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

Ans. With the help of functions, we can avoid rewriting the same logic or code again and again in a program.In a single Program, we can call Python functions anywhere and also call multiple times.We can track a large Python program easily when it is divided into multiple functions.The main achievement of Python functions is its Reusability.However, In a Python program, Function calling is always overhead.

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

Ans. When it’s called.

1. What statement creates a function?

Ans. To define the function in Python, it provides the def keyword

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

Ans. A function is a block of code that does a particular operation and returns a result. It usually accepts inputs as parameters and returns a result. The parameters are not mandatory.  
  
E.g:  
def add(a,b):  
 return a+ b  
  
A function call is the code used to pass control to a function.  
  
E.g.:  
  
b = add(5,6)  
  
Now b will have the value 11.

1. 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.

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. In Python language, those local variables are gone forever.The memory that they occupied is then given to other variables.

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

Ans. The Python return statement is a special statement that you can use inside a function or [method](https://realpython.com/python3-object-oriented-programming/" \l "instance-methods) to send the function’s result back to the caller. A return statement consists of the [return keyword](https://realpython.com/python-keywords/" \l "returning-keywords-return-yield) followed by an optional return value.

Like any value, a return value can be used as part of an expression.

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

Ans. Without return statement it returns nothing .

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

def myfunc():  
   global x  
  x = "fantastic"  
 myfunc()  
  
 print("Python is " + x)

1. What is the data type of None?

Ans. 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. spam.bacon()

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

Ans. Place the line of code that might cause an error in a try clause.

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

Ans. The code that could potentially cause an error goes in the try clause.The code that executes if

an error happens goes in the except clause.