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

Ans.) Various advantage of function to have in programs:

1. **Modularity and Reusability:** Functions allow you to break down your program into smaller, modular pieces of code. Functions also promote code reuse, as you can call the same function from different parts of your program without duplicating code. This modularity and reusability make programs more maintainable and scalable.
2. **Code Organization and Readability:** Functions help in organizing code by grouping related instructions together. By dividing a program into functions, you can enhance its readability and understandability.
3. **Testing and Modifiability:** Functions facilitate unit testing, as you can test each function independently with different inputs and expected outputs. This helps in identifying issues and validating the correctness of individual components. If you need to change the behaviour of a certain functionality, you can modify the corresponding function without affecting other parts of the program, as long as the function's interface remains unchanged.

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

**Ans.)** The code in a function executes when the function is called, not when the function is defined.

3. **What statement creates a function?**

**Ans.)** The **def** statement is used to creates a function.

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

**Ans.)**

|  |  |
| --- | --- |
| **Function** | **Function Call** |
| A function is a block of code that performs a specific task or set of tasks. | A function call is the act of executing a function to perform its specified task. |
| eg.) def add\_numbers(a, b):  return a + b | eg.) result = add\_numbers(5, 3)  print(result) # Output: 8 |

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

**Ans.)** Python programs have one global scope, accessible throughout the program. Each time a function is called or a code block is entered (such as within a loop or conditional statement), a new local scope is created.

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

**Ans.)** When a function call returns in Python, the local scope created during the execution of that function is destroyed. As a result, the variables defined within that local scope cease to exist. They are no longer accessible or usable outside of the function.

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

**Ans.)** The concept of a return value refers to the value that a function can send back to the caller once it has completed its execution. When a function reaches a return statement, it evaluates the expression following the return keyword and returns the resulting value to the caller. Like any value, a return value can be used as part of an expression.

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

**Ans.)** If there is no return statement for a function, its return value is **None**.

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

**Ans.)** To make a function variable refer to the global variable of the same name, you can use the **global** keyword in Python. The **global** keyword allows you to indicate that a variable within a function should be treated as a global variable, rather than creating a new local variable with the same name.

Eg.)

x = 10 # Global variable

def my\_function():

global x

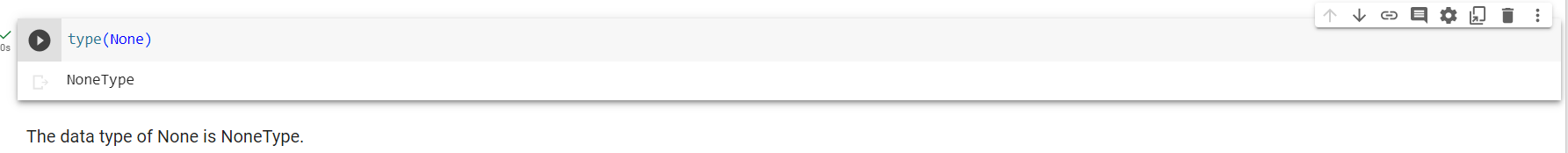
x = 20 # Update the global variable

print("Before function call:", x) # Output: 10

my\_function()

print("After function call:", x) # Output: 20

10. **What is the data type of None?**

**Ans.) **

11. **What does the sentence import areallyourpetsnamederic do?**

**Ans.)**  import statement imports a module named followed by module name.

However, "areallyourpetsnamederic" is not a valid or existing Python module that could be imported. Therefore, attempting to execute the statement "import areallyourpetsnamederic" would raise a **ModuleNotFoundError**, indicating that the specified module could not be found.

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

Ans.) This function can be called with spam.bacon(). Using following syntax:

import spam

spam.bacon()

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

Ans.) To save a program from crashing when encountering an error in Python:

1. Use try-except blocks to catch and handle exceptions.
2. Specify specific exceptions in the except block for customized error handling.
3. Utilize a finally block for cleanup actions that should always execute.
4. Consider logging errors for debugging and troubleshooting purposes.

By implementing these techniques, you can gracefully handle errors, prevent crashes, and provide more robust and user-friendly programs.

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

Ans.) The purpose of the try clause in Python is to enclose a block of code that might raise an exception or error. It allows you to specify a section of code that you want to monitor for potential errors during execution.

The structure of the try clause is as follows:

try:

# Code that might raise an exception

...

except SomeException:

# Actions to handle the exception

...