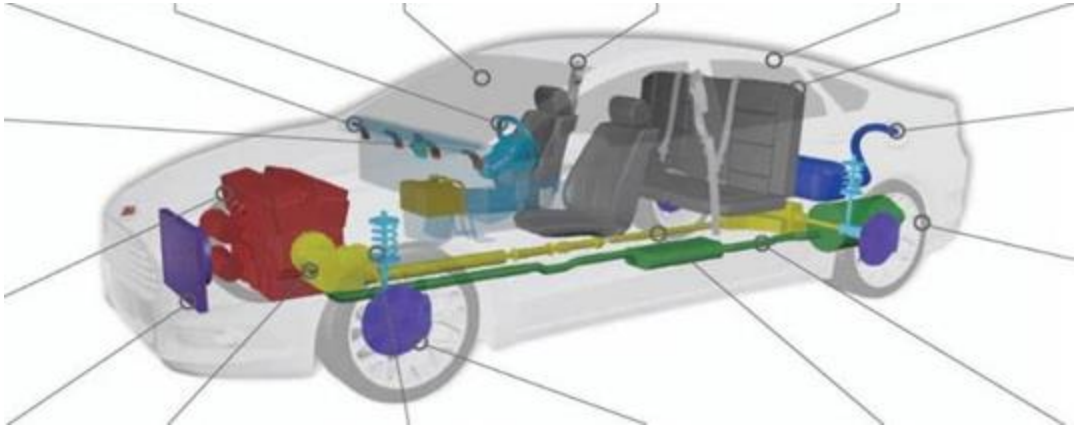
เนื้อหาวันนี้

1. สามารถอธิบายได้ว่า function คืออะไรและมีประโยชน์อย่างไร
2. สามารถประกาศและสร้าง function ได้
3. สามารถใช้ return keyword ได้
4. สามารถใช้ parameter และ argument ได้
5. สามารถเข้าใจ global และ local scope ได้

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

ตัวอย่างการใช้ function กับรถยนต์

- Functions อธิบายการทำงานต่าง ๆของรถยนต์ในแต่ละส่วน ซึ่งการทำงานนี้ถูกเรียกใช้ได้หลายครั้ง



Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

ตัวอย่างการใช้ function ในการรับค่าและแสดงผล

- These are built-in library functions เช่น print, input

```
1.# Input w and h
2.w=input()
3.h=input()
4.# Calculate area: h x w
5.area=float(h)* float(w)
6.# Display the area
7.print(area)
```

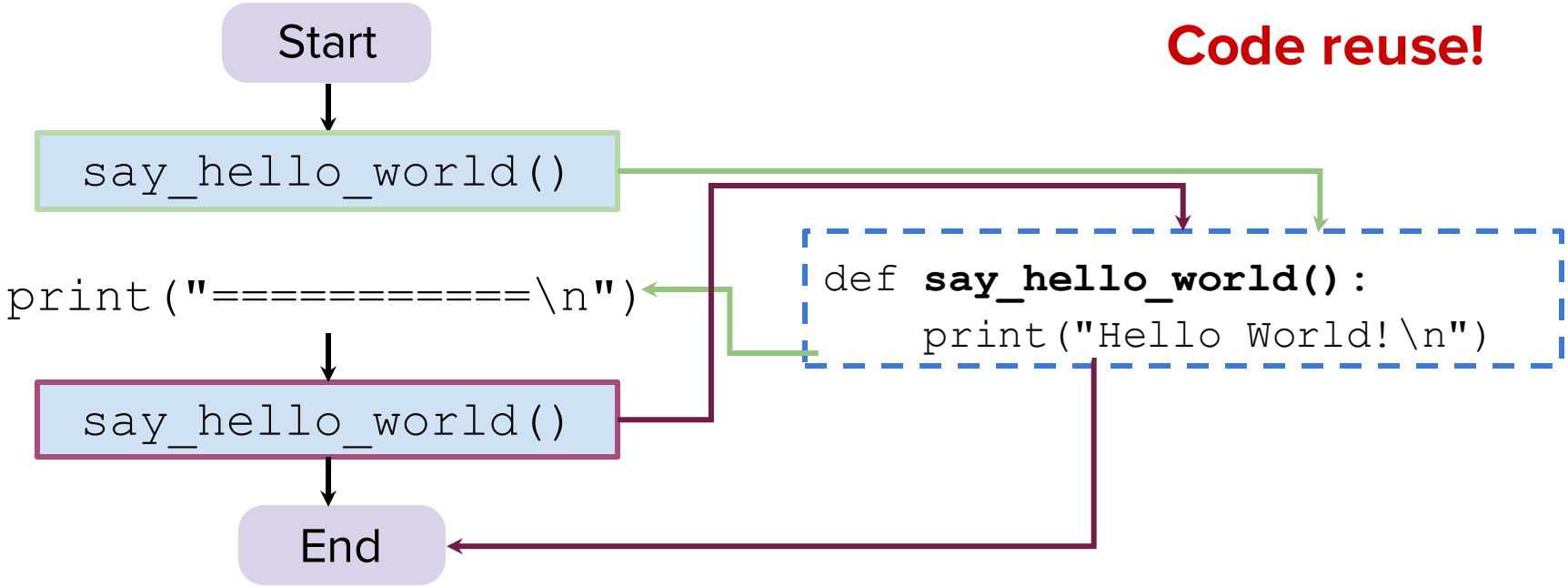
Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

