1. What exactly is []?

- This is the notation used when we write the list. The representation for writing the elements as a list is to write in a Square brackets as shown.

1. 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, 4, 6, 8, 10]

spam[2] = 'hello'

- We can add the Hello as the third value using indexing function.

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)]?

- ‘d’

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

- ‘d’

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

- ‘c’

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')?

1

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

- If we use bacon.append(99)

List will look look like as shown below.

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

**Note - If we use append function the value will be added automatically as last element in a list.**

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

- **bacon = [3.14, 11, 'cat,' , True]** - For List mentioned in question.

If we use the appended list mention in question no 7.

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

**This will remove only the first occurrence of the cat in the list.**

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

-List concatenation operator is done using + operator . when the two list of any element say heterogeneous to be added the concatenation operator is used.

- List replication operator is denoted by \* operator when the new list to be created by exactly replicating the elements in the original list this operator is used.

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

- In list method append () . The element in the list automatically to the last element (position) in the list.

- Whereas, In insert() . We can add the element at the specific index. It takes two arguments, index to need to be added and the element.

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

- We can use the **pop** method and **remove** method. We can also use **del** keyword to remove the element from the list.

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

- The list and strings values are identical in below ways.

* Both the list and string are placed in sequential order. We can access the elements in list or we can access the string char using the slicing and indexing.
* We can access the elements in list and can access the characters in the string using the [] square brackets.
* We can use for loop for both the list and strings as they allows the iteration for the elements and characters respectively.

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

- List : They are the number of elements assigned to variable. They are written or declared in a square brackets. They are mutable & heterogeneous in nature. List are more preferred in collection of various elements as we can perform many actions such as add, remove as they are mutable.

-tupples : They are the number of elements assigned to variable. They are written or declared in bracetts (). They are immutable & heterogeneous in nature. Tupples has a advantage of high performance as they are immutable.

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

aniket = (42,)

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

**A list to a tuple:**

aniket = [a,b,c,d]

result = tuple(aniket)

Print(result)

# Output: (a,b,c,d)

**A tuple to a list:**

aniket = (1, 2, 3, 4)

result = list(aniket)

print(result)

#Output: [1, 2, 3, 4]

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

- Variables doesn’t directly contains the list. If we assign the list to variable. Variable store the reference of memory location not a list.

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

- In copy.copy() if we create a copy of original list using copy.copy() and if we modified the newly copied list it will affect the original list as well. It is shallow copy.

- In copy.deepcopy() if we create a copy of original list using copy.deepcopy() and if we modified the newly copied list it will not affect the original list as well. It is deep copy. Elements in new Copies are independent of original elements in a list.