1. A built-in function in Python is a function that is pre-defined and provided by the Python language itself. Examples of built-in functions include print(), len(), and type(). These functions are available for use without requiring any additional code or definition. For example, print(“Hello World”) here print() is a built in function.

A user-defined function is created by the user to perform a specific task or set of instructions. These functions are defined using the def keyword followed by a function name, parentheses for parameters (if any), and a colon. For example,

def add(a,b):

print(a+b)

add(4,5)

The answer would be 9.

2. Arguments can be passed to a function by providing them inside the parentheses when calling the function. There are two types of arguments: positional arguments and keyword arguments.

Positional arguments are passed based on their position and order in the function call. The values are assigned to the parameters in the function definition based on their position.

Keyword arguments, on the other hand, are passed with the parameter name followed by the value. This allows specifying the values for specific parameters, regardless of their position.

3. The return statement in a function is used to specify the value that should be returned from the function when it is called. It is used to send a value back to the user. A function can have multiple return statements, but only one will be executed when the function is called.

The purpose of the return statement is to provide the output or result of the function's computation to the user, allowing the user to use that result in further calculations or operations.

def multiply(a, b):

return a \* b

result = multiply(3, 4)

print(result)

The result would be 12.

4. Lambdas are anonymous functions; these functions can be defined without a function name.

Lambda functions are different from regular functions in that they don't require a formal def statement, function name, or return statement. They are commonly used when a small function is needed as an argument for another function, such as in map(), filter(), or reduce().

This reduces the size of the program.

sq= lambda x: x\*\*2

print(sq(3))

This results in 9.

5. In Python, the concept of "scope" refers to the visibility and accessibility of variables in different parts of the code. When it comes to functions, there are two main types of scope: local scope and global scope.

Local scope refers to variables that are defined within a specific function. These variables are only accessible within the function, and they cannot be accessed or modified outside of the function.

Global scope, on the other hand, refers to variables that are defined outside of any function or within the global scope itself. These variables are accessible from any part of the code, including inside functions.

6. To return multiple values from a Python function, we can use the return statement with comma-separated values. The values will be returned as a tuple.

def get\_full\_name():

first\_name = "John"

last\_name = "Smith"

return first\_name, last\_name

result = get\_full\_name()

print(result) # Output: ('John', 'Smith')

print(result[0]) # Output: John

print(result[1]) # Output: Smith

7. Function arguments are passed by object reference. It means that when a variable is passed as an argument to a function, a reference to the object is passed. The behavior of argument passing depends on whether the object is mutable or immutable.

For immutable objects like integers, strings, and tuples, they are passed by value. It means that any changes made to the parameter within the function will not affect the original object.

For mutable objects like lists and dictionaries, they are passed by reference. It means that any changes made to the parameter within the function will affect the original object.

8.

import math

def math\_operations(num):

logarithm = math.log(num)

exponential = math.exp(num)

power\_of\_two = math.pow(2,num)

square\_root = math.sqrt(num)

return logarithm, exponential, power\_of\_two, square\_root

num=int(input(“Enter a number:” ))

result = math\_operations(num)

print(result)

9.

def get\_first\_and\_last\_name():

full\_name = input("Enter your full name: ")

names = full\_name.split()

first\_name = names[0]

last\_name = names[-1]

return first\_name, last\_name

result = get\_first\_and\_last\_name()

print(result)