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

```
list
```

```
data=()
type(data)
```

```
tuple
```

```
data={}#dict not set
type(data)
```

```
set
```

```
data={0}#set
type(data)
```

```
set
```

```
data=set()#way to create blank set
type(data)
```

```
set
```

LIST

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

```

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

```
#indexed access
#positive   #negative

data=[11,22,33,44,55,66]
print(data[0])
print(data[-1])
print(len(data))
```

```
11
66
6
```

```
#indexed access
data=[11,22,33,44,55,66]
```

```
for i in range(len(data)):
    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
```

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

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

```
data1=[11,22,33,44]
data2=[1,2,3]
data3=data1+data2
data4=data1*2
print(data3)
print(data4)
```

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

```
data=[11,22,33,44,55,66,77,88,99,111]
print(data[::2])
print(data)
data2=data[::-1]
```

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

```
data2
```

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

```
list2d=[[10,20,30],[40,50],[60,70,80,90,100]]
```

```
len(list2d)
```

```
3
```

```
for row in list2d:
    print(row)
```

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

```
len(list2d[2])
```

```
5
```

```
'''
print each element using index only
10 20 30
40 50
...
'''
for row in range(0,len(list2d)):
    for column in range(0,len(list2d[row])):
        #print("at",row,",","column:",column,":",list2d[row][column])
        print(list2d[row][column]," ",end="")
    print()
```

```
10  20  30
40  50
60  70  80  90  100
```

```
data=[[11,22,33,44],[55,66,77,88],[99,111,222,333],[444,555,666,777]]
#print in zigzag manner on norMAL AND REVERSAL
# 11 22 33 44
# 88 77 66 55
# 99 111 222 333
# 777 666 555 444
```

```
data=[[11,22,33,44],[55,66,77,88],[99,111,222,333],[444,555,666,777]]
for index in range(0,len(data)):
    temp=data[index]
    if index%2!=0:
        print(temp)
    else:
        print(temp[::-1])
```

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

```
datalist=[]
```

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

```
Enter data:-99
```

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

```
datalist has: 6 [333, 1000, 19, 6, -99, 212]
```

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

```
d1=[11,22,33]
d2=[10,20,30]
d1=d1+d2#temp changes
print("d1:",d1)
```

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

```
d1=[11,22,33]
d2=[10,20,30]
d1.extend(d2)#permanent changes
print(d1)
```

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

```
d1
```

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

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

```
d.pop(2)
```

```
11
```

```
print(d)
```

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

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

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-67-efef4923fc3f> in <cell line: 1>()
----> 1 d.remove(11)
      2 print(d)

ValueError: list.remove(x): x not in list
```

SEARCH STACK OVERFLOW

```
print(data)
```

```
[]
```

```
data.clear()
```

```
data=[11,22,33,11,44,11,55]
print(data.index(11))#searches from 0th index
print(data.index(11,1))#searches from 1st index
print(data.index(420))
```

```
0
3
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-87-cfe734d64b05> in <cell line: 4>()
      2 print(data.index(11))#searches from 0th index
      3 print(data.index(11,1))#searches from 1st index
----> 4 print(data.index(420))

ValueError: 420 is not in list
```

SEARCH STACK OVERFLOW

```
print(data)
data.count(420)
```

```
[11, 22, 33, 11, 44, 11, 55]
0
```

```
data
```

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

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

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

```
data.reverse()#inplace
```

```
print(data)
```

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

```
data
```

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

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

```
data
```

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

```
data
```

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

```
data=[22,11,44,55,22,77,6]
print("sorted:",sorted(data))#temp
print(data)

```

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

```
sum(data)
```

```
237
```

```
#Enter all elements in list till first blank
#then print sum of all use sum()
dlist=[]
while True:
    data=input("Enter data:")
    if data=='':
        break
    dlist.append(float(data))
print("Sum of all",len(dlist),"elements is:",sum(dlist))
```

```
Enter data:34
Enter data:11
Enter data:78
Enter data:90
Enter data:34
Enter data:1
Enter data:
Sum of all 6 elements is: 248.0
```

```
#Enter all elements in list till first blank
#then print sum of all use sum()
#print all elements lesser than avg of all
dlist=[]
while True:
    data=input("Enter data:")
    if data=='':
        break
    dlist.append(float(data))
avg=sum(dlist)/len(dlist)
print("Average is:",avg)
for i in dlist:
    if i<avg:
        print(i)


