Lists

- What is list
- List vs Array
- Create list
- Access list
- Edit list
- Add list
- Delete list
- Operations
- Fucntions

Array vs List

--> Array : homogenous, List : heterogeneous

- Array is homogeneous : All data type inside the array must be of the same data type
- Ex. such as. Int list

- Whereas, List can contain heterogeneous values such as integers, floats, strings, tuples, lists, and dictionaries but they are commonly used to store collections of homogeneous objects

---> Array : continue memory locn item store, List : mai aisa kuch nhi hia

- Array's are stored in continuous memory location where as list can be or cannot be it's not compulsory

---> Array are faster, List are slower when compared

- Because of the continuous memory locn of the array the array are faster. Faster access

---> List are more programmer friendly in comparison to Array

```
In [ ]:
```

List

- In Python, a list is created by placing elements inside square brackets [] , separated by commas.
- Lists are used to store multiple items in a single variable.

```
In [2]: # Empty list
        l1 = []
        l1

Out[2]: []
```

```
In [11]: l4 = list()
         l4

Out[11]: []
```

In [3]:
```
# homogeneous list
l2 = [1,2,3,4]
l2
```

Out[3]:
```
[1, 2, 3, 4]
```

In [4]:
```
# heterogenous list
l3 = ['am',1,'doc',2]
l3
```

Out[4]:
```
['am', 1, 'doc', 2]
```

In [ ]:

Mutli-dimensional List

2D - List :

- list inside a list [[]]
- are always heterogenous

3D List :

- list inside a list, another list inside a list [[[]]]
- are always heterogenous

In [5]:
```
# 2D List

l1 = [1,2,3,[4,5]]
l1
```

Out[5]:
```
[1, 2, 3, [4, 5]]
```

In [6]:
```
# 3D List

l2 = [[[1,2],[2,3],[3,4]]]
l2
```

Out[6]:
```
[[[1, 2], [2, 3], [3, 4]]]
```

In [ ]:

Converting str to list

In [9]:
```
l3 = list('NewYork')
l3
```

Out[9]:
```
['N', 'e', 'w', 'Y', 'o', 'r', 'k']
```

In [10]:
```
l4 = list()
l4
```

Out[10]:
```
[]
```

In [ ]:

Accessing Items from the list

```
In [12]: l1 = [1,2,3,4,5]
```

```
In [13]: l1[0]
```

```
Out[13]: 1
```

```
In [14]: l1[-1]
```

```
Out[14]: 5
```

```
In [15]: l1[::-1]
```

```
Out[15]: [5, 4, 3, 2, 1]
```

```
In [16]: l1[-1::-1]
```

```
Out[16]: [5, 4, 3, 2, 1]
```

```
In [ ]:
```

Accessing items from a 2D List

```
In [20]: l2 = [1,2,3,4,[5,6,7]]
```

```
In [21]: l2[4]
```

```
Out[21]: [5, 6, 7]
```

```
In [22]: l2[4][0]
```

```
Out[22]: 5
```

```
In [ ]:
```

Accessing items from a 3D List

```
In [26]: l3 = [ [[1,2],[3,4]],
                [[5,6],[7,8]] ]
         l3
```

```
Out[26]: [[[1, 2], [3, 4]], [[5, 6], [7, 8]]]
```

```
In [29]: # accessing 7 from l3 list

         l3[1][1][0]
```

```
Out[29]: 7
```

```
In [ ]:
```

Edit items inside the list

- list is mutable

```
In [30]: l1 = [1,2,3,4,5]
```

```
In [31]: l1[0]='xx'
         l1

Out[31]: ['xx', 2, 3, 4, 5]
```

```
In [32]: l1[-1]=100
         l1

Out[32]: ['xx', 2, 3, 4, 100]
```

```
In [ ]:
```

```
In [33]: # making changes to the list using slicing

         l1 = [1,2,3,4,5]
         l1[0:3]=['a','b','c']
```

```
In [34]: l1

Out[34]: ['a', 'b', 'c', 4, 5]
```

```
In [ ]:
```

Add new items to the list

- append : add item to the end of the lst
- insert : pass (posn, value)
- extend
- String +=

```
In [35]: l1 = [1,2,3,4,5]
         l1

Out[35]: [1, 2, 3, 4, 5]
```

```
In [36]: l1.append(100)
         l1

Out[36]: [1, 2, 3, 4, 5, 100]
```

```
In [ ]:
```

Insert:

