1. Advantage of Functions in Programs:

Modularity: They allow you to break down complex tasks into smaller, more manageable sub-tasks.

Reusability: Once a function is defined, it can be used multiple times throughout the program without rewriting the same code.

Maintainability: If there's a bug or an improvement to be made, you only have to change the code in one place (within the function), rather than everywhere the code is used.

2. Code Execution in a Function:

The code in a function runs when the function is called, not when the function is specified.

3. Statement to Create a Function:

The def keyword is used to declare or create a function in Python. For example: def functionName():.

4. Difference Between a Function and a Function Call:

A function is a block of organized, reusable code that performs a specific task. A function call is the code used to pass control to a function.

5. Global and Local Scopes:

There's one global scope in a Python program. A new local scope is created whenever a function is called.

6. Variables in Local Scope:

When the function call returns, the variables in the local scope are destroyed, and their memory is freed up.

7. Concept of a Return Value:

A return value is the value that a function gives back (or "returns") to the part of the program that called it. Yes, a return value can be used in an expression.

8. Return Value without a Return Statement:

If a function does not have a return statement, the return value of a call to that function is None.

9. Refer to the Global Variable in a Function:

You can use the global keyword before the variable name to refer to a global variable. For example:

def someFunction():

global someVar

someVar = "someValue"

10. Data Type of None:

The data type of None is NoneType.

11. Sentence import areallyourpetsnamederic:

This sentence attempts to import a module named areallyourpetsnamederic. The module must exist and be discoverable by Python for this to work.

12. Calling bacon() in spam Module:

After importing the spam module, you'd call the bacon() function using: spam.bacon().

13. Preventing Program from Crashing:

You can prevent a program from crashing by handling exceptions and errors gracefully using try-except blocks.

14. Purpose of try and except Clauses:

Try Clause: The code inside the try block is the code that might raise an exception or cause an error. Python tries to run this code.

Except Clause: If the code inside the try block causes an error, the code inside the except block will be executed. It's where you handle the exception or provide a fallback.

For example:

try:

# code that might raise an exception

result = 10 / 0

except ZeroDivisionError:

# code to handle the exception

print("You can't divide by zero!")