Agenda : 12-04-2025

Advanced Data Structure

List

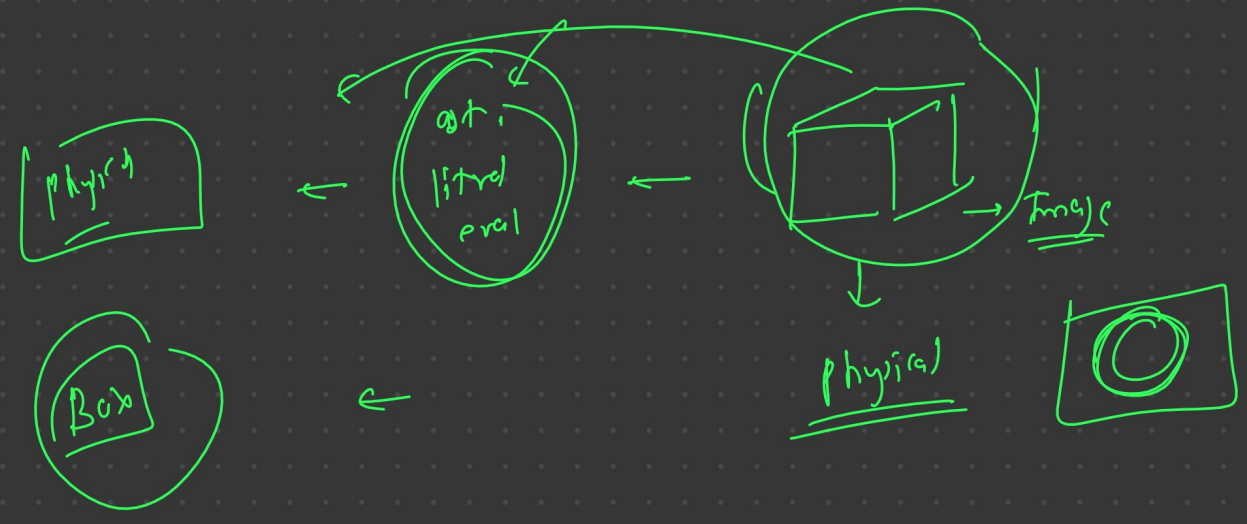
Loop

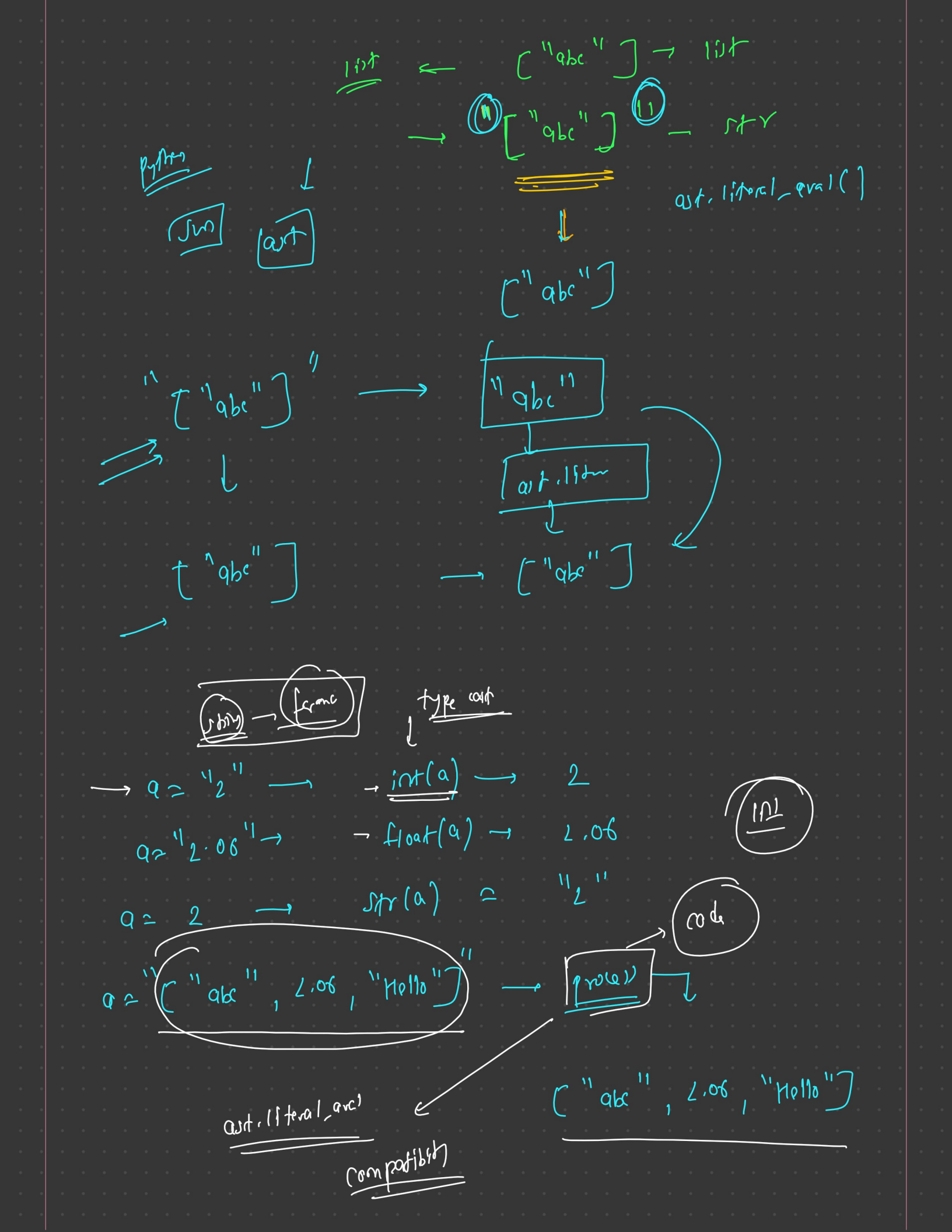
* Variable → a=3

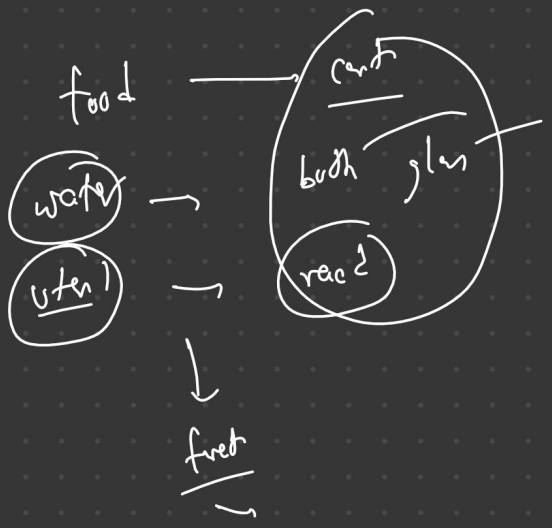
Assignment 2

Problem:
1. Create a single variable called patient_details:
2. Take a list as input & store patient info: name, place, email, age.
