Python Assignment - 22

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

**&gt;&gt;&gt; X = &#39;iNeuron&#39;**

**&gt;&gt;&gt; def func():**

**print(X)**

**&gt;&gt;&gt; func()**

=>

Output: iNeuron

Explanation:

* The code defines a variable X and assigns it the string value 'iNeuron'.
* Then, a function named func() is defined.
* Inside the func() function, the print(X) statement is executed.
* Since X is defined outside the function, it is considered a global variable, accessible from within the function.
* When func() is called, it executes the print(X) statement, which prints the value of X, resulting in the output 'iNeuron'.

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

**&gt;&gt;&gt; X = &#39;iNeuron&#39;**

**&gt;&gt;&gt; def func():**

**X = &#39;NI!&#39;**

**&gt;&gt;&gt; func()**

**&gt;&gt;&gt; print(X)**

=>

Output: iNeuron

Explanation:

* The code first defines a global variable X and assigns it the string value 'iNeuron'.
* Then, a function named func() is defined.
* Inside the func() function, a new local variable X is defined and assigned the string value 'NI!'.
* When func() is called, it creates and modifies the local variable X within its scope.
* After the execution of func(), the scope of the function ends, and the local variable X is destroyed.
* Finally, the print(X) statement is executed outside the function. Since there is no local variable X in the global scope, the value of the global variable X is printed, resulting in the output 'iNeuron'.

**3. What does this code print, and why?**

**&gt;&gt;&gt; X = &#39;iNeuron&#39;**

**&gt;&gt;&gt; def func():**

**X = &#39;NI&#39;**

**print(X)**

**&gt;&gt;&gt; func()**

**&gt;&gt;&gt; print(X)**

=>

Output:

NI

iNeuron

Explanation:

* The code first defines a global variable X and assigns it the string value 'iNeuron'.
* Then, a function named func() is defined.
* Inside the func() function, a new local variable X is defined and assigned the string value 'NI'.
* When func() is called, it executes the print(X) statement within its scope, printing the value of the local variable X, which is 'NI'.
* After the execution of func(), the scope of the function ends, and the local variable X is destroyed.
* Finally, the print(X) statement is executed outside the function. Since there is a global variable X defined, its value 'iNeuron' is printed. The local variable X within func() does not affect the value of the global variable X.

**4. What output does this code produce? Why?**

**&gt;&gt;&gt; X = &#39;iNeuron&#39;**

**&gt;&gt;&gt; def func():**

**global X**

**X = &#39;NI&#39;**

**&gt;&gt;&gt; func()**

**&gt;&gt;&gt; print(X)**

=>

Output: NI

Explanation:

* The code first defines a global variable X and assigns it the string value 'iNeuron'.
* Then, a function named func() is defined.
* Inside the func() function, the line global X is used to specify that the variable X being referenced is the global variable, not a local variable.
* The next line X = 'NI' assigns the string value 'NI' to the global variable X.
* When func() is called, it modifies the global variable X with the value 'NI'.
* Finally, the print(X) statement outside the function is executed. Since the global variable X has been modified within func(), its value 'NI' is printed.

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

**&gt;&gt;&gt; X = &#39;iNeuron&#39;**

**&gt;&gt;&gt; def func():**

**X = &#39;NI&#39;**

**def nested():**

**print(X)**

**nested()**

**&gt;&gt;&gt; func()**

**&gt;&gt;&gt; X**

=>

Output:

NI

'iNeuron'

Explanation:

* The code defines a global variable X and assigns it the string value 'iNeuron'.
* The function func() is defined.
* Inside func(), a local variable X is defined and assigned the string value 'NI'.
* Within func(), another function named nested() is defined.
* Inside nested(), the print(X) statement is executed.
* When func() is called, it executes nested() function, which prints the value of the local variable X within func(), resulting in the output 'NI'.
* After the execution of func(), the global variable X is not affected, so its value remains 'iNeuron'.
* Finally, when X is printed outside the functions, it displays the value of the global variable, which is 'iNeuron'.

**6. How about this code: what is its output in Python 3, and explain?**

**&gt;&gt;&gt; def func():**

**X = &#39;NI&#39;**

**def nested():**

**nonlocal X**

**X = &#39;Spam&#39;**

**nested()**

**print(X)**

**&gt;&gt;&gt; func()**

=>

Output: Spam

Explanation:

* The code defines a function func().
* Inside func(), a local variable X is defined and assigned the string value 'NI'.
* Within func(), another function named nested() is defined.
* Inside nested(), the nonlocal X statement is used to indicate that X is a nonlocal variable, referring to the X variable in the immediate enclosing scope, which is the func() function.
* The line X = 'Spam' assigns the string value 'Spam' to the nonlocal variable X within func().
* When nested() is called within func(), it executes the assignment statement X = 'Spam', modifying the value of the nonlocal variable X.
* After nested() is executed, the print(X) statement within func() is executed, resulting in the output 'Spam'.
* The reason for this output is that the nonlocal keyword allows the inner nested() function to modify the variable X in the outer func() function's scope.