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

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

3. What statement creates a function?

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

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

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

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

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

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

10. What is the data type of None?

11. What does the sentence import areallyourpetsnamederic do?

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

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

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

1. Functions are advantageous to have in programs because they allow you to break up large, complex programs into smaller, more manageable pieces. Functions can also make code more readable, reusable, and easier to debug.
2. The code in a function runs when the function is called.
3. The **def** statement creates a function.
4. A function is a block of code that performs a specific task, while a function call is the code used to invoke that function and pass any required arguments.
5. There is one global scope in a Python program, which can be accessed from anywhere in the program. Local scopes are created whenever a function is called, and are destroyed when the function returns.
6. Variables in a local scope are destroyed when the function call returns.
7. A return value is the value that a function returns to the caller. It is possible to use a return value in an expression.
8. If a function does not have a return statement, the return value of a call to that function is **None**.
9. To make a function variable refer to the global variable, you can use the **global** keyword. For example:

x = 10

def func():

global x

x = 5

func()

print(x) # prints 5

1. The data type of **None** is **NoneType**.
2. The sentence **import areallyourpetsnamederic** is a syntax error. It is not a valid Python import statement.
3. If you had a **bacon()** feature in a **spam** module, you could call it after importing **spam** using the following syntax:

import spam

spam.bacon()

1. You can use error handling techniques, such as **try** and **except** statements, to save a program from crashing if it encounters an error.
2. The purpose of the **try** clause is to enclose code that may raise an exception. The purpose of the **except** clause is to handle the exception if it is raised.

Top of Form