24-01-2026

Python - 4

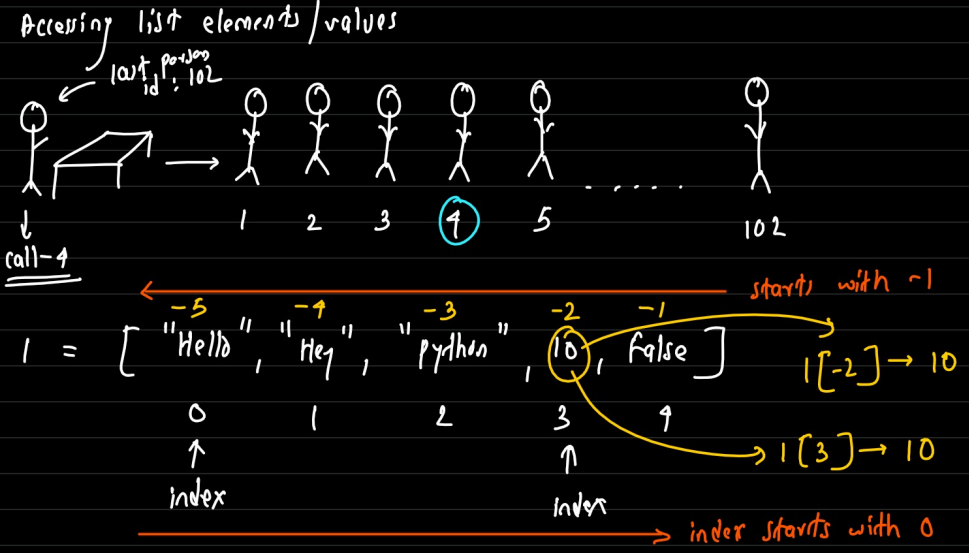
- Discussion on resource
- Non-Primitive Data Types

Non-primitive Data types

l = [0] → l = [0, 2, 2]

l = [ ], l.append (0), append(2), append(2)

List :
1) When we append something to the list, it maintain the order to append.
2) Accepts any data type, duplicates are allowed.
3) Methods :
- append() : add an element/value to the end
- pop() : remove by index



l[3] → index-3 of list l
→ 10

Q. problem statement: i want to know the last value of value.

→ get the first value → index-0
→ get the last value → index-?
index → -1
↓
last element

["Hello", "Hey", "python", 10, False]

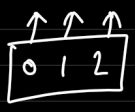
len() ← string → count in characters

len() ← list → count total number of elements/value

↓

total number of elements

1, 2, 3 → (3) index ≠ no. of elements



len("Data")

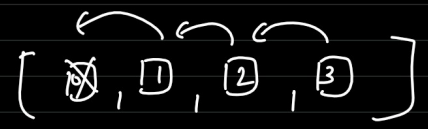
a = "Data" → len(a) → len("Data")

append() → add element

len() → count

index → used to access value of list

pop() → remove elements from list()



→ [□, □, □, □ ... □] → 1000's element

↳ remove "Hey"

pop(index)

append(value)

remove(value) → removes the first occurance

[0, 10, 5, 4]

append(4)

insert() → 4 → index 1

→ [0, 4, 10, 5, 4]

0 1 2 3 4

list :—

append(), index-concept, pop(), insert(), remove()

Dictionary :

Dictionary (The key-value pair) :
1) To store data with labels (keys) instead of positions
2) Key must be (String/Numbers/Tuples) values can be anything.
3) Keys should always be unique, duplicate values are accepted.

person ⟷ Dob, phone, city, pob, occupation

↑ labelled — values

register →

Person	Dob	phone	city	pob	occupation
Monal	—	—	—	—	—
Rohit	—	—	—	—	—
pritam	—	—	—	—	—

