

# PYTHON PROGRAMMING

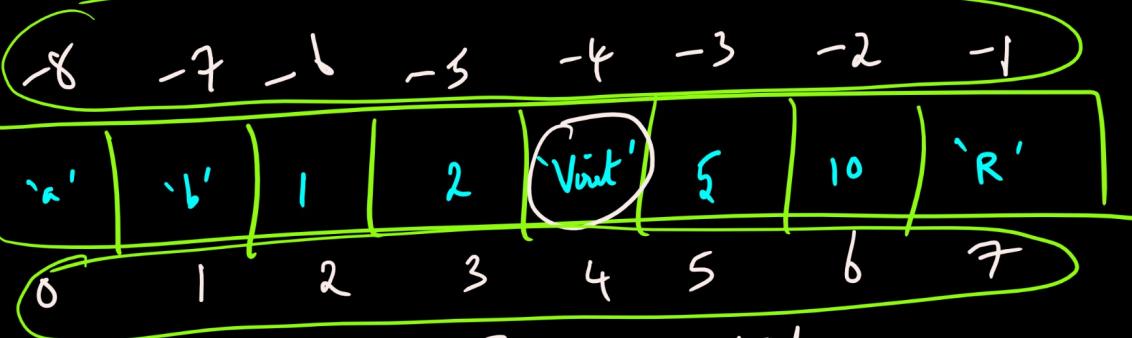
## GATE DA/DSA

Agenda:

→ List & Tuple Operations

$\left\{ \begin{array}{l} \text{Strings} \\ \quad \hookrightarrow \text{Operations} \\ \quad \hookrightarrow \text{inbuilt} \end{array} \right.$

List:



Index

$\text{list}_{-1}[4] \rightarrow \text{Vireet}$

$\text{list}_{-1}[-4] \rightarrow \text{Vireet}$

Let say,  $n = \text{len}(\text{list}_{-1}) \Rightarrow n = 8$

+ve indexing:  $0 \rightarrow n-1 : 0 \text{ to } 7$

-ve indexing:  $-1 \rightarrow -n : -1 \text{ to } -8$

$[[1, 2, 3, 4, 5], [23, [12, 34], 9], 30, 21]] \rightarrow \text{len}(l) \rightarrow 4$

