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

Ans-In Python, a function is a group of related statements that performs a specific task. With the help of Functions, we can break our program into smaller and modular chunks. So, when our program grows larger and larger, functions help us to make it more organized and manageable.

2. When does the code in a function run: when it&#39;s specified or when it&#39;s called? A function is a block of code which 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 def statement defines (that is, creates) a function. What is the difference between a function and a function call? A function consists of the def statement and the code in its def clause.

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

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?

Python Global variables are those which are not defined inside any function and have a global scope whereas Python local variables are those which are defined inside a function and their scope is limited to that function only.

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

The variable y only exists while the function is being executed — we call this its lifetime. When the execution of the function terminates (returns), the local variables are destroyed. Codelens helps you visualize this because the local variables disappear after the function returns.

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

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.

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

If no return statement appears in a function definition, control automatically returns to the calling function after the last statement of the called function is executed. In this case, the return value of the called function is undefined.

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

Normally, 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?

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?

What does the import areallyourpetsnamederic statement do? That import statement imports a module named areallyourpetsnamederic. (This isn't a real Python module, by the way.)

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

This function can be called with spam.bacon().

13. What can you do to save a programme from crashing if it encounters an error?It's important to handle exceptions properly in your code using try-except blocks or other error-handling techniques, in order to gracefully handle errors and prevent the program from crashing.1

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

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.