

In Python, variables can have either global or local scope, and this determines where in the program they can be accessed.

**\*\*Global Variable:\*\***

- A global variable is defined outside of any function or block of code.
- It can be accessed from anywhere in the program, both inside and outside functions.
- It retains its value between function calls.

**\*\*Local Variable:\*\***

- A local variable is defined within a function or a block of code.
- It is only accessible within that specific function or block.
- It is created when the function is called and is destroyed when the function exits.

Here's a simple example to illustrate the difference:

```
# Global variable
global_var = 10

def my_function():
    # Local variable
    local_var = 5
    print("Inside function:", global_var, local_var)

# Accessing global variable
print("Outside function:", global_var)

# Calling the function
my_function()
```

In this example:

- `global\_var` is a global variable, accessible both inside and outside the function.
- `local\_var` is a local variable, only accessible inside the `my\_function` function.

When you run this program, you'll see that you can access the global variable from anywhere, but attempting to access the local variable outside its function will result in an error.