Sequences:

There are 2 important sequence types: lists and tuples.

Common functions of sequence types:

This table lists the sequence operations sorted in ascending priority. In the table, *s* and *t* are sequences of the same type, *n*, *i*, *j* and *k* are integers and *x* is an arbitrary object that meets any type and value restrictions imposed by *s*.

x in s	True if an item of *s* is equal to *x*, else False
x not in s	False if an item of *s* is equal to *x*, else True
s + t	the concatenation of *s* and *t*
s * n or n * s	equivalent to adding *s* to itself *n* times
s[i]	*i*th item of *s*, origin 0
s[i:j]	slice of *s* from *i* to *j*
s[i:j:k]	slice of *s* from *i* to *j* with step *k*
len(s)	length of *s*
min(s)	smallest item of *s*
max(s)	largest item of *s*
s.index(x[, i[, j]])	index of the first occurrence of *x* in *s* (at or after index *i* and before index *j*)
s.count(x)	total number of occurrences of *x* in *s*

Python lists

- Python list is the most versatile feature.
- They are written as a list of comma-separated values between square brackets.
- **Lists might contain items of different types, but usually the items all have the same type.**
- They are indexed and sliced.

- They support concatenation
- **Lists are mutable .**
- Addition of new elements are done with the help of append().

```
>>> b= [1,2,3,4,5] #a list
>>> a=[1,2,3,'coding blocks', 'mentor','jatin'] #also a list

>>> a[0] #indexed
1

>>> a[-3:] #sliced
['coding blocks', 'mentor', 'jatin']

>>> a[:]
[1, 2, 3, 'coding blocks', 'mentor', 'jatin']

>>> [5,6,7] + a #concatenation
[5, 6, 7, 1, 2, 3, 'coding blocks', 'mentor', 'jatin']

>>> a[3]= 64 #mutable
>>> print(a)
[1, 2, 3, 64, 'mentor', 'jatin']

>>> a.append('food') #use of append()
>>> print(a)
[1, 2, 3, 64, 'mentor', 'jatin', 'food']
```

Lists can perform the following actions:-

Method Name	Action
append(x)	Add an item to the end of the list.
insert(i,x)	Insert an item at a given position.
remove(x)	Remove the first item from the list whose value is *x*. It is an error if there is no such item.
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.
clear()	Remove all items from the list.
index(x,[, *start*[, *end*]])	Return zero-based index in the list of the first item whose value is *x*. Raises a 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.

count(x)	Return the number of times *x* appears in the list.
sort(*key=None*, *reverse=False*)	Sort the items of the list in place
copy()	Return a shallow copy of the list.

Python tuple

- Just like lists, they can have different kinds of variable
- Difference between lists and tuples is that tuples are immutable.
- They are enclosed in parentheses () and elements are separated by commas.
- Slicing, indexing are done.

```
>>> a=('apple',2,'banana',3) #tuple
>>> type(a)
<class 'tuple'>


>>> b=() #empty tuple
>>>print(type(b))
<class 'tuple'>
>>> a[2] #indexing
'banana'

>>> a[1:] #slicing
(2, 'banana', 3)
```