1. What exactly is []?

Ans:

“[]” it is an empty list. We can add, remove, and modify items in 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.)

Ans:

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

spam[2] = 'hello'

Print(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)]?

Ans:

The value of spam[int(int('3' \* 2) / 11)] is 'd'

Explanation:

'3' \* 2 creates the string '33' (here 33 is in string)

int('33') converts the string to an integer, which is 33

33 / 11 is 3.0 which is float value

Now we need to convert float into integer,so int(3.0) is 3

spam[3] is 'd' (3 is the index of value d)

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

Ans:

Negative indexing is used to access elements from the end of a list. -1 refers to the last element in the list, which is 'd'.

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

Ans:The value of spam[:2] is ['a', 'b'] (by slicing method).

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')?

Ans: The value of bacon.index('cat') is 1.

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

Ans: After append(99) to bacon list, the list value becomes [3.14, 'cat', 11, 'cat', True, 99]

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

Ans:

After removing “cat” from bacon list, the list value becomes [3.14, 11, 'cat', True] (the first occurrence of `'cat'` is removed from the list)

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

Ans:

The list concatenation operator is “+” which combines two lists into a new list.  
The list replication operator is “\*” which creates a new list by repeating an existing list a specified number of times.

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

Ans:

The difference between append() and insert() is, append() adds an element to the end of a list, whereas insert() adds an element to a specific position in the list.

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

Ans:

The two methods for removing items from list are “remove()” and “pop()”

The remove() method removes a specified element from a list, whereas the pop() method removes an element from a list at a specified index position and shows the removed element.

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

Ans:

Both lists and strings can be indexed and sliced to access individual elements.Also calculating length of the elements.

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

Ans:

The difference between tuples and lists is that tuples are immutable, while lists are mutable. This means that once you create a tuple, you cannot change its contents, while with a list you can add, remove, or modify elements.

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

Ans:

Keep comma at the end that indicates, this is a tuple with only one element.

Tuple1=(42,)

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

Ans:

We can convert a list into a tuple by using the tuple() function and we can convert a tuple into a list by using the list() function.

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

Ans:

A variables that contain list values are actually pointers to the memory locations where the list objects are stored.

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

Ans:

The difference between copy.copy() and copy.deepcopy() is copy.copy() means shallow copy of an object creates a new object with the same contents as the original object. However, the new object contains references to the same objects as the original object.Whereas copy.deepcopy means deep copy of an object creates a new object with the same contents as the original object. However, the new object contains new objects that are copies of the objects in the original object.