1. What exactly is []?

Ans: Empty 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: l = [2,4,6,8,10]  
l[2] = "hello"  
print(l[2])

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: ‘d’

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

Ans: ‘d’

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

Ans: [‘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')?

Ans: 1

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

Ans: [3.14, 'cat', 11, 'cat', True, 99] adds 99 at the end of the list

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

Ans: [3.14, 11, 'cat', True] removes available ‘cat’ string

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

Ans:

* (+) operator, using ‘extend’, (\*) operator for concatenation of lists
* (\*=) operator

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

Ans: Append only allows us to append the element at the end of the list. Whereas insert allows us to insert the element at any index in the said list.

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

Ans:

* Remove an item by index and get its value: pop()
* Remove an item by value: remove()

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

Ans: both are sequences of data.

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

Ans: Tuples are immutable, whereas lists are mutable, and this is the main distinction between the two. That is the values in a list can be changed or modified, while the values of a tuple cannot.

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

Ans: tuple\_b = (42)

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

Ans: tuple to list

#tuple b  
b = (42,4,5,6,7)  
# list a  
a = list(b)

List to tuple

# list a  
a = [2,3,4,5,6]  
# tuple c  
c = tuple(a)

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

Ans: The variables contain the references to the list values rather than the values themselves

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

Ans: copy.copy() is shallow copy: after running this function, the copied object only copies the reference to the original object and any changes made in the original object reflects in the copy object. In case of deep copy, the new references are created and each sub element is copied to the new references, therefore, in case we make any changes in the original object will not reflect in the copied object.