Chapter 4

Lists and Tuples

Python lists are containers to store a set of values of any data type

```
a =["Axe", "PK" , "DK"]
```

List Indexing

A list can be indexed just like a string .

```
L1 = [7,9, "PK"]
L1[0] => 7   , L1[ 4] -> error
```

Eg:

```
a = [1,2,3,4]
Print(a)
Print (a[2])
a[0] = 9      -> change the value .
Print(a)
```

Note: we can create a list with items of different ~~ty~~ types

```
C = [45, PK ,6.9]
```

Note : List slicing is as same as String slicing

* List Methods

Consider the following list :

L1 = [1, 8, 7, 2, 21, 15]

1.) L1.sort() : udates list to [1, 2, 7, 8, 15, 21]
2.) L1. reverse() : updates the list to [15, 21, 2, 7, 8, 1]
3.) L1. append(8) : adds 8 ~~at~~ the end of the list
4.) L1. insert(3, 8) : This will add 8 at 3 index.
5.) L1. pop(2) : will delete element at index 2 & return its value
6.) L1. remove(21) : will remove 21 from the list

* Tuples In Python

A tuple is an immutable data type in Python.
↳ cannot change

a = ( ) → Empty tuple
a = (1,) → Tuple with only one element needs a comma
a = (1, 7, 2) → Tuple with more than one element

eg: t=(1,2,3,4)

```
Print(t[0])
#t[0]=6   -> error (on replacing)
```

once defined a tuple's elements can't be altered or manupulated.

t1 = (1) -> Prints '1' but it's wrong not the part of tuple because it's a defined value, so Put comma,
t1=(1,)

* Tuples Methods

Consider the tuple:
a = (1,7,2)

1.) a.Count(1): a.Count(1) will return no. of times 1 occurs in a.

2.) a.index(1): a.index(1) will return the index of first occurrence of 1 in a

eg: t=(1,2,3,5,1,1,6)

```
Print(t.count(1))  -> 3
Print(t.index(1))  -> 0
```