PYTHON FUNCTION

FUNCTION #1

CS310 : Computer Programming I

Sirinthorn Cheyasak, Asst.Prof.

Python Functions

Python Functions

A **function is a block of code** that is executed as a unit. While there are many inbuilt functions in Python, you can also create your own function. This is known as user-defined functions. Once created, you can reuse a function at other parts of your program. Before you can use a function however, you must define it.

Function

Function

…

Python Functions

1. Pre-defined/Build-in Functions
2. User-defined Functions
3. Lamda Functions
4. Recursion Functions

Python Built-in Functions

- The Python interpreter has a number of functions and types built into it that are always available. They are listed here in alphabetical order

		Built-in Functions		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

https://docs.python.org/3/library/functions.html

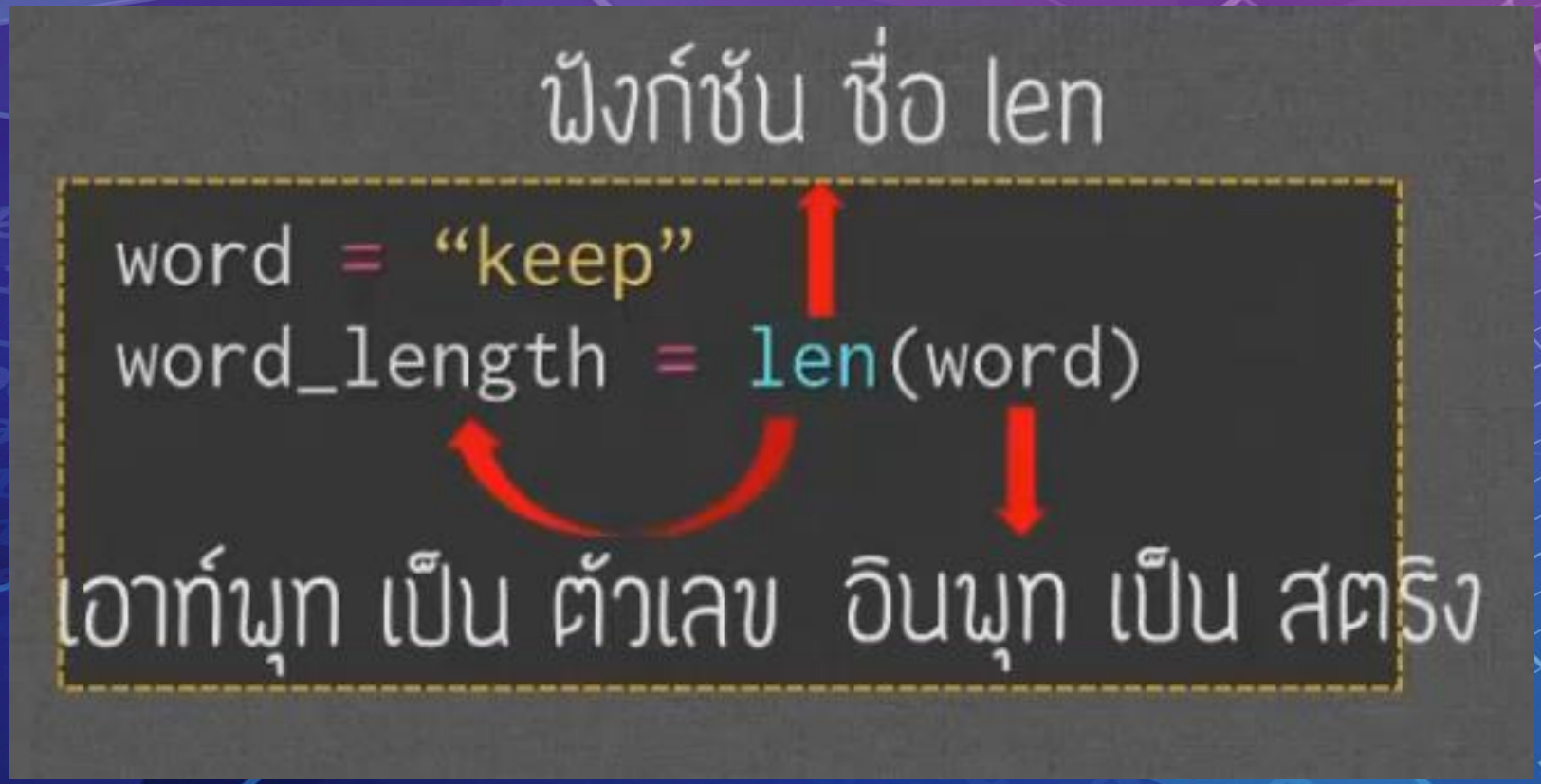
BUILD-IN FUNCTIONS

BUILT-IN FUNCTION EXAMPLE

```
print("Hello")
print("Num = ",10)
num = 200
print(num)
print("num = %d, avg = %0.2f"%(num,num/3))
```

