**1. What exactly is []?**

**Ans.**

[] is 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.)**

**Ans.**

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)]?**

**Ans.**

The value of **spam[int(int('3' \* 2) / 11)]** will be **d** because the evaluated index in the spam given above will be 3 and the in the list, the value that present in 3rd index is **d.**

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

**Ans.**

The value of spam[-1] will be **d.**

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

**Ans.**

The value of spam[:2] will be **[‘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.**

The value of bacon.index(‘cat’) will be **1**.

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

**Ans.**

After appending **99** to the list bacon, the list will be **[3.14, 'cat,' 11, 'cat', True,99]**.

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

**Ans.**

After removing ‘cat’ from the bacon list the list will be look like **[3.14, 11, 'cat', True,99]**.

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

**Ans.**

‘**+**’ is the list concatenation operation and ‘**\***’ is the list replication operation.

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

**Ans.**

Append() will a value to the end of a list and insert() can add a value anywhere in a list if we pass an index to insert() method.

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

**Ans.**

Remove() , pop() and del. are the three methods for removing items from a list

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

**Ans.**

list values and string values are identical because,

* Both list and string have index and slices so both are iterable.
* Some operations like **in**, **not in, concatenation, replication** can be applicable in both of them.
* len() is also applicable in both of the case.

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

**Ans.**

The difference between tuples and lists is, lists are mutable and tuples are 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.**

we can get a list values in tuple form by tuple() function and tuple values in list form by list() function.

Ex- lis=[1,2,3,4,5,6] # type(lis) = list

Tup = tuple(lis)

o/p- (1,2,3,4,5,6) # type(Tup) = tuple

And

Lis = list(Tup)

o/p- [1,2,3,4,5,6] # type(Lis) = list

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

**Ans.**

Variables that "contain" list values are not necessarily lists themselves. Instead, they contain references to list values.

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

**Ans.**

The **copy.copy()** function will do a shallow copy of a list and the **copy.deepcopy()** function will do a deep copy of a list.

That is only the copy.deepcopy() can duplicate nested lists.