```

```
Enter data:11
Enter data:66
Enter data:100
Enter data:22
Enter data:88
Enter data:33
Enter data:93
Enter data:17
Enter data:
Average is: 53.75
11.0
22.0
33.0
17.0
```

```
#Enter all elements in list till first blank
#then print sum of all use sum()
#print all elements lesser than avg of all in sorted order
dlist=[]
while True:
    data=input("Enter data:")
    if data=='':
        break
    dlist.append(float(data))
avg=sum(dlist)/len(dlist)
dlist.sort()
print("Average is:",avg)
for i in dlist:
    if i<avg:
        print(i)

```

```
Enter data:33
Enter data:11
Enter data:77
Enter data:34
Enter data:89
Enter data:67
Enter data:
Average is: 51.833333333333336
11.0
33.0
34.0
```

```
#Enter all elements in list till first blank
#print second largest and second smallest
dlist=[]
while True:
    data=input("Enter data:")
    if data=='':
        break
    dlist.append(float(data))
dlist.sort()
print("Second minimum:",dlist[1],"\nSocond Largest:",dlist[-2])
```

```
Enter data:33
Enter data:11
Enter data:66
Enter data:88
Enter data:44
Enter data:99
Enter data:55
Enter data:
Second minimum: 33.0
Socond Largest: 88.0
```

```
#[1,2,3,4,5]
#rotate content for n time given by user
'''
1:[2,3,4,5,1]
2:[3,4,5,1,2]
3:[4,5,1,2,3]
'''
d=[1,2,3,4,5]
n=int(input("Number of cycles:"))
while n>0:
    t=d.pop(0)
    d.append(t)
    print(d)
    n-=1
```

```
Number of 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]
```

```
#[1,2,3,4,5]
#rotate content for n time given by user
'''
1:[5,1,2,3,4]
2:[4,5,1,2,3]

'''
d=[1,2,3,4,5]
n=int(input("Number of cycles:"))
while n>0:
    t=d.pop()
    d.insert(0,t)
    print(d)
    n-=1
```

```
Number of 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]
```

```
t=(10,20,30)
print(t,type(t),id(t))
t=(11,22,33)
print(t,id(t))
```

```
(10, 20, 30) <class 'tuple'> 140657059278080
(11, 22, 33) 140656714481792
```

```
(11, 22, 33)
```

```
t[0]=t[0]*100
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-6-e83bd1d9a7a8> in <cell line: 1>()
----> 1 t[0]=t[0]*100

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

SEARCH STACK OVERFLOW

```
for i in t:
    print(i)
```

```
10
20
30
```

```
t=(10,20,[11,22,33])
```

```
t[2].append(100)
```

```
print(t)
```

```
(10, 20, [11, 22, 33, 100])
```

```
t[2].append([1000,2000,3000])
```

```
print(t)
```

```
(10, 20, [11, 22, 33, 100, [1000, 2000, 3000]])
```

```
(t[2][-1])[-1]
```

```
3000
```

```
s={11,22,11,33,11,22,44,11,22}
```

```
print(s)
```

```
{33, 11, 44, 22}
```

```
s
```

```
{11, 22, 33, 44}
```

```
s[1]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-20-f8bb2b116405> in <cell line: 1>()
----> 1 s[1]

TypeError: 'set' object is not subscriptable
```

SEARCH STACK OVERFLOW

```
for i in s:
    print(i)
```

```
33
11
44
22
```

```
a={1,2,3,4}
b={3,4,5,6}
b-a
```

```
{5, 6}
```

```
emp={1,2,3,4,5,6,7,8,9}
drama={2,4,6,7}
sing={1,3,6,8}
sports={1,3,6,8}
```

```
drama|sing|sports
```

```
{1, 2, 3, 4, 6, 7, 8}
```

```
drama&sing&sports
```

```
{6}
```

```
emp-drama-sing-sports
```

```
{5, 9}
```

```
s=set()
```

```
s.add(int(input("Enter data:")))
```

```
Enter data:6
```

```
print(s,len(s))
```

```
{33, 66, 11, 6} 4
```

```python
s
```

```
{6, 11, 33, 66}
```

```python
print(s)
```

```
{33, 66}
```

```python
s.remove(11)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-44-05b54e818299> in <cell line: 1>()
----> 1 s.remove(11)

KeyError: 11
```

SEARCH STACK OVERFLOW

```python
s.discard(6)
```

```python
s={11,22,33,11,44,33,66,55,77,88,11,22}
```

```python
print(s)
```

```
{22, 55, 88}
```

```python
s.pop(1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-61-f06e91dfbaaa> in <cell line: 1>()
----> 1 s.pop(1)

