1. What exactly is []?

**Answer: [] is empty list in python It is a fundamental data structure in Python and can be used to store collections of elements.**

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.)

**Answer: we can use square bracket notation to access the third element which is 2nd index and then assign the value 'hello' to it. Here's an example:**

**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.

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

**Answer: the value of [int(int('3' \* 2) / 11)] will be 3 as ‘3’\*2 is equals to ‘33’ then it will be converted into int then 33 /11 which is equal 3**

**And we know spam = ['a', 'b', 'c', 'd'] Therefore value of spam[3] is ‘d’**

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

**Answer: the value of spam[-1] is ‘d’ as negative indexing starts from end**

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

**Answer: The value of spam[:2] is ['a', 'b']. It returns the values from index 0 to index 1 (as index 2 is excluded).**

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')?

**Answer: The value of bacon.index(‘cat’) is 1 it returns the index of first occurrence ‘cat’ element**

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

**Answer: bacon.append(99) adds int value 99 to the end of bacon list so the updated bacon list is [3.14, 'cat', 11, 'cat', True, 99].**

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

**Answer: bacon.remove('cat') removes the first occurrence of the 'cat' from the bacon list. so the updated bacon list is [3.14, 11, 'cat', True].**

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

**Answer: The list concatenation operator is +, which joins two lists, and the list replication operator is \*, which creates a new list by repeating the existing list number of times."**

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

**Answer: append(): This method adds a new element to the end of a list for example my\_list.append('new element') will append the new element to the end of the list.**

**insert(): This method inserts a new element at a specified index of the list for example my\_list.insert(2, 'new element') takes the first argument as an index where the new element should be inserted.**

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

**Answer:**

**remove(): removes the first occurrence of an element from the list.**

**pop(): removes the element at a specified index from the list and returns the removed element. If no index is specified, it removes and returns the last element of the list.**

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

**Answer: List values and string values are similar in some ways. Both are ordered sequences of values. Both can be indexed and sliced (strings are immutable, but we can create a new string using the existing string). Both can be concatenated and replicated using operators. Both have a len() function and can be iterated over using a loop.**

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

**Answer: The main difference is that lists are mutable, meaning their values can be changed, added or removed, while tuples are immutable and cannot be modified once created. Additionally, lists are defined using square brackets [ ] and tuples are defined using parentheses ( ).**

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

**Answer: A comma is needed after the value to indicate that it is a tuple. To create a tuple with the integer 42, we can use the following code: t = (42,)**

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

**Answer: To get a list values tuple form, we use the tuple() function and pass the list as an argument. For example, tuple([1, 2, 3]) will return (1, 2, 3). To get a tuple value's list form, we use the list() function and pass the tuple as an argument. For example, list((1, 2, 3)) will return [1, 2, 3]**

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

**Answer: Variables that "contain" list values are references to the list object. They do not actually contain the list itself, but rather a pointer to where the list is stored in memory.**

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

**Answer: The copy.copy() creates a shallow copy of a list, meaning it only copies the top-level elements of the list and any nested elements are still referenced. Changes to the nested elements will affect the original list. The copy.deepcopy() creates a deep copy of a list, meaning it creates new copies of all the nested elements as well, so changes to nested elements do not affect the original list.**