1. What exactly is []?

**Solution1:-**

Type of [] is list in python.

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.)

**Solution2:-**

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

spam[2]="hello"

spam

output:- [2, 4, 'hello', 8, 10]

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)]?

**Solution3:-**

Value of spam[int(int('3' \* 2) / 11)] is 3rd index value i.e. “d”.

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

**Solution4:-**

Value of spam[-1] is “d”

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

**Solution5:-** ['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')?

**Solution6:-** index is 1 because it will give index of 1st element if same elements are present in the list.

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

**Solution7:-** List will look like this [3.14, 'cat', 11, 'cat', True, 99].

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

**Solution8:-** List will look like this [3.14, 11, 'cat', True, 99].

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

**Solution9:-**

Consider two lists:

a = [1, 2, 3]

b = [4, 5, 6]

If we need to concatenate them into a list, we could use the + operator. This will create a new list c:

c = a + b

c

[1, 2, 3, 4, 5, 6]

You can also use the in-place version += which will effectively extend the list with the contents of b:

a += b

a

[1, 2, 3, 4, 5, 6]

If you need to replicate the list a say 3 times, you could use the \* operator. This will create a new list d:

d = a \* 3

d

[1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]

You can also use the in-place version \*= which will modify the original list (a) by storing the replicated result into itself:

a \*= 3

a

[1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6, 1, 2, 3, 4, 5, 6]

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

**Solution10:-**

The difference is that with append, you just add a new entry at the end of the list. With insert(position, new\_entry) you can create a new entry exactly in the position you want.

Example:

Spam=[1,2,3,4]

Spam.append(99)

spam

Output:-[1,2,3,4,99]

Spam.insert(0,”hello”)

Spam

Output:[‘hello’,1,2,3,4]

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

**Solution11:-**

spam.remove() will remove the very first given element matching from the list.

Spam.pop() will removes an element from the list based on the index given

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

**Solution12:-**

The similarity between Lists and Strings in Python is that both are sequences.

Spam=[1,2,3,4,”hello”]

Str=”my name is khan”

In the list spam elements 1,2,3,4,hello are separated by comma and can be extracted one by one.

Whereas in the string str , every character is an element including space as well.

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

**Solution13:-**

List elements are bounded by [] and tuples elements are bounded by ().

The key difference between the tuples and lists is that while the tuples are immutable objects the lists are mutable. This means that tuples cannot be changed while the lists can be modified. ... If you have data which is not meant to be changed in the first place, you should choose tuple data type over lists.

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

**Solution14:-**

tp=(42,)

type(tp)

output: tuple

comma is mandatory.

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

**Solution15:-**

We can use list(tuple name) to convert tuple into list

And tuple(list name) to convert list into tuple

tpl=tuple(spam)

type(tpl)

output:tuple

lst=list(tpl)

typle(lst)

output:list

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

**Solution16:-**

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.

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

**Solution17:-**

copy.copy() will create a shallow copy. It creates a new variable and any changes made to this variable will also reflect in the original data (mutable)

copy.deepcopy() will create a deep copy. It creates a copy of original data and any changes made to this variable will NOT reflect in the original data.