register = { } → curly brackets

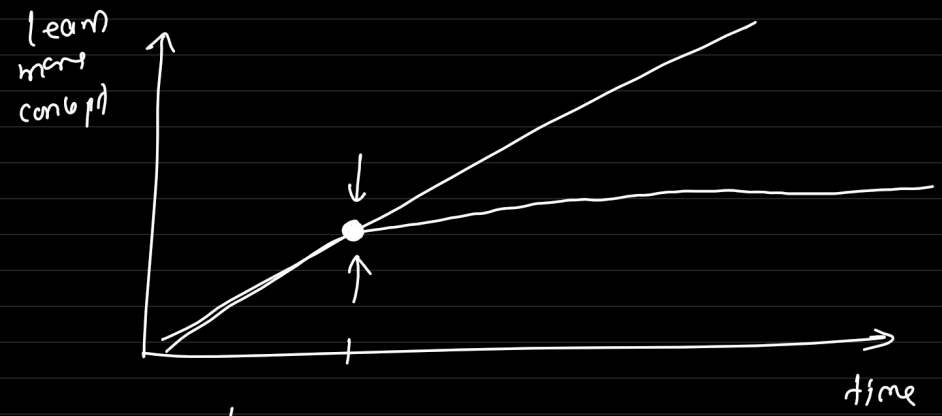
register = { "pritam" : [dob, phone, city, pob, occupation] }

key : value

syntax → { key : value }

r = { "pritam" : [dob, phone, city, pob, occupation],
"rohit" : [dob, phone, city, pob, occupation],
"Ankit" : [dob, phone, city, pob, occupation],
"Amit" : [dob, phone, city, pob, occupation] }

r["Amit"] → [dob, phone, city, pob, occupation]
↑
key



Arun → google

Register :

talk: I want to note down details of visitors.

Name	Address	Phone	ID
Monal	—	—	—
x	—	—	—
y	—	—	—
z	—	—	—

Monal → Name

monal = [Address, ph.no, ID]
⋮
1000)

→ lots of variable

register = [["monal", Address, ph.no, ID], [ ], ["Amit"]]

↳ this solves the problem but not efficient

dictionary = { key : value }

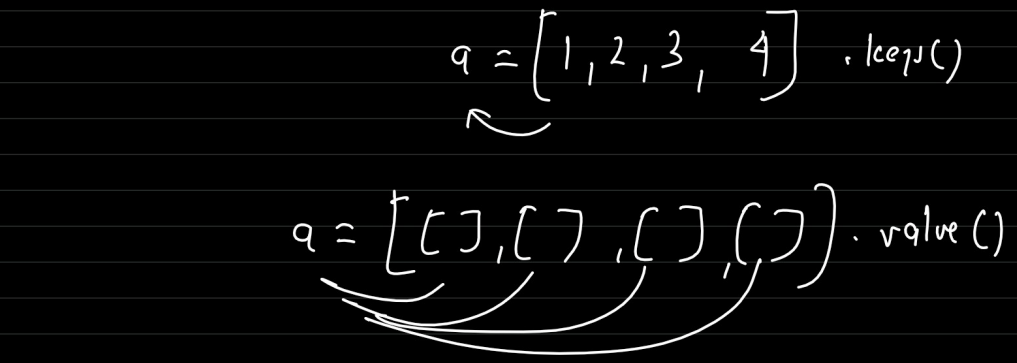
= { "monal" : [Address, Ph.no, ID],
"amit" : [Address, phone] }

dict()
→ .keys() → all the keys in a dict
→ .values() → all the values in a dict
→ .get() → provide key to get value

a = [1, 2, 3, 4]

[101, 102, 103, 104] .keys()

a = [[ ], [ ], [ ], [ ]]



mutable → changes are allowed
immutable → changes are not allowed
list = [] → append, pop → mutable

Tuple (Immutable) :
1) Data that should not change (ID)
2) Prevents accidental cases where data can be overwritten.
3) It accepts anything
4) Limitation : append(), pop(), no re-assignment after creation.

dob = (10, 'March', 1920)

dob[0] = 20

Bangalore = (x.1289, y.8329) → tuple

a = 1,2,3,4 → a = [1,2,3,4]

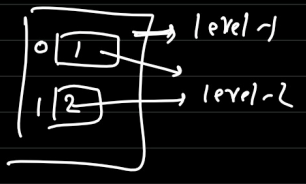
[{}, {}, {}]

