NAMESPACES- KEYWORDS -COMMENTS-ASSIGN VALUE TO THE VARIABLE

NAMESPACES:

1. Python itself maintains a namespace in the form of a python dictionary.
2. Unique name for each object in python.
3. Multiple directories – can have a same file name in that case to direct the user to the requested file we use the concept of namespace.
4. Example its like surname for a name- which make a person with same name differ from other. Like there can be two alice in a class but there are differed by there surname like lee or chake.
5. Python understand the variable with the help of the namespace. Name(unique identifier) +space(talk about the scope).
6. The life time of the namespace is until the scope of the object end. Such as if we create a variable globally and u don’t use after 10 th line than its scope gets over. It mean we cannot access to the local namespace from the outside.
7. Like u cannot access to the variable inside the function, in outside the function.

TYPES OF NAMESAPCES:

1. Built in namespace—namespace that’s already present for print() id() function in the system.
2. Global namespace—when user creates a module then namespace created for that is called global namespace.
3. Local namespace—namespace created for local functions in the program.

# var1 is in the global namespace

var1 = 5

def some\_func():

    # var2 is in the local namespace

    var2 = 6

    def some\_inner\_func():

        # var3 is in the nested local

        # namespace

        var3 = 7

we can use the global namespace within a function like bellow:

# global variable

count = 5

def some\_method():

    global count

    count = count + 1

    print(count)

some\_method()

output of the above code is 6 as the count is 5 now its increased to count+1= 5+1=6.

SCOPE OF THE OBJECTS:

1. As we can see namespace also deal with scope. It is defined as the region from which the object is accessed. Like you access the local namespace(variable x within function). Then the scope of the x is within the function.
2. Bellow we can see the var=10 is declared inside the function but you are trying to access to it outside too.. print("Try printing var from outer function: ",var) this line will throw an error that var is not defined.

def some\_func():

    print("Inside some\_func")

    def some\_inner\_func():

        var = 10

        print("Inside inner function, value of var:",var)

    some\_inner\_func()

    print("Try printing var from outer function: ",var)

some\_func()

output of this is

Inside some\_func

Inside inner function, value of var: 10

Traceback (most recent call last):

File "/home/1eb47bb3eac2fa36d6bfe5d349dfcb84.py", line 8, in

some\_func()

File "/home/1eb47bb3eac2fa36d6bfe5d349dfcb84.py", line 7, in some\_func

print("Try printing var from outer function: ",var)

NameError: name 'var' is not defined

KEYWORDS:

1. It’s a set of predefined words are called keywords.
2. These types of keywords cannot be used as the names of varibales, fuctions and classes.
3. These can be also said as reserved words that cannot be used.

* IMPORT
* FROM
* GLOBAL
* NONLOCAL
* DEL
* EXCEPT

1. Logical operations--- or, and, not, true, false,none.
2. Iteration keywords—for, while, break, continue.
3. Loops—for, while.
4. Conditional keywords—if,else,elif
5. Structure keywords—def,class,with,as,pass,lamba,in,is.
6. Return keywords—return,yield.
7. Exception handling keywords—try,except,finally,raise,assert.

import keyword

# printing all keywords at once using "kwlist()"

print("The list of keywords is : ")

print(keyword.kwlist)

output:

The list of keywords are:

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break',

'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if',

'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

COMMENTS:

1. Useful information that help in understand the source code easier.
2. Explain the logic or part of it used in the code.

TYPES OF COMMENTS:

1. Single-line comment--# hash.
2. Multiline comment—use multiple # or “”” 3 double quotes or ‘’’ 3 single quotes.- used to show how a block of code works.
3. Docstring—surrounded by triple quotes in python””” “””
4. Shows how a class and functions works.

''’This article on geeksforgeeks gives you a

perfect example of

multi-line comments'''

#single line comment

“””this is docstring”””

#multline line

#multiline

print("GeeksForGeeks")

output: GeeksForGeeks

ASSIGNING VALUES OF VARIBLES :

Steps and methods to assign values to variables-

* Direct initialization.

# initialising variable directly

a = 5

# printing value of a

print ("The value of a is: " + str(a))

Assign multiple values:

# Assigning multiple values in single line

a,b,c="geeks","for","geeks"

print(a+b+c)

#output is geeksforfeeks

* Using conditional operator.

*condition? True\_value : False\_Value*

a = 1 if 20 > 10 else 0

# Printing value of a

print("The value of a is: " , str(a))

output: The value of a is: 1.