$$l[0] = [1, 2, 3, [4, 5]] \rightarrow l[0]$$

$$l[1] = [23, [12, 34], 9] \rightarrow l[0][3]$$

$$l[2] = 30$$

$$l[3] = 21$$

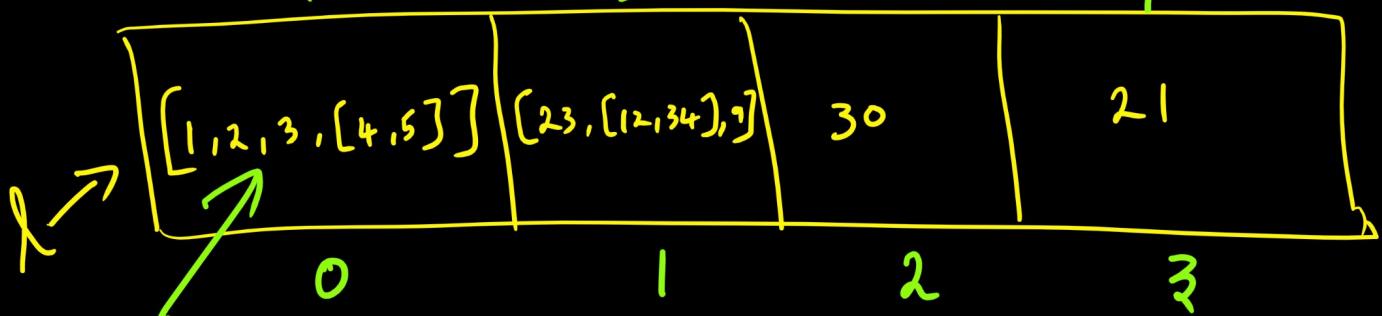
$$l[0][0] = 1 ; l[0][2] = 3$$

$$l[0][1] = 2$$

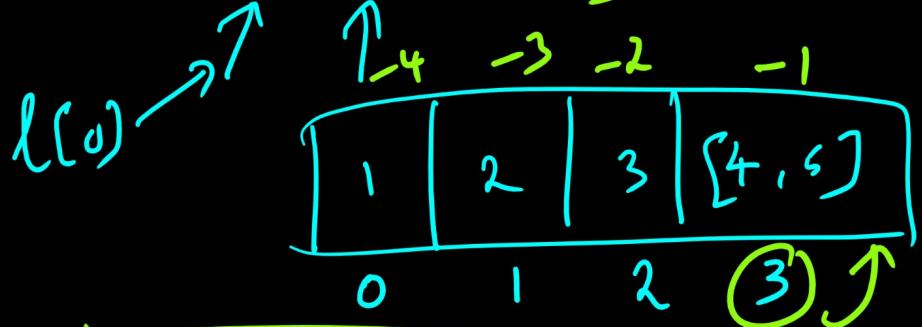
$$[4, 5]$$

$$l[0][3] = \begin{bmatrix} -2 & -1 \\ 4 & 5 \end{bmatrix}$$

$$\frac{l[0][3][0] + l[0][3][-2]}{4}$$

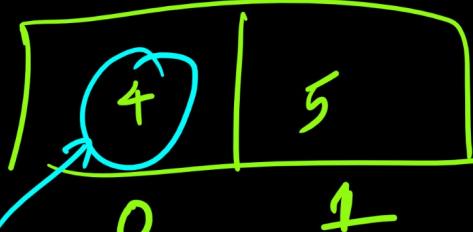


$$l[0] \rightarrow [1, 2, 3, [4, 5]]$$



$$l[0][3] = [4, 5]$$

$$l[0][3][0] \\ l[0][3][1] \\ l[0][3][2]$$



$$l[-4][3][0]$$

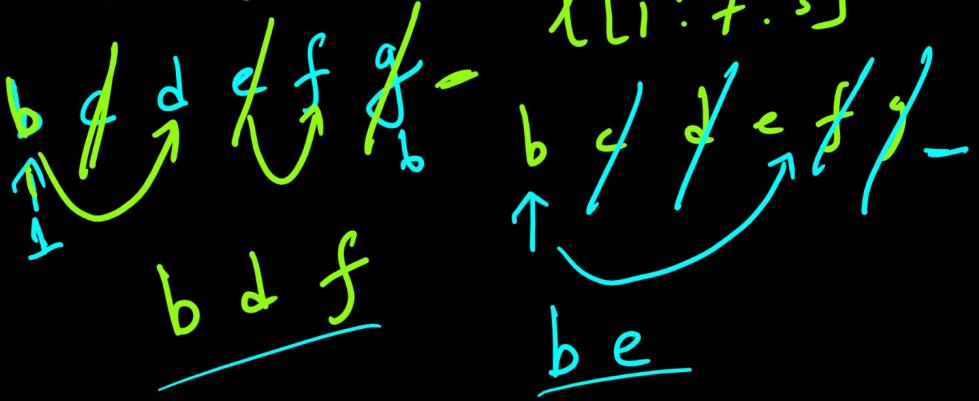
$$l[0]: \text{start index} \quad : \text{end index} \quad : \text{step size}$$

$$l[0]: \text{start index} \quad : \text{step size} \quad : \text{end index}$$

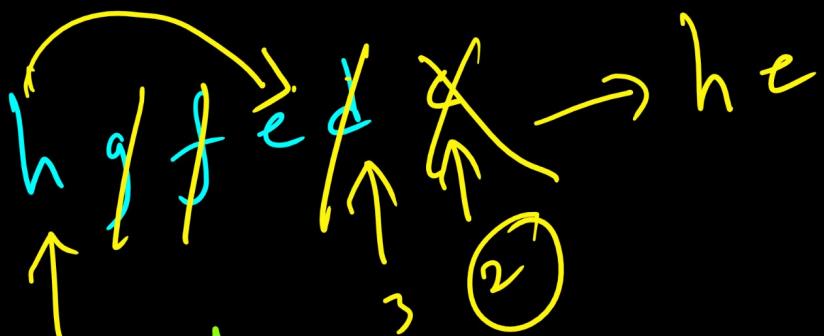
$-10$	$-9$	$-8$	$-7$	$-6$	$-5$	$-4$	$-3$	$-2$	$-1$
a	b	c	d	e	f	g	h	i	j

