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

Reduced Redundancy, eased Repeatability and Preciseness.

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

When it's called.

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

**def**

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

**Function** is the set of predefined instructions, specified by the user.

**Function call** is the signal to execute the predefined instructions that are in Function.

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

**One Global Scope** per Python Program.

**One or More Local Scope(s)** per Python Program.

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

Variable value inside the function will be used for executing the commands within the Function.

In [1]:
```
r=10                    #GLOBAL variable
def test3(a,b):
    r=a/b               #LOCAL variable
    print(r)

test3(100,10)
```

```
10.0
```

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

- To end the execution of Function, return keyword is used
- If **return** is used in place of “print”, the datatype is properly retained. But the “print” will convert any datatype to “NoneType” datatype.
- While defining a function, if any line of code is written after return line, that will not be executed.

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

**None() or NoneType()**

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

First, it needs to be declared globally, before the Function. And then, by calling it without assigning a new value to it.

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

**NoneType**

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

**It will throw a ModuleNotFoundError.**

```
In [1]: 1 import areallyourpetsnamederic

---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-1-b1ea9cb2228f> in <module>
----> 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?

Following Code will be used to call the feature.

**import spam**

**spam.bacon()**

```
In [1]: 1 import spam        # Spam is the module
        2 spam.bacon()       # "bacon()" is the Function or Feature

---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-1-3b6dc785b515> in <module>
----> 1 import spam        # Spam is the module
      2 spam.bacon()       # "bacon()" is the Function or Feature

ModuleNotFoundError: No module named 'spam'


In [2]: 1 import numpy       # Numpy is the module or library
        2 numpy.__version__  # "__version__"  is the Function or Feature

Out[2]: '1.19.2'
```

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

Introducing **Try and Except Clauses**, will avoid the crashing of the program.

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

**Try Clause:**

- Its purpose is to find out whether a block of code is throwing a error or not.
- If there is no Error, code in the 'try block' is executed.
- If the Error exists, code in the 'try block' is terminated and is passed on to the Except Clause.

**Except Clause:**

- Its purpose is to handle the exception (i.e Error) by executing the user defined Except Clause, thereby avoiding the crashing of the code/program.