3.2 Python Lists:

Python lists are just like dynamically sized arrays, declared in other languages (vector in C++ and ArrayList in Java).

Lis is a sequence data type used to store and collect data enclosed in "[]" and each element is separated by comma.

3.2.1: List Creation:

Lists can be created by placing oth data

inside square brackets [].
Each element must be separated by comma ",".

- Lists are mutable
- Lists can contain duplicate and different data types.

Example:
See List Creation.ipynb

3.2.2 Acessing Elements:

Elements in a list are accessed using index operator []. Index must be an integer just like in the string. Negative indexing is also used in accessing an element from list.

Examples
See List Accessing.ipynb

3.2.3 Size of List:

Python len() funetion is used to get length of the list.

Example:
See List Len.ipynb

3.24. Adding elements:

Elements in a list can be added by following 3 methods.

Method 1: append():

Elements can be added to the list using built-in append() method. Lists, Tuples and sets can also be added to the list.

append() always adds elements at the end of the list.

See Adding Elements to the list.ipynb

Method 2: Insert().

insert() also add elements to the list but index is given as argument in the curly brackets, with the value,

See Adding Elements to the list.ipynb

append(pos, val)

Method 3: Extend():

Extend() also works like append() method except that extend() method can add multiple elements and append() can only add one.

See Adding elements to a list.ipynb

3.2.5 Reversing List:

A list can be reversed by using double slicer.

See Reversing List.ipynb

Assignment 3.3:

Use two other methods to reverse lists

See

Assignment 33.py

3.2.6 Removing Elements:

Elements can be removed from lists by following two methods.

Method 1: remove()

Elements can be removed from list using built-in remove() function. It takes the value that is to be removed. If element does not exist, it causes an error.

It will only remove first occurance of that element.

See List Removal.ipynb

Method 2: pop():

pop() function is also used to remove an element from list and return that element to our screen as output

By default element at last index is poped/removed except when index is specified.

See List Removal.ipynb

3.2.7 List Slicing:

We can get substrings and sublists using a slice. When we want to print a certain range of elements from list instead of just one element or whole list, we use slicing operations.

To print elements from beginning to a range use:

[:Index]

To print elements from end:

[:-Index]

To print elements from specific index till end:

[Index:]

To print whole list in reverse order

[::-1]

Note:
To print elements of list from rear-end use negative indexing.

See Slicing List.JPEG ↓
~~and read below points.~~

Understanding Slicing of List

- pr[0] accessing first item, 2.
- pr[-4] acce⁄ fourth item from end, 5.
- pr[2:] accesses a list of items from third to last, [5, 7, 11, 13]
- pr[:4] accesses a list of items from first to four, [2, 3, 5, 7]
- pr[2:4] accesses a list of items from third to fifth
- pr[1:6:2] accesses [3, 7, 13], alternate items, starting from second item.

See List Slicing.ipynb.