

Python Namespace

Namespace in python are implemented as python dictionaries, this means it is a mapping from names to object. the user doesn't have to know this to write a python program and when using namespace. the following some namespace in python.

global names of a module

local names in a function or method invocation

built in names

Python Scope

Although there are various unique namespaces defined, we may not be able to access all of them from every part of the program. the concept of scope comes into play. scope is the portion of the program from where a namespace can be accessed directly without any prefix. at any given moment. there are at least three nested scopes.

1. Scope of the current function which has local names
2. Scope of the module which has global names
3. Outermost scope which has built-in names

Example

```
def scope_test():
    def do_local():
        spam = "local spam"

    def do_nonlocal():
        nonlocal spam
        spam = "nonlocal spam"

    def do_global():
        global spam
        spam = "global spam"

    spam = "test spam"
    do_local()
    print("After local assignment:", spam)
    do_nonlocal()
    print("After nonlocal assignment:", spam)
    do_global()
    print("After global assignment:", spam)

scope_test()
print("In global scope:", spam)
```

Output

```
After local assignment: test spam  
After nonlocal assignment: nonlocal spam  
After global assignment: nonlocal spam  
In global scope: global spam
```