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

ANS Functions therefore aid us in better structuring and managing our programme as it grows. With the help of functions, we can also get rid of duplication and broaden the scope of our code. Essentially, there are two types of functions: User-defined functions – The user defines these functions to carry out a certain task.

1. When does the code in a function run: when it&#39;s specified or when it&#39;s called?

ANS When does the code in a function execute: when the function is defined or when the function is called? The code in a function executes when the function is called, not when the function is defined.

1. 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 - depends on what we want to do.

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

ANS a function is a block of code that performs a specific task, while a function call is an expression that invokes a function and executes the code in the function.

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

ANS At any given time during execution, you'll have at most four active Python scopes—local, enclosing, global, and built-in—depending on where you are in the code. On the other hand, you'll always have at least two active scopes, which are the global and built-in scopes.

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

ANS 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.

1. 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.

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

ANS Regardless of how long and complex your functions are, any function without an explicit return statement, or one with a return statement without a return value, will return None .

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

ANS 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.

1. 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.

1. What does the sentence import areallyourpetsnamederic do?

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

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

ANS If you had a function named bacon() inside a module named spam, how would you call it after importing spam? This function can be called with spam. bacon().

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

ANS Try blocks can help programmers to categorize exception objects. The try bracket contains the code that encounters the exception and prevents the application from crashing.

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

ANS A Try-Except statement is a code block that allows your program to take alternative actions in case an error occurs. Python will first attempt to execute the code in the try statement (code block 1). If no exception occurs, the except statement is skipped and the execution of the try statement is finished.

FINISH\*\*\*\*