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

Ans.Functions reduce the need for duplicate code. This makes programs shorter, easier to read, and easier to update.It’s frequently difficult to read a large program. Breaking the code down into smaller functions keeps the program structured, understandable, and reusable. The function can be reused countless times after it is defined.

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

Ans.The code in a function executes when the function is called, not when the function is defined.

3. What statement creates a function?

Ans.To create a function, we must first declare it and give it a name, the same way we’d create any variable, and then we follow it by a function definition: var sayHello = function() { }; We could put any code inside that function – one statement, multiple statements.

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

Ans.A function is a piece of code which enhanced the reusability and modularity of your program. It means that piece of code need not be written again. A function call means invoking or calling that function. Unless a function is called there is no use of that function.

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

Ans.There’s only one global Python scope per program execution. This scope remains in existence until the program terminates and all its names are forgotten. Otherwise, the next time you were to run the program, the names would remember their values from the previous run.

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

Ans.Each call of the function creates new local variables, and their lifetimes expire when the function returns to the caller.

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

Ans.A return is a value that a function returns to the calling script or function when it completes its task. A return value can be any one of the four variable types: handle, integer, object, or string. The type of value your function returns depends largely on the task it performs.

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

Ans.Every function in Python returns something. If the function doesn't have any return statement, then it returns None .

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

Ans.When you create a variable inside a function, that variable is local, and can only be used inside that function. To create a global variable inside a function, you can use the global keyword.

10. What is the data type of None?

Ans.The None keyword is used to define a null value, or no value at all. None is not the same as 0, False, or an empty string. None is a data type of its own NoneType and only None can be None.

11. What does the sentence import areallyourpetsnamederic do?

Ans.That import statement imports a module named areallyourpetsnamederic.

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

Ans.This function can be called with spam. Bacon().

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

Ans.If an error occurs in a program, we don't want the program to unexpectedly crash on the user. Instead, error handling can be used to notify the user of why the error occurred and gracefully exit the process that caused the error.

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

Ans.The try block lets you test a block of code for errors. The except block lets you handle the error. The else block lets you execute code when there is no error. The finally block lets you execute code, regardless of the result of the try- and except blocks.