Python Datastructures are 2 types 1.Inbuid data structures 2.User defined data structures

What is Data structure?

-->Data structures is collection of data types(we can declare more than one value) -->ex:List,tuple,set,dictionary.

List:

1.we can define list with [] 2.List conatains inbuild methods 3.List allows multile data types. 4.duplicate values are allowed. 5.covered appened(),copy(),reverse(),string indexing()

indexing is 2 types:

forward indexing(left to right) backward indexing(right to left) step indexing

What is Matrix?

-->Collection of Data structures is called matrix.

```
i=5
type(i)
```

```
int
```

```
l=[]
l
```

```
[]
```

```
type(l)
```

```
list
```

```
len()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-45-adf3103c7c3e> in <cell line: 0>()
----> 1 len()

TypeError: len() takes exactly one argument (0 given)
```

Archana Kapu
Jan 27, 2025

will get error message as we didnt pass any arguments

```
len(l)
```

```
0
```

append(): Adds an element to the end of the list.

```
l.append(10)
l.remove(10)
l
```

```
[]
```

```
l
```

```
[]
```

remove(): Removes the first occurrence of a specified element.

```
l.remove(10)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-56-dc80452e70d5> in <cell line: 0>()
----> 1 l.remove(10)

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

```
l
```

```
[]
```

```
l.remove(10)
```

```
l
```

```
l.remove(10)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-58-dc80452e70d5> in <cell line: 0>()
----> 1 l.remove(10)

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

```
l
```

```
[]
```

```
l.remove(10)
```

```
l
```

```
len(l)
```

```
l.append(10,20,30,40)
```

```
l.append(10)
l.append(20)
l.append(30)
l.append(40)
```

```
l
```

```
l1=[]
l1
```

```
[]
```

```
l1.append(70)
l1.append(2.3)
l1.append(True)
l1.append(10+2j)
l1.append([10,20,30])
```

```
l1
```

```
[70, 2.3, True, (10+2j), [10, 20, 30]]
```

```
print(l)
print(l1)
```

```
[]
[70, 2.3, True, (10+2j), [10, 20, 30]]
```

```
print(id(l))
```

```
137788748694784
```

```
print(id(l1))
```

Archana Kapu
Jan 27, 2025

remove method deletes the value present in the first

Archana Kapu
Jan 27, 2025

append methos takes only 1 argument at time, as we passed 4 arguments, error occured.

Archana Kapu
Jan 28, 2025

in data structures id values are different

```
137788747412864
```

how to find length of list?

```
print(len(l))
print(len(l1))
```

```
0
5
```

```
l1
```

```
[70, 2.3, True, (10+2j), [10, 20, 30]]
```

copy(): Returns a shallow copy of the list.

copying 1 list into another>list with in the another list is called nested list.

```
l2=l1.copy()
```

```
l2
```

```
[70, 2.3, True, (10+2j), [10, 20, 30]]
```

Comparing 2 lists using assignment operators

```
l1==l2
```

```
True
```

```
l1!=l2
```

```
False
```

```
l
```

```
[]
```

```
l==l2
```

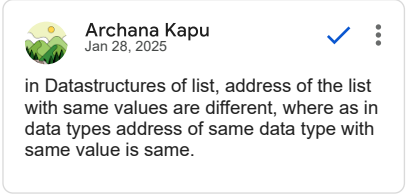
```
False
```

```
l!=l2
```

```
True
```

```
l1==l2
```

```
True
```




```
print(l1)
print(l2)
```

```
[70, 2.3, True, (10+2j), [10, 20, 30]]
[70, 2.3, True, (10+2j), [10, 20, 30]]
```

```
print(id(l1))==print(id(l2))
```

```
137788747412864
137788747786048
True
```

```
a=4
b=4
```

```
print(id(a))==print(id(b))
```

```
10750952
10750952
True
```

Archana Kapu
Jan 28, 2025
need to pass atleast 1 value in remove method

```
l
```

```
[]
```

```
l.remove()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-82-6e57923871b5> in <cell line: 0>()
----> 1 l.remove()

TypeError: list.remove() takes exactly one argument (0 given)
```

```
l.remove(10)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-83-dc80452e70d5> in <cell line: 0>()
----> 1 l.remove(10)

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

Archana Kapu
Jan 28, 2025
remove method removes 1st occurences of specified value in list

