Local scope vs Global scope

# Global scope:

This refers to variables:

-Declared outside a function block

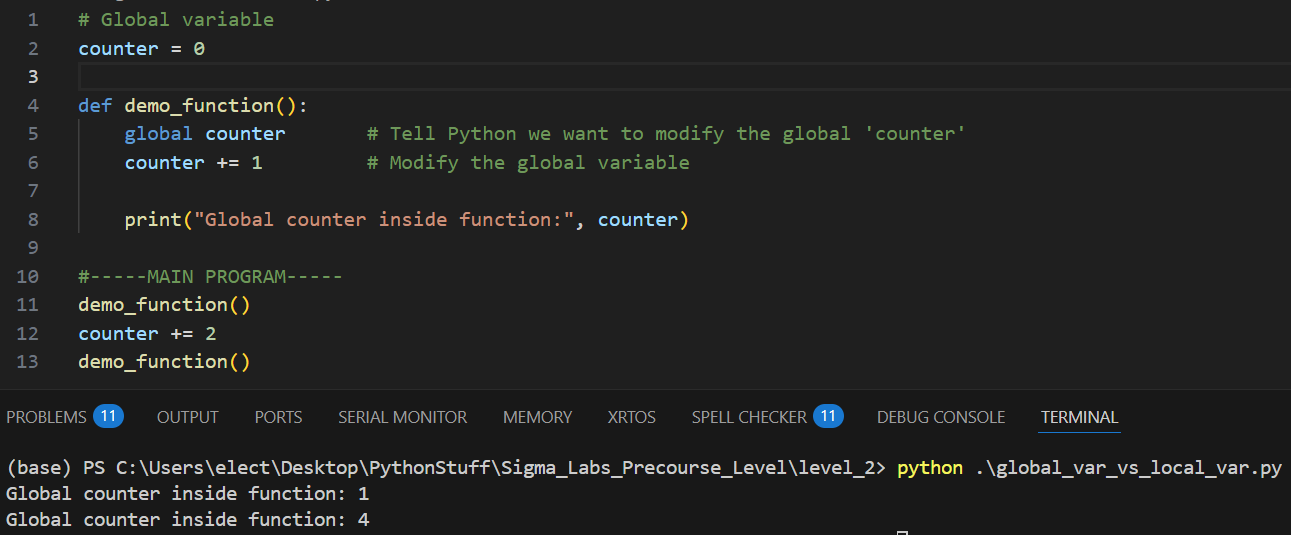
-Accesible from any part of the program including functions, loops or other blocks of code

-Live throughout the entirety of the code’s runtime

-Can lead to name conflicts if not used carefully

-Ideal for values that need to be accessed by multiple parts of the program

## Example snippet (python):



This snippet showcases a global variable in python being accessed both within a function and outside a function. The variable can be accessed from anywhere because it has global scope.

# Local Scope:

This refers to variables:

-Declared within a function block

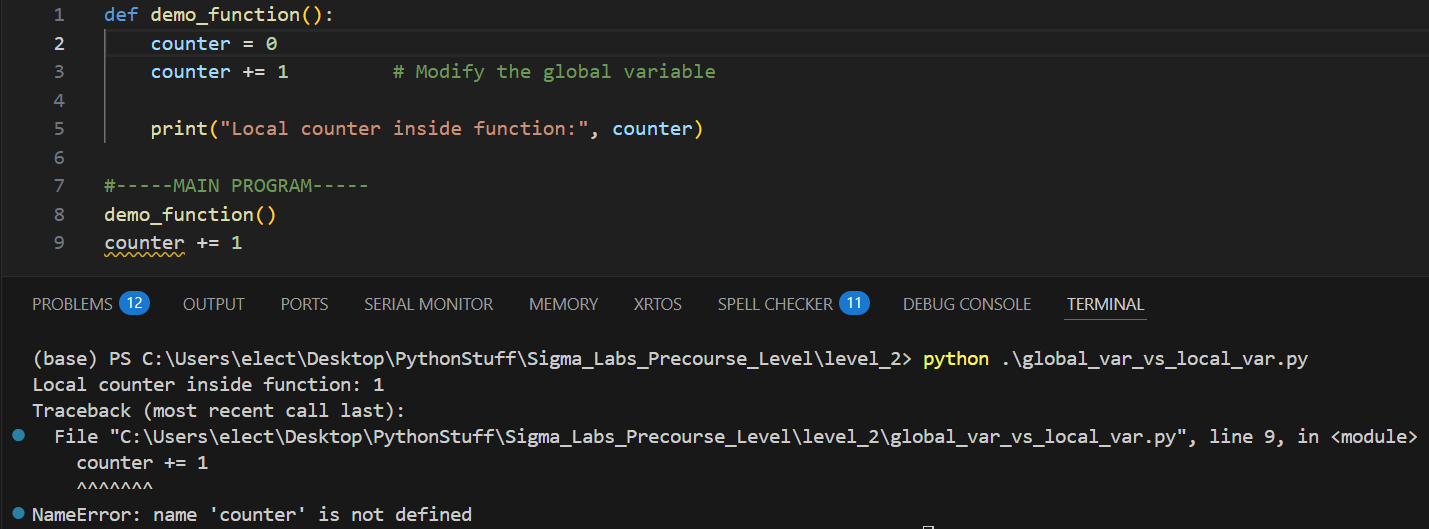
-Only accessible within the function or scope where it was declared (e.g: a variable declared within a for loop is only accessible within the for loop).

-Deleted once once they are not needed anymore (e.g: function finished executing)

-Can have the same name as variables in other blocks of code (even global variables outside the program)

-Best used for temporary storage for a specific block of code

## Example snippet (python):



As the snippet shows, the “counter” local variable is inaccessible outside the function and leads to an error in execution. This error happens because the “counter” variable is declared within a function, therefore it only has local scope within “demo\_function” and is therefore inaccessible outside the function. “counter” does not exist outside “demo\_function”.