- insert(posn,item)
-

```
In [39]: l1 =[1, 2, 3, 4, 5, 100]
         l1.insert(0,'inserted')
         l1

Out[39]: ['inserted', 1, 2, 3, 4, 5, 100]
```

```
In [41]: l1 = [1, 2, 3, 4, 5, 100,1]
         l1.insert(1,'-')
         l1

Out[41]: [1, '-', 2, 3, 4, 5, 100, 1]
```

```
In [ ]:
```

```
In [42]: l1 =[1, 2, 3, 4, 5, 100]
         l1.extend('str')
```

```
In [43]: l1

Out[43]: [1, 2, 3, 4, 5, 100, 's', 't', 'r']
```

```
In [46]: l1 =[1, 2, 3, 4, 5, 100]
         l1.extend([11,22,33,1])
```

```
In [47]: l1

Out[47]: [1, 2, 3, 4, 5, 100, 11, 22, 33, 1]
```

```
In [ ]:
```

```
In [50]: l1 =[1, 2, 3, 4, 5, 100]
         l1+=['1x1',22,33]
         l1

Out[50]: [1, 2, 3, 4, 5, 100, '1x1', 22, 33]
```

```
In [ ]:
```

Append vs Extend

- Append : single item append to list
- Extend : multiple items append to list

```
In [51]: l1 =[1, 2, 3, 4, 5, 100]
         l1.append([1,3,4])
```

```
In [52]: l1

Out[52]: [1, 2, 3, 4, 5, 100, [1, 3, 4]]
```

```
In [ ]:
```

```
In [56]: l1 =[1, 2, 3, 4, 5, 100]
         l1.extend([11,33,55])
```

```
In [57]: l1

Out[57]: [1, 2, 3, 4, 5, 100, 11, 33, 55]
```

```
In [ ]:
```

Deleting items from the list

- del
- remove
- pop
- clear

```
In [58]: # deleting an entore list
         l1 =[1, 2, 3, 4, 5, 100]
         l1
```

```
Out[58]: [1, 2, 3, 4, 5, 100]
```

```
In [59]: del l1
```

```
In [60]: l1
```

```
---------------------------------------------------------------------------
----
NameError                                 Traceback (most recent call l
ast)
Input In [60], in <module>
----> 1 l1

NameError: name 'l1' is not defined
```

```
In [ ]:
```

```
In [62]: # deleting an item inside the list
         l1 =[1, 2, 3, 4, 5,[6,7], 100]
         l1
```

```
Out[62]: [1, 2, 3, 4, 5, [6, 7], 100]
```

```
In [64]: del l1[-2]
         l1
```

```
Out[64]: [1, 2, 3, 4, 100]
```

```
In [ ]:
```

```
In [65]: l1 =[1, 2, 3, 4, 5,[6,7], 100]
         l1
```

```
Out[65]: [1, 2, 3, 4, 5, [6, 7], 100]
```

```
In [66]: del l1[0]
         l1
```

```
Out[66]: [2, 3, 4, 5, [6, 7], 100]
```

```
In [ ]:
```

```
In [71]: l1 =[1, 2, 3, 4, 5,[6,7], 100]
         l1

Out[71]: [1, 2, 3, 4, 5, [6, 7], 100]
```

```
In [72]: del l1[-2][0]
         l1

Out[72]: [1, 2, 3, 4, 5, [7], 100]
```

```
In [ ]:
```

```
In [78]: # del a series of item in list using slicing

         l1 =[1, 2, 3, 4, 5,[6,7], 100]
         l1

Out[78]: [1, 2, 3, 4, 5, [6, 7], 100]
```

```
In [79]: del l1[:3]
```

```
In [80]: l1

Out[80]: [4, 5, [6, 7], 100]
```

```
In [ ]:
```

remove

- remove is used when we dont know the index of the element
- but we know the element exist
- it only removes the element found in first occurance

```
In [81]: l1 =[1, 2, 3, 4, 5,[6,7,1], 100,1]
         l1

Out[81]: [1, 2, 3, 4, 5, [6, 7, 1], 100, 1]
```

```
In [82]: l1.remove(1)
```

```
In [83]: l1

Out[83]: [2, 3, 4, 5, [6, 7, 1], 100, 1]
```

```
In [ ]:
```

pop

- removes the last element of the list

```
In [84]: l1 =[1, 2, 3, 4, 5,[6,7,1]]
         l1

Out[84]: [1, 2, 3, 4, 5, [6, 7, 1]]
```

```
In [85]: l1.pop()
         l1

Out[85]: [1, 2, 3, 4, 5]

In [ ]:

In [86]: l1.pop()
         l1

Out[86]: [1, 2, 3, 4]

In [ ]:
```

clear()

- used to empty the list

```
In [1]: l1 =[1, 2, 3, 4, 5,[6,7,1]]
        l1

Out[1]: [1, 2, 3, 4, 5, [6, 7, 1]]

In [3]: l1.clear()

In [4]: l1

Out[4]: []

In [ ]:
```

Operations on list

- addition of lists
- multiplication of lists

```
In [5]: # additon of list
        l1 = [1,2,3]
        l2 = ['a','b','c']
        l3 = [4,5,6]

        l1+l2+l3

Out[5]: [1, 2, 3, 'a', 'b', 'c', 4, 5, 6]

In [6]: l2+l1+l3

Out[6]: ['a', 'b', 'c', 1, 2, 3, 4, 5, 6]

In [ ]:
```

