```
list1=['physics','maths','sachin','virat',7,3.5,2,6.80]
```

```
len(list1)
```

```
8
```

```
list1[4]
```

```
7
```

```
list1[8]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-7-434e8353cbf8> in <module>()
----> 1 list1[8]

IndexError: list index out of range
```

SEARCH STACK OVERFLOW

```
list1[1:5]
```

```
['maths', 'sachin', 'virat', 7]
```

```
list1[:6]
```

```
['physics', 'maths', 'sachin', 'virat', 7, 3.5]
```

```
list1
```

```
['physics', 'maths', 'sachin', 'virat', 7, 3.5, 2, 6.8]
```

```
list1[6:]
```

```
[2, 6.8]
```

```
list1[1:10]   #index=1,2,3,4,5,6,7,8,9                    #homework
```

```
['maths', 'sachin', 'virat', 7, 3.5, 2, 6.8]
```

Modifying the list content

```
list1=['virat',1,3,4.6,'ABC']
```

```
list1
```

```
['virat', 1, 3, 4.6, 'ABC']
```

```
list1[2]='XYZ'          #modifying the value
```

```
list1
```

```
['virat', 1, 'XYZ', 4.6, 'ABC']
```

```
del list1[-1]          #deleting the entry
```

```
list1
```

```
['virat', 1, 'XYZ', 4.6]
```

```
len(list1)
```

```
4
```

```
list2=['sachin','3',4,2.5]
```

```
list1+list2+list1     #concatenation
```

```
['virat', 1, 'XYZ', 4.6, 'sachin', '3', 4, 2.5, 'virat', 1, 'XYZ', 4.6]
```

```
['DAIICT']*5        #repetition over list
```

```
['DAIICT', 'DAIICT', 'DAIICT', 'DAIICT', 'DAIICT']
```

```
'DAIICT'*5            #repitition over string
```

```
'DAIICTDAIICTDAIICTDAIICTDAIICT'
```

```
'-'*100
```

```
'----------------------------------------------------------------------------------------------------
```

```
#membership operator
3 in [1,2,3]
```

```
True
```

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

```
False
```

```
for i in [1,2,3]:
  print(i)
```

```
1
2
3
```

Built-in functions in list

```
list1=[1,4,2,0,6,7,8,3,1]
```

```
len(list1)
```

```
9
```

```
max(list1)
```

```
8
```

```
min(list1)
```

```
0
```

Methods of list

```
list1
```

```
[1, 4, 2, 0, 6, 7, 8, 3, 1]
```

```
list1.append(11)
```

```
list1
```

```
[1, 4, 2, 0, 6, 7, 8, 3, 1, 11]
```

```
len(list1)
```

```
10
```

```
list1.append([11,15,18,19])  #append considers the parameter entry as the singleton entity
```

```
list1
```

```
[1, 4, 2, 0, 6, 7, 8, 3, 1, 11, [11, 15, 18, 19]]
```

```
list1[10]
```

```
[11, 15, 18, 19]
```

```
list1[10][1]
```

```
15
```

```
list1.extend([11,15,18,19])  #extend appends all the elements separately to the list
```

```
list1
```

```
[1, 4, 2, 0, 6, 7, 8, 3, 1, 11, [11, 15, 18, 19], 11, 15, 18, 19]
```

```
list1.insert(4,'PQR')
```

```
list1
```

```
[1, 4, 2, 0, 'PQR', 6, 7, 8, 3, 1, 11, [11, 15, 18, 19], 11, 15, 18, 19]
```

```
list1.insert(0,'XYZ')
```

```
list1
```

```
['XYZ', 1, 4, 2, 0, 'PQR', 6, 7, 8, 3, 1, 11, [11, 15, 18, 19], 11, 15, 18, 19]
```

```
list1.insert(0,[1,2,3,4])
```

```
#pop function
```

```
list1.pop()
```

```
19
```

```
list1
```

```
[[1, 2, 3, 4],
 'XYZ',
 1,
 4,
 2,
 0,
 'PQR',
 6,
 7,
 8,
 3,
 1,
 11,
 [11, 15, 18, 19],
 11,
 15,
 18]
```

```
list1.pop(3)
```

```
4
```

```
list1.pop()
```

```
18
```

```
list1
```