0 ① 2 3 4 5 6 7 8 9

$\lambda [1 : 7 : 2]$



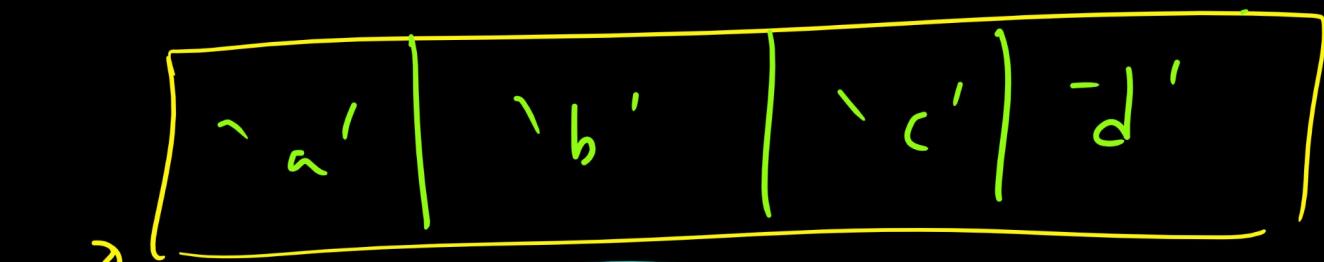
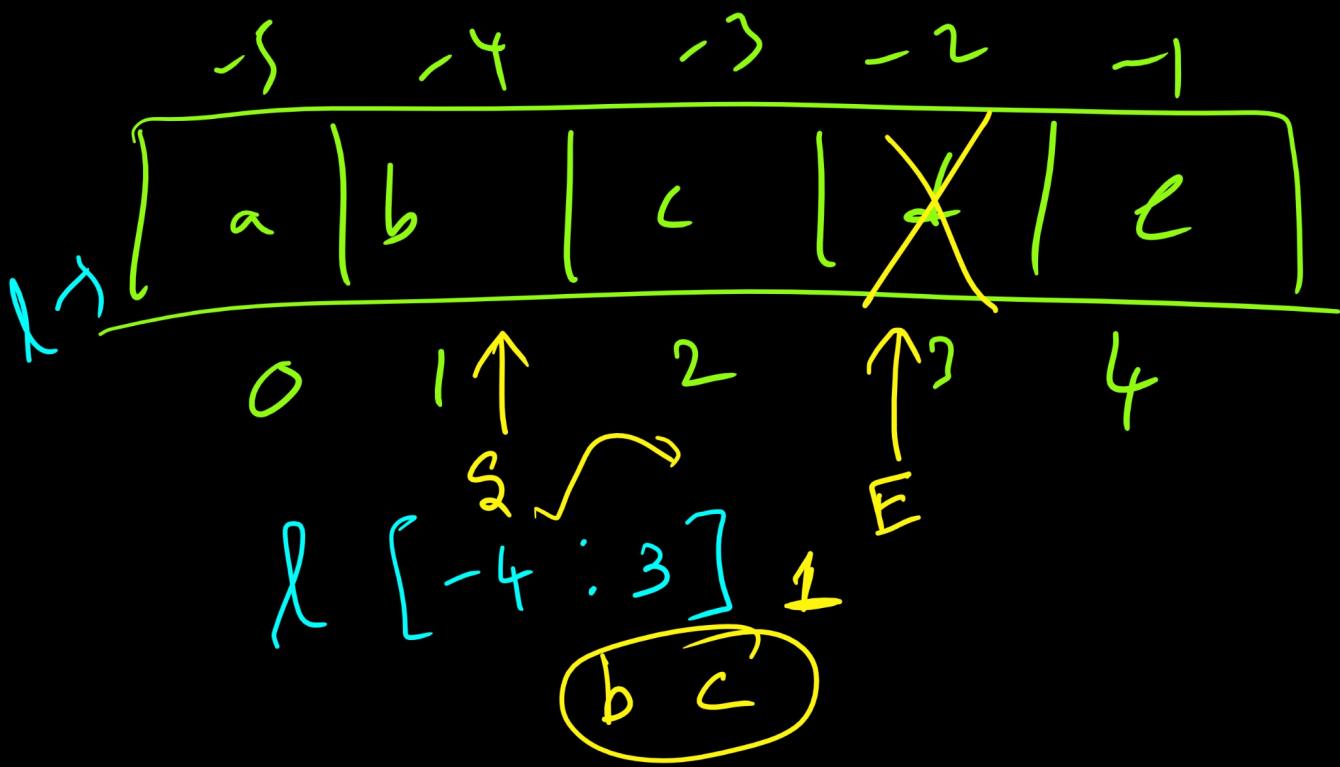
$\lambda [-7 : 2 : -3]$



