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

Functions allow the same piece of code to run multiple times by calling the name of the function. Functions break long programs up into smaller components. This allows us to reduce a complicated program into smaller, more manageable chunks, which reduces the overall complexity of our program.

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

Function does not execute the code immediately upon defining it. The code in a function runs when the in-built or pre-defined function is called.

3. What statement creates a function?

The "def" is statement used to create a function along with the name of function and parenthesis. for eg: def my_function():

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

A function is a block of code that performs a specific task or set of instructions. While a function call is the actual execution of those instructions at a specific point in the program.

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

Global scopes refer to the variables defined outside the body of the function. These variables and functions are accessible throughout the entire module. Local scopes, on the other hand, are created when a function is called. The exact number of global and local scopes in a Python program cannot be determined.

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

When a function call returns in Python, the local scope associated with that function is destroyed.

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 (value of the expression following the return keyword) to the caller. The statements after the return statements are not executed.

Yes, it is 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?

In [26]:
```python
# Let's demonstrate this by an example:
def add_num(num1,num2):
    num1 + num2
print(add_num(5,6))

# As per the above example it returned None when we did not use return sta
```

```
None
```

In [21]:
```python
# Let's check it again by using return statement
def add_num(num1,num2):
    return num1 + num2
print(add_num(5,6))
# It has succefully returned the correct value.
# Summary,If a function does not have a return statement, it returns None

```

```
11
```

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

You can use the 'global' keyword to indicate that a variable within a function should refer to the global variable.

10. What is the data type of None?

'None' is often used as a default value for variables or to represent the absence of a return value in functions.

In [23]:
```python
# example:
result = None
print(type(result))
```

```
<class 'NoneType'>
```

11. What does the sentence import areallyourpetsnamederic do?

In [1]:
```python
import areallyourpetsnamederic
#It gives an error No module named 'areallyourpetsnamederic'
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[1], line 1
----> 1 import areallyourpetsnamederic

ModuleNotFoundError: No module named 'areallyourpetsnamederic'
```

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

If there is a module named spam and I've to call it after importing, I'd call it as below:

import spam spam.bacon()

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

To save a program from crashing when encountering an error, we can implement exception handling techniques. By using appropriate error handling techniques, you can handle exceptions and take actions to keep the program running.

You can catch and handle exceptions by using following try-except block: try: # Code that might cause an error ... except ExceptionType: # Code to handle the specific exception ...

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

The 'try' block encloses the error causing code that could raise exceptions. The 'except' blocks define the appropriate code to handle each respective exception type.