3. When asked for a patient's details, use the patient ID to get the name and age only.

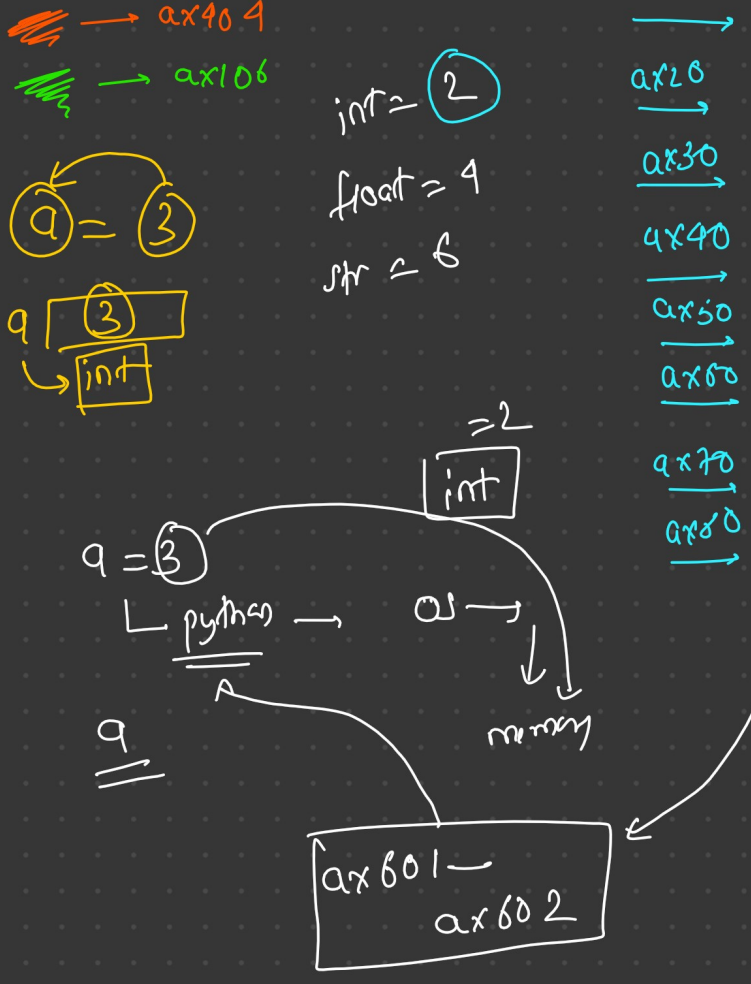
Hint* - This requires a new code / function that we have not learned yet. You are free to use any resources available to you to solve the problem.

