

1. What exactly is []?

[] → denotes empty list in python

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.)

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)]?

We got **d** as output → spam[int(int('3' * 2) / 11)] is **equal to spam[3]**

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

Output is **'d'**

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

Output is **['a', 'b']**

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

bacon = [3.14, 'cat', 11, 'cat', True]

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

Index of **bacon.index('cat')** is **1**

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

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

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

bacon = [3.14, 11, 'cat', True]

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

The operator for list concatenation is **‘+’**, while the operator for replication is **‘*’**.

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

Insert() function will insert the data at **particular Index** and whereas **append() function** will append **the data at last position in list**

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

The two methods to removing the data in list are **remove()** and **pop()**

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

Strings can only consist of characters, while lists can contain any data type. **But while accessing the data(traversing the data) is identical** for both list and strings.

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

In list we mutable(we can insert the values) and tuples are immutable we can't insert the data.

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

```
t = (42)
```

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

Tuple to list

```
my_tuple = (1,2,3,4,5,6)
```

```
my_list = list(my_tuple)
```

list to tuple

```
my_list = [1,2,3,4,5,6]
```

```
my_tuple1= tuple(my_list)
```

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

Variables will **contains reference to lists**

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

The `deepcopy()` will create a different memory location and so if we update any data in one list, it won't update in another list

`Copy.copy()` will be referring to same location. So if we update one list and other list data also updated.