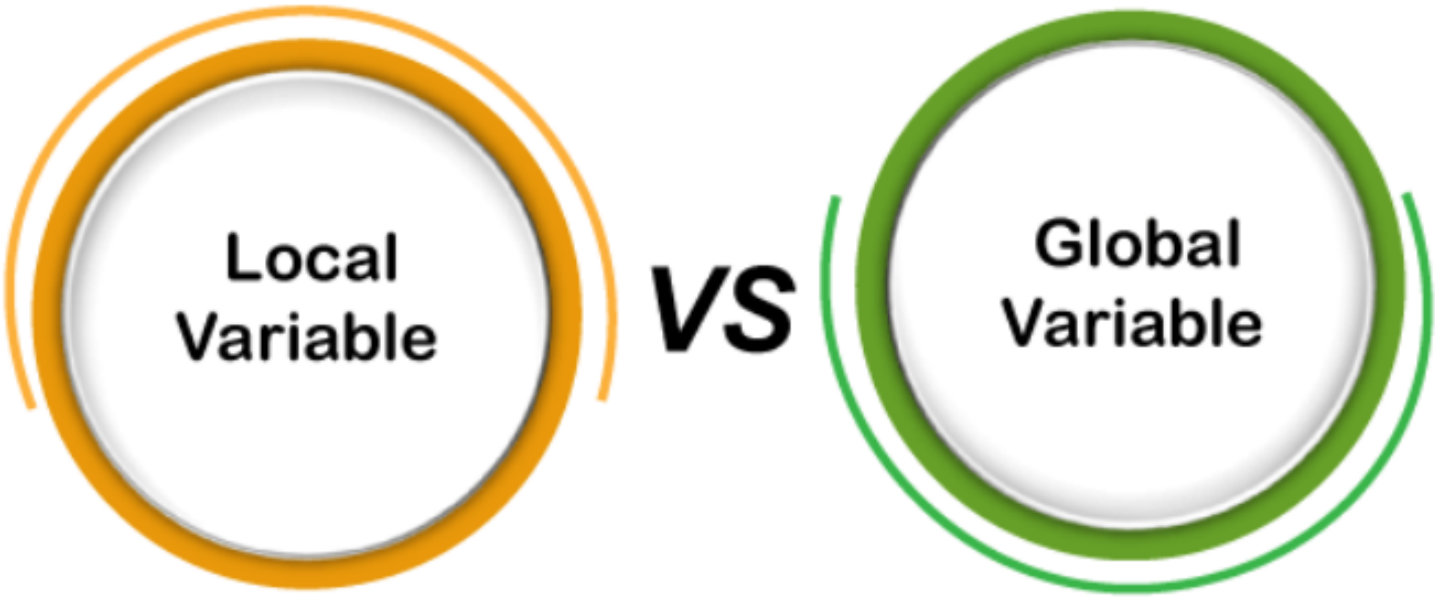
Python

Python Variable Scope

Variables in any programming language have a crucial role. Variables are classified into Global variables and Local variables based on their scope. The main difference between Global and local variables is that global variables can be accessed globally in the entire program, whereas local variables can be accessed only within the function or block in which they are defined. In this topic, we will first understand what are the variables and scope, along with local variables, global variables, and then differences between both the variables.



What is the Difference Between Local and Global Variable in Python?

Variables have a very important role in storing data and information. Before discussing the Difference Between Local and Global Variable in Python, let's see what the scope of variable really is.

Scope of Variable

A variable's scope is basically the lifespan of that variable. It is the section of code to which a variable is alive. Depending on their scope, variables are divided into:

- Global variables
- Local variables

Local variables can only be accessed within the function or module in which they are defined, in contrast to global variables, which can be used throughout the entire program.

In Python, Global variable can be defined using global Keyword, also we can make changes to the variable in local context.

