

1. What exactly is []?

Ans: It is used for creation of 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.)

Ans: `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)]`?

Ans: 'd'

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

Ans: 'd'

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

Ans: ['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')`?

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, 99]

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

Ans: The operator for list concatenation is +, while the operator for replication is \*.

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

Ans: `Append()` is used to add element at the end of the list and `insert()` is used to insert a element in specific position in list.

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

Ans: `remove()`, `pop()`

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

Ans: Both are iterable, both the elements can be accessible by index, slicing is possible. Each character in a string as well as each item in a list has a position, also called an index.

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

Ans: List is mutable and tuple is immutable.

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

Ans: 42

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

Ans: tuple(list), list(tuple)

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

Ans: Tuple

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

Ans: The copy() returns a shallow copy of the list, and deepcopy() returns a deep copy of the list. In the case deepcopy() any changes in the copy of the object that will not reflect in the original object.