1. What exactly is []?

ANSWER:

It is an empty list with no items in it..

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.)

ANSWER:

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

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)]?

ANSWER:

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

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

ANSWER:

the value of spam[-1] is 'd'.

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

ANSWER:

the value of spam[:2] is ['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')?

ANSWER:

bacon.index('cat') will return 1.

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

ANSWER:

bacon.append(99) changes bacon by appending 99 to its end. the list bacon will become [3.14, 'cat', 11, 'cat', True, 99].

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

ANSWER:

the list bacon will become [3.14, 11, 'cat', True].

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

ANSWER:

The list concatenation operator in Python is the plus sign (+). It is used to combine two or more lists into a single list. For example:

```
list1 = [1, 2, 3]

list2 = [4, 5, 6]

result_list = list1 + list2

print(result_list)  # Output: [1, 2, 3, 4, 5, 6]
```

The list replication operator in Python is the asterisk (*) symbol. It is used to create a new list by repeating the elements of an existing list a certain number of times. For example:

```
original_list = [1, 2, 3]

repeated_list = original_list * 3

print(repeated_list)  # Output: [1, 2, 3, 1, 2, 3, 1, 2, 3]
```

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

ANSWER:

append() adds an element to the end of the list, while insert() inserts an element at a specified index in the list.

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

ANSWER:

the two methods for removing items from a list are remove() (removes the first occurrence of a specific element) and pop() (removes and returns an element from a specified index).

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

ANSWER:

Lists are mutable collections that allow you to modify their elements, while strings are immutable sequences of characters that cannot be changed after creation.

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

ANSWER:

Tuple start and end with () however list start and end with[]

Once the tuples are declared , they can not be added or modified while lists can be added and modified.

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

ANSWER:

tuple1 = (42,)

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

ANSWER:

my_list = [1, 2, 3, 4, 5]

my_tuple = tuple(my_list) **# to get a list value's tuple form**

print(my_tuple) # Output: (1, 2, 3, 4, 5)

my_tuple = (1, 2, 3, 4, 5)

my_list = list(my_tuple) **# to get a tuple value's list form**

print(my_list) # Output: [1, 2, 3, 4, 5]

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

ANSWER:

variables that "contain" list values store references to the list's memory location rather than the list itself.

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

ANSWER:

copy.copy() creates a shallow copy, which copies the top-level object and its references to nested objects. copy.deepcopy() creates a deep copy, which recursively copies the entire object hierarchy, resulting in a completely independent copy with no shared references to nested objects.