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

```
 0,
 'PQR',
 6,
 7,
 8,
 3,
 1,
 11,
 [11, 15, 18, 19],
 11,
 15]
```

```
#to remove some element i
list1.remove(1)
```

```
list1
```

```
[[1, 2, 3, 4], 'XYZ', 2, 0, 'PQR', 6, 7, 8, 3, 1, 11, [11, 15, 18, 19], 11, 15]
```

```
list1.reverse()
```

```
list1
```

```
[15, 11, [11, 15, 18, 19], 11, 1, 3, 8, 7, 6, 'PQR', 0, 2, 'XYZ', [1, 2, 3, 4]]
```

```
list1.sort()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-72-73031ee8c861> in <module>()
----> 1 list1.sort()

TypeError: '<' not supported between instances of 'list' and 'int'
```

SEARCH STACK OVERFLOW

```
list1=[2,1,4,2,56,8,3,6]
```

```
list1.sort()
```

```
list1
```

```
[1, 2, 2, 3, 4, 6, 8, 56]
```

```
list1.sort(reverse=True)
```

```
list1
```

```
[56, 8, 6, 4, 3, 2, 2, 1]
```

```
list1.clear()
```

```
list1
```

```
[]
```

```
#addition of the values in front
list1=[1,2,3,4,5,6,7]
```

```
list1.extend(['ABC','PQR','XYZ'])
```

```
list1
```

```
[[1.2, 3.4, 4.6], 1, 2, 3, 4, 5, 6, 7, 'ABC', 'PQR', 'XYZ']
```

```
A=[1.2,3.4,4.6]
```

```
list1.insert(0,A)   #A is considered as the singleton entry
```

```
list1
```

```
[1, 2, 3, 4, 5, 6, 7]
```

```
A.reverse()
for i in A:
  list1.insert(0,i)
  print(list1)
```

```
[4.6, 1, 2, 3, 4, 5, 6, 7]
[3.4, 4.6, 1, 2, 3, 4, 5, 6, 7]
[1.2, 3.4, 4.6, 1, 2, 3, 4, 5, 6, 7]
```

```
list1
```

```
[4.6, 3.4, 1.2, 1, 2, 3, 4, 5, 6, 7]
```

Tuple

```
tup1=('physics','maths','chemistry','DAIICT',1,2,3)
```

```
tup1
```

```
('physics', 'maths', 'chemistry', 'DAIICT', 1, 2, 3)
```

```
tup2="a","b","c","d",1,2,3
```

```
tup2
```

```
('a', 'b', 'c', 'd', 1, 2, 3)
```

```
a=1,2
```

```
a
```

```
(1, 2)
```

```
tup3=()
```

```
#single value in tuple
tup3=(50)
```

```
tup3
```

```
50
```

```
type(tup3)
```

```
int
```

```
tup3=(50,)
```

```
tup3
```

```
(50,)
```

```
type(tup3)
```

```
tuple
```

```
tup1=(1,2,3,4,5,6,7)
```

```
tup1[0]
```

```
1
```

```
tup1[1:4]
```

```
(2, 3, 4)
```

```
tup1[0]=0
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-120-929b97c75225> in <module>()
----> 1 tup1[0]=0

TypeError: 'tuple' object does not support item assignment
```

SEARCH STACK OVERFLOW

```
tup1+tup2
```

```
(1, 2, 3, 4, 5, 6, 7, 'a', 'b', 'c', 'd', 1, 2, 3)
```

```
tup1*3
```

```
(1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7)
```

```
del tup1[-1]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-124-5acdc279e7b9> in <module>()
----> 1 del tup1[-1]

TypeError: 'tuple' object doesn't support item deletion
```

SEARCH STACK OVERFLOW

```
len(tup1)
```

```
7
```

```
tup1
```

```
(1, 2, 3, 4, 5, 6, 7)
```

```
max(tup1)
```

```
7
```

```
min(tup1)
```

```
1
```

```
tup1[-2]
```

```
6
```

```
3 in tup1
```

```
True
```

```
for i in (1,2,3,4,5,6,7):
  print(i)
```

```
1
2
3
4
5
6
7
```

Colab paid products - Cancel contracts here