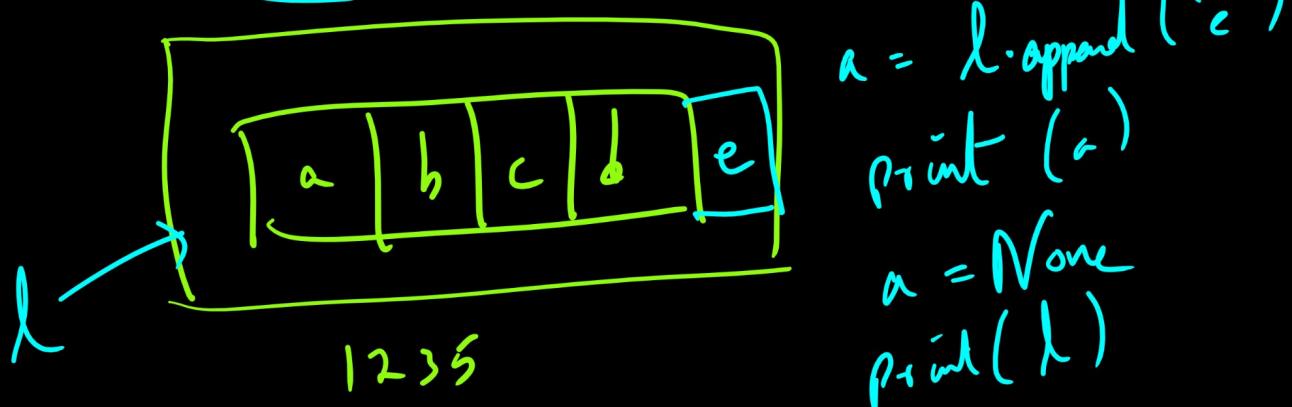
$-10$	$-9$	$-8$	$-7$	$-6$	$-5$	$-4$	$-3$	$-2$	$-1$
a	b	c	d	e	f	g	h	i	j

0 1 2 3 4 5 6 7 8 9

Q1:  $\lambda [-4 : 8]$  |  $\lambda [-4 : -8]$  Q2:  
g h [ ]



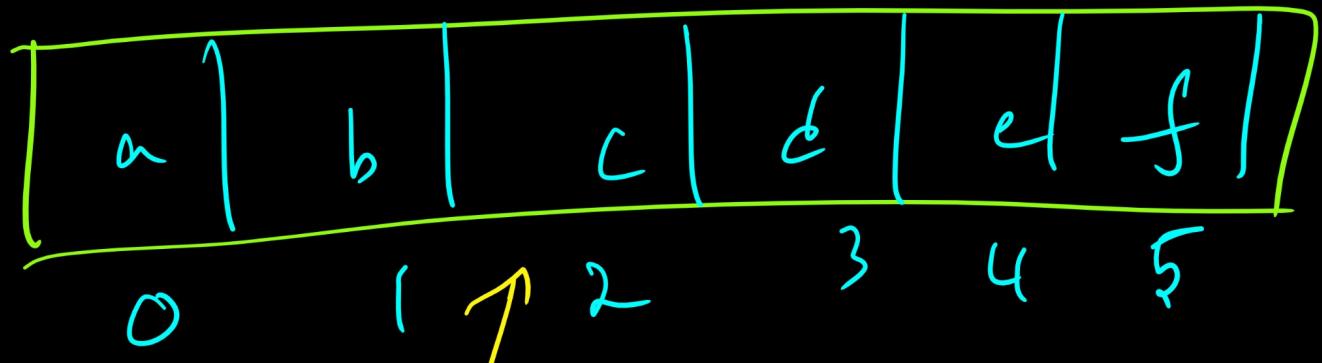
$l = l.append('e')$  → return None



$a = l.append('e')$   
 $print(a)$   
 $a = None$   
 $print(l)$

$l = l.append('e')$  →  $l = None$

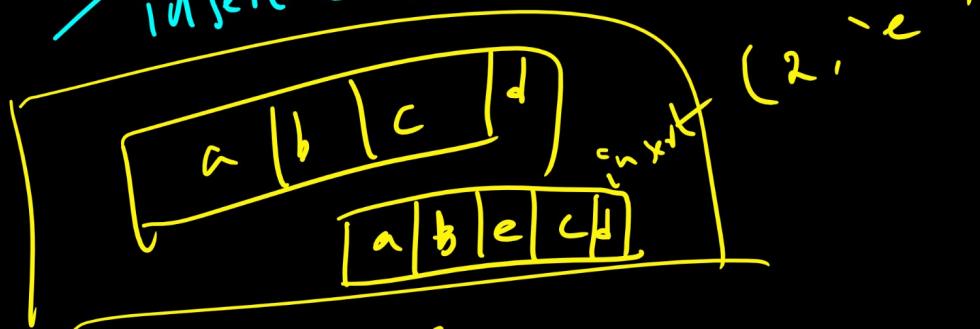
insert (position\_to\_insert, value\_to\_insert)



