What is the global keyword

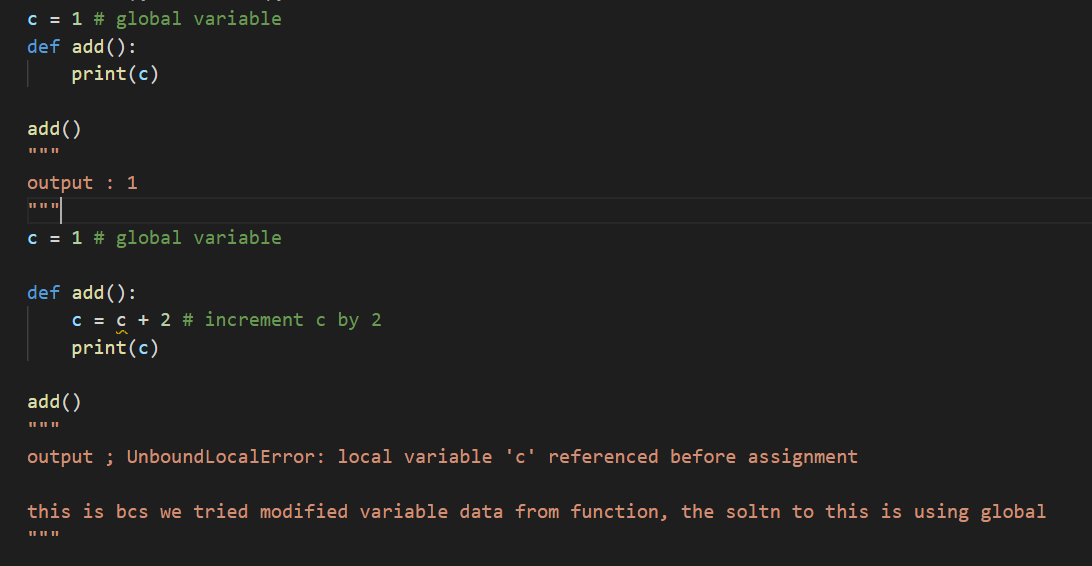
In Python, global keyword allows you to modify the variable outside of the current scope. It is used to create a global variable and make changes to the variable in a local context.

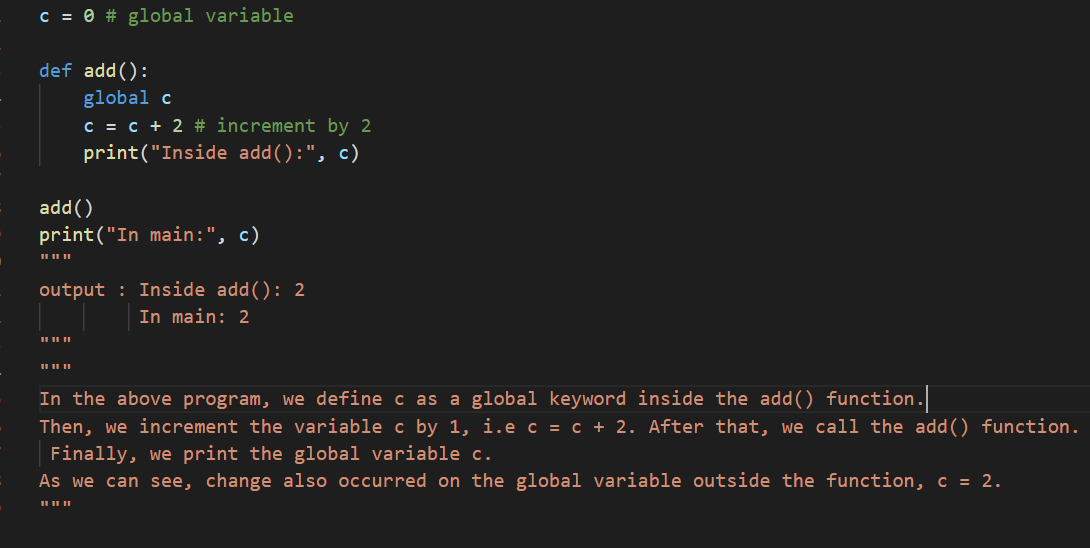
## Rules of global Keyword

The basic rules for global keyword in Python are:

* When we create a variable inside a function, it is local by default.
* When we define a variable outside of a function, it is 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.

