1. What exactly is []?

[] represents an 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.)

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

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)]?

Spam=[2, 4, 6, 8, 10,['a', 'b', 'c', 'd']]

Answer =8

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

Spam=[2, 4, 6, 8, 10,['a', 'b', 'c', 'd']]

Answer =['a', 'b', 'c', 'd']

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

Spam=[2, 4, 6, 8, 10,['a', 'b', 'c', 'd']]

Answer = [2, 4]

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')?

Answer = 1

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

Answer = [3.14, 'cat', 11, 'cat', True, 99]

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

Answer = [3.14, 11, 'cat', True, 99]

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

The list concatenation operator (+) is used to combine or concatenate two or more lists into a single list.

The list replication operator (\*) allows you to create a new list by replicating the elements of an existing list a specified number of times.

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

**Append() -** The append() method is used to add an element at the end of a list

**Insert() -** method is used to add an element at a specific position within the list. It also modifies the list in place by shifting the existing elements to accommodate the new element.

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

1.remove() - method is used to remove the first occurrence of a specific value from a list

2.pop() - method is used to remove and return an element at a specified index position from a list.

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

Lists and strings have similarities in terms of their sequential nature, indexing, slicing, and iteration, they differ in mutability and the types of values they can store. Lists are mutable and can contain diverse types of objects, while strings are immutable and are specifically designed to hold textual data as a sequence of characters.

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

The main differences between tuples and lists are their mutability, syntax, usage, immutability, performance characteristics, and suitability for different scenarios. Lists are mutable and commonly used for dynamic collections, while tuples are immutable and often used for fixed data or as return values from functions.

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

tuble = (42)

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

**List values tuple form**

my\_list = [1, 2, 3, 4]

my\_tuple = tuple(my\_list)

**tuble value’s list form**

my\_tuple = (1, 2, 3, 4)

my\_list = list(my\_tuple)

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

Instead, they contain a reference to the list object in memory. In other words, the variable stores the memory address where the list is stored.

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

copy.copy() creates a shallow copy, which copies the object but not its nested objects, while copy.deepcopy() creates a deep copy, which recursively copies the object and all its nested objects, resulting in completely independent copies.

Example

original\_list = [1, 2, 3]

shallow\_copy = copy.copy(original\_list)

deep\_copy = copy.deepcopy(original\_list)

original\_list.append(4)

print(original\_list) # Output: [1, 2, 3, 4]

print(shallow\_copy) # Output: [1, 2, 3, 4]

print(deep\_copy) # Output: [1, 2, 3]