1.What exactly is []?

Answer -

The empty list value, which is a list value that contains no items.

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 = [2,4,6,8,10]
spam[2] ="hello"
print(spam)
```

output:

```
[2, 4, 'hello', 8, 10]
```

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

```
spam = ["a", "b", "c", "d"]
spam[int(int("3" * 2) / 11)]
```

output:

```
'd'
```

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

```
spam[-1]
```

output:

```
'd'
```

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

```
spam[:2]
```

output:

```
['a', 'b']
```

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

```
bacon =[3.14, "cat" , 11, "cat", True]
bacon.index("cat")
```

output:

```
1
```

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

Answer:-

```
bacon.append(99)
print(bacon)
```

output;

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

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

Answer:-

```
bacon.remove("cat")
print(bacon)
```

output:

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

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

Answer:-
The operator for list concatenation is +, while the operator for replication is *.

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

Answer -
While append() will add values only to the end of a list, insert() can add them anywhere in the list.

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

Answer:-
The del statement and the remove() list method are two ways to remove values from a list.

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

Answer:-
Both lists and strings can be passed to len(), have indexes and slices, be used in for loops, be concatenated or replicated, and be used with the in and not in operators.

13. What is the difference between tuples and lists?

Answer-
Lists are mutable; they can have values added, removed, or changed. Tuples are immutable; they cannot be changed at all. Also, tuples are written using parentheses, (and), while lists use the square brackets, [and].

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

Answer- (42)

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

Answer -
The tuple() and list() functions, respectively

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

Answer:-
Variables that "contain" list values in Python are not the actual lists themselves but rather references to the lists.
In Python, lists are mutable objects, and when you assign a list to a variable, the variable holds a reference to the memory location where the list is stored. It means that the variable doesn't store the list's actual elements directly but points to the memory location where the list data is stored.

This concept is essential to understand because it affects how variables work with lists and how they behave when passed as function arguments or assigned to other variables. When you assign a list to multiple variables, all those variables reference the same list object in memory.

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

Answer-
copy.copy() creates a shallow copy that shares references to nested objects, while copy.deepcopy() creates a deep copy that recursively duplicates all nested objects, providing a completely independent copy.