Say, x is equal to 4 and i will give y is equal to 5 ok now i can access x also, y also.

i can get 4 also, 5 also ……ok, so then what is the work of namespace ok?, I'll simply make sure that whatever the names i use are unique, If I take care about that then why we need namespace?

That is because if you are writing a 10 or 20 lines of code, then that's ok. you can take

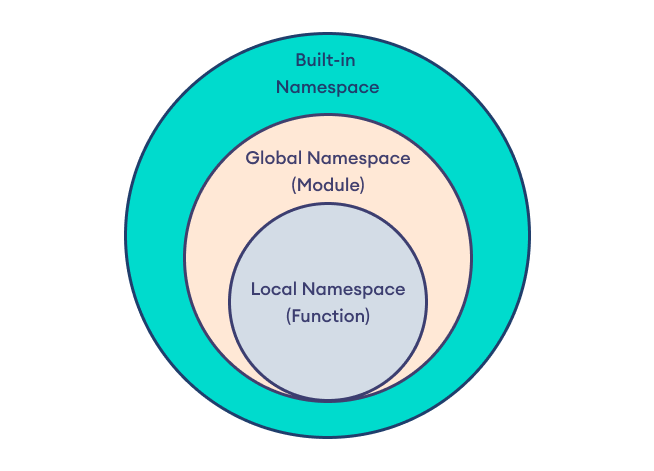
care of the variable names and you can make sure that whatever the names you used are unique ……..but but but….. if you are writing a long program and you are using external functions also like you are importing the modules, then **in that module you don't know which names you used, In that case you may reused the names in the different modules.**

In another case, **half of the code is written by your friend and half is written by you, then you don't know which names he used or she used in her part of code right?, so you may have reused that same names**.

So in that case you can't sit and check all the names right?

In such cases namespace comes to play and it will allow us to reuse the names.

3 types of namespace:



When you call a function it will create its own namespace that is called as local namespace.

When you import any module, when you use any module that time it will create its own Global namespace, so each module has its own namespace that's why you can reuse names in different

modules and ….

Built-in is when you start the Python interpreter built-in namespace is created

it contains all the built-in names

How does it work?

Say my name is Amulya Shetty and some other student may have same name Amulya but different surname like Sharma Ok, so using that surname we can identify them right?, Here also we will do the same thing, here when we have different names, we will identify them through namespace

In the module,

Import module\_name

Module\_name.function()

we can see when we import module like this like any module name, we will call the function as module name dot function name. so that’s us telling Python that this function belongs to this module ok so each module has its own namespace even though they have same names but module name is different that's why they can be reused,

VARIABLE SCOPE:

Part of the program where variable is accessible is called as variable scope. Here this is the scope of x, inside the function .

There will be 4 different types

So there are four type in the variable scope first one is local second one is global

third one is enclosed fourth one is built in.

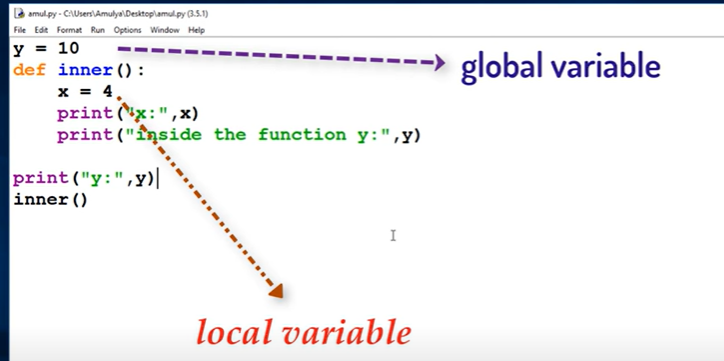
1. Local scope: Local scope, it contains names defined inside the current function ok if we define any variable inside a function then the scope of that variable is inside that function…… when the function ends the availability of the variable also ends
2. Global Scope: It contains names defined at the top level of the module or you can say beginning of the script and these names are available for the whole script, the lifetime of the names ends when the program ends .

NOTE: Whatever the variable we use in the local scope are called as local variables, if we use functions then they are called as local functions.

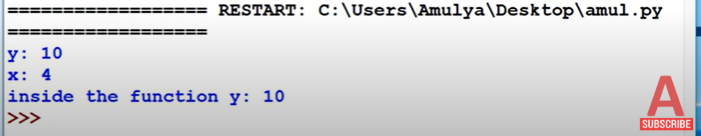
Whatever the variable we use in the global scope is called as global variables and the

functions are called as global functions.

1. Built in Scope:The last one is built in scope it contains names built into the Python language through the special built\_in module ok so it contains built-in names.
2. Enclosed Scope: is the names with in the enclosed functions

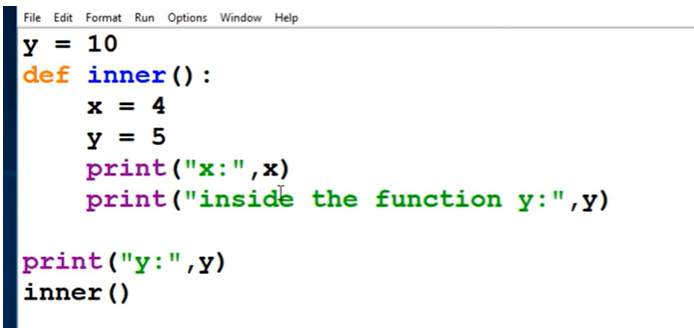


Output: note we can access the

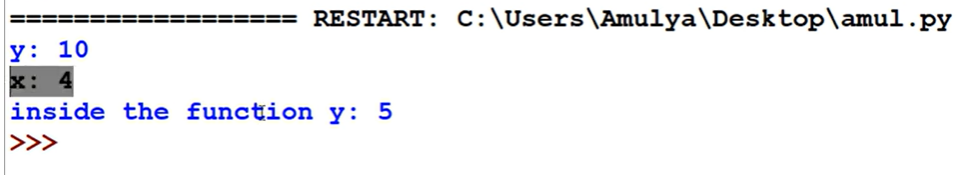


In above example,“x” is the local variable, “y” is the global variable, and I can print the “y” value inside the function also outside the function also but “x” can be accessed only in the function body.

