

1. What exactly is []?

This implies 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.)

```
spam[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)]?

'd'

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

'd'

5. What is the value of 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')?

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?

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

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

+ is the list concatenation and * is the list replication operators.

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

append() function is used to add elements at the end of the list while insert() is used to insert the elements at a specified position

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

Remove() and pop() are the two methods for removing items from a list.

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

List and String are sequences. List is a sequence of different datatype while string is a sequence of characters.

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

Lists are mutable while tuples are immutable.

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

(42,)

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

Using zip and iter function.

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

This will contain reference to list values.

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

A shallow copy means constructing a new collection object and then populating it with references to the child objects found in the original. In essence, a shallow copy is only one level deep. The copying process does not recurse and therefore won't create copies of the child objects themselves.

A deep copy makes the copying process recursive. It means first constructing a new collection object and then recursively populating it with copies of the child objects found in the original. Copying an object this way walks the whole object tree to create a fully independent clone of the original object and all of its children.