There are some key Difference Between Local and Global Variable in Python:

- Global variables are declared outside the functions whereas local variables are declared within the functions.
- Local variables are created when the function starts its execution and are lost when the function ends. Global variables, on the other hand, are created as execution of the program begins and are lost when the program is ended.
- In contrast to global variables, local variables do not offer data sharing.
- While global variables are kept in a fixed location selected by the compiler, local variables are kept on the stack.
- For local variables, parameter passing is necessary, but not for global variables.
- In Python, Global variables can be defined using global Keyword whereas local variables can be defined directly.

Local Variable

Local variables are declared inside the function blocks. In Python, local variables can be declared at any location in the code block.

Only statements that are written inside a function can access local variables. They are secure in the way that no other function or variable of that program can access them.

Local variables are created during the execution of the function and are destroyed once the block has finished. As soon as the execution leaves the block in which a local variable is declared, it loses its content. It happens because local variables are always stored on the stack.

Example:

Let's see a short example of how local variables can be defined.

```
def fun():
    a = 10
    print(a)

fun()
```

Output:

```
10
```

Global Variable

Global variables are the types of variables that are declared outside of every function of the program. The global variable, in contrast to local variables, is accessible by all functions in a program. Global variables are not very reliable because any function in the program can alter their value.

They continue to exist until the entire program has been ended. Global variables hold onto their values throughout the program execution. The compiler-determined fixed region of memory where they are stored is the cause.

A global variable is useful when many functions are using the same set of data. Utilizing a lot of global variables could be challenging because they could undergo unwanted changes in value.

Example:

Let's see a short example of how global variables can be defined. Here, we are simply accessing the global variable using the global keyword.

```
a = 10

def fun():
    global a
    print(a)

fun()
```

Output:

```
10
```

Local Variable Vs. Global Variables

Lets see the tabular difference Between Local and Global Variable in Python

Comparision Basis	Global Variable	Local Variable
Definition	Global variables are declared outside the functions	Local variables are declared within the functions
Lifetime	Global variables are created as execution of the program begins and are lost when the program is ended	Local variables are created when the function starts its execution and are lost when the function ends
Data Sharing	Global Variables Offers Data Sharing	Local Variables doesn't offers Data Sharing
Scope	Accessible throughout the code	Accessible inside the function
Storage	Global variables are kept in a fixed location selected by the compiler	Local variables are kept on the stack
Parameter Passing	For global variables, parameter passing is not necessary	For local variables, parameter passing is necessary
Changes in a variable value	Changes in a global variable is reflected throughout the code	Changes in a local variable doesn't affect other functions of the program

Advantages and Disadvantages of Global and Local variables

We have discussed Difference Between Local and Global Variable in Python, now lets discuss about their advantages and disadvantages:

Advantages of Local Variables

- The main advantage of a local variable is that the data is not accidentally changed. Variable declared inside a function makes use of that variable while avoiding adverse side effects.
- When the block containing the variable is executed, a local variable only uses memory for a brief period of time.

Advantages of Global Variables

- When dealing with multiple functions in the program that are manipulating the same data, global variables are very helpful.
- The use of a global variable makes it simpler to make changes that are needed to be made throughout the entire programme.
- For global variables, access is available from anywhere or via any program function at random.

Disadvantages of Local Variables

- The local variable's scope is restricted.
- Local Variable prevents sharing of data.
- Since local variables are created and destroyed with each entry and exit from the block, they cannot save the data in-between the function calls.

Disadvantages of Global Variables

- The use of numerous global variables may lead to the creation of programming errors.
- The main issue it creates is the accidental changes in data that arise as a result of the program's multiple global variables.
- It might also necessitate performing code refactoring, a time-consuming process that involves reorganizing the entire program's code.

Which One is More Useful?

In Python programming, both local and global variables are crucial when writing programs. However, many global variables could take up a lot of memory. It is getting harder to spot an undesireable change in global variables. As a result, it is wise to avoid declaring pointless global variables and using local variables for passing and manipulating data. It is generally good practice to make use of local variables in Python.