1. What is the result of the code, and explain?

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

Ans:

The line, func() Call the function we defined which prints the value of X.

In [2]:

X **=** 'iNeuron'

**def** func():

print (X)

fun()

2. What is the result of the code, and explain?

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

Result of the code is iNeuron

Explanation:

X=’NI’ value will only have the scope inside of the fun()

so outside of the fun(),x value will be ‘iNeuron’

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

Ans:

X **=** 'iNeuron'

**def** func():

X **=** 'NI!'

print (X)

func()

o/p: NI!

print(X)

o/p : iNeuron

Explanation:

here x is printing inside of the fun() as the local variable.

and in the last statement X is printing as a global vaiable

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

Ans:

o/p:

‘NI’

Explanation:

here we are making X as a global variable manually inside the fun() .by doing thing that assigned

value can be accessed all over the program.

5. What about this code—what’s the output, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func()

>>> X

Ans:

X **=** 'iNeuron'

**def** func():

X **=** "NI"

**def** nested():

print(X)

nested()

o/p: iNeuron

func()

X

o/p: 'iNeuron'

6. How about this code: what is its output in Python 3, and explain?

>>> def func():

X = 'NI'

def nested():

nonlocal X

X = 'Spam'

nested()

print(X)

>>> func()

aNS:

The nonlocal keyword is used to work with variables inside nested functions, where the variable

should not belong to the inner function.Use the keyword nonlocal to declare that

the variable is not local.

**def** func():

X **=** 'NI'

**def** nested():

**nonlocal** X

X **=** 'Spam'

nested()

print(X)

O/P: iNeuron

func()

print(myfunc1())

O/P: hello