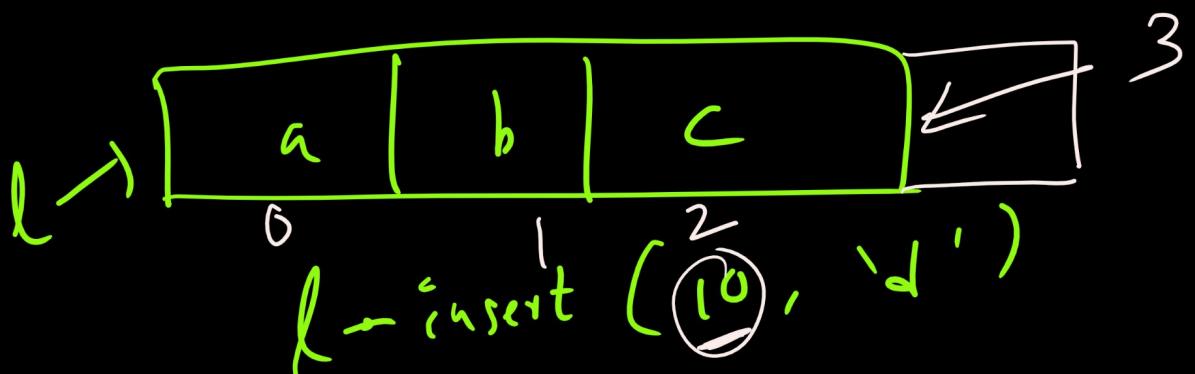
insert (2, z)

a b z c d e f  
0 1 2 3 4 5 6

→ insert(5, y)



1235  
a b z c d e f



l.insert(5, y)

a	b	c	d
$\tau_1$	$\tau_2$	$\tau_1$	$\tau_3$

-3 -2 -1

a	b	c
0	(	2

insert(-2, d)

a d b c

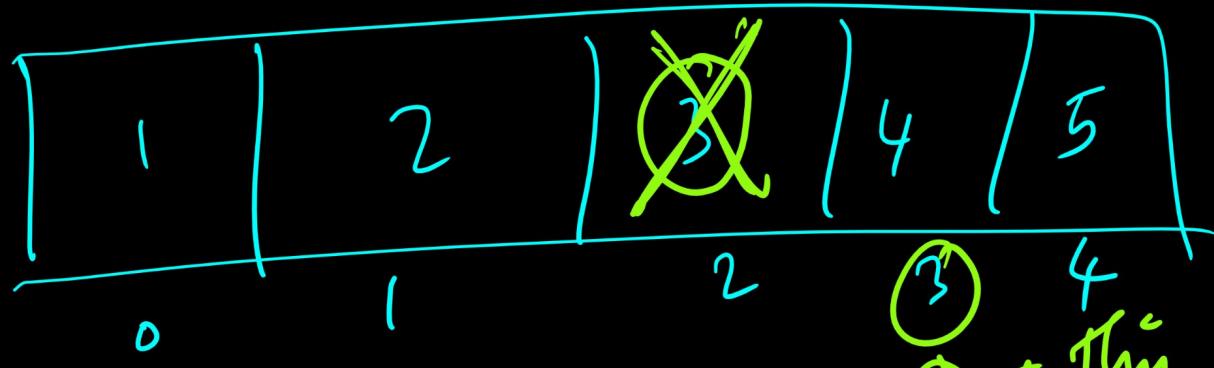
-5 -4 -3 -2 -1

1	2	3	4	5
0	1	2	3	4

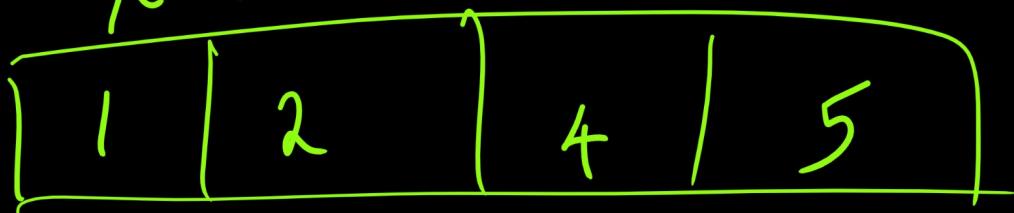
l.insert(2, 'a') l.insert(-2, 'a')

