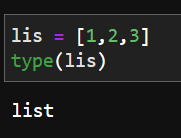
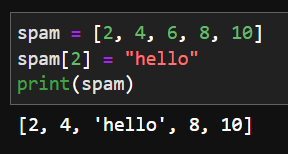
**1. What exactly is []?**

Ans: [] are square brackets which are used for to represent lists in python. For example :



Here [] is used to define the a new list called lis.

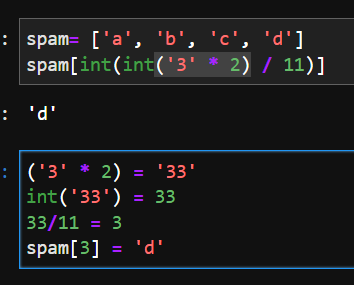
**2. In a list of values stored in a variable called spam, how would you assign the value 'hello' as the third value? (Assume [2, 4, 6, 8, 10] are in spam.)**

Ans : 

**Let's pretend the spam includes the list ['a', 'b', 'c', 'd'] for the next three queries.**

**3. What is the value of spam[int(int('3' \* 2) / 11)]?**

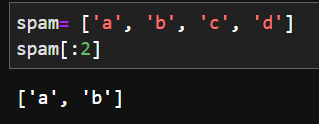
Ans:



**4. What is the value of spam[-1]?**

Ans: ‘d’

**5. What is the value of spam[:2]?**

Ans : 

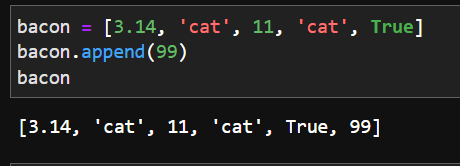
‘c’ not included as the element at index 2 is excluded and all before it is printed

**Let's pretend bacon has the list [3.14, 'cat,' 11, 'cat,' True] for the next three questions.**

**6. What is the value of bacon.index('cat')?**

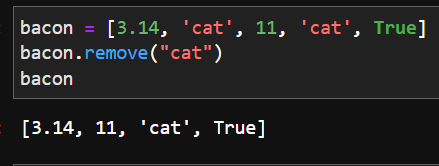
Ans: 1

**7. How does bacon.append(99) change the look of the list value in bacon?**

Ans : 

**8. How does bacon.remove('cat') change the look of the list in bacon?**

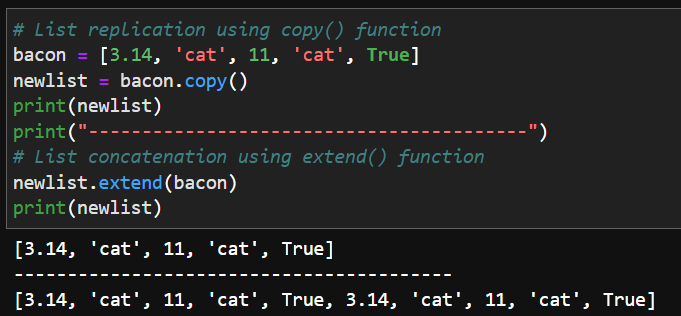
Ans:



The second “cat” is not removed as the remove() function removes only the first value it encounters

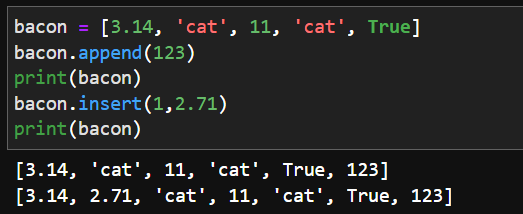
**9. What are the list concatenation and list replication operators?**

Ans:



**10. What is difference between the list methods append() and insert()?**

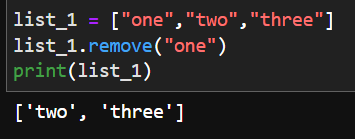
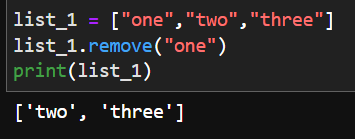
Ans:



Append() adds the element to the end of the list whereas using insert() we can add the element as a specific position as we desire.

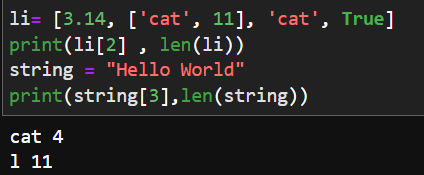
**11. What are the two methods for removing items from a list?**

Ans: remove() and pop().

