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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

Ans. iNeuron

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

Ans. iNeuron

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

Ans.

'NI'

'iNeuron'

when x is printer inside the function then the x which is local variable of the fumction is printed then when x is printed again then the x which is declared which is outside of the function is printed.

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

Ans. 'NI'

As X is declared global that's why the value of x is changed and it's reflected outside the function also.

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.

iNeuron

NI

iNeuron

In, nested the x which is declared the function is printed and inside func the the local x is printed and atlast the x which is declared outside the function is printed.

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. Spam

Here the value of the parent x is changed by spam, therefore it's printed.