1 2 'a' 3 4 5

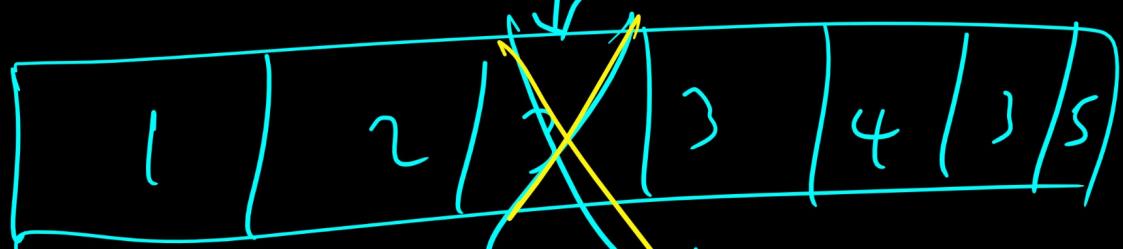
1 2 3 'a' 4 5



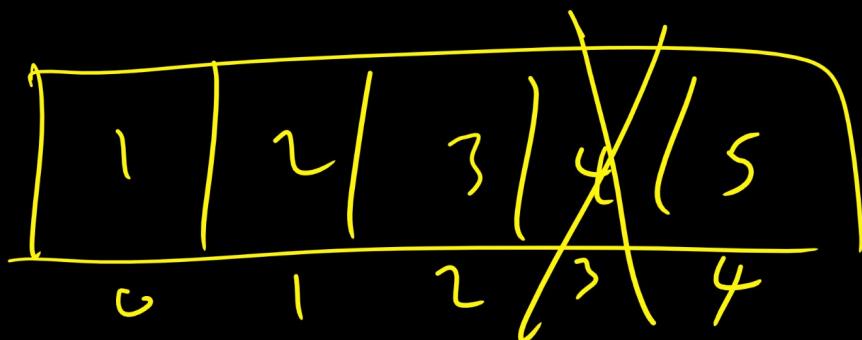
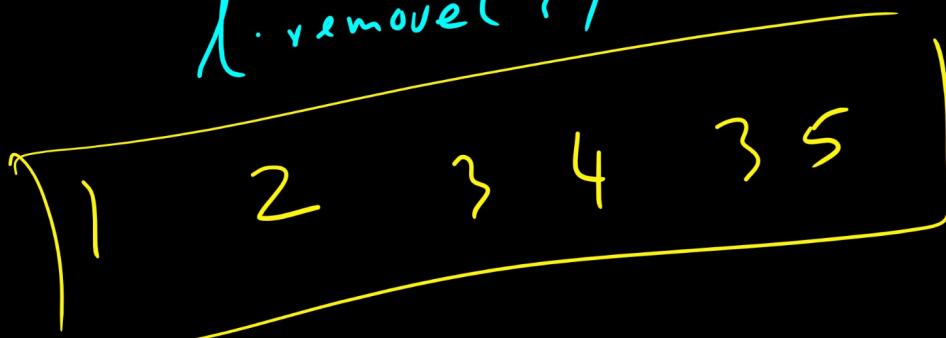
$\lambda.$  remove(3)