```
l
```

```
[]
```

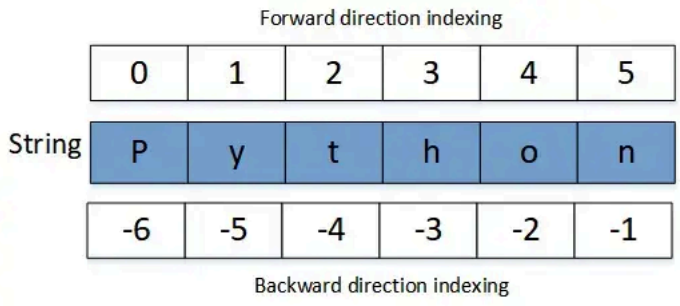
```
l.remove(10)
l
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-85-dcd5764900bd> in <cell line: 0>()
----> 1 l.remove(10)
      2 l

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

list indexing and slicing

String Indexing:




```
s7='ArchanaDatascientist'
s7
```

```
'ArchanaDatascientist'
```

Archana Kapu
Jan 28, 2025
as we pass index out of rage, string index out of range error occured.

```
s7[0]
```

```
'A'
```

```
s7[1]
```

```
'n'
```

```
s7[20]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-89-6610ae8d39fa> in <cell line: 0>()
----> 1 s7[20]

IndexError: string index out of range
```

```
s7[-1]
```

```
't'
```

```
s7[-13]
```

```
'D'
```

```
for i in s7:
  print(i)
```

```
A
r
c
h
a
n
a
D
a
t
a
s
c
i
e
n
t
i
s
t
```

slicing,in pyton slicing defines with[:]

slicing are 3 types

forward slicing

backward slicing

step slicing

Forward slicing:

[2:7]---->here 2 is left indexing and 7 is right indexing.

2nd index:(n-1) formua right index(7-1)

output will print as 2th to 6th

backward slicing:

[-7:-1]--->here left indexing is -7:-1-1==-7:-2 p/o:-7,-6,-5,-4,-3,-2

```
s9='nareshit'
s9
```

```
'nareshit'
```

```
s9[0:9]
s9
```

```
'nareshit'
```

```
s9[1:8]
```

```
'areshit'
```

Archana Kapu
Jan 28, 2025
(edited Jan 28, 2025)

n=n-1
n=-3-1=-4

Archana Kapu
Jan 28, 2025

n=n-1 every time applies to right indexing

```
s9[1:-3]
```

```
'ares'
```

```
s9[1:-4]
```

```
'are'
```

step slicing

```
step_indexing=[1,2,3,4,5,6,7,8,9,10]
step_indexing[0:10:4]#here this mean print values from oth to 10th indexing by stepping 4steps
```

```
[1, 5, 9]
```

```
step_indexing[1:7:2]
```

```
[2, 4, 6]
```

```
step_indexing
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
step_indexing[0:10:5]
```

```
[1, 6]
```

```
step_indexing
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
#empty slice means all elemnets will print
step_indexing[:]
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
l
```

```
[]
```

```
#reverse method will print values in reverse order of the string
l.reverse()
```

```
l
```

```
[]
```

```
t= 5
r = 2
print(t // r)
```

```
2
```

```
l
```

```
[]
```

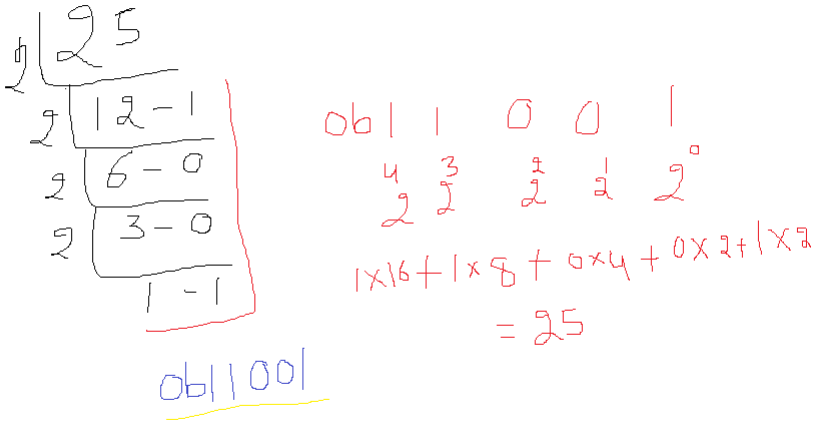
29th Dec

1.number system 2.list data structure 3.deepseek lllm model

```
25
```

```
25
```

