1. Why are functions advantageous to have in your programs?

ANS - Functions are advantageous to have in your programs because they help reduce the duplication of code. Instead of writing the same code again and again for different inputs, we can do the function calls to reuse code contained in it over and over again. This increases code reusability and readability.

Functions also help write clearer code. By breaking down a program into smaller functions, it becomes easier to understand and debug .

2. When does the code in a function run: when it's specified or when it's called?

ANS - The code in a function runs when it is called. A function is a block of code that only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result.

3. What statement creates a function?

ANS- The statement that creates a function in Python is the def keyword.

4. What is the difference between a function and a function call?

ANS- A function is a block of code that only runs when it is called. It is defined using the def keyword.

A function call is what we use to tell Python to execute the code inside the function.

5. How many global scopes are there in a Python program? How many local scopes?

ANS- There is only one global scope in a Python program.

Local scopes are created whenever a function is called. There can be as many local scopes as there are function calls.

6. What happens to variables in a local scope when the function call returns?

ANS- When a function call returns, the local scope is destroyed and all the variables in it are forgotten.

7. What is the concept of a return value? Is it possible to have a return value in an expression?

ANS- The concept of a return value is that it is the value that a function returns to the caller.

Yes, it is possible to have a return value in an expression.

8. If a function does not have a return statement, what is the return value of a call to that function?

ANS- If a function does not have a return statement, the return value of a call to that function is None.

9. How do you make a function variable refer to the global variable?

ANS - To make a function variable refer to the global variable in Python, you can use the global keyword.

Here’s an example:

x = 10

def my\_function():

global x

x = 5

my\_function()

print(x)

This will output 5.

10. What is the data type of None?

ANS- The data type of None is NoneType .

11. What does the sentence import areallyourpetsnamederic do?

ANS- The sentence import areallyourpetsnamederic does not do anything useful in Python. It imports a module named areallyourpetsnamederic, but this is not a real Python module.

12. If you had a bacon() feature in a spam module, what would you call it after importing spam?

ANS-If you had a bacon() feature in a spam module, you would call it using the following code after importing spam:

import spam

spam.bacon()

13. What can you do to save a programme from crashing if it encounters an error?

ANS- To save a program from crashing if it encounters an error, you can use a try and except block .

example:

try:

# some code here

except:

# code to handle the error here

14. What is the purpose of the try clause? What is the purpose of the except clause?

ANS- The purpose of the try clause is to test a block of code for errors .

The purpose of the except clause is to handle the error if the try clause encounters one .