→ List store primitive or Noprimitive datatypes.
→ List will store Like, Integer, Float, character, Boolean value, Imaginary datatype.

07/06/23

M T W T F S S
Page No.:
Date:
YOUVA

→ List Datatype:-

Create List:-

L = [1, 345, 45, "sudh", True, 5+8j, 345]

→ A List in python is used to store the sequence of various types data.

→ A List can be defined as a collection of values or items of different types.

characteristics of Lists.

① The Lists ~~has the~~ are ordered.

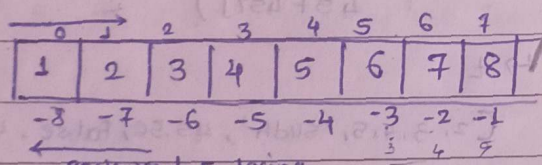
② The element of the List can access by Index.

④ The Lists are the mutable type.

⑤ A List can store the Number of various elements.

Import Note:

List work on Forword Indexing & Backword Indexing.



L = [1, 35, 45, 345, "sudh", True, 3+5j, 345.456]
0 1 2 3 4 5 6 7
List variable

→ type(L) # check the data type of List.

L[0]
→ 1

L[2]
→ 45

L[90]
→ throw error [Index out of bound range]

reverse Indexing

L[0:4]
→ [1, 35, 45, 345

L[-1]
→ 345.456.

reverse the List command.

L[::-1]
→ [345.456, (3+5j), True, 'sudh', 345, 45, 35, 1]

Find data on even Number of Index.

L[::2]
→ [1, 45, 'sudh' (3+5j)]

Note :- List & String are does Not Concatenate

L1 = [3, 4, 5]

Insert() :- (Insert object before Index)

L1.Insert(1, "sudh")
→ [3, 'sudh', 5]

Note → Insert(-1, 45), this function Insert the 45 value Insert after the Last value
→ [3, 'sudh', 45, 5]

pop function (pop()) :- remove element Like Index=-1 position

L1.pop()
→ [3, 'sudh', 45, 5]

L1
→ [45, 3, 'sudh', [2, 3, 4], 4]

Remove Function (Remove : Remove first occurance of value).

L1.Remove("sudh")
→ [45, 3, [2, 3, 4], 4]

mutable :- changeable
immutable :- No changeable

* List is mutable.
* String is immutable

L4

L1.remove(234)
↳ List.remove(x): x not in List.

condition : एका List च्या आती List ला कसे remove करायचे.

L1[2].remove(3)
↳ [45, 'sudh', [2,4], 4]

append() → Insert the element after the end of List.
ex. L1.append
[45, 'sudh', [2,4], 4]
L1.append(4)
↳ [45, 'sudh', [2,4], 4, 4]

List reverse करण्याच्या दोन method.
① temporary ② permanent.
① ex. L1[::-1]
↳ [4, [2,4], 'sudh', 45]
② ex. L1.reverse()
↳ [4, [2,4], 'sudh', 45]

sort() :- It is work on similar datatype, Like Integer.
ex. L2 = [45,23,90,0,3,1,4,5,6]
L2.sort()
↳ [0,3,4,5,6,23,45,90]

sort on descending orrder.
L2.sort(reverse=True)
↳ [90,45,23,6,5,4,3,0]

↳ ['sudh', 'pwskill', 'kumar', 'Data Scien
L4.index("sudh")
↳ 0
#

replace() : It is work on old string to New string and store on New memory Location.

tuples :-
→ A tuple is a collection of objects which ordered and immutable.
→ Tuples are sequences, just Like Lists. the differences between tuples & List are, the tuples cannot be changed UnLike List. & tuples use parentheses whereas List use square brackets.

ex. t = (2,3,4,5, "sudh", 45.56, false, 45+457j)
↳ t
[2, 3, 4, 5, "sudh", 45.56, false, 45+457j)

type(t)
↳ tuple

t[0]
↳ 2

len(t)
↳ 9

t[-1]
↳ 45+457j

t[::-1]
↳ (45+457, false, 45.56, 'sudh', 5, 4, 3, 2

tuples have only two function
① count
② Index

* Why are you use set?
→ set is gives the unique Data.
ex.

t=
ex. (2, 3, 4, 5, 'sudh', 45.56, false, 45+45tj)

t.count(3)
↳ 1

t.Index('sudh')
↳ 4

ex. S4 = {2, 3, 4, 5, 4, 5, 2, 8}
↳ {2, 3, 4, 5, 8}

S5
↳ {12, 2, 23, 234, 342, 45, 456, 567, 789, 'abc'}
S5.add(4)
↳ {12, 2, 23, 4, 234, 342, 45, 456, 567, 'abc', 789}

Set It will build an unordered collection of Unique element.

Set :-

A python set is the collection of the unordered items. each elements in the set must be unique, Immutable and the sets remove the duplicate elements.
→ sets are mutable which means we can modify it after its creation.
→ we cannot directly access any element of the set by the Index.

S1 = { }
type(S1)
↳ dict
S2 = {2, 3, 4, 55, 6}
type(S2)
↳ set
S3 = {324, 456, 456, "sudh", 45+45j, 34.465, [2, 3, 4]}
↳ typer error :- unhashable type: 'List'.
S3 = {324, 456, 456, "sudh", 45+45j, 34.465}
↳ {324, 'sudh', 456, 34.465, 45+45j}

comprehension function :-

there are two types of comprehension.
① List comprehension
② set comprehension :-

→ comprehension It is method to reduce more Line of code Few Line of code.

① ex. # L = [1, 2, 3, 4, 45, 5]
L1 = []
for i in L
L1.append(i**2)
L1
↳ [1, 4, 9, 16, 2025, 25]

② # L
↳ [1, 2, 3, 4, 45, 5]
[i**2 for i in L] # Example List comprehension function.
↳ [1, 4, 9, 16, 2025, 25]

③ # statement :- Find the even Number in given a List. ex. Of List L.
[i for i in L if i%==0]
↳ [2, 4]