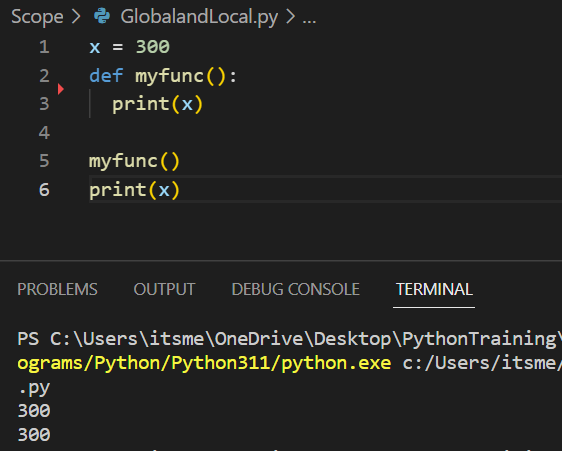
**SCOPE OF A VARIABLE**

The term **"scope"** refers to the part of a program where a particular variable can be accessed. A variable's scope is determined by where it is defined within the program. There are two main types of scopes**: global scope** and **local scope.**

**Global** scope refers to variables that are defined outside of any function or class. These can be accessed from anywhere in the program, including from within functions and classes. They are typically used to store values that need to be accessed by multiple parts of the program. It is better to limit the number of global variables used in a program.

Global variables are available from within any scope, global and local.

example,

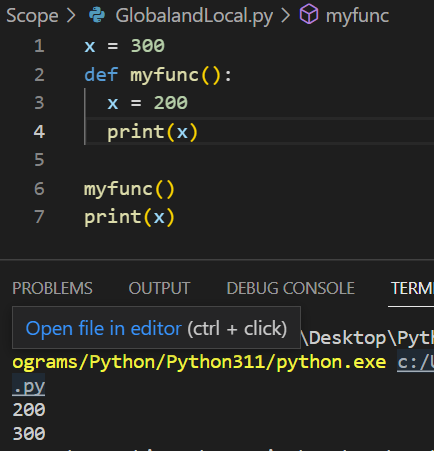


Here, the variable is declared globally. So, it can be accessed inside and outside the function.

**Local scope** refers to variables that are defined within a function or method. These variables can only be accessed from within a block of code (such as a loop or conditional statement) also have local scope and can only be accessed within that block of code. Once the function or method has completed its execution, the local variables are no longer accessible. The garbage collector deletes them after the execution of the function/method is completed.

If we operate with the same variable name inside and outside of a function, the compiler will treat them as two separate variables, one available in the global scope (outside the function) and one available in the local scope (inside the function):

Consider the following example:

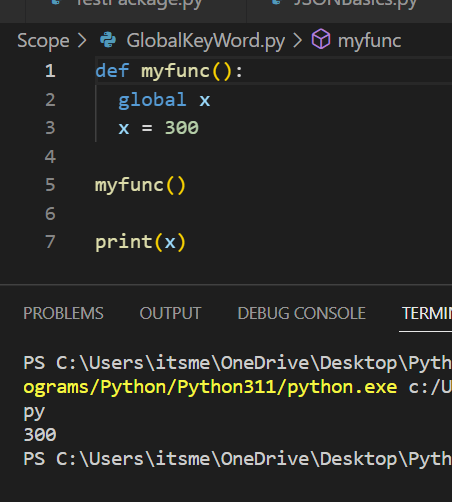


Here, a variable called 'x' is declared globally with value 300.Inside the function, we created another variable with same name as global variable with value 200.Even though they have same name, inside the function, the declaration inside the block is considered. i.e, the value of x inside the function is 200, which is not the same as the global variable. As soon as the function execution is completed, the locally declared variable is destroyed. Now the value of x = 300.

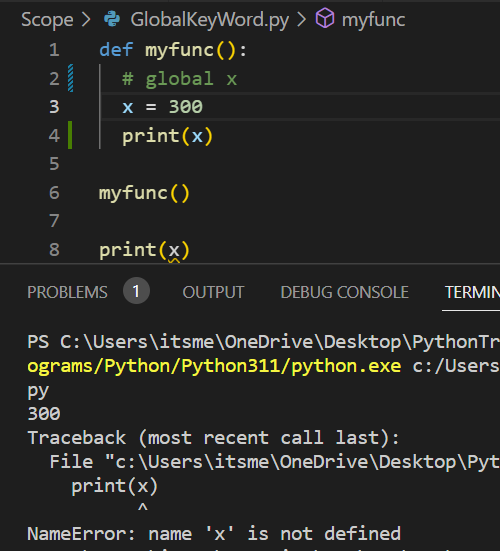
**Make a local variable into a global variable**

Variables declared in a block can be made into global variables using the keyword global.

If we need to create a global variable, but are stuck in the local scope, you can use the global keyword.



If x was not declared as global in the above code, we would meet an exception called Name Error



Accessing variables outside of their scope can lead to errors and unexpected behaviour in your program. So we should reduce the use of global variables and to carefully manage the scope of your local variables.