Python list are container to store a set of values of any data-types.

Friends=[‘kriti’,”karuna”,”mango”,4,true]

List indexing:

A list can be indexed just like a string.

L1=[3,4,”golf”]

L1[0]=3

L1[1]=>4

L1[0:2]=>[3,4]🡺list slicing

List methods::

Consider the following list:

M1=[3,2,12,76,45]

* M1.sort()=updates the list to

[2,3,12,45,76]

* M1.reverse()=updates the list to[76,45,12,3,2]
* M1.append(8)=adds 8 at the end of list

M1.insert(3,8)=this will add 8 at index 3

list.**append**(*x*)

Add an item to the end of the list. Equivalent to a[len(a):] = [x].

list.**extend**(*iterable*)

Extend the list by appending all the items from the iterable. Equivalent to a[len(a):] = iterable.

list.**insert**(*i*, *x*)

Insert an item at a given position. The first argument is the index of the element before which to insert, so a.insert(0, x) inserts at the front of the list, and a.insert(len(a), x) is equivalent to a.append(x).

list.**remove**(*x*)

Remove the first item from the list whose value is equal to *x*. It raises a [ValueError](https://docs.python.org/3/library/exceptions.html" \l "ValueError" \o "ValueError) if there is no such item.

list.**pop**([*i*])

Remove the item at the given position in the list, and return it. If no index is specified, a.pop() removes and returns the last item in the list. (The square brackets around the *i* in the method signature denote that the parameter is optional, not that you should type square brackets at that position. You will see this notation frequently in the Python Library Reference.)

list.**clear**()

Remove all items from the list. Equivalent to del a[:].

list.**index**(*x*[, *start*[, *end*]])

Return zero-based index in the list of the first item whose value is equal to *x*. Raises a [ValueError](https://docs.python.org/3/library/exceptions.html" \l "ValueError" \o "ValueError) if there is no such item.

The optional arguments *start* and *end* are interpreted as in the slice notation and are used to limit the search to a particular subsequence of the list. The returned index is computed relative to the beginning of the full sequence rather than the *start* argument.

list.**count**(*x*)

Return the number of times *x* appears in the list.

list.**sort**(*\**, *key=None*, *reverse=False*)

Sort the items of the list in place (the arguments can be used for sort customization, see [sorted()](https://docs.python.org/3/library/functions.html#sorted) for their explanation).

list.**reverse**()

Reverse the elements of the list in place.

list.**copy**()

Return a shallow copy of the list. Equivalent to a[:].

An example that uses most of the list methods:

>>>

**>>>** fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']

**>>>** fruits.count('apple')

2

**>>>** fruits.count('tangerine')

0

**>>>** fruits.index('banana')

3

**>>>** fruits.index('banana', 4) *# Find next banana starting a position 4*

6

**>>>** fruits.reverse()

**>>>** fruits

['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange']

**>>>** fruits.append('grape')

**>>>** fruits

['banana', 'apple', 'kiwi', 'banana', 'pear', 'apple', 'orange', 'grape']

**>>>** fruits.sort()

**>>>** fruits

['apple', 'apple', 'banana', 'banana', 'grape', 'kiwi', 'orange', 'pear']

**>>>** fruits.pop()

'pear'

* M1.pop(2):will delete element at index 2 and returns its value
* M1.remove(12):will remove 12 from the list

Tuples in python

A tuple is a( immutable🡺can’t change) data type in python ..

a=()🡺empty tuple

a=(1,)🡺tuple with only one element needs a comma

a=(1,2,3)🡺tuple with more than one element

Once defined a tuple element can’t be altered and manipulated

Tuples method

Consider the following tuples a=(2,3,4)

1. a.count(2):it will return number of times 2 occur in a
2. a.index(2):will return index of first occurrence of 1 in a

practice sets

* a program to store 7 countries in a list entered by the user
* program to accept marks of 6 students and display them in a sorted manner
* check that the tuple can’t be changed in python
* program to sum the list with 4 numbers
* program to count the number of 0’s in following tuple a=(3,4,5,0,4,0,4,0,0)