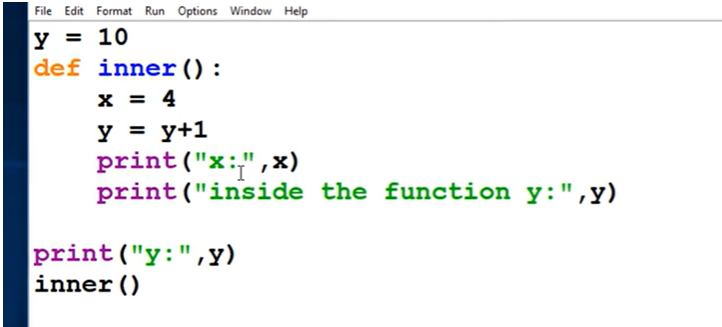
Example 2:



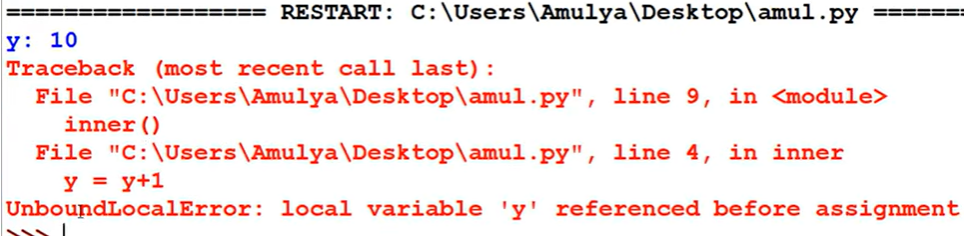
Output:



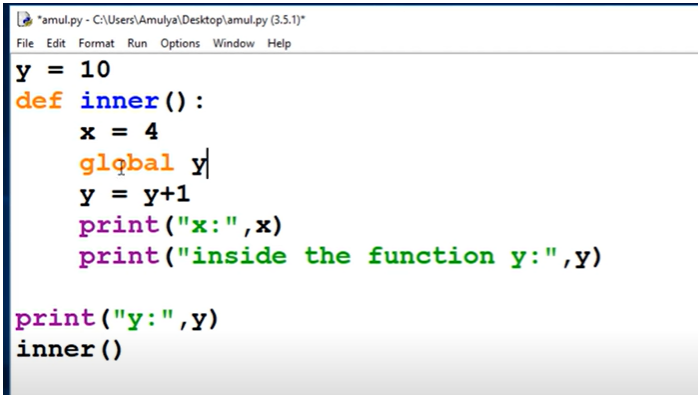
Modifying the global variable in local scope:

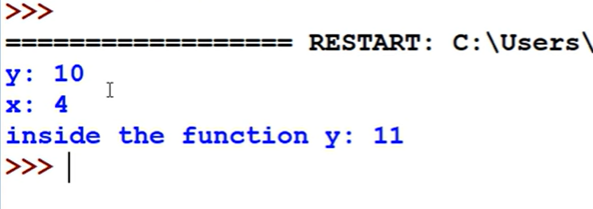


Output: is an error b/c we can’t modify the global variable inside the local variable DIRECTLY.

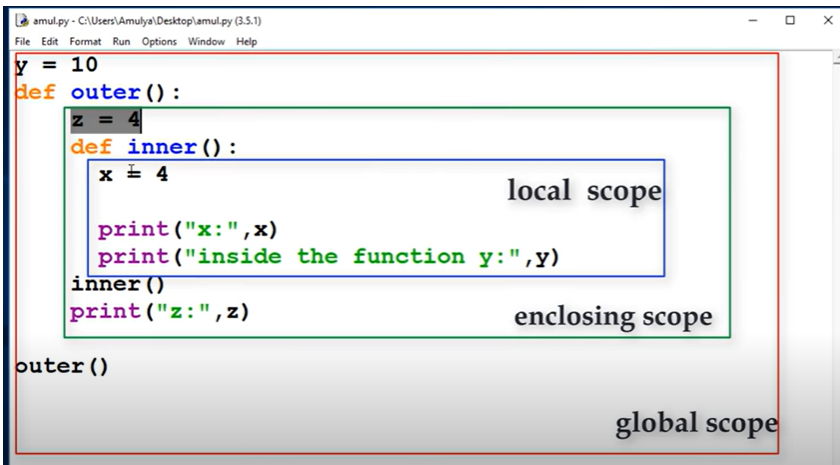


So we need to use a keyword “ global y ” .





Example 3 : Enclosed scope



Here, “z” is local to the outer function but it is non local to inner function, for

the inner function “**z” is not the local variable as well as it is not the global variable**

**so that's why it is called as non local variable and this is also called as “Enclosing variable”.** Here “z” is the enclosed variable.