## Use of global Keyword

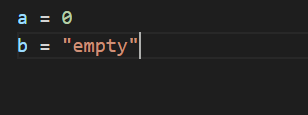




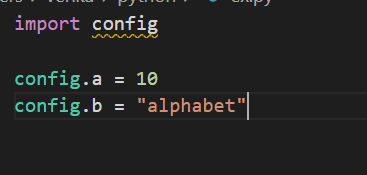
## Global Variables Across Python Modules

In Python, we create a single module config.py to hold global variables and share information across Python modules within the same program.

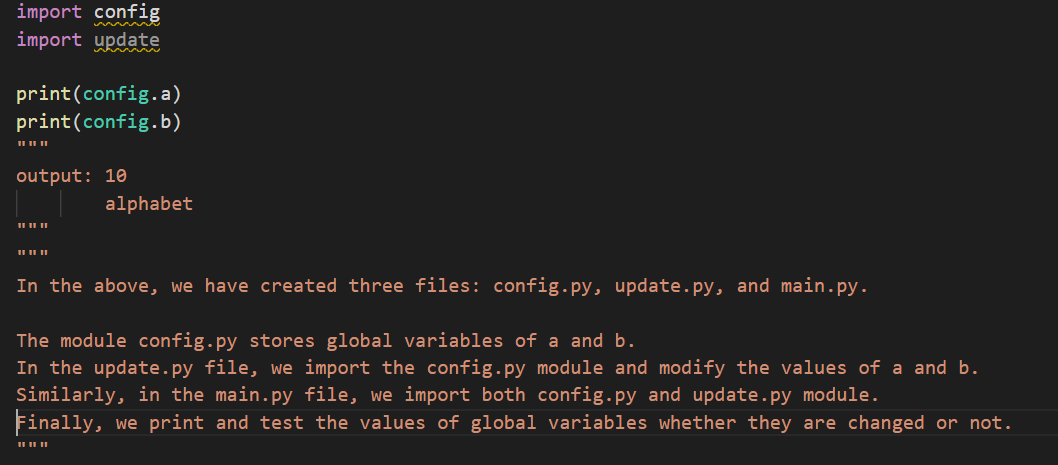
Create a config.py file, to store global variables



Create a update.py file, to change global variables

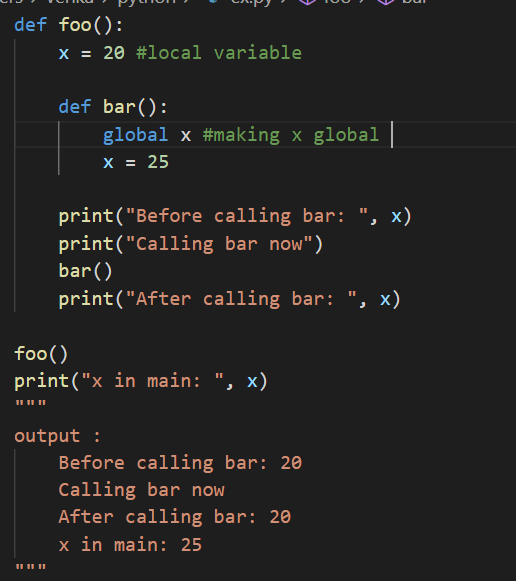


Create a main.py file, to test changes in value



## Global in Nested Functions

Here is how you can use a global variable in nested function.



In the above program, we declared a global variable inside the nested function bar(). Inside foo() function, x has no effect of the global keyword.

Before and after calling bar(), the variable x takes the value of local variable i.e x = 20. Outside of the foo() function, the variable x will take value defined in the bar() function i.e x = 25. This is because we have used global keyword in x to create global variable inside the bar() function (local scope).

If we make any changes inside the bar() function, the changes appear outside the local scope, i.e. foo().