Datatypes.

- Numeric
  - integral — int.
  - non-integral — float.
- Boolean → bool
  - True / False
- Collections.
  - Sequences
    - Mutable — list. [ ]
    - Immutable — tuple, str
      ( ) " "
  - Mapping — dict. { }
  - Set
    - Mutable — Set
    - Immutable — Frozenset.

Sequence-

-6	-5	-4	-3	-2	-1
15	25	37	54	76	93
0	1	2	3	4	5

→ index.

list:

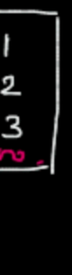
add → append()
extend([])
insert(idx, elem).

remove → pop()
pop(index)
remove(elem)
clear().

X = 10 → 10 (x)

✓ l1 = [1,2,3]

✓ l2 = l1 → one obj.



[1,2,3].

l2.append(100)

1,2,3,100 l1 → 1,2,3

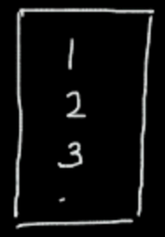
l1 → 1 2 3 ✓

l2 = l1.copy() → 2 obj

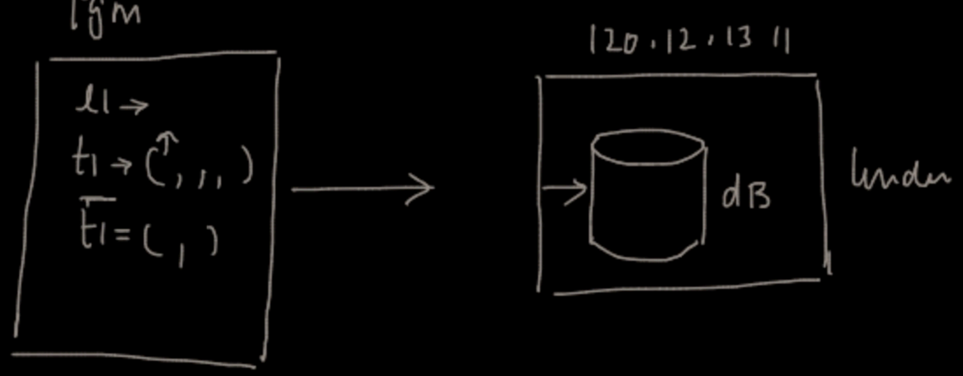
l2.append(100)

l2 → 1 2 3 100

Immutable — Once created, its contents cannot be changed.

$t_1 = (1,2,3) \rightarrow$ 

← Memory cannot be touched



tup() ← fn(,,)

Str → Sequence of characters

s1 = 'hello' s1 → hello

s1.upper()

new_s1 → HELLO → displayed

dictionary → { }

→ Key-Value pairs

s1 = { 'Name' : 'Thomas',
'Age' : 25,
'Course' : 'Analytics' }

Keys ← Immutable
tuple, str, Nr

Values (any dtype)

s1 →

Thomas	25	Analytics
Name	Age	Course

s1['Age'] → 25

s1['Course'] → 'Analytics'

{ 'Tom': 0
'Jon': 0
'Tim': 0 }

list list

try try

l1 = ['Tom', 'Tim', 'Jon', 'Jim']

l2 = [90, 57, 63, 75]

Zip (l1, l2) → Combine

(Tom, 90) →

(Tim, 57) →

(Jim, 63)

(Jim, 75)

dictdata →

Status →

Capital →