1. What exactly is []?

**Ans:** 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=[2, 4, 6, 8, 10]

spam[2]='hello'

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

1. 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’]

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

1. 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:** [3.14, 11, 'cat', True, 99]

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

**Ans:** List concatenation operators: +

List replication operators: \*

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

**Ans**: With insert method, we can add an element to a list at any arbitrary position.

Eg:

bacon.insert(3,'guitar')

With append method, we can add an element only in the end.

Eg:

bacon.append('Audi')

Output:

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

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

**Ans:** remove and pop

Eg:

bacon.remove('cat')

bacon.pop(1)

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

**Ans:** Following operations make lists and string almost identical:

* Concatenation
* Can be used in for loops
* Can be sliced (string slicing and list slicing)
* Have indexes.

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

**Ans:**

* Lists are mutable. Tuples are immutable.
* Lists consume more memory. Tuples on the other hand consume less memory.
* Lists have several built-in methods. Tuple does not have many built-in methods.
* Operations involving iterations for lists are time-consuming. For tuples it is comparatively faster.

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

**Ans:**

tup=(42,)

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

**Ans:** By using the following commands:

tuple()

list()

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

**Ans**: They contain references to list values.

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

**Ans:** The copy.copy() function will do a shallow copy of a list, while the copy.deepcopy() function will do a deep copy of a list. 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. 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.