python data structure

--> data structure is the collection of data types

--> data structure is the way of storing, managing,organizing tha data make it easier to perform operations like insertion , deletion etc operations

data structure is divided into two types

1)built in data structure

list

tuple

set

dictionary

2) user defined data structure

stack

queue

linked list

tree

graph

LIST

1) list is a collection of different data type

2) list is denoted by square brackets[]

3) list is mutable

4) list allowed duplicates

we can perform many operations in a list

append()

copy()

remove()

pop()

index()

insert()

extend()

count()

sort()

clear()

reverse()

In [25]:
```
l=[]
```

In [26]:
```
l
```

Out[26]:
```
[]
```

In [27]:
```
type(l)
```

Out[27]:
```
list
```

In [28]:
```
len(l)
```

Out[28]:
```
0
```

In [29]:
```
id(l)
```

Out[29]:
```
2272104206656
```

In [30]:
```
l.append(10)
```

In [31]:
```
l
```

Out[31]:
```
[10]
```

In [32]:
```
l.append(20,30)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-32-00ccc9a7a83d> in <module>
----> 1 l.append(20,30)

TypeError: append() takes exactly one argument (2 given)
```

we can add only argument to the list

In [33]:
```
len(l)
```

Out[33]:
```
1
```

append function add the element at the last of the list

In [34]:
```
l.append(20)
l.append(30)
l.append(40)
l.append(50)
```

In [35]:
```
l
```

Out[35]:
```
[10, 20, 30, 40, 50]
```

In [36]:
```
len(l)
```

Out[36]:
```
5
```

In [37]:
```
l1=l.copy()
l1
```

Out[37]:
```
[10, 20, 30, 40, 50]
```

In [38]:
```
l==l1
```

Out[38]:
```
True
```

In [39]:
```
l!=l1
```

Out[39]:
```
False
```

In [40]:
```
l1
```

Out[40]: [10, 20, 30, 40, 50]

In [41]:
```
l1.append(100)
```

In [42]:
```
l1
```

Out[42]: [10, 20, 30, 40, 50, 100]

In [43]:
```
1==l1
```

Out[43]: False

In [44]:
```
print(len(l))
print(len(l1))
```

```
5
6
```

In [45]:
```
print(l)
print(l1)
```

```
[10, 20, 30, 40, 50]
[10, 20, 30, 40, 50, 100]
```

In [46]:
```
id(l1)
```

Out[46]: 2272106060416

clear function removes all the elements from the list

In [47]:
```
l1.clear()
```

In [48]:
```
l1
```

Out[48]: []

In [49]:
```
len(l1)
```

Out[49]: 0

In [50]:
```
id(l1)
```

Out[50]: 2272106060416

the id will be same when the list with elements without elements

In [51]:
```
l
```

Out[51]: [10, 20, 30, 40, 50]

In [52]:
```
l.append("akshitha")
l.append(2.3)
l.append(1+2j)
l.append(True)
l.append([1,2,3])
```

In [53]:
```
l
```

Out[53]: [10, 20, 30, 40, 50, 'akshitha', 2.3, (1+2j), True, [1, 2, 3]]

list can allow all the data types

In [54]:
```
l.append(10)
l
```

Out[54]: [10, 20, 30, 40, 50, 'akshitha', 2.3, (1+2j), True, [1, 2, 3], 10]

list can allow duplicates

In [55]:
```
l1
```

Out[55]: []

In [56]:
```
l1=l.copy()
l1
```

Out[56]: [10, 20, 30, 40, 50, 'akshitha', 2.3, (1+2j), True, [1, 2, 3], 10]

In [57]:
```
l==l1
```

Out[57]: True

In [58]:
```
l1.count(2.3)
```

Out[58]: 1

count function counts the number of mentioned elements present in the list

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

Out[59]: [10, 20, 30, 40, 50, 'akshitha', 2.3, (1+2j), True, [1, 2, 3], 10]

In [60]:
```
l1.remove(1+2j)
```

In [61]:
```
l1
```

Out[61]: [10, 20, 30, 40, 50, 'akshitha', 2.3, True, [1, 2, 3], 10]

remove funtion deletes the particular element

In [62]:
```
l1.remove(10)
```

In [63]:
```
l1
```

Out[63]: [20, 30, 40, 50, 'akshitha', 2.3, True, [1, 2, 3], 10]

we have 10 2 duplicates elements when we perform remove operation it deletes first occurance element

In [64]:
```
l1.pop()
```

Out[64]: 10

In [65]:
```
l1
```

Out[65]: [20, 30, 40, 50, 'akshitha', 2.3, True, [1, 2, 3]]

pop function removes the last element in the list by default(last in first out)

In [66]:
```
l1.pop()
l1
```

Out[66]: [20, 30, 40, 50, 'akshitha', 2.3, True]

In [67]:
```
l1.pop(1)
```

Out[67]: 30

when we want to remove particular element from the list using pop function we pass the index of that element

In [68]:
```
l1.remove(True)
```

In [69]:
```
l1
```

Out[69]: [20, 40, 50, 'akshitha', 2.3]

In [70]:
```
l
```

Out[70]: [10, 20, 30, 40, 50, 'akshitha', 2.3, (1+2j), True, [1, 2, 3], 10]

In [71]:
```
l.pop(2)
```

Out[71]: 30

In [72]:
```
l
```

Out[72]: [10, 20, 40, 50, 'akshitha', 2.3, (1+2j), True, [1, 2, 3], 10]

In [73]:
```
l.index(2.3)
```

Out[73]: 5

index function used to return the index of a value

In [74]:
```
l.index(10)
```

Out[74]: 0

In [75]:
```
l.insert(5,3)
```

In [76]:
```
l
```

Out[76]: [10, 20, 40, 50, 'akshitha', 3, 2.3, (1+2j), True, [1, 2, 3], 10]

