**1. What exactly is [ ]?**

These are the square brackets used in syntax to represent data type List in python. [] represents the empty list, which is data structure to store ordered collection of 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] = 'hello'

Index of the list starts from the 0. So third element will have index 2.

**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)]?**

Value of the expression '3'\*2 = '33'

int('33')/11=3

int(3)=3

spam[3] means fourth value in the list as index of the list starts from 0

So value of the

spam[int(int('3' \* 2) / 11)]='d'

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

spam[-1] = ‘d’

Negative index starts from last element from the list.

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

spam[:2]= [‘a’,’b’]

Slicing operation on list in python is given as follows,

List\_name[start\_index:end\_index:step\_value]

By default start index is 0, and step\_value is 1which moves from left to right of the list.

**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')?**

Bacon.index(‘cat’) = 1

Even element ‘cat’ is at two different positions in list but during slicing operation first element will be considered for the output, to give index of the element.

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

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

New element added at (last\_index+1) position using append operation.

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

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

First element which matches with ‘cat’ will be removed from list when sliced from left to right using remove operation.

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

List concatenation operator: + is the list concatenation operator, it joins two list together into single list,

Ex. List1=['a’,’b’,’c’]

List=['d’,’e’,’f’]

List1+List2=['a’,’b’,’c’,'d’,’e’,’f’]

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

|  |  |
| --- | --- |
| **append()** | **Insert()** |
| Adds the element at the end of the list  Ex.  List\_1=[1,2,3]  List\_1.append(’new’)  print(List\_1)  Output: [1,2,3,’append’] | Adds the element at specific index  Ex.  List\_1=[1,2,3]  List\_1.insert(1,’new’)  print(List\_1)  Output: [1,’new’,2,3] |

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

|  |
| --- |
| **remove() :**  Removes the first occurrence of a specified value as given input to remove.  Example:  List\_1= [1, 2, 3, 2, 4, 5]  list\_1.remove(2)  print(list\_1)  Output: [1, 3, 2, 4, 5] |
| **pop() :**  Removes and returns the element at a given index.  If index not given then by default it will take last element of the list  Example:  List\_1= [1, 2, 3, 2, 4, 5]  List\_1.pop(2)  print(List\_1)  Output: [1, 2, 2, 4, 5] |

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

* Both are sequences of values, accessible by indexing and slicing.
* Both support operations like concatenation and replication.

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

|  |  |
| --- | --- |
| **LIST** | **TUPLE** |
| List is a group of comma separated values within square brackets and square brackets are mandatory.  Ex. List=[1,2,3,4] | Tuple is a group of comma separated values within parenthesis and parenthesis are optional |
| Objects are mutable | Objects are immutable |
| Cannot be used as keys for dictionary | Can be used as key for dictionary |
| Used when content is not fixed and keep on changing | Used when content is fixed and not changing |
| Lists consume more memory | Tuple consumes less memory as compared to the list |

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

* Tuple\_42=(42,)

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

* List to tuple: tuple(list\_name)
* Tuple to list: list(tuple\_name)

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

They contain references to list objects in memory.

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

* copy.copy() creates a shallow copy, where the top-level elements are copied, but nested objects are shared references.
* copy.deepcopy() creates a deep copy, where all elements and nested objects are recursively copied.