**12. Describe how list values and string values are identical.**

Ans:



They both have a certain length and values that can be accessed using [] brackets and the idea of indexing.

**13. What's the difference between tuples and lists?**

Ans:

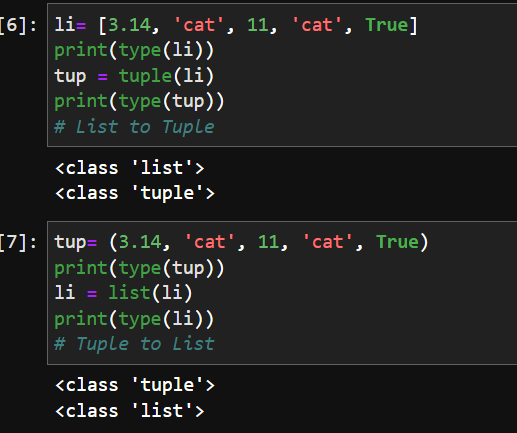
|  |  |
| --- | --- |
| List | Tuple |
| Mutable : they can be changed | Immutable : the values cannot be changed |
| List is defined using [] brackets | Tuples are defines using () brackets |

**14. How do you type a tuple value that only contains the integer 42?**

Ans: 

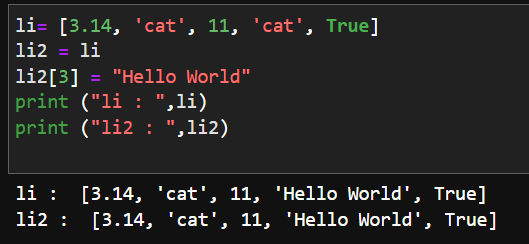
**15. How do you get a list value's tuple form? How do you get a tuple value's list form?**

Ans:



**16. Variables that "contain" list values are not necessarily lists themselves. Instead, what do they contain?**

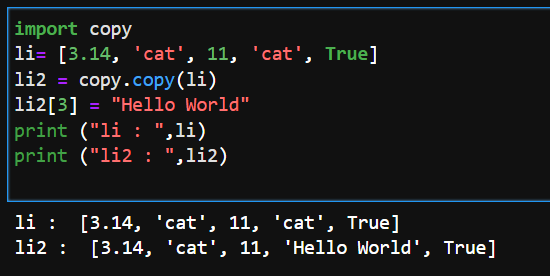
Ans:



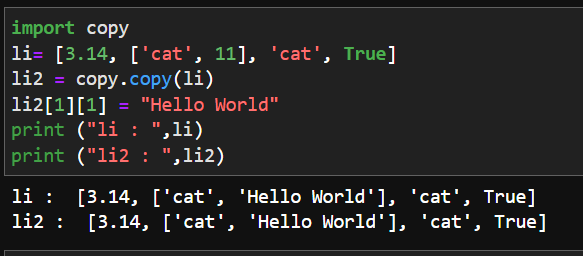
From this example we can see that even though the value of li2 is changed at a particular index the value of the original list li changes too. This is because the list li2 is not a copy that just “contain” the list values of li. In Python the values of list when copied get passed on as **references** and that is why the value of the li changes when li2 is changed.

**17. How do you distinguish between copy.copy() and copy.deepcopy()?**

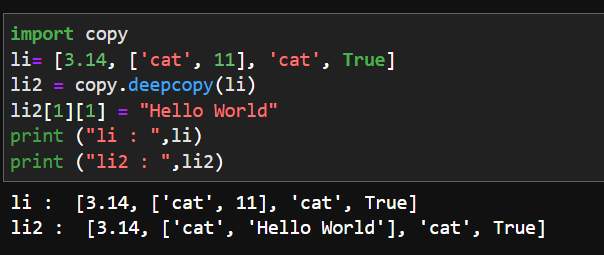
Ans:



The copy.copy() function lets us make a make a copy of a list without having a reference to the previous list which as a result let us change the value of new list. This can be seen above. However we have a problem if we have a list inside another list.

 The problem

To solve this problem we use the deepcopy() function which makes sure that even lists inside the list have a separate copy of themselves and not just references which can get changed like above.



This is how deepcopy() solves our problem.