ทำไมต้อง functions?

- ลดการมี code ซ้ำกัน (Avoid duplicate code) และสามารถใช้ code ชุดนั้นซ้ำได้อีกในอนาคต (Promote code reuse)
- ลดความซ้ำซ้อนของโปรแกรมโดยแบ่งสัดส่วน code ตามการใช้งาน (Reduce complexity)

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

ทำไมต้อง functions?



Code reuse!

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

ทำไมต้อง functions?


```
a1, b1, a2, b2, a3, b3, a4, b4, a5, b5 = 5,5,6,6,7,7,8,8,9,9
result  = a1+b1
result2 = a2+b2
result3 = a3+b3
result4 = a4+b4
result5 = a5+b5
print(f"the result is {result}")
print(f"the result is {result2}")
print(f"the result is {result3}")
print(f"the result is {result4}")
print(f"the result is {result5}")
```

มีการซ้ำกันของ codes

- `result = var1 + var2`
- Display `result`

```
the result is 10
the result is 12
the result is 14
the result is 16
the result is 18
```

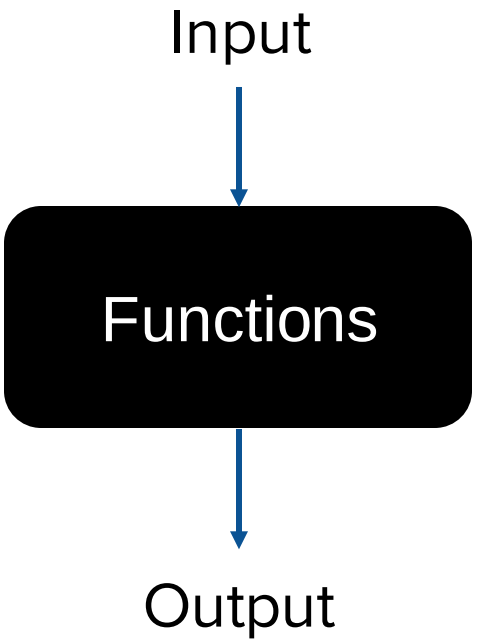
ถ้าผู้ใช้ต้องการเปลี่ยนรูปแบบการแสดงผลเป็น Result = 10 เราต้องแก้ข้อความใน print ทั้งหมด

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

ถ้าใช้ function เข้ามาแก้ปัญหานี้


```
def add(a,b):
    sum = a + b;
    print("the result: "+str(sum));
    return sum;

result  = add(5,5);
result2 = add(6,6);
result3 = add(7,7);
result4 = add(8,8);
result5 = add(9,9);
```

```
the result: 10
the result: 12
the result: 14
the result: 16
the result: 18
```



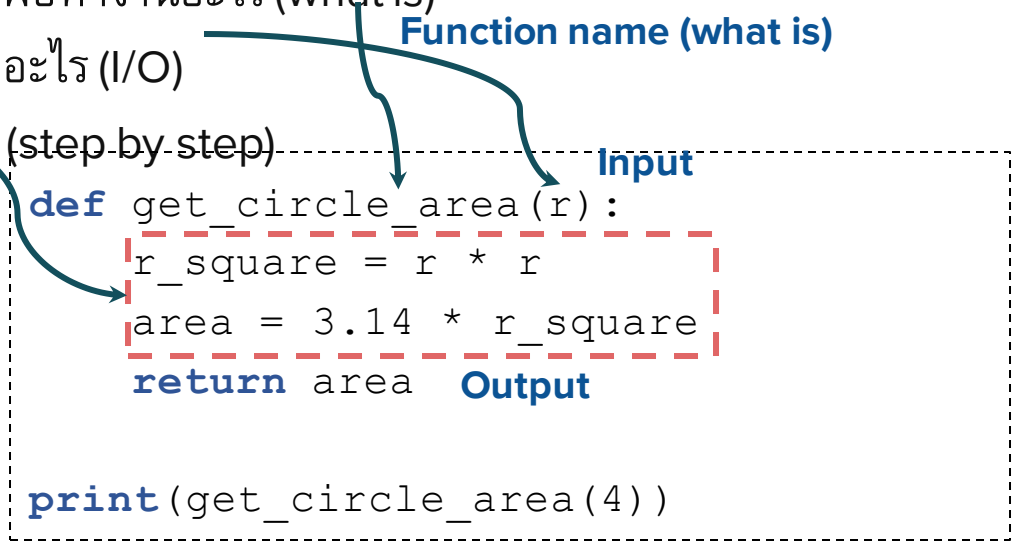
Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

