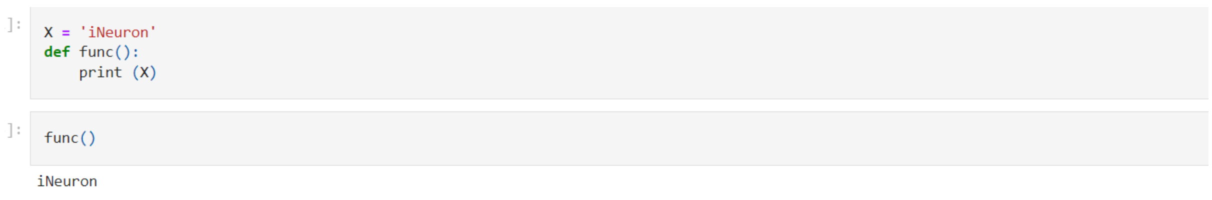
1. What is the result of the code, and explain?

```
>>> X = 'iNeuron'
>>> def func():
print(X)


>>> func()
```

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



```
X = 'iNeuron'
def func():
    print (X)
```

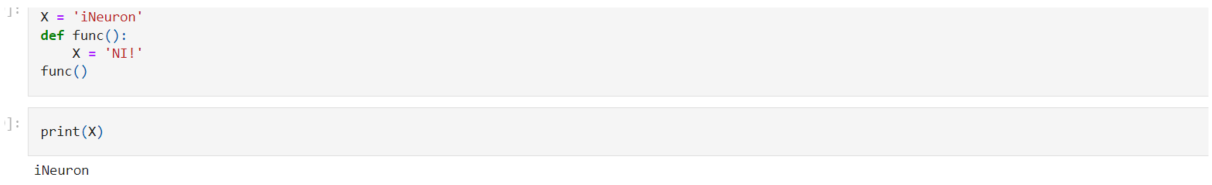
```
func()
```

```
iNeuron
```

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

```
>>> X = 'iNeuron'
>>> def func():
X = 'NI!'


>>> func()
>>> print(X)
```

Ans:



```
X = 'iNeuron'
def func():
    X = 'NI!'
func()
```

```
print(X)
```

```
iNeuron
```

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.

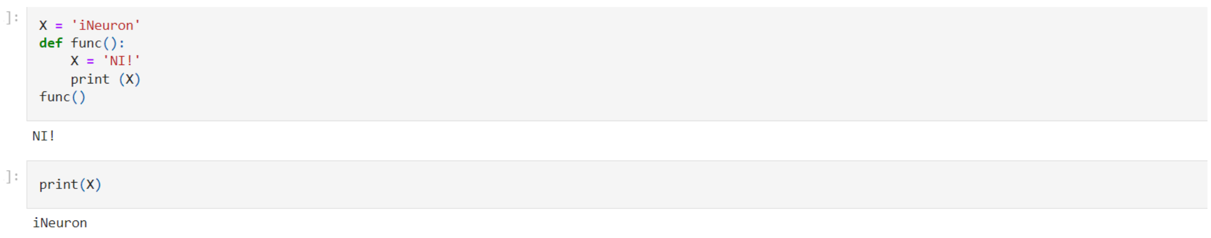
The line, print(X), prints the value of X, which is "iNeuron", which is outside func().

3. What does this code print, and why?

```
>>> X = 'iNeuron'
>>> def func():
X = 'NI'
print(X)


>>> func()
>>> print(X)
```

Ans: The line, func() Call the function we defined which prints the value of X, which is "NI" inside the funtion.

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



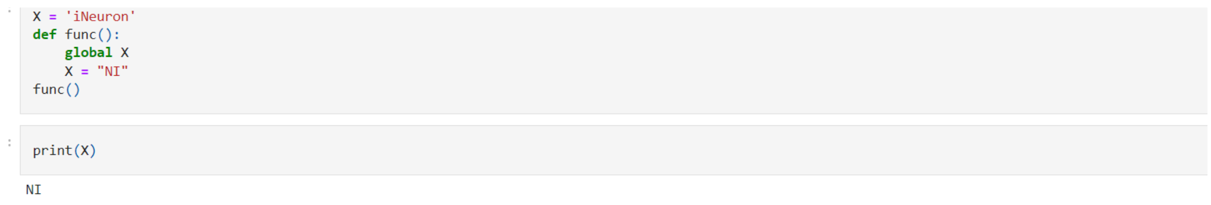
```
]:  X = 'iNeuron'
    def func():
        X = 'NI!'
        print (X)
    func()

    NI!

]:  print(X)

    iNeuron
```

4. What output does this code produce? Why?

```
>>> X = 'iNeuron'
>>> def func():
global X
X = 'NI'


>>> func()
>>> print(X)
```

Ans: 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, and we have used global keyword, which means, global keyword allows us to modify the variable, that is "X", outside of the current function.