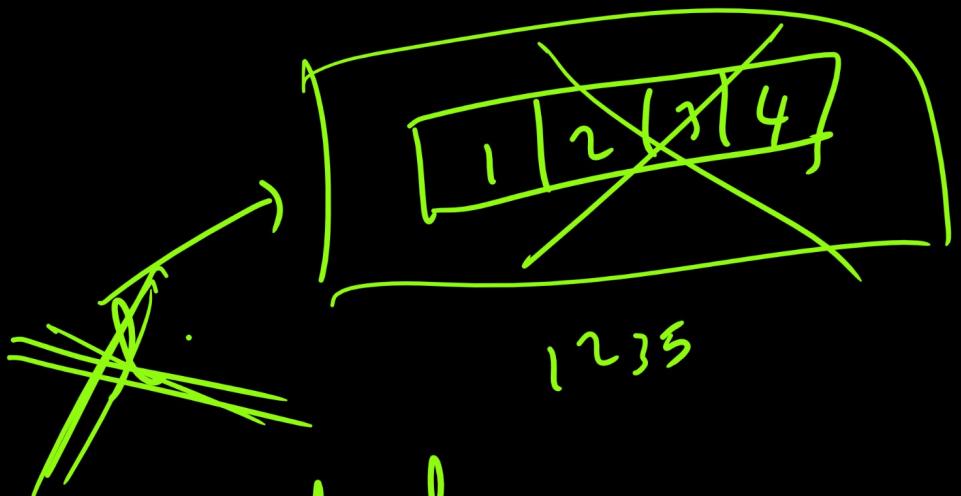
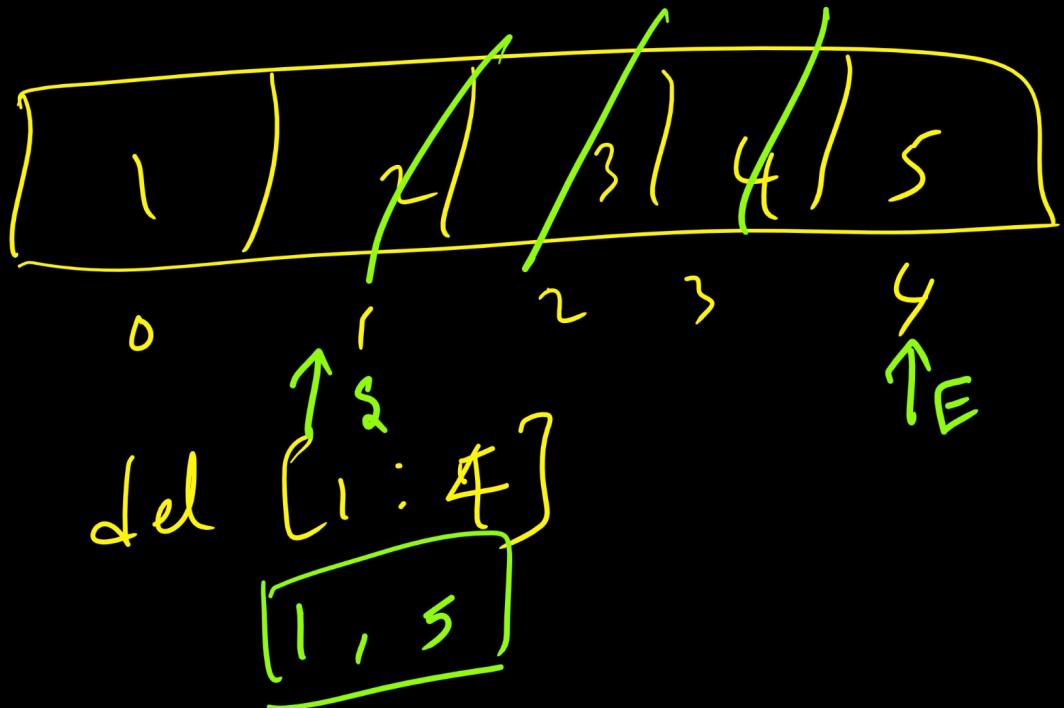
$\lambda.$  remove( $elm$ )  $\hookrightarrow$  removes first occurrence.



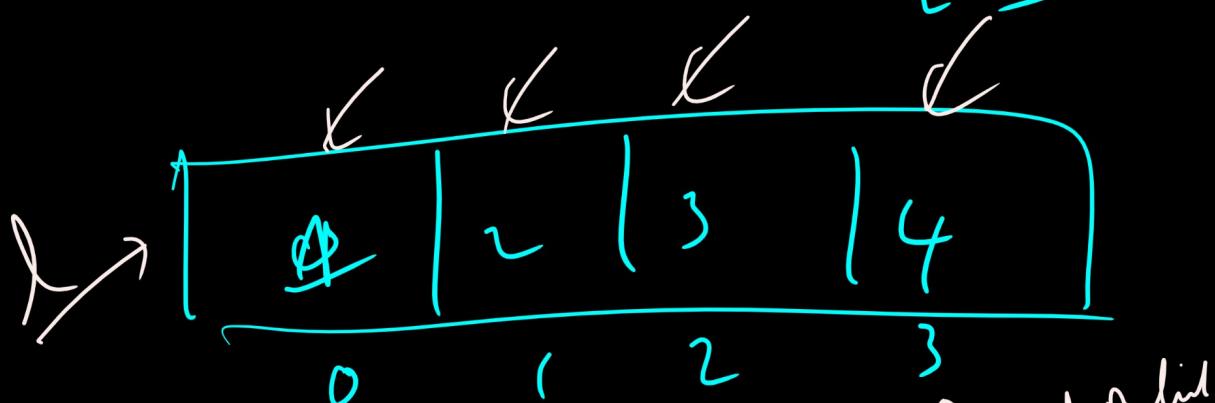
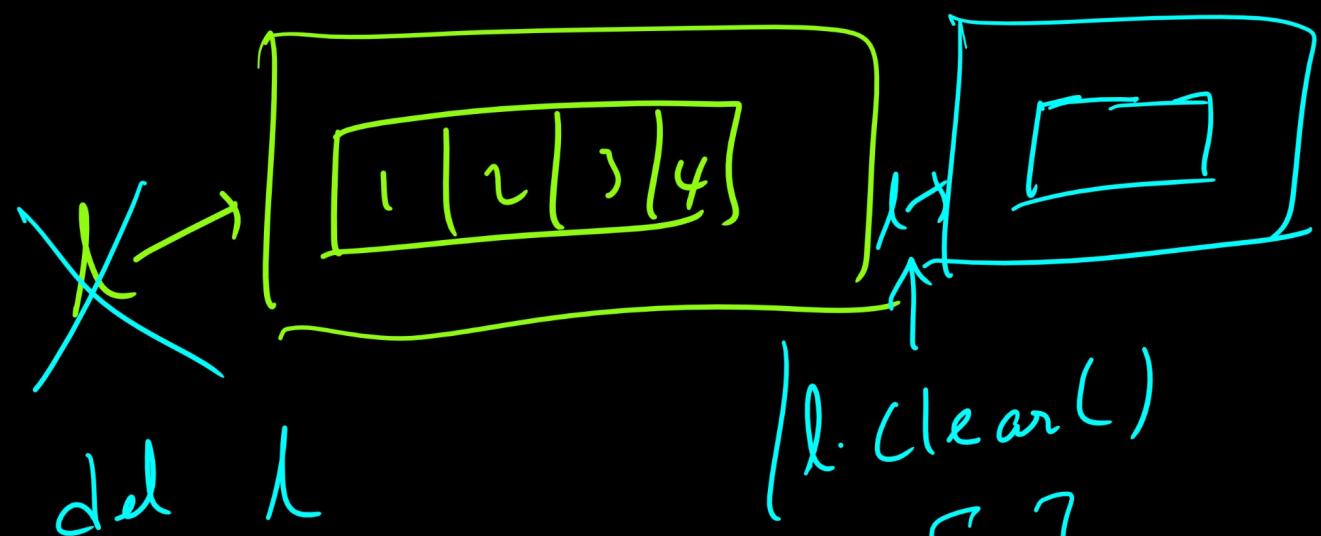
$\lambda.$  remove( $i$ )



