	0	1	2	3	4	5	6	7	8	9	10	11
L	10	20	30	40	50	60	70	80	90	100	110	120
	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

L[10 : 1 : -2] == [110, 90, 70, 50, 30]

i = 10, j = 1, k = -2

←

	0	1	2	3	4	5	6	7	8	9	10	11
L	10	20	30	40	50	60	70	80	90	100	110	120
	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

L[1 : 11 : -3] == [ ]

i = 1, j = 11, k = -3

	0	1	2	3	4	5	6	7	8	9	10	11
L	10	20	30	40	50	60	70	80	90	100	110	120
	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

L[-3 : -11 : -2] == [100, 80, 60, 40]

i = -3, j = -11, k = -2.

−10

	0	1	2	3	4	5	6	7	8	9	10	11
L	10	20	30	40	50	60	70	80	90	100	110	120
	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

L[-12 : -4 : -4] == [ ]

i = -12, j = -4 k = -4

$k < 0$ and $i > j$ => len(L[i:j:k]) > 0

$k < 0$ and $i \leq j$ => len(L[i:j:k]) = 0

Valid positive index $0 \leq i <$ len(L)

corross ponding -ve index i − len(L)

Valid -ve index : $-\text{len}(L) \leq i \leq -1$

Corrossponding +ve index : $i + \text{len}(L)$

	0	1	2	3	4	5	6	7	8	9	10	11
L	10	20	30	40	50	60	70	80	90	100	110	120
	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

L[0:10:2]   L[1::2]

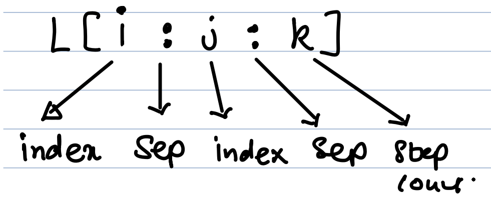
L[::2]

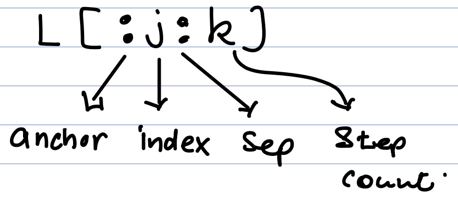
>>> L[::-1]

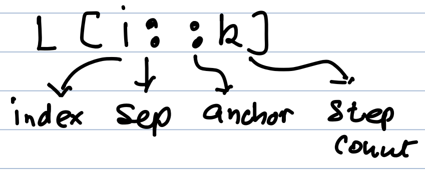
L[::-1]

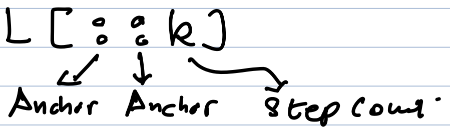
L[0:12:-1] == []

		0	1	2	3	4	5	6	7	8	9	10	11	
L	:	10	20	30	40	50	60	70	80	90	100	110	120	:
		-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	









If a particular : is an anchor then whether it is the first anchor or the last anchor depends on the sign of k.

k > 0	k < 0
$L[\underset{\text{first anchor}}{\boxed{:}}\, j \underset{\text{sep}}{:} k]$	$L[\underset{\text{last anchor}}{\boxed{:}}\, j \underset{\text{sep}}{:} k]$
$L[i \underset{\text{sep}}{:} \underset{\text{last}}{\boxed{:}} k]$	$L[i \underset{\text{sep}}{:} \underset{\text{first}}{\boxed{:}} k]$
$L[\underset{\text{first}}{\boxed{:}} \underset{\text{last}}{\boxed{:}} k]$	$L[\underset{\text{last}}{\boxed{:}} \underset{\substack{\text{first}\\\text{last}}}{\boxed{:}} k]$

L $\boxed{:}$

0	1	2	3	4	5	6	7	8	9	10	11
10	20	30	40	50	60	70	80	90	100	110	120
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

:

$L[\underset{\text{first anchor}}{\boxed{:}}\, 6 : 2]$ [10, 30, 50]

$L[: 6 : -2]$ == $\underline{[120, 100, 80]}$

[ ]

Range is a special case of slice

```
L[i : j]  == L[i : j : 1]
```

1) How to fetch first 'r' elements in the given list?

Answer: L[: r]

2) How to fetch last 'r' elements in the given list?

Answert: L[-r :]

```
L[-1 :]   -> last element
L[-2 :]   -> last two elements
L[-3 :]   -> last three elements

.
.
L[-r :]   -> last r elements
```

```
s = "hello.c"
s[-2 : ] == ".c"    -> s is a C file name
```

```
s = "hello.py"
s[-3 : ] == ".py"   -> s is a Py file name
```

3) How to get all elements from index r to the end

```
   L[r : ]
L[4:] -> [50 -> 120]
```

4) How to get all elements in list with even indices?

Answer: L[::2]

5) How to get all elements in list having odd indices?

Answer: L[1 : : 2]

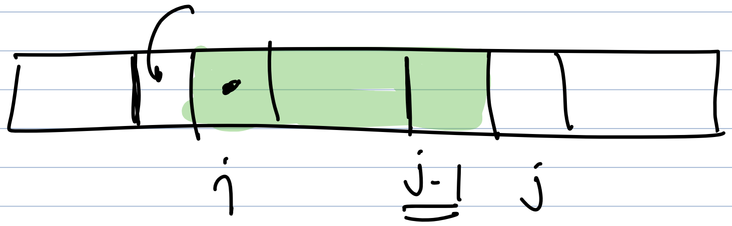
6) How to reverse L[i:j]

```
L[i:j][::-1]

L = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]

L[2 : 6]

L[2:6][::-1]

[60, 50, 40, 30]
```

L[i:j] $\boxed{i<j}$ Given

How will you reverse this range in a single slice syntax?

L[i: j]



$\boxed{\text{L[j-1 : i-1 : -1]}}$

L[i:j][::-1]

L[2:6]

✓ L[5:1:-1]

✓ L[2:6][::-1]