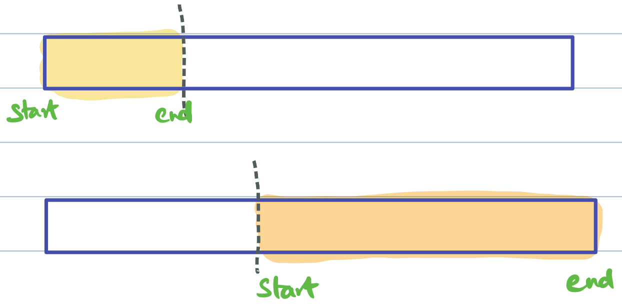
List Slicing :

Today's Highlights :

i) Slicing
ii) Quest^n on slicing
iii) Inbuilt Methods for list
iv) Quest^n
v) map

★ List Slicing :



range(start , end , jump)

$$\begin{array}{ccccc} & 0 & 1 & 2 & 3 \\ \text{i) } l = & [2, & 5, & 7, & 8] \\ & -4 & -3 & -2 & -1 \end{array}$$

len = $\underline{4}$

syntax of list Slicing

=> l [start : end : jump] ∴ These are indexes

i) end is exclusive
ii) By default jump is 1
iii) By default start if not given then 0th index

=> l [0] => 2

=> l [start : end]

=> l [: 4] => l [0], l [1], l [2], l [3], ~~l[4]~~

∴ end is excluded

=> [2, 5, 7, 8]

start end jump => 1

ii) l [1 : 3] => l [1] , l [2], l [~~3~~]

=> l [1] , l [2]

iii) l[:]

⇒ start by default is 0
⇒ end - - - - - len(l)

⇒ Here I will get the whole list

★ **Jump :** Jump will give increment in index values !!

⇒ l[0 : len(l) : 1]

o/p is whole list

⇒ -ve indexing in slicing :

⇒ l[-1] ⇒ l[len(l)-1]

+ve index	0	1	2	3	4	
l ⇒	[2,	5,	7,	8,	10]	len ⇒ 5
-ve index	-5	-4	-3	-2	-1	

⇒ l[0 : -1] ⇒ start = 0
end = -1/4

⇒ l[0], l[1], l[2], l[3]

↱ -(len+1)

⇒ l[4:-6:-1]

start = 4
end ⇒ -6
jump ⇒ -1

⇒ l[4], l[3], l[2], l[1], l[0]

⇒ l[::] ⇒ entire list

⇒ l[::-1] ⇒ entire list but in reverse

By default start = last index
" " end ⇒ including 0th index
-(len(l)+1)

+ve index	0	1	2	3	4
l ⇒	[2,	5,	7,	8,	10]
-ve index	-5	-4	-3	-2	-1

=> l [4 : -6 : -1]

start = 4
end = -6
jump = -1