BUILD-IN FUNCTIONS

BUILT-IN FUNCTION EXAMPLE



USER-DEFINED FUNCTIONS

1. Defining/Creating a Function
2. Calling a Function
3. Pass by reference vs value
4. Function Arguments

USER-DEFINED FUNCTIONS

```
print("Program starts")
print("Hi Peter")
print("Nice to see you again!")
print("Enjoy our video!")
# some lines of codes
the_answer = 42
print("Hi Sarah")
print("Nice to see you again!")
print("Enjoy our video!")
width, length = 3, 4
area = width * length
print("Hi Sirinthorn")
print("Nice to see you again!")
print("Enjoy our video!")
```

1

2

3

USER-DEFINED FUNCTIONS

1. Defining/Creating a Function
2. Calling a Function
3. Pass by reference vs value
4. Function Arguments

USER-DEFINED FUNCTIONS

- Defining a Function in Python
  - To define your own Python function, you use
  - the '**def**' keyword before its name. And its name is to
  - Input parameters to the function are placed inside the braces
  - After the braces, then you must add colon (:) before you start writing the function body
  - The function body must be indented
  - The last statement in a function is optionally a return statement.

HOW TO CREATE FUNCTION

Function Syntax for **creating**:

```
def function_name(<Parameters>):
    <Block>
    return result
```

Function Syntax for **calling**:

```
function_name(<Parameters>)
```

USER-DEFINED FUNCTIONS

- Syntax of defining a function in Python

Syntax of creating

```
def functionName() :
    statement(s)
```

colon

indent

Syntax of calling

```
functionName([param])            #calling function without return
var = functionName([param])      #calling function with return
```

HOW TO CREATING A FUNCTION

1. Creating a function without parameters
2. Creating a function with parameters
3. Creating a function without return()
4. Creating a function with return()

USER-DEFINED FUNCTIONS EXAMPLE

```
def greet() :
    print("Hi Sirinthorn")
    print("Nice to see you again!")
    print("Enjoy our video!")
print("Program starts")
greet() #calling function
the_answer = 42
greet()  #calling function
width, length = 3, 4
area = width * length
greet()  #calling function
```

Creating funciton without parameter

```
Program starts
Hi Sirinthorn
Nice to see you again!
Enjoy our video!
Hi Sirinthorn
Nice to see you again!
Enjoy our video!
Hi Sirinthorn
Nice to see you again!
Enjoy our video!
```

USER-DEFINED FUNCTIONS EXAMPLE

```
def greet(name) :
    print("Hi " + name)
    print("Nice to see you again!")
    print("Enjoy our video!")
print("Program starts")
greet("Ning") #calling function
the_answer = 42
greet("Poj")  #calling function
width, length = 3, 4
area = width * length
greet("Lily")  #calling function
```

Creating funciton with parameter

```
Program starts
Hi Ning
Nice to see you again!
Enjoy our video!
Hi Poj
Nice to see you again!
Enjoy our video!
Hi Lily
Nice to see you again!
Enjoy our video!
```

CALLING A FUNCTION

After creating a function, we can call it by using the name of the function followed by parenthesis containing parameters of that particular function

Syntax of creating

```
def functionName() :
    statement(s)
```

colon

indent

Syntax of calling

```
functionName([param])            #calling function without return
var = functionName([param])      #calling function with return
```

EXAMPLE : CALLING A USER-DEFINED FUNCTION

- Calling Function : A simple Python function to check whether x is even or odd

```
# A simple Python function to check
# whether x is even or odd
def evenOdd(x):
   if (x % 2 == 0):
      print("Even")
   else:
      print("Odd")
# Driver code to call the function
evenOdd(2)
evenOdd(3)
```

Defining a Function

- For example, the function below calculates the sum of two numbers

```
# Function to calculate the sum of two numbers
def calculatesum1(num1, num2):
  sum = num1 + num2
  print("Sum = ",sum)
```

Calling a Function

To use a function, you must call it. If it requires parameters, then you must also provide the parameters. For example, we can call the calculatesum function using the code below.

```
# Function to calculate the sum of two numbers
def calculatesum1(num1, num2):
  sum = num1 + num2
  print("Sum = ",sum)
# Calling the function calculatesum1
calculatesum1(4, 6)
```

```
Sum = 10
```