In the insert() function, we pass two parameters

The index where we want to insert the element

The value of the element to be inserted

In [77]:
```
l.insert(1,15)
l
```

Out[77]: [10, 15, 20, 40, 50, 'akshitha', 3, 2.3, (1+2j), True, [1, 2, 3], 10]

In [78]:
```
print(l)
print(l1)
```

```
[10, 15, 20, 40, 50, 'akshitha', 3, 2.3, (1+2j), True, [1, 2, 3], 10]
[20, 40, 50, 'akshitha', 2.3]
```

In [79]:
```
l.reverse()
```

In [80]:
```
l
```

Out[80]: [10, [1, 2, 3], True, (1+2j), 2.3, 3, 'akshitha', 50, 40, 20, 15, 10]

In [81]:
```
l5=[300,3,34,9,100]
l5
```

Out[81]: [300, 3, 34, 9, 100]

In [82]:
```
l5.sort()
```

In [83]:
```
l5.sort(reverse=True)
```

In [84]:
```
l5
```

Out[84]: [300, 100, 34, 9, 3]

In [85]:
```
l
```

Out[85]: [10, [1, 2, 3], True, (1+2j), 2.3, 3, 'akshitha', 50, 40, 20, 15, 10]

In [86]:
```
l5[0]=3000
```

In [87]:
```
l5
```

Out[87]: [3000, 100, 34, 9, 3]

forward indexing and backward indexing

In [88]:
```
l5
```

Out[88]: [3000, 100, 34, 9, 3]

In [89]:
```
l5[2]
```

Out[89]: 34

In [90]:
```
l5[-1]
```

Out[90]: 3

In [91]:
```
l5[-4]
```

Out[91]: 100

slicing

In [92]:
```
l
```

Out[92]: [10, [1, 2, 3], True, (1+2j), 2.3, 3, 'akshitha', 50, 40, 20, 15, 10]

In [93]:
```
l1
```
Out[93]: [20, 40, 50, 'akshitha', 2.3]

In [94]:
```
l1[0:3]
```
Out[94]: [20, 40, 50]

In [95]:
```
l1[1:]
```
Out[95]: [40, 50, 'akshitha', 2.3]

In [96]:
```
l1[:3]
```
Out[96]: [20, 40, 50]

In [97]:
```
l1[-3:]
```
Out[97]: [50, 'akshitha', 2.3]

In [98]:
```
l1[:-3]
```
Out[98]: [20, 40]

In [99]:
```
l1[:]
```
Out[99]: [20, 40, 50, 'akshitha', 2.3]

In [100]:
```
mylist=["one",'two','three','four','five','six','seven','eight']
```

In [102]:
```
mylist[0:3]
```
Out[102]: ['one', 'two', 'three']

In [103]:
```
mylist
```
Out[103]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [104]:
```
mylist.append('ten')
```

In [105]:
```
mylist
```
Out[105]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'ten']

In [106]:
```
mylist.insert(4,'eleven')
mylist
```

Out[106]:
```
['one',
 'two',
 'three',
 'four',
 'eleven',
 'five',
 'six',
 'seven',
 'eight',
 'ten']
```

In [107]:
```
mylist.insert(1,'ONE')
mylist
```

Out[107]:
```
['one',
 'ONE',
 'two',
 'three',
 'four',
 'eleven',
 'five',
 'six',
 'seven',
 'eight',
 'ten']
```

In [108]:
```
mylist.remove('ONE')
```

In [109]:
```
mylist
```

Out[109]:
```
['one',
 'two',
 'three',
 'four',
 'eleven',
 'five',
 'six',
 'seven',
 'eight',
 'ten']
```

In [111]:
```
mylist.pop()
```

Out[111]:
```
'ten'
```

In [112]:
```
mylist
```

Out[112]:
```
['one', 'two', 'three', 'four', 'eleven', 'five', 'six', 'seven', 'eight']
```

In [113]:
```
mylist.pop(4)
```

Out[113]:
```
'eleven'
```

In [114]:
```
mylist
```

Out[114]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [115]:
```
mylist[0]=1
mylist[1]=2
mylist[2]=3
mylist
```

Out[115]:
```
[1, 2, 3, 'four', 'five', 'six', 'seven', 'eight']
```

In [116]:
```
mylist.clear()
mylist
```

Out[116]:
```
[]
```

In [117]:
```
del mylist
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-117-e3998fd6e4af> in <module>
      1 del mylist
----> 2 mylist

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

In [119]:
```
mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine'
```

In [120]:
```
mylist1=mylist
```

In [121]:
```
id(mylist1), id(mylist)
```

Out[121]:
```
(2272106016896, 2272106016896)
```

In [123]:
```
mylist2=mylist.copy()
```

In [124]:
```
mylist2
```

Out[124]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [125]:
```
id(mylist2)
```

Out[125]:
```
2272106161088
```

In [126]:
```
mylist[0]
```

Out[126]: 'one'

In [127]:
```
list1=['one','two','three','four']
list2=['five','six','seven','eight']
```

In [128]:
```
list3=list1+list2
list3
```

Out[128]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [129]:
```
list1.extend(list2)
list1
```

Out[129]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [130]:
```
list1
```

Out[130]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [131]:
```
"one" in list1
```

Out[131]: True

In [133]:
```
"ten" in list1
```

Out[133]: False

In [134]:
```
"ONE" in list1
```

Out[134]: False

In [135]:
```
list1
```

Out[135]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [136]:
```
list1.reverse()
list1
```

Out[136]: ['eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one']

In [ ]:
```

```