1.In Python, what is the difference between a built-in function and a user-defined function? Provide an example of each.

Answer:

Python libraries contain built-in functions that are already pre-defined and you can access it by simply calling out the function and you can also define your function to your wish these are called user-defined functions.

Pre-defined function:

list1=[1,2,3,4,5,6,7]

print(len(list1))

Output:

7

Here, len() is a built-in function.

User-defined function:

def add(x,y):

print(x+y)

x=4

y=5

add(x,y)

Output:

9

Here, we created user-defined function add().

2. How can you pass arguments to a function in Python? Explain the difference between positional

arguments and keyword arguments.

Answer:

Positional arguments:

Arguments that are passed in order of parameters. The order that is defined in the order declaration function.

Keyword arguments:

Parameter names are used to pass the argument in the function call.

def greet(name):

print("welcome! have a great day",name)

name="rihan"

greet(name)

Output:

welcome! have a great day rihan

3. What is the purpose of the return statement in a function? Can a function have multiple return

statements? Explain with an example.

Answer:

The return statement stops the execution of the function and returns function to calling function.

We can have multiple return functions in python.

4. What are lambda functions in Python? How are they different from regular functions? Provide an

example where a lambda function can be useful.

Answer:

Lambda is a keyword used in python. Lambda functions are functions without name.

x=lambda a:a+10

print(x(5))

Output:

15

5. How does the concept of "scope" apply to functions in Python? Explain the difference between local scope and global scope.

Answer:

Local variables – local variables are declared inside a function and can be accessed only inside that function.

Global variables – global variables can be declared in two ways. Using a global keyword or by declaring outside the function. These can be accessed anywhere in the program.

x=100

def division():

y=50

print(x/y)

division()

Output:

2.0

Here, x is a global variable that can be accessed anywhere in the program and the local variable y can only be accessed inside the function.

6. How can you use the "return" statement in a Python function to return multiple values?

Answer:

We can return multiple values by a separating them with a comma.

def fun():

str = "hello world"

x = 20

return str, x

str, x

fun()

Output:

('hello world', 20)

7. What is the difference between the "pass by value" and "pass by reference" concepts when it

comes to function arguments in Python?

Answer:

Pass by value:

student = {'tom': 12, 'Anna': 14, 'Preethi': 10}

def test(student):

student = {'Sam':20, 'Steve':21}

print("Inside function", student)

return

test(student)

print("Outside function:", student)

Output:

Inside function {'Sam': 20, 'Steve': 21}

Outside function: {'tom': 12, 'Anna': 14, 'Preethi': 10}

Pass by reference:

def marks(list):

list.append([11, 12, 13, 14, 15])

print("Value inside the function: ", list)

return

list = [10,20]

marks(list)

print("Value outside the function: ", list)

Output:

Value inside the function: [10, 20, [11, 12, 13, 14, 15]]

Value outside the function: [10, 20, [11, 12, 13, 14, 15]]

8. Create a function that can intake integer or decimal value and do following operations:

a. Logarithmic function (log x)

import math

print("logarithm of 14 is : ", end="")

print(math.log(14))

Output:

logarithm of 14 is : 2.6390573296152584

b. Exponential function (exp(x))

import math

x=23.5

print("exponent of 23.5 is:",end="")

print(math.exp(23.5))

Output:

exponent of 23.5 is:16066464720.622478

c. Power function with base 2 (2\*)

base = 2

exponent = 2

result = pow(base, exponent)

print("Answer:",end="")

print(result)

d. Square root

import math

print(math.sqrt(2))

Output:

1.4142135623730951

9. Create a function that takes a full name as an argument and returns first name and last name.

Answer:

full\_name=" riya thomson"

full\_name.split()

first\_name=full\_name.split()[0]

last\_name=full\_name.split()[-1]

print("first name is",first\_name)

print("last name is",last\_name)

Output:

first name is riya

last name is thomson