CALLING A FUNCTION

```
# Function without return to calculate the sum of two numbers
def calculatesum1(num1, num2):
  sum = num1 + num2
```

VS.

```
# Function with return to calculate the sum of two numbers
def calculatesum2(num1, num2):
  sum = num1 + num2
  return sum
```

Defining/Calling FUNCTIONS

```
# Function to calculate the sum of two numbers
def calculatesum2(num1, num2):
   sum = num1 + num2
   print("In function...")
   print("Sum = " ,sum)
   return sum

# Calling the function calculatesum2
calculatesum2(100,10) #calling function without a variable to keep a return value

answer = calculatesum(4, 6) #calling function with a variable to keep a return value
print("\nIn global...")
print("Answer = ",answer)
```

Defining/Calling Functions

Function with return() Example

```
def function1() :
    num1 = int(input("Enter a number : "))
    num2 = int(input("Enter a number : "))
    return(num1+num2)

def function2(num1) :
    num2 = int(input("Enter a number : "))
    return(num1+num2)

def function3(num1,num2) :
    ans = num1 + num2
    return(ans)

num1 = 10
num2 = 30
ans = 0
ans = function1()
print("Number1 = %d, Number2 = %d, Summation = %d"%(num1,num2,ans))
ans = function2(num1)
print("Number1 = %d, Number2 = %d, Summation = %d"%(num1,num2,ans))
ans = function3(num1,num2)
print("Number1 = %d, Number2 = %d, Summation = %d"%(num1,num2,ans))
```

CLASS EXERCISE : FIND AREA OF A RECTANGLE

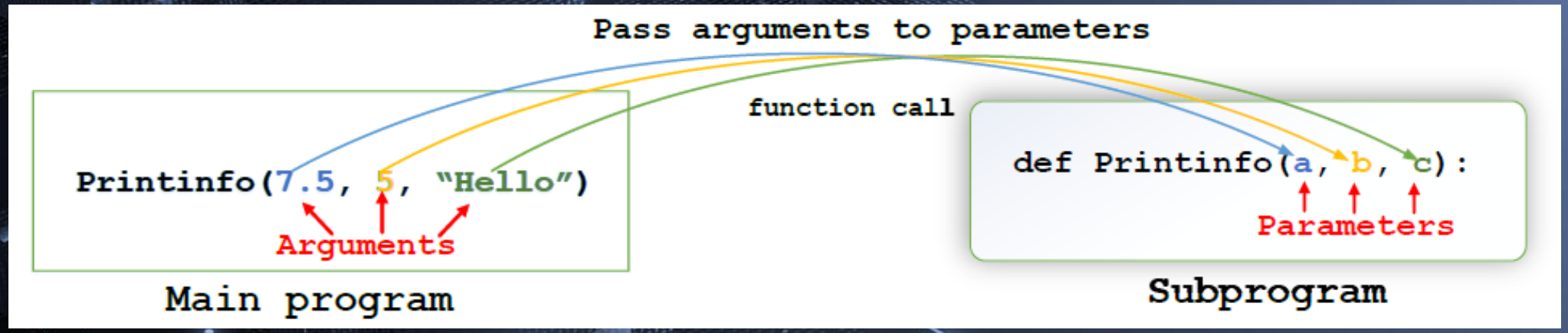
- Write a python program and define function finding an area of triangle
  1. Function name : find_area
  2. Parameter : height, base
  3. Return value : area of triangle

  Formular :

  area of triangle = 0.5 * base * height

Argument and Parameter

- **An arguments** is a value that is passed to a method when it is invoked.
- **A Parameters** is a variable defined by a method that receives a value when the method is called.




```
def Printinfo(a, b, c):
  print("a = %d\nb = %d\nc = %s"%(a,b,c))

Printinfo(7.5,5,"Hello")
```

Argument and Parameter

Example : Argument and parameter example

```
def fn1(name, age):
print(name,age)

def fn2(id, name, year, gpa):
print(id,name,year,gpa)

def fn3(code, name, salary):
print(code,name,salary)

fn1("beauty verymuch",18)
fn2("S1001","Beauty Verymuch",1,3.55)
fn3("E001","Iam Mybigboss",99000)
```

FUNCTION ARGUMENTS

1. Required Arguments
2. Keyword Arguments
3. Default Arguments
4. Variable-length Arguments

1 REQUIRED ARGUMENTS

- Required arguments are the arguments passed to a function in correct positional order. Here, the number of arguments in the function call should match exactly with the function definition.
- To call the function printme(), you definitely need to pass one argument, otherwise it gives a syntax error as follows –

```
def printme(str,num ):
   print (str,",lucky number is",num )

str = "My name is ning"
num = 99
printme(str,num)
```

```
My name is ning ,lucky number is 99
```

