

```
1 data=[]
2 type(data)

list
```

```
1 data=()
2 type(data)

tuple
```

```
1 data={} # creates empty dict, not a empty set
2 type(data)

dict
```

```
1 data={0} # to create a set, it should have atleast some element
2 type(data)

set
```

```
1 data=set() # way to create a blank set
```

▼ List

```
1 #create var = [] var=[v1, v2, v3, v4]
2 data=[11, 22, "amar", True, 3.14] #heterogeneous
3 print(data)

[11, 22, 'amar', True, 3.14]
```

```
1 # indexed access
2 # positive
3 data=[11, 22, 33, 44, 55, 66]
4 print(data[0])
5 print(data[-1])
6 print(len(data)) # len(mentod works on all, list, tuple, list, dict & even string)

11
66
6
```

```
1 # indexed access
2 data=[11, 22, 33, 44, 55, 66]
3 for i in range(len(data)):
4     print("at", i, "we have: ", data[i])

at 0 we have: 11
at 1 we have: 22
at 2 we have: 33
```

```
at 3 we have: 44
at 4 we have: 55
at 5 we have: 66
```

```
1 # auto iterated access
2 data=[11, 22, 33, 44, 55, 66]
3 for i in data:
4     print(i)
```

```
11
22
33
44
55
66
```

```
1 data1=[11, 22, 33, 44]
2 data2=[1, 2, 3]
3 data3=data1+data2 # will concatenate two lists
4 print(data3)
5 data4=data1*2 # multiply(*) with an integer number will repeat list 'n' times
6 print(data4)
```

```
[11, 22, 33, 44, 1, 2, 3]
```

```
1 data=[11, 22, 33, 44, 55, 66, 77, 88, 99, 111]
2 data[3:6:1]
3 data[:]
```

```
[11, 22, 33, 44, 55, 66, 77, 88, 99, 111]
```

data[start:end:step

data[:] data[::-1]

```
1 data=[11, 22, 33, 44, 55, 66, 77, 88, 99, 111]
2 print(data[::-1])
3 print(data)
4 print(data[::2])
5 print(data)
```

```
[111, 99, 88, 77, 66, 55, 44, 33, 22, 11]
[11, 22, 33, 44, 55, 66, 77, 88, 99, 111]
[11, 33, 55, 77, 99]
[11, 22, 33, 44, 55, 66, 77, 88, 99, 111]
```

```
1 list2d=[[10, 20, 30], [40, 50], [60, 70, 80, 90, 100]]
2 print(list2d[2][4])
3 print(len(list2d))
```

```
100
3
```

```

1 list2d=[[10, 20, 30], [40, 50], [60, 70 ,80, 90, 100]]
2 for i in list2d:
3     print(i)

[10, 20, 30]
[40, 50]
[60, 70, 80, 90, 100]

```

```

1 '''
2 print each element using index only
3 10 20 30
4 40 50
5 '''
6 list2d=[[10, 20, 30], [40, 50], [60, 70 ,80, 90, 100]]
7 for row in range(len(list2d)):
8     for col in range(len(list2d[row])):
9         print(list2d[row][col], end=" ")
10    print()

10 20 30
40 50
60 70 80 90 100

```

```

1 data=[[11, 22, 33, 44], [55, 66, 77, 88], [99, 111, 222, 333], [444, 555, 666, 777]]
2 # print in zig-zag manner of normal and reversal
3 for i in range(len(data)):
4     temp=data[i]
5     if i%2==0:
6         print(temp)
7     else:
8         print(temp[::-1])

[11, 22, 33, 44]
[88, 77, 66, 55]
[99, 111, 222, 333]
[777, 666, 555, 444]

```

```
1 datalist=[]
```

```
1 datalist.append(int(input("Enter data: ")))
```

Enter data: -19

```
1 print("datalist has: ", len(datalist), datalist)
```

datalist has: 6 [333, 1000, 19, 6, -19, 212]

```
1 datalist.insert(0, 1000)
```

```
1 datalist.insert(100, 212)
```

```
1 datalist.insert(-100, 333)
```

```
1 d1=[11, 22, 33]
2 d2=[10, 20, 30]
3 d1+d2 # temporary change, does not reflect any change in the argument list
4 print("d1: ", d1)
```

```
d1: [11, 22, 33]
```

```
1 d1=[11, 22, 33]
2 d2=[10, 20, 30]
3 d1.extend(d2) # permanently changes, reflects change in argument list
4 print("d1: ", d1)
5
```

```
d1: [11, 22, 33, 10, 20, 30]
```

```
1 d1=[11, 22, 33]
2 d2=[10, 20, 30]
3 d1+=d2 # equivalent to extend method
4 print("d1: ", d1)
```

```
d1: [11, 22, 33, 10, 20, 30]
```

```
1 d=[11, 22, 33, 44, 55]
```

```
1 d.remove(11)
2 print(d)
```

```
[22, 33, 44, 55]
```

```
1 d.pop()
2 print(d)
```

```
[11, 22]
```

```
1 d.pop(2) # pops out element at index 2
2 print(d)
```

```
[11, 22, 55]
```

```
1 print(data)
```

```
[[11, 22, 33, 44], [55, 66, 77, 88], [99, 111, 222, 333], [444, 555, 666, 777]]
```

```
1 data.clear()
```

```
1 print(data)
```

```
[]
```

```
1 data=[11, 22, 33, 11, 44, 11, 55]
2 data.index(33)
```

```
2
```

```
1 data.index(11) # searches from 0th index
```

```
0
```

```
1 data.index(11, 1) # searches from 1st index
```

```
3
```

```
1 print(data)
2 data.index(420) #throws error as 420 is not present
```

```
[11, 22, 33, 11, 44, 11, 55]
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-87-2b800982ef10> in <cell line: 2>()
      1 print(data)
----> 2 data.index(420) #throws error as 420 is not present
```

```
ValueError: 420 is not in list
```