Conversion of numeric to binary-->25 to binary conversion)




```
bin(25)
```

```
'0b11001'
```

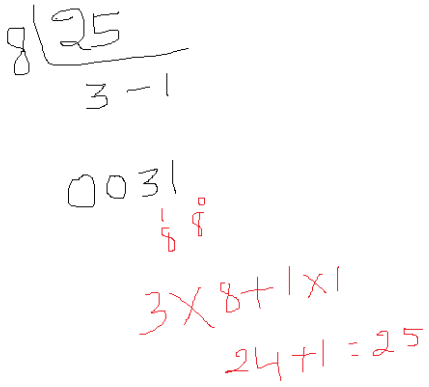
```
int(0b100011)
```

```
35
```

```
bin(35)
```

```
'0b100011'
```

numeric to octal conversion




```
oct(25)
```

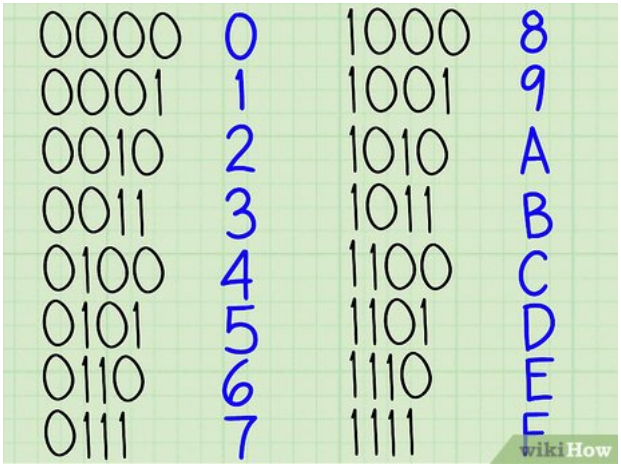
```
'0o31'
```

```
int(0o31)
```

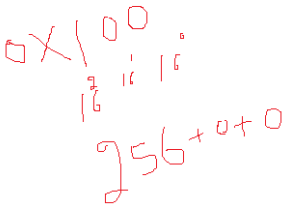
```
25
```

```
bin(7)
```

```
'0b111'
```



number to hexagonal conversion



number to hexagonal conversion

```
hex(1)
```

0x1

```
hex(2)
```

0x2

```
hex(8)
```

0x8

```
hex(10)
```

0xa

```
hex(11)
```

oxb

```
hex(256)
```

ox100

```
hex(1)
```

```
'0x1'
```

```
hex(9)
```

```
'0x9'
```

```
hex(10)
```

```
'0xa'
```

