1. What exactly is []?

[] is the 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.)

spam = [2, 4, 6, 8, 10]

print(spam)

spam[2] = 'hello'

print(spam)

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)]?

spam = ['a', 'b', 'c', 'd']

spam[int(int('3'\*2)/11)]

'd'

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

spam[-1] 🡪 'd'

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

spam[:2]

[['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')?

bacon = [3.14, 'cat', 11, 'cat', True]

bacon.index('cat') 🡪 1

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

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

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

bacon = [3.14, 'cat', 11, 'cat', True]

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

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

List concatenation operator is ‘+’

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

Append method appends the given object to the end of the list.

Insert method accepts two arguments, index at which the element needs to be inserted and the object we want to insert.

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

List.remove(object) 🡪 Removes the first occurrence of the given object from the list

List.pop(index = -1) 🡪 Removes and returns item at the index of the last element

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

list is a collection of values in Python, string is a collection of characters. Both list and strings are iterable in Python

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

Tuples are immutable where as lists in Python are mutable. So once created, the items inside a tuple cannot be changed.

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

tup = tuple([42])

print(tup)

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

b = tuple([42, 3, 5])

type(b) 🡪 tuple

c = list(b)

type(c) 🡪 list

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

Variables that "contain" list values in Python do not actually contain the list itself. Instead, they contain a reference or pointer to the list object in memory.

In Python, variables are essentially labels or names that refer to objects in memory. When a variable is assigned a list value, it does not directly store the list elements but rather stores a reference to the memory location where the list object is stored.

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

deepcopy: In case of deep copy, a copy of object is copied in other object. It means that any changes made to a copy of object do not reflect in the original object.

shallow copy: In case of shallow copy, a reference of object is copied in other object. It means that any changes made to a copy of object do reflect in the original object.