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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(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

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()

Answers:

1. The result of the code is 'iNeuron'. The function func() doesn't have any local variable X defined, so it refers to the global variable X, which has the value 'iNeuron'.
2. The code doesn't produce any output. When the function func() is called, it assigns the value 'NI!' to the local variable X, which doesn't affect the global variable X. When print(X) is called after the function, it prints the value of the global variable X, which is still 'iNeuron'.
3. The code prints 'NI'. When the function func() is called, it assigns the value 'NI' to the local variable X, which is then printed. However, this doesn't affect the value of the global variable X, which is still 'iNeuron', so when print(X) is called after the function, it prints 'iNeuron'.
4. The output of the code is 'NI'. When the function func() is called, it declares that the global variable X will be used, and assigns it the value 'NI'. This changes the value of the global variable, so when print(X) is called after the function, it prints 'NI'.
5. The output of the code is 'NI'. When the function func() is called, it assigns the value 'NI' to the local variable X, and then defines a nested function called nested(). When nested() is called, it prints the value of the local variable X, which is 'NI'. After nested() completes, the function func() doesn't have a return statement, so it returns None. When X is printed after the function, it still has the value 'iNeuron', since the local variable X defined in the function func() doesn't affect the global variable X.
6. The output of the code is 'Spam'. When the function func() is called, it defines a local variable X with the value 'NI', and then defines a nested function called nested(). When nested() is called, it declares that the variable X will be a nonlocal variable (i.e., it's defined in an outer scope), and assigns it the value 'Spam'. This changes the value of the variable X that's defined in the outer scope of nested(), which is the local variable X defined in the function func(). After nested() completes, the value of the local variable X is 'Spam', so when print(X) is called, it prints 'Spam'.