$\ell$ .remove(3)  $\rightarrow$  1 2 4 5  
del  $\ell$ (3)  $\rightarrow$  1 2 3 {



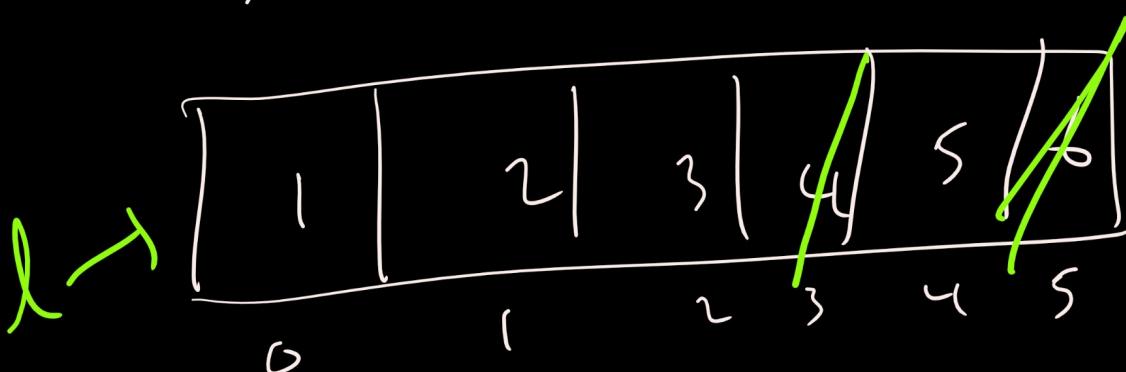
`del  $\ell$   
print( $\ell$ )` - it's not defined.



`del l`

deletes entire list

`del l[:]` equivalent to clear.



`l.pop()`  $\rightarrow$  1, 2, 3, 4, 5

`l.pop(3)`  $\rightarrow$  1, 2, 3, 5, 6

`l.pop(index)`

remove → return None  
pop → remove (element) - first occurrence is removed  
del → returns value  
clear → del l → delete complete list  
del l[:] → delete the range of indices specified.  
→ empty the list, not deleting the list

List Comprehension:

→ Concise way to create a list  
in Python

[ expression for item in iterable ]

[ i for  $\textcircled{i}$  in range(1, 10) ]

[ expression ]      for  $i$  in iterable if [condit.]

expression  
in appended.

Satisfies.

[ $i$ ]      for  $i$  in range(1, 11) if  $i \% 2 == 0$

1       $1 \% 2 == 0$   
2       $2 \% 2 == 0$   
3       $3 \% 2 == 0$   
4

[2, 4]