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

=> set to define mutable data types - lists, index etc. it is also used for dictionaries too.

**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”. Because the first value starts from zero(0).

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)]?**

=>’d’

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

=> “d”. because negative values comes from the last.

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

=> [“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')?**

=> “1”

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

=> [3.14,’cat’,’11’,’cat’,’True’,’99’]

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

=>[3.14,’cat’,’11’,’True’]

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

=> The operator for list concatenation is +, while the operator for replication is \*. the concatenation operator connects one string at the end of the other. Replication Operator. The multiplication operator acts as a replication operator when we have one string and one integer as value.

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

=> The difference is that with append, you just add a new entry at the end of the list. With insert (position, new entry) you can create a new entry exactly in the position you want. The append method adds a new item to the end of a list.

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

=> In Python, there are many methods available on the list data type that help you remove an element from a given list. The methods are remove(), pop() and clear() .

Besides the list methods, you can also use a del keyword to remove items from a list.

e.g. list.remove(element) or list.pop(index)

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

=> A list is a sequential collection of Python data values, where each value is identified by an index. The values that make up a list are called its elements. Lists are similar to strings, which are ordered collections of characters, except that the elements of a list can have any type and for any one list, the items can be of different types.

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

=>

|  |  |
| --- | --- |
| **List** | **Tuple** |
| It is mutable. | It is immutable. |
| The implication of iterations is time-consuming in the list. | Implications of iterations are much faster in tuples. |
| Operations like insertion and deletion are better performed. | Elements can be accessed better. |
| Consumes more memory. | Consumes less memory. |
| Many built-in methods are available. | Does not have many built-in methods. |
| Unexpected errors and changes can easily occur in lists.   . | Unexpected errors and changes rarely occur in tuples.. |

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

=> (42,)

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

=> by using tuple() and list() itself only.

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

=>They contain reference to list values.

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

=> the copy.copy() will do shallow copy of a list while copy.deepcopy() will do deep copy of the list.