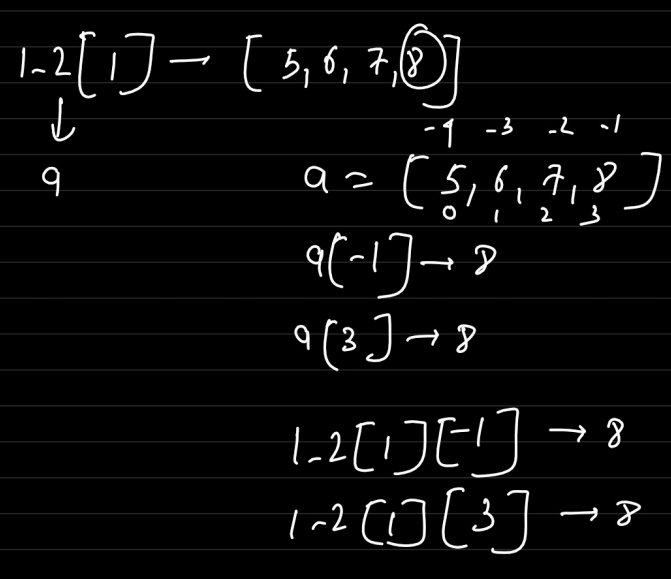
key :

l = [1, 2, 3, 4]
0 1 2 3

l[2] → 3

l_2 = [[1,2,3,4], [5,6,7,8]]
0 1
level-1 level-2





→ register[101] →
→ register.get(101)

```
register_v3 = {
   101 : {
         "Name" : {
            "First_name" : "Monal",
            "Last_name" : "S"
         },
         "DOB" : (10, "feb", 1920),
         "Phone.no" : "+91-1111111111"
      }
}
```

```
{
   "Name" : {
      "First_name" : "Monal",
      "Last_name" : "S"
   },
   "DOB" : (10, "feb", 1920),
   "Phone.no" : "+91-1111111111"
}
```

↓

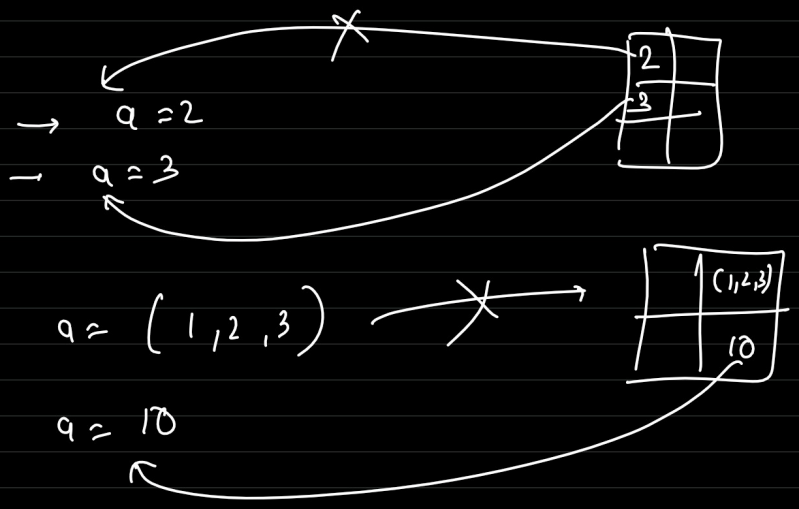
register[101]["DOB"]

↓

(10, "feb", 1920)

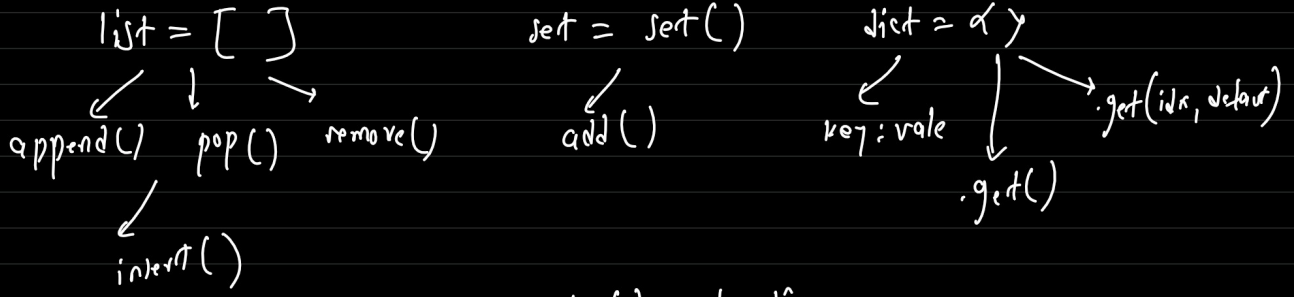
if not able to understand → * watch recording
* practice

if you are able to understand → * practice



a[0] =

SET (The unique values) :
1) To handle uniqueness
2) Removes duplicates automatically
3) NOTE : Unordered



tuple = tuple()
= ( )

len() → function
→ str
→ iterable

sorted → list → o/p → list
→ set