The line, print(X), prints the value of X, which is now "NI", as we used global keyword inside the function.

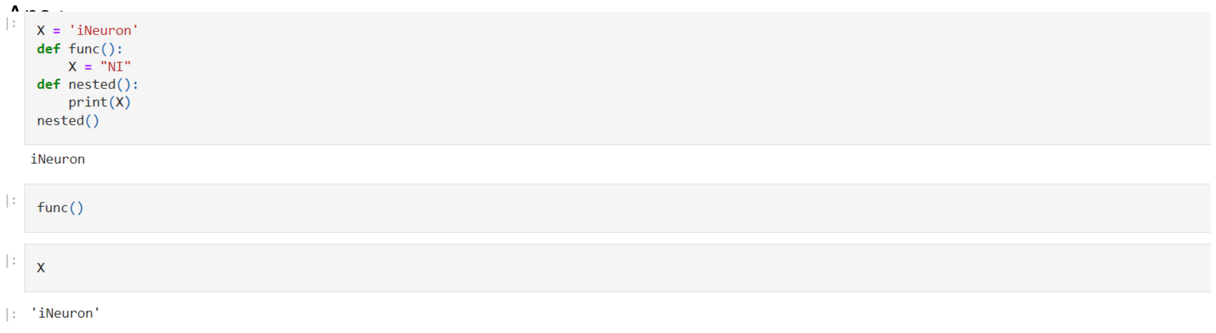


```
X = 'iNeuron'
def func():
    global X
    X = "NI"
func()

:  print(X)

NI
```

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

```
>>> X = 'iNeuron'
>>> def func():
X = 'NI'
def nested():
print(X)
nested()


>>> func()
```

>>> X

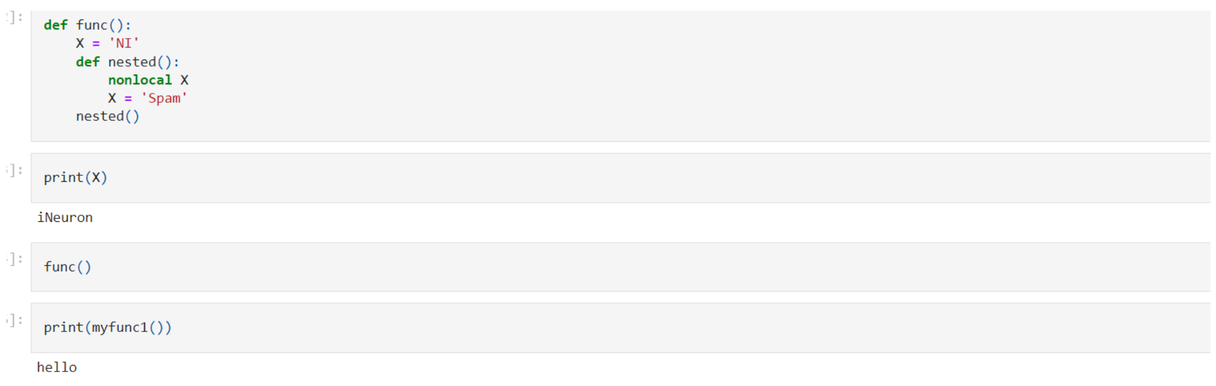


```
[ ]: X = 'iNeuron'
     def func():
         X = "NI"
     def nested():
         print(X)
     nested()

     iNeuron

[ ]: func()

[ ]: X

[ ]: '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()
variable is not local.belong to the inner function.Use the keyword nonlocal to declare that the



```
[ ]: def func():
         X = 'NI'
         def nested():
             nonlocal X
             X = 'Spam'
         nested()

[ ]: print(X)

     iNeuron

[ ]: func()

[ ]: print(myfunc1())

     hello
```