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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

Solution1:-

iNeuron

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

Solution2:- The line, func() Call the function we defined, with "NI" as the value of X inside the funtion, but doesn't prints it, as there is no print statement inside the function.

X = 'iNeuron'

def func():

X = 'NI!'

func()

print(X)

iNeuron

The line, print(X), prints the value of X, which is "iNeuron", which is outside func().

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

Solution3:- NI

iNeuron

because print() is inside the function.

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

Solution4:-

NI

Because we declared x=ni as global.

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func()

>>> X

Solution5:- iNeuron

'iNeuron'

Because fun() has no print() statement.

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()

Solution6:- 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)

iNeuron

func()