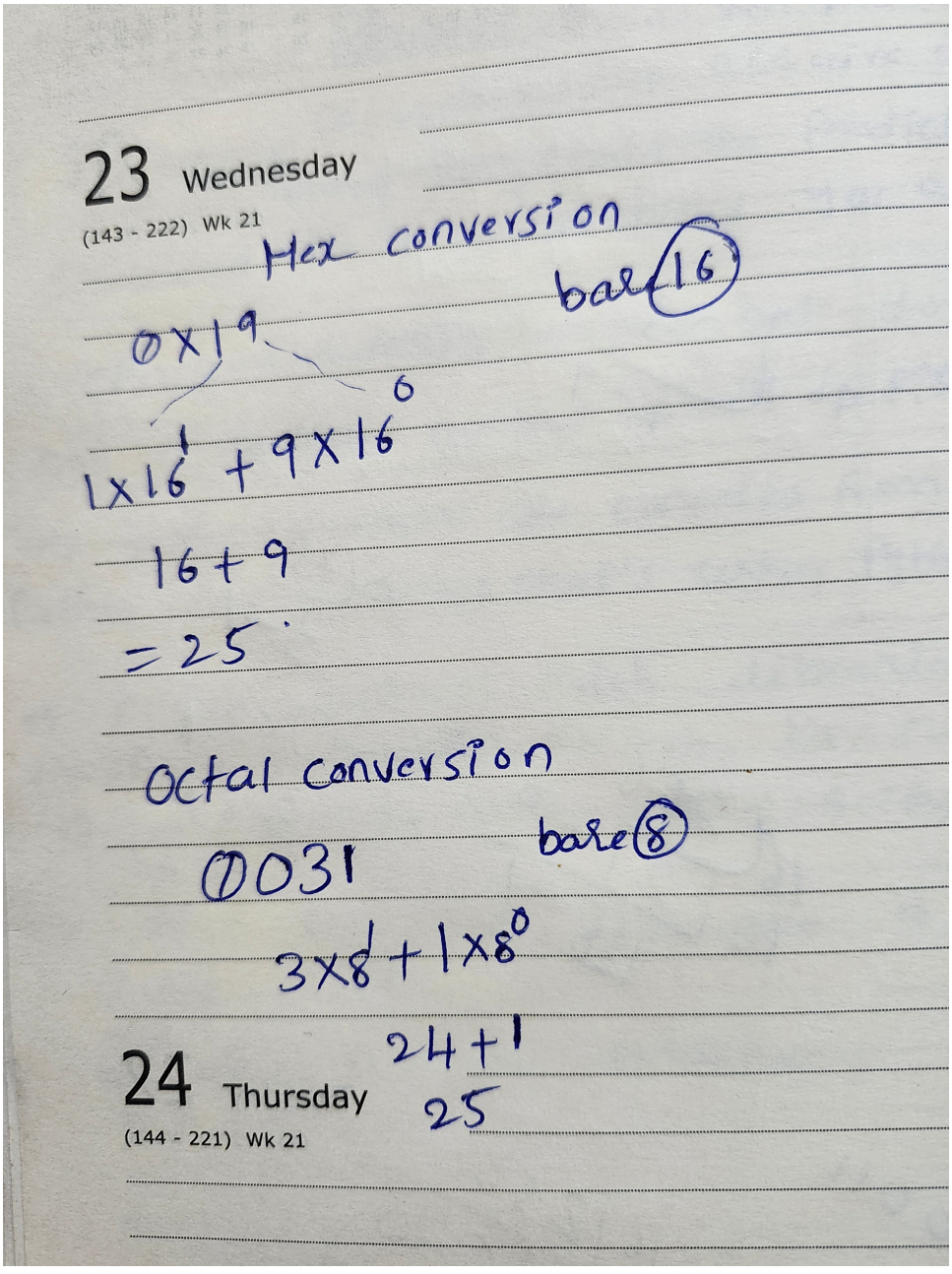
```
hex(11)
```

```
'0xb'
```

```
hex(256)
```

```
'0x100'
```

hexagonal convesrison and octal conversion




```
0o31
```

```
25
```

```
0x19
```

```
25
```

```
0x15
```

```
21
```

Q)how to swap 2 variables in python: a,b=5,6 after swap a=6 b=5

```
#1.using temp variable
a1=5
b1=6


temp=a1
a1=b1
b1=temp


print(a1)
print(b1)
```

```
6
5
```

Archana Kapu
Feb 1, 2025
3*8^1+1*8^0=24+1=25

Archana Kapu
Feb 1, 2025
1*16^1+5*16^0=16+5=21

```
#2.swapping variables using sum
a3=5
b3=6


a3=a3+b3
b3=a3-b3
a3=a3-b3


print(a3)
print(b3)
```

```
6
5
```

```
bin(5)
```

```
'0b101'
```

```
bin(6)
```

```
'0b110'
```

```
#3.using binary valueswe can swap
print(0b101)
print(0b110)
```

```
5
6
```

```
print(0b110)
print(0b101)
```

```
6
5
```

```
#4th way of swapping
d,e=5,6


e,d=d,e


print(d)
print(e)
```

```
6
5
```

```
l1
```

```
[70, 2.3, True, (10+2j), [10, 20, 30]]
```

```
l2
```

```
[70, 2.3, True, (10+2j), [10, 20, 30]]
```

```
l2.count(True)
```

```
1
```

```
l2.count(70)
```

```
1
```

```
l2.append(70)
```

```
l2
```

```
[70, 2.3, True, (10+2j), [10, 20, 30], 70]
```

```
l2.count(70)
```

```
2
```

```
l2[:]
```

```
[70, 2.3, True, (10+2j), [10, 20, 30], 70]
```

```
l2[:5]
```

```
[70, 2.3, True, (10+2j), [10, 20, 30]]
```

```
l2[5:]
```

```
[70]
```

```
l
```

```
[]
```

```
l.append(40)
```

```
l
```

```
[40]
```

```
l.append(50)
```

```
l
```

```
[40, 50]
```

```
l[1:]
```

```
[50]
```

```
l[2:]
```

```
[]
```

```
l
```

```
[40, 50]
```

```
l.append(20)
```

```
l
```

```
[40, 50, 20]
```

```
l[:-1]
```

```
[40, 50]
```

```
l[::-1]
```

```
[20, 50, 40]
```

```
l7=[40,30,20,10,40,30,20]
```

```
l7
```

