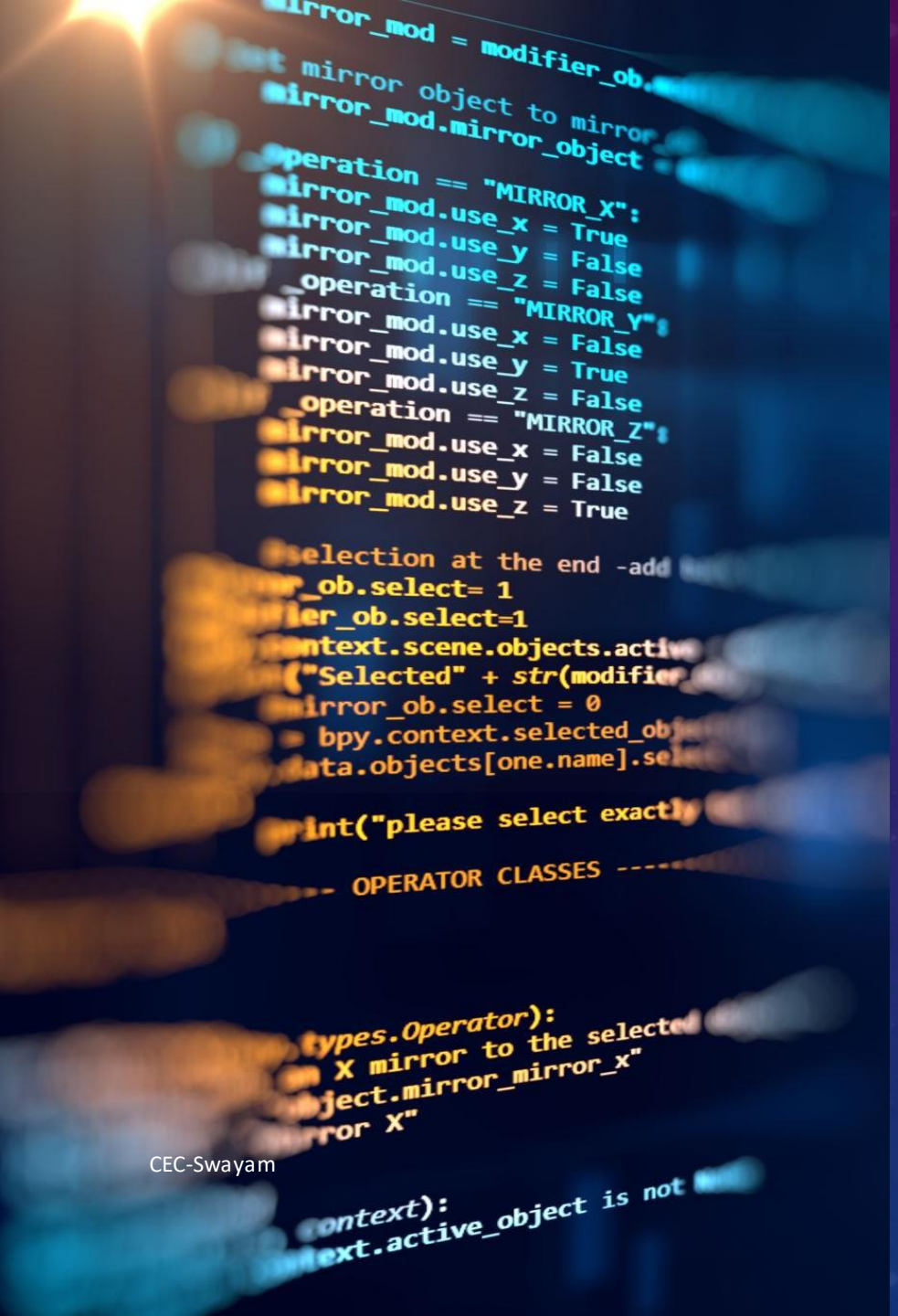
CEC-Swayam / EMRC Dibrugarh University
Course- Programming in Python
Week-9

