Boolean operations — AND, OR
— Combine two conditions.
NOT → Negate a condition.

T/F T/F.
↑ ↑
C1 AND. C2
OR

OR C1 → B.Tech
C2 → 5 yrs exp. AND.

C1	C2	OR.	AND.
True	False	True	False
False	True	True	False
True	True	True.	True
False	False	False	False

X = 10, Y = 3

X > Y AND. Y > 5 → False
True False

OR. → True

C1	NOT.
True	False.
False	True

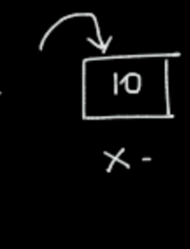
NOT (X > Y)
↳ X ≤ Y

Basic types

Numeric
- integral — int
- non-integral — float

Boolean → bool
True / False

x = 10



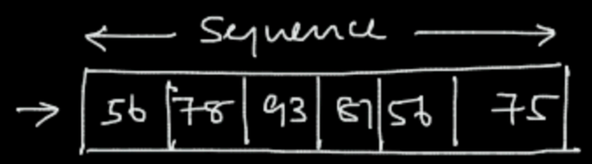
a = True

Collections
- Sequence
  - Mutable → list / [ ]
  - Immutable → tuple, string
    ( ) " "

M1 = 45
M2 = 73
M6 = 57
M24 = 95

M25 → 83
M26 →

Mf2 ⇒ 51



lst = [45, 78, 96, 64, 59]
-5 -4 -3 -2 -1
0 1 2 3 4 → index.

tup1 = (11, 12, 13, 14, 15)
0 1 2 3 4

tup1[3] → 14.

1. Positional Indexing

lst[2] → 96.

lst → Extractor

st1 = "python3.0"

p	y	t	h	o	n	3	.	0
0	1	2	3	4	5	6	7	8

'3'

→ Homogeneous.
→ Characters.

Heterogeneous

lst2 = ['Apple', 10, 10.5, True, [1,2,3], (1,2,3)]

st1[3] → 'h'.

lst[1:4] → 78, 96, 64.

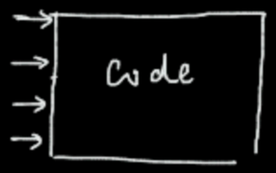
45
↳ grade

2. Sequences are iterable

Loops → Execute code repeatedly

```
lst = ['Apple', 'Orange', 'Grapes']
```

```
for x in lst :
```



→ repeated multiple times (3)

```
for x in lst :
    print(x)
```

→ 3 times


```
lst = [0, 1, 2, 4, 5, [4, 5]]
```

```
extend([4, 5])
```

```
append([4, 5])
```