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

```
l7[::-1]#adance slicing
```

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

```
l7
```

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

```
l7[::-2]
```

```
[20, 40, 20, 40]
```

```
l7.index(20)
```

```
2
```

```
l2
```

```
[70, 2.3, True, (10+2j), [10, 20, 30], 70]
```

Archana Kapu
Feb 1, 2025
prints list of the variables in reverse order

```
id(l2)
```

```
137788747786048
```

```
l2.clear()
```

```
l2
```

```
[]
```

Archana Kapu
Feb 1, 2025
index fun writtens first occurence of index value of attribute

```
id(l2)
```

```
137788747786048
```

```
del l2
```

```
l2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-171-ea320d2ace30> in <cell line: 0>()
----> 1 l2

NameError: name 'l2' is not defined
```

```
l1
```

```
[70, 2.3, True, (10+2j), [10, 20, 30]]
```

Archana Kapu
Feb 1, 2025
after clearing the list of varibales also shows same address of the list

```
l1.pop()
```

```
[10, 20, 30]
```

Archana Kapu
Feb 1, 2025
deleting the list here.

```
l1
```

```
[70, 2.3, True, (10+2j)]
```

diff between pop and remove

remove ---we need to define value of an element

pop--need to mention index value

```
l
```

```
[40, 50, 20]
```

```
l2=l.copy()
```

```
l2
```

```
[40, 50, 20]
```

```
l2.remove(50)
```

```
l2
```

```
[40, 20]
```

```
l2.pop(-1)
```

```
20
```

```
l2
```

```
[40]
```

```
l2.append(10)
```

```
l2
```

```
[40, 10]
```

```
l2.append(40)
```

```
l2
```

```
[40, 10, 40]
```

```
l2.append(30)
```

```
l2
```

```
[40, 10, 40, 30]
```

```
l2.insert(2,25)
```

```
l2
```

```
[40, 10, 25, 40, 30]
```

```
l
```

```
[40, 50, 20]
```

```
l[0]
```

```
40
```

```
l[0]=400
```

```
l
```

```
[400, 50, 20]
```

- List is muatble
- appened()-appened the element/value at te last
- remove()--remove the value from the list and we need pass argument value which we need to remove.
- pop()--pop the element oor value by the index(by default last)

Archana Kapu
Feb 1, 2025

in insert we need to pass 2 values 1st one is index and 2nd value as value

- copy()-copy the list
- insert()--insert function requires 2 values at which index we need to insert an elemnet and at what place.ex:(2,25) 1st arg--index position 2nd arg---element of a value
- clear()----clear the elements from the list
- del--we can delete the list using del reverse()----reverse the list.

Archana Kapu
Feb 1, 2025

list is muatble

```
l
```

```
[400, 50, 20]
```

```
l1
```

```
[70, 2.3, True, (10+2j)]
```

```
len(l1)
```

```
4
```

```
l2
```

```
[40, 10, 25, 40, 30]
```

```
len(l2)
```

```
5
```

```
l1
```

```
[70, 2.3, True, (10+2j)]
```

```
for i in l1:
  print(i)
```

```
70
2.3
True
(10+2j)
```

```
for i in enumerate (l1):
  print(i)
```

```
(0, 70)
(1, 2.3)
(2, True)
(3, (10+2j))
```

30Jan

```
l
```

```
[400, 50, 20]
```

```
l.sort()#parameter tuning
```

```
l
```

```
[20, 50, 400]
```

```
l.sort(reverse=True)#hyper parameter tuning
```

```
l
```

```
[400, 50, 20]
```

```
l5=['m','a','g']
```

```
l5.sort()
```

```
l5
```

```
['a', 'g', 'm']
```

```
l1
```

```
[70, 2.3, True, (10+2j)]
```

```
l1.sort()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-228-b8f5f256bbcf> in <cell line: 0>()
----> 1 l1.sort()

TypeError: '<' not supported between instances of 'complex' and 'bool'
```

```
a=2
b=3
a+b
```

```
5
```

```
int.__add__(2,3)
```

```
5
```

Archana Kapu
Feb 1, 2025

sort applcable for only with in same data type

```
int.__sub__(2,3)
```

```
-1
```

```
int.__mul__(2,3)
```

```
6
```

```
float.__sub__(2.3,3.3)
```

```
-1.0
```

list is completed