Local and Global variable

All variables in a program may not be accessible at all locations in that program. This depends on where you have declared a variable.

The scope of a variable determines the portion of the program where you can access a particular identifier. There are two basic scopes of variables in Python

- Local Variable
- Global Variable

Local Variable

Variables declared inside a function body is known as Local Variable. These have a local access thus these variables cannot be accessed outside the function body in which they are declared.

Example	Output

def abc():
    #local Variable
    a=200
    print "value of a is",a
abc()
	value of a is 200

Global Variable

Variable defined outside the function is called Global Variable. Global variable is accessed all over program thus global variable have widest accessibility.

Example	Output

#global Variable
b=100
def abc():
    #local Variable
    a=200
    print "Value of a is",a
    print "Value of b is",b
abc()
print "Value of b is",b
	Value of a is 200
	Value of b is 100
	Value of b is 100

Global Keyword

In Python, global keyword allows you to modify the variable outside of the current scope. It is used to create a global variable and make changes to the variable in a local context.

- When we create a variable inside a function, it’s local by default.
- When we define a variable outside of a function, it’s global by default. You don’t have to use global keyword.
- We use global keyword to read and write a global variable inside a function.
- Use of global keyword outside a function has no effect.

Example	Output

def abc():
    #local Variable
    a=200
    #global Variable using global keyword
    global b
    b = 100
    print "Value of a is",a
    print "Value of b is",b
abc()
print "Value of b is",b
	Value of a is 200
	Value of b is 100
	Value of b is 100