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

* Functions are techniques used to combine a set of statements within a program.
* They prove to be a useful tool when the operations are coded in it and can be used in a variety of scenarios.
* They form an alternative method of cutting-and-pasting codes rather than typing redundant copies of the same instruction or operation, which further reduces the future work
* To sum up it can be said that its technique for code re-use.

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

* The code in the function runs when it is called

Eg:

def greetings():

"""This function displays 'Hello World!'"""

print('Hello World!')

* The greetings function is created with a message to be printed in it. Function is just defined over here. It doesn’t return a value unless being called.
* To call a function, it needs to be specified with a parentheses like **greetings()**

To run the function

3. What statement creates a function?

To define a function, Python provides the **def**keyword followed by a key word and a colon. The “def” call creates the function object and assigns it to the name given.

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

* Using a function is to do a particular task at any point in program. That is called as function call.
* A function is procedure to achieve a particular result while function call is using this function to achieve that task.

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

**Global scope** once defined remains in the entire program, until the program ends. Whereas **local scope** is restricted only in the function its been defined into.

x = "Global Variable"

def func():

print(x)

s = "local variable"

print(s)

func()

print(x)

‘x’ can be called even after the function is been closed. It scope remains in the entire program, whereas ‘s’ scope remains restricted only in the defined function and cant be called outside the function

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

* A local variable retains its value until the next time.
* The function is called a local variable becomes undefined
* After the function call completes

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

* A return statement is used to end the execution of the function call and “returns” the result once called.
* The statements after the return statements are not executed.
* If the return statement is without any expression, then the special value None is returned.

8. If a function does not have a return statement, what is the return value of a call to that function?  
If a function doesn't specify a return value, it returns None.

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

To refer a global variable in a function, we can use the **global** keyword to declare which variables are global inside a function variable.

10. What is the data type of None?

* None is a data type of its own (NoneType) and only None can be None.
* The None keyword is used to define a null value, or no value at all.

11. What does the sentence import areallyourpetsnamederic do?

That import statement imports a module named areallyourpetsnamederic.

(This isn’t a real Python module)

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?

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

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

* 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.