SNS COLLEGE OF ENGINEERING

Kurumbapalayam (PO), Coimbatore – 641 107
Accredited by NAAC-UGC with 'A' Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

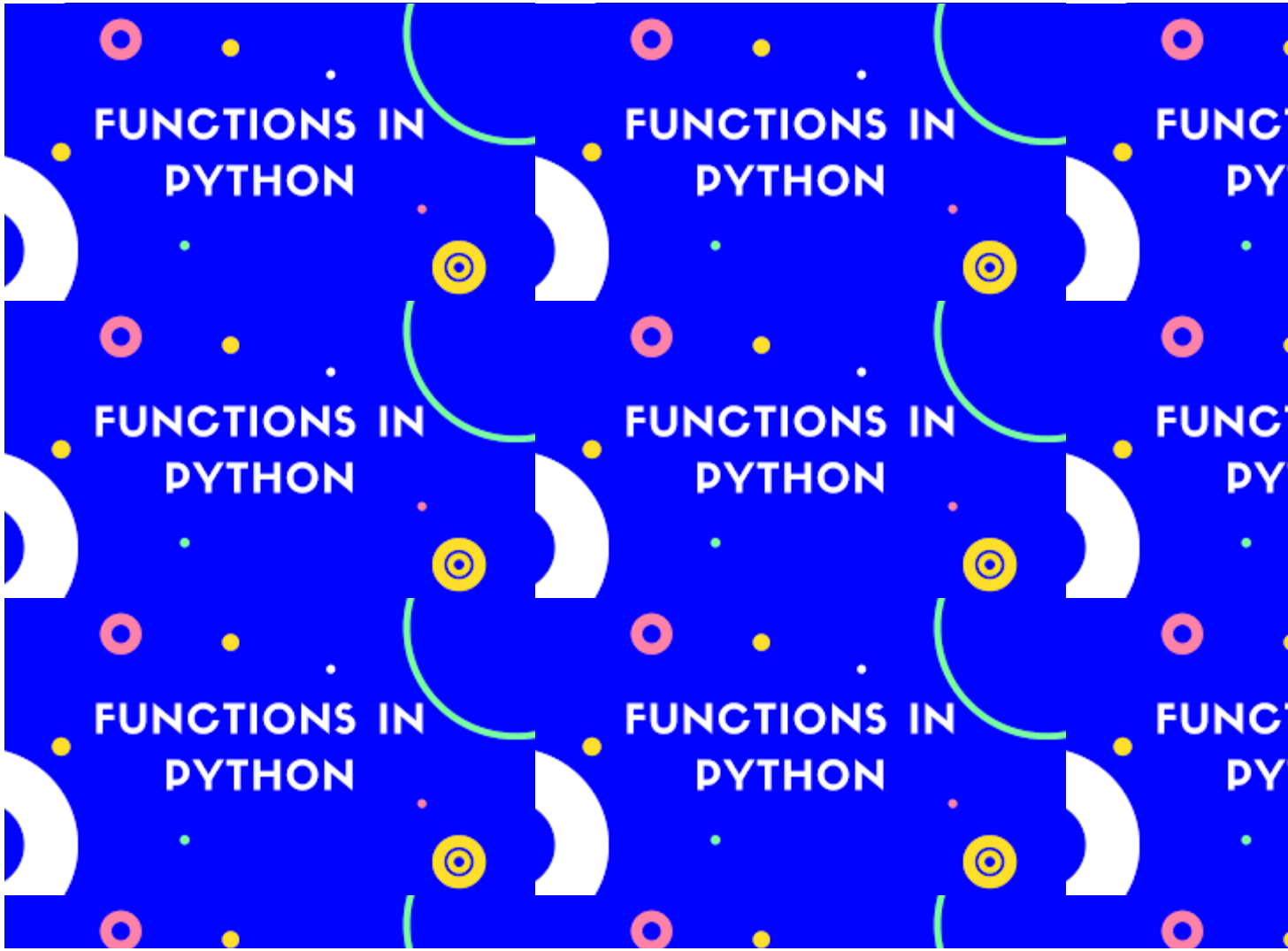
DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE NAME: 23ITB202-PYTHON PROGRAMMING

II YEAR/ III SEM

Unit : MEMORY SYSTEM

Topic : Semiconductor RAM memories


FUNCTIONS IN PYTHON
FUNCTIONS IN PYTHON
FUNC PY
FUNCTIONS IN PYTHON
FUNCTIONS IN PYTHON
FUNC PY
FUNCTIONS IN PYTHON
FUNCTIONS IN PYTHON
FUNC PY

FUNCTIONS

- A function is a programming block of codes which is used to perform a single, related task. It only runs when it is called.
- We can pass data, known as parameters, into a function.
- A function can return data as a result.
- We have already used some python built in functions like print().
- We can also create our own functions. These functions are called user-defined functions.