การประกาศ function

เมื่อใดก็ตามที่เราใช้ function, เราควร

Syntax `def function_name():`

- รู้ว่า function นี้ถูกสร้างขึ้นมาเพื่อทำงานอะไร (what is)
- รู้ว่า INPUT และ OUTPUT คืออะไร (I/O)
- รู้ว่า function นี้ทำงานอย่างไร (step by step)



Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

การประกาศ function

```
def name_of_function():
    print('do something') #indent the code

def say_hi():
    print('Hi')

def sing_happy_birthday(): #define fn
    print('Happy Birthday to you')
    print('Happy Birthday to you')
    print('Happy Birthday to you')
    print('Happy Birthday to you')
```

```
sing_happy_birthday() #call fn
say_hi()
name_of_function()
```

```
Happy Birthday to you
Happy Birthday to you
Happy Birthday to you
Happy Birthday to you
Hi
do something
```

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

การใช้ return

- return ใช้เพื่อคืนค่าจาก function
- เมื่อ return ถูกเรียกใช้ เราจะถือว่าเป็นการจบการทำงานของ function นั้นๆ
- เราสามารถ return หลายค่าได้

```
nums = [1,2,3,4]
length = len(nums)
print(length)
```

```
4
```

```
def square_of_7():
    print('before fn')
    return 7**2
     print('after fn')

result= square_of_7()
print(result)
```

```
before fn
49
```

ลองทาย output กันดูสิ?


```
from random import random
def flip_coin(): #diff output each time
    # generate random numbers 0-1
    r = random()
    if r>0.5:
        return 'head'
    else:
        return 'tail'

print(flip_coin())
print(flip_coin())
print(flip_coin())
```

```
head
tail
head
```

```
def hello():
    print("hello\n")

hello()
hello()
```

```
hello

hello
```

parameters vs argument

1. parameter คือตัวแปรที่ถูกระบุไว้ใน function definition
2. argument คือค่าที่ programmer ใส่เมื่อเรียกใช้ function ซึ่งค่าที่ใส่ไปนี้ จะถูกเก็บโดย parameter ตามลำดับ
3. ลำดับในการใส่ค่า parameter และ argument มีความสำคัญ

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

parameter vs argument

```
def divide(num1, num2): # parameters

    return num1/num2

print(divide(2,5))# arguments
print(divide(5,2))
```

```
0.4
2.5
```

```
def divide(num2, num1):
    return num1/num2

print(divide(2,5))
print(divide(5,2))
```

```
2.5
0.4
```

common mistakes when returning -> return too soon


```
def sum_odd_numbers(numbers):
    total =0
    for num in numbers:
        if num%2 !=0:
            total += num
        #return total   #end execution of the loop
    return total

print(sum_odd_numbers([1,2,3,4,5,6,7])) # 1+3+5+7 = 16
```

```
16
```

ถ้าเราใส่ return ตรงนี้ output จะเป็น 1

จงแก้ function ต่อไปนี้ให้ถูกต้อง

- Function count_dollar_signs นี้ควรจะ return จำนวน character ‘$’ จาก argument ที่ function ได้รับ เช่น count_dollar_signs(‘upper$ $ize’) ควร return 2 แต่ปัจจุบัน function return 0 หรือจำนวนที่ไม่ถูกต้อง จงแก้ code ต่อไปนี้

```
def count_dollar_signs(word):
    count = 0
    for char in word:
        if char == '$':
            count += 1
        return count
```

```
def count_dollar_signs(word):
    count = 0
    for char in word:
        if char == '$':
            count += 1
    return count
```

default parameters

- default parameter มีไว้เพื่อกำหนดค่าเริ่มต้นใน function
- default parameter จะถูกเรียกใช้เมื่อ ไม่มีการส่งค่าไปที่ function นั้นๆ