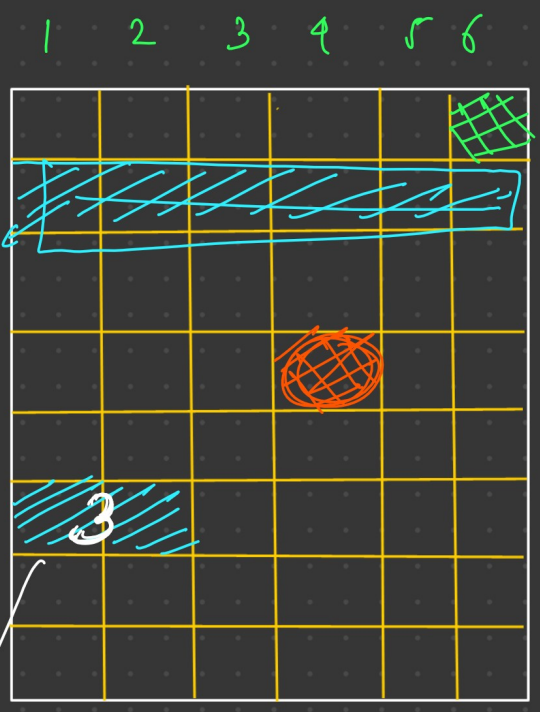



list
["abc"]
list
"abc"
str
Python
Json
ast
ast.literal_eval()
["abc"]
"["abc"]"
"abc"
ast.literal_eval
["abc"]
["abc"]
Json
Python
type cast
a = "2"
int(a)
2
a = "2.06"
float(a)
2.06
int
a = 2
str(a) = "2"
a = "["abc", 2.06, "Hello"]"
eval()
code
["abc", 2.06, "Hello"]
ast.literal_eval
compatible



ast.literal_eval

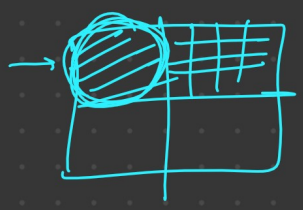




RAM → Not able to retain memory

Harddrive
SSD →

a holds address/memory location of the cell where value is stored.



Data types

Mutable

→ list
Dictionaries
sets

immutable

→ int, float, string, tuples, Bool

int

a → 3

last session → class

→ a = 3

a = a+1

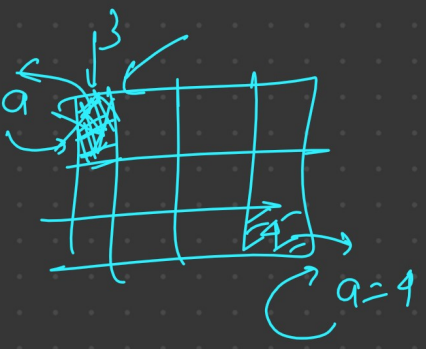
a = 3+1

→ a = 4

a = 3

a → 3

a = a+1

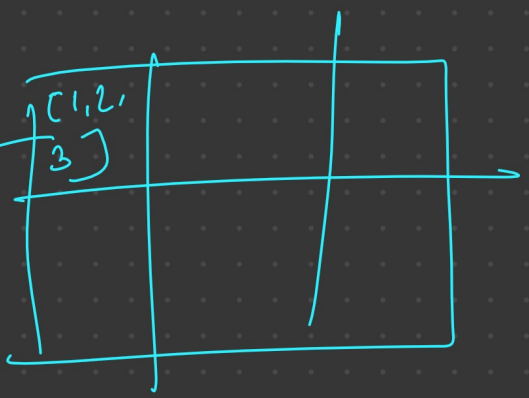


→ Java

C, C++

→ garbage collection

a.append(3)



Chrome → lot of tabs

↓

our system stuck. hang

↓ low

CPU ↑

90 100

Tuples

a = ("abc", 101) → immutable

id:

Name: → change

biometric:

list

→ ① a = (("Mon", "Tue", "Wed" . . .))

→ ② a = (["Mon", "Tue" , , "Hello"])

1 - 29 ← 26

Sets

Unordered collection of unique objects.
↳ sorts the value automatically

does not allow duplicate values.

Syntax

list :

a = [ ] / a = list( )

a.append(value)
a.extent(b) → list

tuple:

a = ( ) / a = tuple( )

sets :

a = sets ( )

a.add(value)
a.update(b) → set

Dictionary

a = { } → key & val pair

a = { 0 : value }

value → dict, list, int, str, tuple, set

0 → int, str, float

Dict :

→ key & value pair,

→ if multiple values : seperate key & value pair by comma.

→ key should be unique, no 2 key should be present with same name, value can be anything.

list → append, extend

set → add, update

dict → keys, values

dict[key] = value

key → add

x duplicate key