1. It is an empty list, ie a list consisting of no elements.
2. spam = [2, 4, 6, 8, 10]

spam[2] = 'hello'

print(spam)

----------------

Output --> [2, 4, 'hello', 8 , 10]

1. if spam = ['a', 'b', 'c', 'd']

Then spam[int(int('3' \* 2) / 11)] will give the output ‘d’.

‘3’ \* 2 --> 33 , 33/11--> 3 .so It will print the third element.

1. if spam = ['a', 'b', 'c', 'd']

then spam[-1] will give the output ‘d’

1. If spam = ['a', 'b', 'c', 'd']

Then spam[:2] will give the output [‘a’ , ‘b’]

1. If bacon= [3.14, 'cat’, 11, 'cat’, True]

Then bacon. index('cat') will give the output 1

1. If bacon= [3.14, 'cat,' 11, 'cat,' True]

Then bacon. append(99) will give the output of [3.14, 'cat', 11, 'cat', True, 99]

ie, it get appended to the existing list.

1. If bacon= [3.14, 'cat,' 11, 'cat,' True]

Then bacon.remove(‘cat ’) will give the output of [3.14, 11, 'cat', True]

1. List concatenation operator--> ‘ + ‘ ie, it adds the values of same data type.

List replication operator ---> ‘ \* ‘ , it replicates the same value for a specified number of times.

1. append() is used to add items by adding them as the last element .

insert() is used to add items anywhere in the list.

1. pop() and remove() can be used to remove items from the list. del() can be used to delete specified objects as well as a range of objects using index.
2. Similarities between **List and String**

• Supports indexing and slicing to access.

• Both can be used along with the conditional statements.

• Can be concatenated and replicated.

1. Difference between **List and Tuple**

|  |  |
| --- | --- |
| **List** | **Tuples** |
| * **Mutable** * **Created by placing items inside […] separated by commas**. | * **Immutable** * **Created by placing items inside parenthesis (…) separated by commas** |

1. Tuple = ( 42 )
2. By using the tuple() and list() functions.
3. They contain the references to list values.
4. Copy.copy() returns a shallow copy of x . ie, when you change the copied object you change the original object.

Copy.deepcopy() returns a deep copy of x .ie, changing the copied object does not effect the original object.