1. What exactly 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.)

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)]?

‘d’

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

‘d’

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

['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')?

1

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

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

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

The only difference between append() and insert() is that insert function allows us to add a specific element at a specified index of the list unlike append() where we can add the element only at end of the list.

Ex:

**Syntax: List\_Name.append(item)**

**Syntax: List\_Name.insert(index, item)**

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

Actually there are three. Those methods are **remove(), pop() and clear()**.

**remove()** helps to remove the very first given element matching from the list.

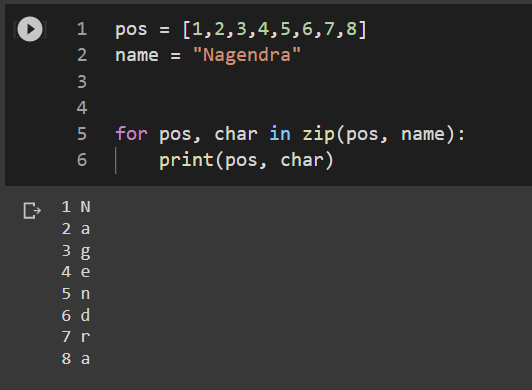
The **pop()** method removes an element from the list based on the index given or last element if no index given.

The **clear()** method will remove all the elements present in the list.

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

Both a list and a string are ordered sequences. But in list we have different data types as sequence members whereas string has only characters in sequence members. We can iterate through both them using for loop and we can access elements in it using indexing. In a broader way we can assume that a string is a list of characters.

Ex:



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

**Lists are mutable whereas tuples are immutable**. That basically means we can make changes in an existing list but we cannot in a tuple. The tuple is faster than the list because of static in nature.

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?

**list((1,2,3,4))**

**tuple([1,2,3,4])**

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

Variables will contain references to list values rather than list values themselves. But for strings and integer values, variables simply contain the string or integer value.

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

copy() create **reference to original object**. If you change a copied object - you change the original object. . deepcopy() creates a new object and **does real copying of original object to new one**. Changing a new deep copied object doesn't affect the original object.