TypeError: set.pop() takes no arguments (1 given)
```

SEARCH STACK OVERFLOW

```python
#Enter all elements in list till first blank
#print only unique data
dlist=[]
while True:
    data=input("Enter data:")
    if data=='':
        break
    dlist.append(float(data))
print("unique data\n")
for i in sorted(set(dlist)):
    print(i)
```

```
Enter data:11
Enter data:22
Enter data:55
Enter data:22
Enter data:77
Enter data:11
Enter data:99
Enter data:22
Enter data:33
Enter data:11
Enter data:99
Enter data:
unique data

11.0
22.0
33.0
55.0
77.0
99.0
```

```
#list all elements in sorted manner with their count.
dlist=[]
while True:
    data=input("Enter data:")
    if data=='':
        break
    dlist.append(float(data))
print("unique data\n")
for i in sorted(set(dlist)):
    print(i,"frequency:",dlist.count(i))
```

```
Enter data:11
Enter data:22
Enter data:33
Enter data:11
Enter data:8
Enter data:3
Enter data:9
Enter data:11
Enter data:6
Enter data:3
Enter data:9
Enter data:11
Enter data:8
Enter data:11
Enter data:11
Enter data:
unique data

3.0 frequency: 2
6.0 frequency: 1
8.0 frequency: 2
9.0 frequency: 2
11.0 frequency: 6
22.0 frequency: 1
33.0 frequency: 1
```

```
#list all elements in sorted manner with their count.
dlist=[]
while True:
    data=input("Enter data:")
    if data=='':
        break
    dlist.append(float(data))
print("unique data\n")
maxelement=0
max=0
for i in sorted(set(dlist)):
   if dlist.count(i)>max:
       max=dlist.count(i)
       maxelement=i
print("Maximum Frequency of",max,"is:",maxelement)
```

```
Enter data:11
Enter data:22
Enter data:11
Enter data:33
Enter data:11
Enter data:55
Enter data:22
Enter data:11
Enter data:66
Enter data:11
Enter data:99
Enter data:11
Enter data:
unique data

Maximum Frequency of 6 is: 11.0
```

```
d={1:"One",2:"Two",3:"Three",4:"Four"}
```

```
d[2]
```

```
'Two'
```

```
d[5]="five"
```

```
print(d)
```

```
{1: 'Ek Number', 2: 'Two', 3: 'Three', 4: 'Four', 5: 'five'}
```

```
d[1]="Ek Number"
```

```
#Print number in words digit by digit
#Example: 123--->one two three
dw={1:"one",2:"two",3:"three",4:"four",5:"five",6:"six",7:"seven",8:"eight",9:"nine",0:"zero"}
no=int(input("Enter a number:"))
rno=0
while no>0:
    rno=rno*10+(no%10)
    no=no//10
while rno>0:
    d=rno%10
    rno=rno//10
    print(dw[d],end=" ")
```

```
Enter a number:8194
eight one nine four
```

```
#take a number and print only unique digits only
#in sorted mannerin words
#input:361421288
#output:one two three four six eight
dw={1:"one",2:"two",3:"three",4:"four",5:"five",6:"six",7:"seven",8:"eight",9:"nine",0:"zero"}
s=set()
no=int(input("Enter a number:"))
while no>0:
    d=no%10
    no=no//10
    s.add(d)
for digit in sorted(s):
    print(dw[digit],end=" ")


```

```
Enter a number:361421288
one two three four six eight
```

```
dw={1:"one",2:"two",3:"three",4:"four",5:"five",6:"six",7:"seven",8:"eight",9:"nine",0:"zero"}
for i in dw:
    print(i)
```

```
1
2
3
4
5
6
7
8
9
0
```

```
print(dw.keys())
```

```
dict_keys([1, 2, 3, 4, 5, 6, 7, 8, 9, 0])
```

```
print(dw.values())

```

```
dict_values(['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'zero'])
```

```
print(dw.items())
```

```
dict_items([(1, 'one'), (2, 'two'), (3, 'three'), (4, 'four'), (5, 'five'), (6, 'six'), (7, 'seven'), (8, 'eight'), (9, 'nine'), (0, 'zero')])
```

```
for k,i in dw.items():
    print("Key:",k,":Item:",i)
```

```
Key: 1 :Item: one
Key: 2 :Item: two
Key: 3 :Item: three
Key: 4 :Item: four
Key: 5 :Item: five
Key: 6 :Item: six
Key: 7 :Item: seven
Key: 8 :Item: eight
Key: 9 :Item: nine
Key: 0 :Item: zero
```