1. What exactly is []?

*It’s a list to store multiple items in a single variable.*

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.insert(2,’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)]?

*d*

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

*d*

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

*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')?

*1*

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

*[3.14, 'cat’, 11, 'cat,' True,99]*

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

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

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

*List concatenation => +*

*List replication => copy ()*

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

*Append() add the new item to the end of list and insert() is used to insert new item at a specified index.*

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

*Remove(), pop(), del()*

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

*They are in sequence*

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

*Tuples are unchangeable and lists are changeable.*

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

*a =(42,)*

*print(a)*

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

*b = (1,2,3)*

*a =list (b)*

*print (type(a)) => list*

*c =tuple(a)*

*print (type(c)) => tuple*

*d =list(c)*

*print (type(d)) => list*

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

*Variables will contain the references.*

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

copy.copy() => Return a shallow copy

copy.deepcopy() => Return a deep copy

In a deep copy, a change made to a copy of object do not effect original object but in shallow copy it does since they are referencing to same objects found in the original.