1. What exactly is []?

Solution:

[] means that it is an empty list.

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.

Solution For the following three questions, lets say spam contains the list ['a', 'b','c','d'].

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)]?

Solution 'd' (Note that '3' * 2 is the string '33', which is passéd to int() before being divided by 11. This eventually evaluates to 3 Expressions can be used wherever it is used.)

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

Solution [-1] (Negative indexes count from the end of the list)

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.

Solution:

For the above three questions, lets say bacon contains the list [3.144, 'cat', 11, 'cat', True].

['a', 'b]

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

Solution : 1

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

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

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

Solution: [3.14, 11, 'cat', True]

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

Solution: 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()?

Solution:

append() will add values only at the end of the list, insert() can add them anywhere in the list.

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

Solution: The two methods for removing items from a list:

1) remove(): It helps you to remove the very first given element matching from the list.
2) Pop(): The pop() method removes an element from the list based on the index given.
3) Clear(): The clear() method will remove all the elements present in the list.
4) Del: To remove an element from a list, you can use del keyword followed by a list. you have to pass the index of the element to the list the index starts with zero.

del list[index]

12)Describe how list values and string values are identical

Solution:

The similarity between lists and tuples are list are mutable while tuples are immutable and secondly

Elements of the list can be of different types where as a string only contains characters that are all of string type.

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

Solution: Difference between a tuple and a list

List	tuple						
List are mutable	tuple are immutable						
List iteration is slower	Tuples processing is faster						
Lists consumes more memory	Tuple consumes less memory						
Operations like insertion and deletion are better performed	Elements can be accessed better						

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

Solution: (42),

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

Solution: We can get a list value's tuple form by using the following methods:

Using zip() function

Using zip() and iter() method

Using map() function

Using list comprehension and tuple method.

We can get a tuples value's list form by using:

List comprehension

Using iteration

Using sum

Using operator and reduce

Using lambda

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

Solution: Variables that "contain' list values are not necessarily list themselves. Instead, they will contain references to list values rather than list values themselves.

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

Solution:

The main difference between copy.copy() and copy.deepcopy() are as follows:

copy.copy() constructs a new command object and then(to the extend possible)

inserts references into it to the objects found in the original.

copy.deepcopy(): constructs a new compound object and then, recursively, inserts copies into it of the objects found in the original.