1. What exactly is []?

Answer: 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.)

Answer: spam.insert(2,"hello")

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)]?

Answer: ‘d’

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

Answer: ‘d’

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

Answer: [‘a’ , ’b’]

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')?

Answer: 1

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

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

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

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

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

Answer: + is used for concatenation and \* is used for replication.

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

Answer: append() – add new element at the end of list.

Insert()= add new element at given index value.

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

Answer: remove(), pop()

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

Answer: both list and string have index values.

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

Answer: tuples are immutable list of objects. Lists are mutable list of objects.

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

Answer: a = tuple([42])

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

Answer:

t = tuple(bacon)

l = list(t)

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

Answer: they contain References, python uses references whenever variables store values of mutable data types, such as list or dictionary.

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

Answer: copy.copy() - Insert reference into it to the objects found in the original.

copy.deepcopy()?- Insert copies into it to the objects found in the original.