1. What exactly is []?

[] represents an empty list in python.

2. 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[2] = 'hello'

Let's pretend the spam includes the list ['a', 'b', 'c', 'd'] for the next three queries.

3. What is the value of spam[int(int('3' \* 2) / 11)]?

‘d’

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

‘d’

5. 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.

6. What is the value of bacon.index('cat')?

1

7. 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, 99]

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

The list concatenation operator is +, which is used to combine two or more lists into a single list.

The list replication operator is \*, which duplicates a list a certain number of times.

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

The append() method adds an element to the end of the list, whereas the insert() method inserts an element at a specific index, moving the existing elements to accommodate the new element.

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

The two methods for removing items from a list are remove() and pop(). The remove() method removes the first occurrence of a specified value, while the pop() method removes the element at a specified index.

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

List values and string values are identical in the sense that they both can contain elements accessible via indices. Both lists and strings are sequences, and you can use indexing and slicing to access their elements.

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

The main difference between tuples and lists is that tuples are immutable, meaning their elements cannot be changed after creation, while lists are mutable, allowing elements to be added, removed, or modified.

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

my\_tuple = (42,)

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

my\_list = [1, 2, 3]

my\_tuple = tuple(my\_list) use the tuple() function

same for list

my\_tuple = (1, 2, 3)

my\_list = list(my\_tuple) use the list() function

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

Variables that "contain" list values are not necessarily lists themselves; they contain references to lists. In Python, variables store references to objects in memory. So, a variable that appears to contain a list actually contains a reference to the list's location in memory.

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

copy.copy() creates a shallow copy of a list, meaning it duplicates the list itself but not the elements inside it. If the list contains other mutable objects like lists or dictionaries, those inner objects will still reference the same objects in memory.

copy.deepcopy() creates a deep copy of a list, meaning it duplicates both the list and all the elements inside it. This includes nested lists and other mutable objects, creating entirely separate copies in memory.