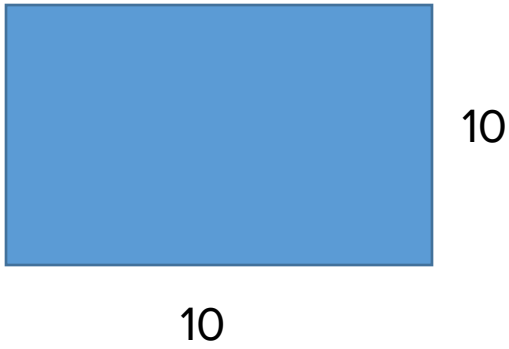
default parameter: ลำดับของ default parameter ควรอยู่หลัง parameter ธรรมดา

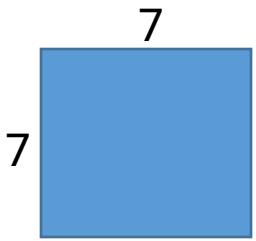
```
def exponent(num, power=2):
    return num ** power

print(exponent(2,3)) #8
print(exponent(3,2)) #9
print(exponent(3)) #9 --> default value for power
print(exponent())
```

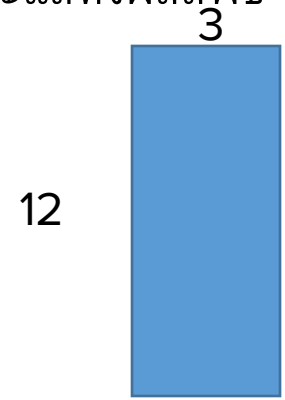
```
TypeError: exponent() missing 1
required positional argument: 'num'
```

In-class Exercise 1

จงเขียนโปรแกรมและสร้าง function เพื่อคำนวณพื้นที่สี่เหลี่ยมและแสดงผลลัพธ์







Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

In-class Exercise 1


```
def get_square_area(width, height):
    return width*height

area1 = get_square_area(10,10)
area2 = get_square_area(7,7)
area3 = get_square_area(3,12)

print(round(area1, 2))
print(round(area2, 2))
print(round(area3, 2))
```

```
100
49
36
```

In-class Exercise 2

จงเขียนโปรแกรมและสร้าง function speak() เพื่อแสดงผลลัพธ์ของเสียงสัตว์แต่ละชนิด โดยเสียงร้องจะถูกกำหนดไว้ดังนี

- If animal is "pig", it should return "oink".
- If animal is "duck", it should return "quack".
- If animal is "cat", it should return "meow"
- If animal is "dog", it should return "woof"
- If animal is anything else, it should return "?"
- If no animal is specified, it should default to "dog"

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

In-class Exercise 2

solution#1

```
def speak(animal="dog"):
    if animal == "pig":
        return "oink"
    elif animal == "duck":
         return "quack"
    elif animal == "cat":
        return "meow"
    elif animal == "dog":
        return "woof"
    else:
        return "?"
```

- If animal is "pig", it should return "oink".
- If animal is "duck", it should return "quack".
- If animal is "cat", it should return "meow"
- If animal is "dog", it should return "woof"
- If animal is anything else, it should return "?"
- If no animal is specified, it should default to "dog"

solution#2

```
def speak(animal="dog"):
    noises = {"dog": "woof", "pig": "oink", "duck": "quack", "cat": "meow"}
    noise = noises.get(animal)
    if noise:
        return noise
    return "?"
```

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

default parameters

- Parameters และ default parameters สามารถเป็นข้อมูลชนิดใดก็ได้

```
def add(a,b):
    return a+b
def math(a,b, fn=add): #default should go the last, order does matter
    return fn(a,b)
def subtract(a,b):
    return a-b

print(math(4,5))
print(math(4,5, subtract))
```

→

```
9
-1
```

Keyword Arguments

- เราสามารถระบุ ค่าของ arguments ที่ถูกส่งไปให้ parameters ใน function ได้
- ใน function เราสามารถตั้งค่า default parameter ได้
- เมื่อเรียกใช้ function เราสามารถตั้งค่า keyword argument ได้

```
def exponent(num,  power=2):
    return num ** power

print(exponent(num = 2,power = 3)) #8
print(exponent(power = 3, num = 2)) #8
print(exponent(2, 3)) #8
```

Scope

- ตัวแปร (variable) ใน python มีขอบเขตการใช้ภายในและภายนอก function (global/local scope)
- ตัวแปร (variable) ที่ถูกสร้างขึ้นใน functions จะมีขอบเขตอยู่ภายใน function นั้นๆ

```
instructor = 'Mock'    #Global variable
def say_hello():
    return 'Hello '+ instructor
print(say_hello()) ⟶ Hello Mock
```