TYPES OF FUNCTIONS

- 1. LIBRARY FUNCTIONS→They are pre-defined, inbuilt functions, used as it. For eg: sqrt(81)→9
- 2. USER DEFINED FUNCTIONS→ They are defined by user or programmer according to the requirement. For eg: def square(a):
  - Print (a*a)

FUNCTION DEFINITION

- A function is defined using the def keyword in
- python.
- E.g. program is given below.

```
def abc():
    print("Hello")
```

- abc() #function calling.
- Hello
- Save the above source code in python file and
- execute it

- **Variable's Scope in function**

There are three types of variables with the view of scope.

- 1. Local variable – accessible only inside the functional block where it is declared.
- 2. Global variable – variable which is accessible among whole program using global keyword.
- 3. Non local variable – accessible in nesting of functions,using nonlocal keyword.

Local variable:

- def fun():
- s = "I love India!" print(s)
- s = "I love World!"
- fun()
- print(s)
- Output:
- I love India!
- I love World!

Global variable :

```
def fun():
global s
fun()
print(s)
s = "I love India!“
print(s)
s = "I love world!"
fun()
print(s)
```

Output:

I love world!
I love India!
I love India!

Variable's Scope in function

```
#Find the output of below program
def fun(x, y):
# argument /parameter x and y
global a
a = 10
x,y = y,x
b = 20
b = 30
c = 30
print(a,b,x,y)
a, b, x, y = 1, 2, 3,4
fun(50, 100)
 #passing value 50 and 100 in parameter x and y of function fun()
 print(a, b, x, y)
```

Variable's Scope in function

```
#Find the output of below program
def fun(x, y):
global a
a = 10
x,y = y,x
b = 20
b = 30
c = 30
print(a,b,x,y)

a, b, x, y = 1, 2, 3,4
fun(50, 100)
print(a, b, x, y)
```

OUTPUT :-

10 30 100 50

10 2 3 4

Visit

Variable's Scope in function

Global variables in nested function

```
def fun1():
    x = 100
    def fun2():
        global x
        x = 200
    print("Before calling fun2: " + str(x))
    print("Calling fun2 now:")
    fun2()
    print("After calling fun2: " + str(x))

fun1()
print("x in main: " + str(x))
```

OUTPUT: Before calling fun2: 100
Calling fun2
now: After calling fun2: 100
x in main: 200

Parameters / Arguments

These are specified after the function name, inside the parentheses. Multiple parameters are separated by comma.

The following example has a function with two parameters x and y. When the function is called, we pass two values, which is used inside the function to sum up the values and store in z and then return the result

```
def sum(x,y): #x, y are formal arguments
z=x+y
return z #return the result
x,y=4,5
r=sum(x,y) #x, y are actual arguments
print(r)
```

Note :- 1. Function Prototype is declaration of function with name ,argument and return type.

2. A formal parameter, i.e. a parameter, is in the function definition. An actual parameter, i.e. an argument, is in a function call.

Functions using libraries:

Mathematical functions: Mathematical functions are available under math module.To use mathematical functions under this module, we have to import the

module using import math.

For e.g.

To use sqrt() function we have to write statements like given below.

```
import math
r=math.sqrt(4)
print(r)
```

OUTPUT :

2.0

MATH FUNCTIONS:

Function	Description	Example
ceil(n)	It returns the smallest integer greater than or equal to n.	math.ceil(4.2) returns 5
factorial(n)	It returns the factorial of value n	math.factorial(4) returns 24
floor(n)	It returns the largest integer less than or equal to n	math.floor(4.2) returns 4
fmod(x, y)	It returns the remainder when n is divided by y	math.fmod(10.5,2) returns 0.5
exp(n)	It returns e**n	math.exp(1) return 2.718281828459045
log2(n)	It returns the base-2 logarithm of n	math.log2(4) return 2.0
log10(n)	It returns the base-10 logarithm of n	math.log10(4) returns 0.6020599913279624
pow(n, y)	It returns n raised to the power y	math.pow(2,3) returns 8.0
sqrt(n)	It returns the square root of n	math.sqrt(100) returns 10.0
cos(n)	It returns the cosine of n	math.cos(100) returns 0.8623188722876839
sin(n)	It returns the sine of n	math.sin(100) returns -0.5063656411097588
tan(n)	It returns the tangent of n	math.tan(100) returns -0.5872139151569291
pi	It is pi value (3.14159...)	It is (3.14159...)
e	It is mathematical constant e (2.71828...)	It is (2.71828...)