1. What exactly is []?

Ans:- The empty list value, which is a list value that contains no items. This is similar to how '' is the empty string value.

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.)

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

Ans:- spam[2] = 'hello' (Notice that the third value in a list is at index 2 because the first index is 0.)

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

Ans:- 'd' (Note that '3' \* 2 is the string '33', which is passed to int() before being divided by 11. This eventually evaluates to 3. Expressions can be used wherever values are used.)

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

Ans:- 'd' (Negative indexes count from the end.)

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

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

Ans:- ['a', 'b']

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:- [3.14, 'cat', 11, 'cat', True, 99]

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

Ans:- [3.14, 11, 'cat', True]

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

Ans:- The operator for list concatenation is +, while the operator for replication is \*. (This is the same as for strings.)

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

Ans:- The only difference between append() and insert() is that insert function allows us to add a specific element at a specified index of the list unlike append() where we can add the element only at end of the list.

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

remove() It helps to remove the very first given element matching from the list.

pop() The pop() method removes an element from the list based on the index given.

clear() The clear() method will remove all the elements present in the list.

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

Ans:- The similarity between Lists and Strings in Python is that both are sequences. The differences between them are that firstly, Lists are mutable but Strings are immutable. Secondly, elements of a list can be of different types whereas a String only contains characters that are all of String type.

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

Ans:- The key difference between the tuples and lists is that while the tuples are immutable objects the lists are mutable. This means that tuples cannot be changed while the lists can be modified

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

Ans:- Also, tuples are written using parentheses, ( and ), while lists use the square brackets, [ and ]. How do you type the tuple value that has just the integer value 42 in it? (42,) (The trailing comma is mandatory.)

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

Ans:- Example code

#Convert list to tuple

listx = [5, 10, 7, 4, 15, 3]

print(listx)

#use the tuple() function built-in Python, passing as parameter the list

tuplex = tuple(listx)

print(tuplex)

Output

[5, 10, 7, 4, 15, 3]

(5, 10, 7, 4, 15, 3)

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

Ans:- Variables will contain references to list values rather than list values themselves. But for strings and integer values, variables simply contain the string or integer value.

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

Ans:- copy() create reference to original object. If you change copied object - you change the original object. . deepcopy() creates new object and does real copying of original object to new one.