**LISTS**

Lists are the collection of elements i.e..,similar (or) dissimilar data items.

Declaring of a list

a[] -------- empty list

a=[1,1,1] ---------with similar data items

a=[‘Hello’,1,2,3,4]

We can access elements using index number.

**Accessing:**

L1=[1,2.4,3+9j,4,’Hello’]

#accessing elements in list

Print (L1[1])

Ex:

L1=[1,2,3,3+j,4,’Hello’]

L2=[5,6,7]

Print(L1+L2)

* Duplicating elements in a single list

L1 \* L2

**Slicing:**

Print(L1[0:3])

**Reverse:**

Print (L1[ :: -1])

Whole list using :: will be reversed

**Delete:**

Del L1[2]

Print[L1]

Elements in list can change (Mutable)

Elements in strin cannot change (Immutable)

L1[2]= ‘I have changed’

Print [L1] Immutable object

Creation of list with the help of string:

a= ‘HELLO’

b=L1[a]

o/p :- ‘H’,’E’,’L’,’L’,’O’

L={1,’i am out’,[‘i am in’,5,6,4.5],’biayta’,6]]

* List inside the list is homogeneous

Print(L[2][1])

-->This is the homogeneous property of list

We can use loop with list or iterating through list

creating a list using for loop

* A=[i for i in range (1,10)]

Holds the value varying from 1 to 10

For creation of list

Ex:

List (i for i in range (1,10)

Print (b)

o/p :- [‘1’,’2’,’3’,’4’,’5’,’6’,’7’,’8’,’9’]

Iterating through list:

For a in l:

Print(A)

**Inbuilt functions in list :**

List () convert anything into list

Len ()  length of the list items

Sum () **** sum of the elements in the list

Ex:

A=[i for i in range (1,10)]

Print (a)

Print(len(a))

Print(sum(a))

Print(min(a))

Print(max(a))

Print(all(a))  it checks all elements are high are not(if we have 0 it returns as false

Print(any(a))  if all elements are low (i.e [0,0,0,0] it returns or any single is high it returns true

**Inbuilt methods:**

.append ()  adding

It works on the architecture of stack (LIFO manner)

Ex:

L1=[1,2,3]

Print (l1)

Print(l1.index(3))

o/p:- 2

**enumerate ():**

 it gives an enumerated object

enumerated object means it gives an reference of the location

Ex:

L2=[4,5,6]

Print(enumerate(L2))

**TUPLE:**

Collection of similar and disimiliar elements but these aree immutable.

-->whenever we declare a tuple we cannot delete , add or change

A=1

B=1,2,3,4,  unpacked tuple

we can slice the tuple

**Iterating through tuple:**

a=”Hello”

b=tuple(a)

for a in b

print (a)

print(type(a))

**Tuple Comprehension:**

For loop in tuple

A=(a for a in range (0,11))

Parint (a)

* Generator object contains elements generated by tuple because tuples are right protected

**Enumerate():**

Enumerate () generates enumerate object

enumerate object contains list of tuple o index number and value at that index number

Ex:

L2=[4,5,6]

A=enumerate(L2)

Print(a)

For i in a

Print(i)

o/p :- (0,4)

(1,5)

(2,6)

**To print as a list:**

Info =[i for i in a ]

Print(info)

[(0,4),(1,5),(2,6)]

**Inbuilt functions:**

All ()

Any()

Sum ()

Len()

Min ()

Max ()

Enumerate()

**DICTIONARIES:**

Collection of keys and values

keys and values collectively known as key value pairs

key value pairs collectively known as item

keys and values separated by :

Use: form data

*Syntax:*

Mydictionary = { }

Print(my dictionary)

o/p :- { }

*representation* : key1 : value1, key2 : value2, key3 : value3

keys  Unique, immutable object

should declare number,string,tuple

values  should not be unique

values can be anything

**Accessing:**

>>> my dictionary[key1]

Value1

>>>my dictionary[key3]=value10

Print mydictionary

o/p :- {key1:value1, key2 : value2 key3 : value10}

>>> del mydictionary[key2]

>>>del mydictionary  entire dictionary deleted

**Methods:**

.keys()  [key,key2,key3]

.values()  [val1,val2,val3]

.items() [ key1 : value1, key2 : value2, key3 : value3]

.popitem()

Ex:

L1=[1,2,3,4]

L2=[‘ty’,’sup’,’php’,’jam’]

zip(L1,L2)  convert two list to a dictionary

[(1,’ty ‘),(2,’sup’),(3,’php’),(4,’jam’)]

D=dict(zip(L1,L2))

**SETS:**

****Unique collections

Heterogeneous

non-iterable,but mutable

*Syntax:*

set= {1,2,3}

myset= set( )empty set

myset= set((1,2,3))

ex:

d1.update(d2)  is used to act like .extend() in lists

Dictionaries are arbitrary and these are not sequential.

Myset={1,2,3}

>>> my set (4)

{1,2,3,4}

>>>myset.discard(2)

{1,3,4}

>>>myset.remove(3)

{1,4}

>>>myset.pop()  can remove any element

**Methods:**

.union()

.intersection()

.difference()

.issuperset()

.issubset()

Ex:

A.union(B)  A | B

A.intersection(B)  A & B

A.difference(B)  A - B

A.issuperset(B)  A < B

A.issubset(B)  A > B