2 KEYWORD ARGUMENTS

- This allows you to **skip arguments or place them out of order** because the Python interpreter is able to use the keywords provided to match the values with parameters. You can also make keyword calls to the printinfo() function in the following ways –

```
def printinfo( name, age ):
   print ("Name: ", name)
   print ("Age: ", age)

printinfo(age=50, name="ning")
```

```
Name: ning
Age: 50
```

3 DEFAULT ARGUMENTS

- A default argument is an **argument that assumes a default value if a value is not provided** in the function call for that argument. The following example gives an idea on default arguments, it prints default age if it is not passed –

```
def printinfo(name, age = 35):
   print ("Name: ", name)
   print ("Age ", age)
printinfo(age=50, name="nung-ning")
printinfo(name="sirinthorn")
```

```
Name:  nung-ning
Age  50
Name:  sirinthorn
Age  35
```

4 VARIABLE-LENGTH ARGUMENTS

In Python, we can pass a variable number of arguments to a function using special symbols. There are two special symbols:

- ***args (Non-Keyword Arguments)**

```
def myFun(*argv):
    for item in argv:
        print(item)

myFun('Hello', 'Welcome', 'to', 'GeeksforGeeks')
```

```
Hello
Welcome
to
Python
```

https://www.geeksforgeeks.org/python-functions/

CLASS ACTIVITY OF WEEK12

จงเขียนโปรแกรมแยกเป็นฟังก์ชันคำนวณเกรดของนักศึกษาที่มาจากคะแนน 3 ส่วน ได้แก่ คะแนน midterm, final และ assignment ดังตัวอย่าง

```
Enter midterm score: 30
Enter final score: 30
Enter assignment score: 5
Total score : 65.0
Grade : C
```

CLASS ACTIVITY OF WEEK12

```
Enter midterm score: 30
Enter final score: 30
Enter assignment score: 5
Total score : 65.0
Grade : C
```

Get Input function

Calculate total score function

Calculate grade function

Display output function

HOMEWORK OF WEEK12

จงเขียนโปรแกรมแยกเป็นฟังก์ชันคำนวณพื้นที่รูปทรงจำนวน 3 รูปทรง คือ

1. วงกลม
2. สามเหลี่ยม
3. สี่เหลี่ยม

กำหนดให้นักศึกษาสร้างฟังก์ชันจำนวนตามความเหมาะสม ให้นักศึกษากำหนดชื่อฟังก์ชันและพารามิเตอร์ที่เกี่ยวข้องเอง

4. โดยกำหนดให้ผู้ใช้กรอกรหัสเมนูพื้นที่ที่ต้องการคำนวณผ่านแป้นพิมพ์ ดังภาพตัวอย่างในหน้าถัดไป

RESULT AFTER SELECT C MENU

```
-----------------------------------------------------------
                          Menu
-----------------------------------------------------------
 C : Circle Area
 T : Triangle Area
 R : Rectangle Area
 E : Exit
-----------------------------------------------------------
Enter menu : c
Enter radius : 10
===========================================================
                   Output of Circle
-----------------------------------------------------------

              Area of Circle = 314.29
===========================================================
```

RESULT AFTER SELECT R MENU

```
-----------------------------------------------------------
                          Menu
-----------------------------------------------------------
 C : Circle Area
 T : Triangle Area
 R : Rectangle Area
 E : Exit
-----------------------------------------------------------
Enter menu : R
Enter width : 10
Enter length : 10
===========================================================
                Output of Rectangle
-----------------------------------------------------------

            Area of Rectangle = 100.00
===========================================================
```

RESULT AFTER SELECT T MENU

```
-----------------------------------------------------------
                          Menu
-----------------------------------------------------------
 C : Circle Area
 T : Triangle Area
 R : Rectangle Area
 E : Exit
-----------------------------------------------------------
Enter menu : t
Enter base : 10
Enter height : 10
===========================================================
                  Output of Triangle
-----------------------------------------------------------

              Area of Triangle = 50.00
===========================================================
```

RESULT AFTER SELECT INVALID MENU

```
-----------------------------------------------------
                          Menu
-----------------------------------------------------
 C : Circle Area
 T : Triangle Area
 R : Rectangle Area
 E : Exit
-----------------------------------------------------
Enter menu : f
ERROR!!!
-----------------------------------------------------
                          Menu
-----------------------------------------------------
 C : Circle Area
 T : Triangle Area
 R : Rectangle Area
 E : Exit
-----------------------------------------------------
Enter menu : g
ERROR!!!
-----------------------------------------------------
```

THANK YOU
Q/A

38