FUNCTIONS

PYTHON PROGRAMMING

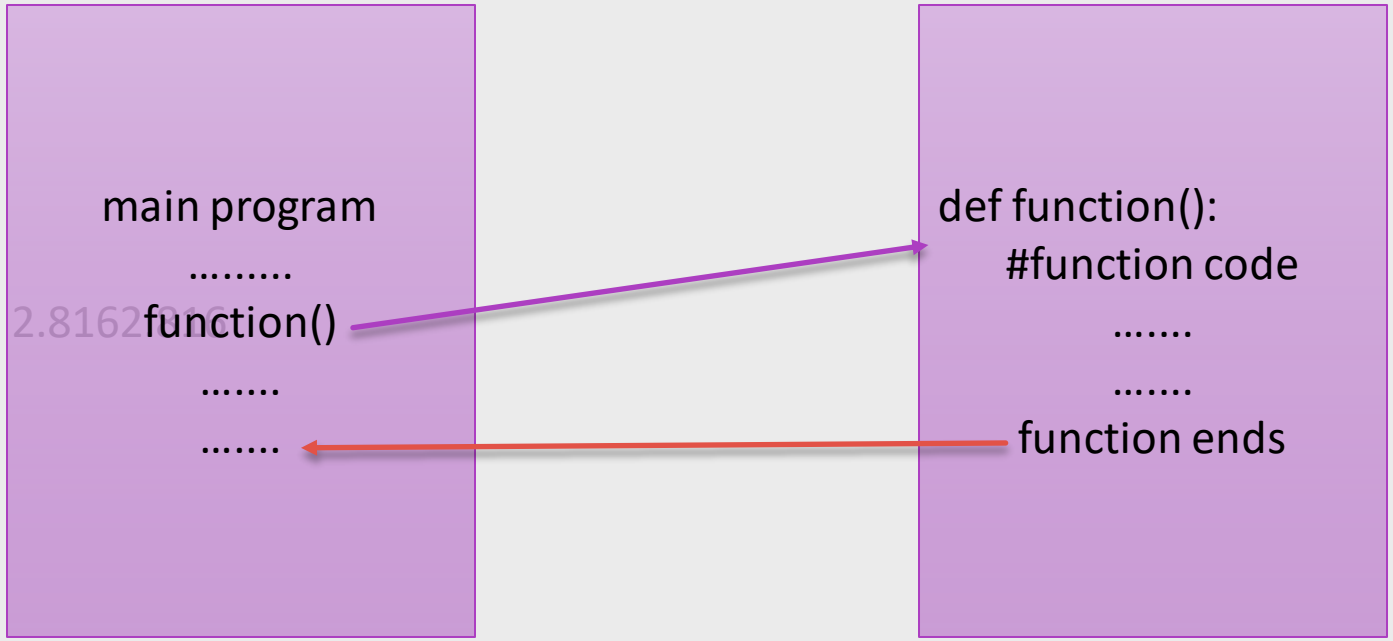


DEFINITION

- A block of program code which can be resued. It is an organized block of code that can perform a single, specific, and a well-defined task

FUNCTION CALLING

- 2.8162816



WHY FUNCTION ?

Code Reusability

Modularity

Easy Debugging

Less Development Time

SYNTAX OF FUNCTION DEFINITION

Function header

parameters

```
def function_name(variable1, variable2,…)
    documentation string
    statement block
    return[expression]
```

Function body

EXAMPLE

```
def function():
    print("Hello World")

function()   #function call
```

FUNCTION PARAMETERS

- The name of the function while calling and the number of arguments must match with the function definition.
- If there is a mismatched in parameters passed and declared, then an error will be returned.
- If the data type of the parameters passed doenot match with that of the function, then an error is generated.

TYPES OF FUNCTIONS IN PYTHON

- Built-in functions
- User-Defined Functions (UDFs)
- Anonymous functions (lambda functions)

LAMBDA OR ANONYMOUS FUNCTION

Lambda functions are not declared using *def* keyword. Instead, *lambda* keyword is used.

Lambda functions are required at place where they have been created in the program.

Any number of arguments ca be supplied to lambda function, but it must a single expression.

SYNTAX

lambda *arguments* :
expression

```
z = lambda x, b : x* b
print(z(7, 18))
```

```
x = lambda m, n, c: m + n + c
print(x(6, 2, 7))
```

POINTS

- Lambda functions have no name.
- Lambda function cannot access variables other than those in their parameter list.
- Lambda function can take N number of arguments.
- Lambda function doesnot have any return statement.

FUNCTION ARGUMENTS

- You can call a function by using the following types of formal arguments –
- Required arguments
- Keyword arguments
- Default arguments
- Variable-length arguments

REQUIRED ARGUMENTS

Arguments must be passed on to a function in correct order.

The number of arguments passed to a function must match with the number with the number of arguments specified in the function definition.

KEYWORD ARGUMENTS

Keyword argument when used helps to identify the arguments by the name of the parameter.

The order of the keyword argument is not important.

The keyword arguments passed must match with one of the arguments of the accepting function.

Keyword arguments make the program code less complex and easy to understand.

DEFAULT ARGUMENTS

- Default argument allow to specify a value for a parameter.
- This allows to call a function with less numeber of arguments than defined.
- Any number of default arguments can be defined.
- Non default argument cannot be followed by the default argument.

VARIABLE LENGTH ARGUMENTS

In cases where it is not known in prior how many arguments will be passed to a function python allows to make function call with arbitrary number of arguments.

An asterisk (*) symbol is used before the parameter name .

VARIABLE SCOPE AND LIFETIME

Scope of a variable is defined by the part of the program where a variable is accessible.

The time for which a varible exist is called its lifetime.

GLOBAL AND LOCAL VARIABLE

- Global variables are defined in the main body of the program and can be used throughout the program. They are also accesible by all the function in the program.
- Local Variables are defined within the function and their scope is within that function only. They are not related with the same name variable defined outside the function.