1. What exactly is []?

Ans:- [] this the declaration of 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.)

Ans:- 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)]?

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

Ans:- ‘d’

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

Ans:- [‘a’,’b’,’c’]

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')?

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

Ans:- add 99 at the end

[3,14,’cat’,11,’cat’,True,99]

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

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

Ans:-

List concatenation:- Joining of two list known as concatenation. Use of + operator.

For ex:- l1=[1,2,3,4]   l2=[5,6,7,8]

After concatenation

L1+l2 = [1,2,3,4,5,6,7,8]

List Replication :- Replication of same list more than one time is known as replication. Use of (*) operator

For ex:- l1=[1,2,3]

After replication

L1*2=[1,2,3,1,2,3]

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

Ans:-

append():- Add the value at the end of the list.

For ex:- l1=[1,2,3,4]

L1.append(5)

L1=[1,2,3,4,5]

Insert():- It also used for insert value in the list but at the particular index.

For ex:- l1=[1,2,3,4]

L1.append(1,5)

L1=[1,5,2,3,4]

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

Ans:- l=[1,2,3,4,5]

1. Pop() :- It take index as reference

For ex:-

l.pop(2) =[1,2,4,5]

2.remove():- It take value as input.

For ex:-

l.remove(4)= [1,2,3,5]

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

Ans:- Both are sequential nature

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

Ans:-

Tuples	Lists
1. Tuples are immutable	1. List are mutable
2.Tuples are faster than list because they are static	2. list is slow because list are dynamic .
3. Tuples are denoted by parenthesis	3. List is denoted by square bracket.

T=(1,2,3,4)	**L=[1,2,3,4,5]**

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

Ans:- (21,)

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

Ans:- l1=[1,2,3,4,5]

T1=tuple(l1)

Print(T1) = (1,2,3,4,5) by using tuple function we can get list values tuple form.

T2=(1,2,3,4,5)

L2=list(T2)

Print(T2)=[1,2,3,4,5] by using list function we can get tuple values list form.

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

Ans:- Variables contain references to list values rather than list values themselves.But for strings and integer values, variables simply contain the string or integer value.

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

Ans:-

Copy .copy()	Copy.deepcopy()
1. A shallow copy creates a new compound object and then references the objects contained in the original within it	**1. A deep copy creates a new compound object before inserting copies of the items found in the original into it in a recursive manner.**
2. , which means it constructs a new collection object and then populates it with references to the child objects found in the original.	**2. First constructing a new collection object and then recursively populating it with copies of the child objects found in the original**
3. . In the case of shallow copy, a reference of an object is copied into another object	**3. . In the case of deep copy, a copy of the object is copied into another object.**

4. It means that **any changes made to a copy of an object do reflect in the original object. In python, this is implemented using the “**copy()**” function.**	**4. It means that any changes made to a copy of the object do not reflect in the original object.**
5. In python, this is implemented using the “copy()**” function.**	**5. In python, this is implemented using the “**deepcopy()**” function.**