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

[] is an empty list. [] are used to create a list.

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.)**

I will simply run the following:

spam= [2,4,6,8,10]

for i in range(len(spam)):

if type(spam[i])==int:

spam.append(str(spam[i]))

for i in range(len(spam)):

if type(spam[i])==int:

spam.pop(i)

spam.append(’hello’)

**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)]?**

This will give us value of ‘**d**’

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

It would return the value as “**d**”

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

It would return the values as **‘a’,’b’**

**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')?**

There are four data types inside this list. Therefore, the list statement would not execute.

If we change the datatypes to string format then the answer would be

**1**

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

As I am assuming that I have changes the elements inside bacon to string format. Therefore, I am using this statement **bacon.append(‘99’)**

this would return the following value:

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

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

As I am assuming that I have changes the elements inside bacon to string format. Therefore, running bacon.remove(‘cat’) will give us the following output:

[‘3.14’,’11’,’cat’,’True’]

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

For list concatenation we can use:

1. “**+**” operator
2. append()
3. extend()

for replication we use:

1. “**\***” operator
2. sorted()
3. **What is difference between the list methods append() and insert()?**

**append()** inserts the value at the end of the list whereas with **insert()**, we can specify the index for the value where we want it to be inserted.

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

Two methods are:

1. pop()
2. remove()
3. **Describe how list values and string values are identical.**

Lists and string values are identical if we convert its values to set. This will remove all the duplicate values in the list/string.

1. **What's the difference between tuples and lists?**
2. In lists we can modify the data inside, like we can perform operations like add, delete elements. Whereas in tuples we cannot perform such operations.

Therefore, we say that the lists are mutable, and tuples are immutable.

1. Lists are created with square braces while tuples are created with parenthesis.
2. Lists consumes more memory than tuple.

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

Tup=(42)

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

List to tuple:

tuple(list\_name)

tuple to list:

list(tuple\_name)

1. **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.**

If the variables are stored in a mutable object, then python uses variable referencing.

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

**copy.copy()** will simply create a shallow copy of the variable, which can be modified if there is modification in the original part.

While **copy.deepcopy()** will create a deep copy which cannot be modified even if there is a change in the original part.