Scope

```
def say_hello():
    instructor = 'Tip' #Local variable
    return 'Hello '+ instructor
print(say_hello())  ⟶  Hello Tip
print(instructor) #NameError
```

global keyword

- global scope นั้นจะไม่ได้ถูกสร้างภายใน function

```
total = 0                    #gloabl variable
def increment():
    global total     #use global keyword to refer
    total +=1
    return total
print(increment())
print(increment())
print(increment())
```

→

```
1
2
3
```

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

- global scope นั้นจะไม่ได้ถูกสร้างภายใน function

```
total = 0                    #gloabl variable
def increment():
    total +=1                #local variable
    return total
print(increment())
print(increment())
print(increment())
```

→

```
UnboundLocalError: local
variable 'total' referenced
before assignment
```

Docstring

ใช้ในการอธิบาย Function

""" คำอธิบาย """

```
def add(a,b):
    """ add two numbers """
    return a+b

print(add.__doc__)
```

→

```
add two numbers
```

In-class Exercise 3

จงเขียน function ‘return_day’ ซึ่งรับค่า 1 parameter (number from 1-7) และ return ชื่อวันโดย (1 is Sunday, 2 is Monday, 3 is Tuesday, etc...) ถ้า ค่าไม่ได้อยู่ในช่วง 1-7 โปรแกรมจะ return None

```
def return_day(num):
    days = ["Sunday","Monday",
  "Tuesday","Wednesday","Thursday","Friday","Saturday"]
    # Check to see if num valid
    if num > 0 and num <= len(days):
        # use num - 1 because lists start at 0
        return days[num-1]
    return None
```


Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

In-class Exercise 4

จงเขียน function ‘number_compare’ ซึ่งรับค่าตัวเลข 2 parameters ถ้าตัวเลขแรกมากกว่าตัวเลขที่สองโปรแกรมจะ return ข้อความ “First is greater” ถ้าตัวเลขที่สองมากก่าตัวเลขแรกโปรแกรมจะ return ข้อความ “Second is greater” ถ้าตัวเลขทั้งสองเท่ากันโปรแกรมจะ return ข้อความ “Numbers are equal”

```
def number_compare(a,b):
    if a > b:
        return "First is greater"
    elif b > a:
        return "Second is greater"
    return "Numbers are equal"
```

In-class Exercise 5

จงเขียน function ‘frequency’ ซึ่งรับค่า 2 parameters (a list, search_term) และ return จำนวนครั้งที่ search_term นั้นถูกพบอยู่ใน list

```
def frequency(collection, searchTerm):
    return collection.count(searchTerm)
print(frequency([1,2,3,4,4,4], 4))
```

return 1 ค่า

```
3
```

```
def frequency(collection, searchTerm):
    return searchTerm, collection.count(searchTerm)
print(frequency([1,2,3,4,4,4], 4))
```

return 2 ค่า

```
(4,3)
```

In-class Exercise 6

จงเขียน function ‘list_manipulation’ ซึ่งรับค่า 1 parameters (command ‘add’ or ‘remove’) และ return list ที่ถูก updated กลับไป

- ถ้ามีการเพิ่มข้อมูลเข้าไปใน list ข้อมูลที่ถูกเพิ่มจะเป็น ‘a’ ที่ตำแหน่งสุดท้าย
- ถ้ามีการลบข้อมูล ข้อมูลตำแหน่งสุดท้ายจะถูกลบ

```
def list_manipulation(command):
    collection = [4,5,6,7,8]
    if command == "remove":
        collection.pop()
        return collection
    elif command == "add":
        collection.append('a')
        return collection
```

```
print(list_manipulation('add'))
print(list_manipulation('remove'))
```

```
[4, 5, 6, 7, 8, 'a']
[4, 5, 6, 7]
```

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu

Wrap-up

1. Function คือการนำเอา code บางส่วนที่มีการทำงานซ้ำกันและวัตถุประสงค์เดียวกันมารวมกันไว้ เพื่อเรียกใช้งานได้ง่าย
2. Function สามารถรับค่า (parameter) และคืนค่าได้เมื่อมี return keyword
3. Function สามารถมี default parameters ได้
4. Variables ที่ถูกสร้างใน Function ถูก scoped ขอบเขตอยู่ภายใน Function นั้นๆ (local scope)
5. เมื่อเรียกใช้งาน Function เราสามารถส่ง keyword arguments ได้

Contact: [jidapa.kra, dolvara.gun, akara.sup, tipajin.tha]@mahidol.edu