In [8]:

```
# multiplication of list
l1 = [1,2,3]
l3 = [4,5,6]
l1*3
```

Out[8]:

```
[1, 2, 3, 1, 2, 3, 1, 2, 3]
```

In [ ]:

In [10]:

```
# iterating throught the lists
l1 = [1, 2, 3, 'a', 'b', 'c',[5,6], 6]

for i in l1:
    print(i,end= ' ')
```

```
1 2 3 a b c [5, 6] 6
```

In [ ]:

In [11]:

```
# using membership operators in list
l1 = [1, 2, 3, 'a', 'b', 'c',[5,6], 6]
'a' in l1
```

Out[11]:

```
True
```

In [12]:

```
5 in l1
```

Out[12]:

```
False
```

fuctions on list

- For using functions on list the list should contain numberical values
- They are :
  - min/max
  - len
  - sort
  - sorted (not permanent changes)
  - index (to find index of item)

In [18]:

```
l1 = [1,2,3,4,4.0]
len(l1)
```

Out[18]:

```
5
```

In [19]:

```
max(l1)
```

Out[19]:

```
4
```

In [20]:

```
min(l1)
```

Out[20]:

```
1
```

In [ ]:

In [21]:

```
l2 = [1,2,3,4,0,-1]
```

```
In [22]: max(l2)

Out[22]: 4
```

```
In [23]: min(l2)

Out[23]: -1
```

```
In [24]: len(l2)

Out[24]: 6
```

```
In [ ]:
```

Sorted fuctn

- Creates a new list into the memory
- does not affect's the original list
- we can also pass (reverse=True) to reverse the order of the list

```
In [25]: l1 = [1,2,3,4,0,-1]
         sorted(l1)

Out[25]: [-1, 0, 1, 2, 3, 4]
```

```
In [26]: l2 = [1,2,3,4,0,-1]
         sorted(l2,reverse=True)

Out[26]: [4, 3, 2, 1, 0, -1]
```

```
In [27]: l1 # does not affects the original list

Out[27]: [1, 2, 3, 4, 0, -1]
```

```
In [28]: l2 # does not affects the original list

Out[28]: [1, 2, 3, 4, 0, -1]
```

Sort fuctn

- Sort fuctn is a permanent operation
- if modifies the original list
- we can also reverse the order by passing teh argument reverse=True

```
In [29]: l1 = [1,2,3,4,0,-1]
         l1

Out[29]: [1, 2, 3, 4, 0, -1]
```

```
In [30]: l1.sort()
```

```
In [31]: l1

Out[31]: [-1, 0, 1, 2, 3, 4]
```

```
In [ ]:
```

```
In [35]: l2 = [4, 3, 0, -1,2, 1]
         l2
```

```
Out[35]: [4, 3, 0, -1, 2, 1]
```

```
In [36]: l2.sort(reverse=True)
```

```
In [37]: l2
```

```
Out[37]: [4, 3, 2, 1, 0, -1]
```

```
In [ ]:
```

Index fuctn

- returns the index of the item present in the list

```
In [38]: l2 = [4, 3, 0, -1,2, 1]
         l2.index(1)
```

```
Out[38]: 5
```

```
In [39]: l2.index(-1)
```

```
Out[39]: 3
```

```
In [ ]:
```

```
In [41]: l1 = [4, 3, 0, -1, 2, 4, 1, -1]
         l2.index(-1)
```

```
Out[41]: 3
```

Creating custom title method

```
In [63]: msg = input('Enter the msg : ')
         msg = msg.split(' ')
         res = []
         for i in msg:
             res.append(i.capitalize())

         print(' '.join(res))
```

```
Enter the msg : hello my name is harsh
Hello My Name Is Harsh
```

or.

In [64]:

```
msg = input('Enter the msg : ')
res = []
for i in msg.split(' '):
    res.append(i.capitalize())

print(' '.join(res))
```

```
Enter the msg : hey buddy
Hey Buddy
```

In [ ]:

Prog. to extract the username from email id's

ex. username@gmail.com (mailto:username@gmail.com)

In [66]:

```
mail =input('Enter ur mail id : ')
result = mail.split('@')
print(result[0])
```

```
Enter ur mail id : yahoo@gmail.com
yahoo
```

or.

In [68]:

```
mail =input('Enter ur mail id : ')
print(mail[:mail.find('@')])
```

```
Enter ur mail id : helloraju@ham.in
helloraju
```

Type *Markdown* and LaTeX: α^2

Program to remove duplicates from the list

In [77]:

```
lst = []

msg = (input())
lst.extend(msg)

print('Before : ',lst)
lst = list(set(lst))
print('After : ',lst)
```

```
11223344
Before :  ['1', '1', '2', '2', '3', '3', '4', '4']
After :  ['1', '4', '3', '2']
```

or.

In [79]:

```
lst = []

msg = (input())
lst.extend(msg)

res = []
for i in lst:
    if i not in res:
        res.append(i)
```

```
11223344
```

In [80]:

```
res
```

Out[80]:

```
['1', '2', '3', '4']
```

In [ ]:

```
```