1. What exactly is []?

**Ans**: It is an empty list.

1. 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[2] = ‘hello’

1. Let's pretend the spam includes the list ['a', 'b', 'c', 'd'] for the next three queries.

What is the value of spam[int(int('3' \* 2) / 11)]?

**Ans**: ‘d’

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

**Ans**: ‘d’

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

**Ans**: [‘a’, ‘b’]

1. Let's pretend bacon has the list [3.14, 'cat,' 11, 'cat,' True] for the next three questions.

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

**Ans**: 1

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

**Ans**: [3.14, 'cat,' 11, 'cat,' True, 99]

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

**Ans**: It removes 1st occurrence of ‘cat’.

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

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

**Ans**: List concatenation is done using the ‘+’ operator, and list replication is done using the ‘\*’ operator.

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

**Ans**: Append always adds value at end of list whereas insert adds value to the list at specified index.

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

**Ans**: The two methods are,

remove(): removes the first occurrence of a specified value.

pop(): removes an element at a specified index, or the last element if no index is specified.

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

**Ans**: Both list values and string values are sequences of items. They support indexing, slicing, and iteration.

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

**Ans**: Lists are mutable, while tuples are immutable. Tuples are defined using parentheses ().

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

**Ans**: (42,)

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

**Ans**: Given list is my\_list and tuple is my\_tuple, below is how we get each others form.

List(my\_tuple)

Tuple(my\_list)

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

**Ans**: They contain references to the list objects

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

**Ans**: copy.copy() creates a shallow copy of an object, meaning it creates a new object but doesn't recursively copy nested objects.

copy.deepcopy() creates a deep copy, copying all objects recursively, even nested ones