SEARCH STACK OVERFLOW

```
1 data.count(11)
2 print(data)
```

```
[11, 22, 33, 11, 44, 11, 55]
```

```
1 data[::-1] # temporary
```

```
[55, 11, 44, 11, 33, 22, 11]
```

```
1 data.reverse() # makes changes in place; doesn't return, makes changes in list itself
```

```
1 print(data)
```

```
[55, 11, 44, 11, 33, 22, 11]
```

```
1 data
```

```
[55, 11, 44, 11, 33, 22, 11]
```

```
1 data.sort() # sorts in ascending order by default; doesn't return, makes changes in list itself
```

```
1 data
```

```
[11, 11, 11, 22, 33, 44, 55]
```

```
1 data.sort(reverse=True) # adding attribute reverse=True, reverses list in descending order
```

```
1 data
```

```
[55, 44, 33, 22, 11, 11, 11]
```

```
1 data=[22, 11, 44, 55, 22, 77, 6]
2 print("sorted: ", sorted(data)) # returns sorted object, list is not affected
3 print(data)
```

```
sorted: [6, 11, 22, 22, 44, 55, 77]
[22, 11, 44, 55, 22, 77, 6]
```

```
1 # enter all elements in a list first blank
2 # then print sum of all , use sum()
3 list1=[]
4 while True:
5     ele=input("Enter element: ")
6     if ele=="":
7         break
8     list1.append(float(ele))
9 print(list1)
10 print("sum: ", sum(list1))
```

```
Enter element: 45
Enter element: 65
Enter element: 41
Enter element: 62
Enter element: 23
Enter element:
[45.0, 65.0, 41.0, 62.0, 23.0]
sum: 236.0
```

```
1 # enter all elements in a list first blank
2 # print all elements lesser than the average
3 list1=[]
4 while True:
5     ele=input("Enter element: ")
6     if ele=="":
7         break
8     list1.append(float(ele))
9 print(list1)
10 average=sum(list1)/len(list1)
11 print("Average: ", sum(list1)/len(list1))
12 for elem in list1:
13     if elem < average:
14         print(elem, end=" ")
```

```
Enter element: 95
Enter element: 12
Enter element: 23
Enter element: 34
```

```

Enter element: 45
Enter element: 56
Enter element:
[95.0, 12.0, 23.0, 34.0, 45.0, 56.0]
Average:  44.166666666666664
12.0 23.0 34.0

```

```

1 # enter all elements in a list first blank
2 # print all elements lesser than the average in sorted order
3 list1=[]
4 while True:
5     ele=input("Enter element: ")
6     if ele=="":
7         break
8     list1.append(float(ele))
9 print(list1)
10 average=sum(list1)/len(list1)
11 print("Average: ", sum(list1)/len(list1))
12 list1.sort()
13 for elem in list1:
14     if elem < average:
15         print(elem, end=" ")

```

```

Enter element: 95
Enter element: 23
Enter element: 12
Enter element: 45
Enter element: 34
Enter element: 56
Enter element:
[95.0, 23.0, 12.0, 45.0, 34.0, 56.0]
Average:  44.166666666666664
[12.0, 23.0, 34.0]

```

```

1 # enter all elements in a list first blank
2 # print second largest & second smallest
3 list1=[]
4 while True:
5     ele=input("Enter element: ")
6     if ele=="":
7         break
8     list1.append(float(ele))
9 print(list1)
10 list1.sort()
11 print(list1)
12 print("second largest:", list1[-2])
13 print("second smallest:", list1[1])

```

```

Enter element: 95
Enter element: 87
Enter element: 45
Enter element: 2
Enter element: 12
Enter element: 34
Enter element: 56
Enter element: 43
Enter element:

```

```
[95.0, 87.0, 45.0, 2.0, 12.0, 34.0, 56.0, 43.0]
[2.0, 12.0, 34.0, 43.0, 45.0, 56.0, 87.0, 95.0]
second largest: 87.0
second smallest: 12.0
```

```
1 # [1, 2, 3, 4, 5]
2 #rotate content for n time given by user
3 # pass1: [2, 3, 4, 5, 1]
4 # pass2: [3, 4, 5, 1, 2]
5
6 list1=[1, 2, 3, 4, 5]
7 print(list1)
8 i=int(input("Enter clock cycles: "))
9 while i>0:
10     n=list1.pop(0)
11     list1.append(n)
12     print(list1)
13     i-=1
```

```
[1, 2, 3, 4, 5]
Enter clock cycles: 5
[2, 3, 4, 5, 1]
[3, 4, 5, 1, 2]
[4, 5, 1, 2, 3]
[5, 1, 2, 3, 4]
[1, 2, 3, 4, 5]
```

```
1 # [1, 2, 3, 4, 5]
2 #rotate content anti-clockwise for n time given by user
3 # pass1: [2, 3, 4, 5, 1]
4 # pass2: [3, 4, 5, 1, 2]
5 list1=[1, 2, 3, 4, 5]
6 print(list1)
7 i=int(input("Enter clock cycles: "))
8 while i>0:
9     n=list1.pop(4)
10     list1.insert(0, n)
11     print(list1)
12     i-=1
```

```
[1, 2, 3, 4, 5]
Enter clock cycles: 5
[5, 1, 2, 3, 4]
[4, 5, 1, 2, 3]
[3, 4, 5, 1, 2]
[2, 3, 4, 5, 1]
[1, 2, 3, 4, 5]
```

1

