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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

Answer:

iNeuron

A variable is assigned to string and displayed when the function is called.

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

Answer:

iNeuron

The variable is assigned to string and displayed when the function is called.

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

Answer:

NI!

iNeuron

Here print (X) is called twice which calls all the values assigned to it.one X is inside the function and another X is outside of the function.Here both X are not same.

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

Answer:

NI!

The global function assigns the new string as a value for X.

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func()

>>> X

Answer: iNeuron

The nested function assigns values one by one, it helps define function inside another function.

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()

Answer:

No output

As null value is newly assigned to the variable.