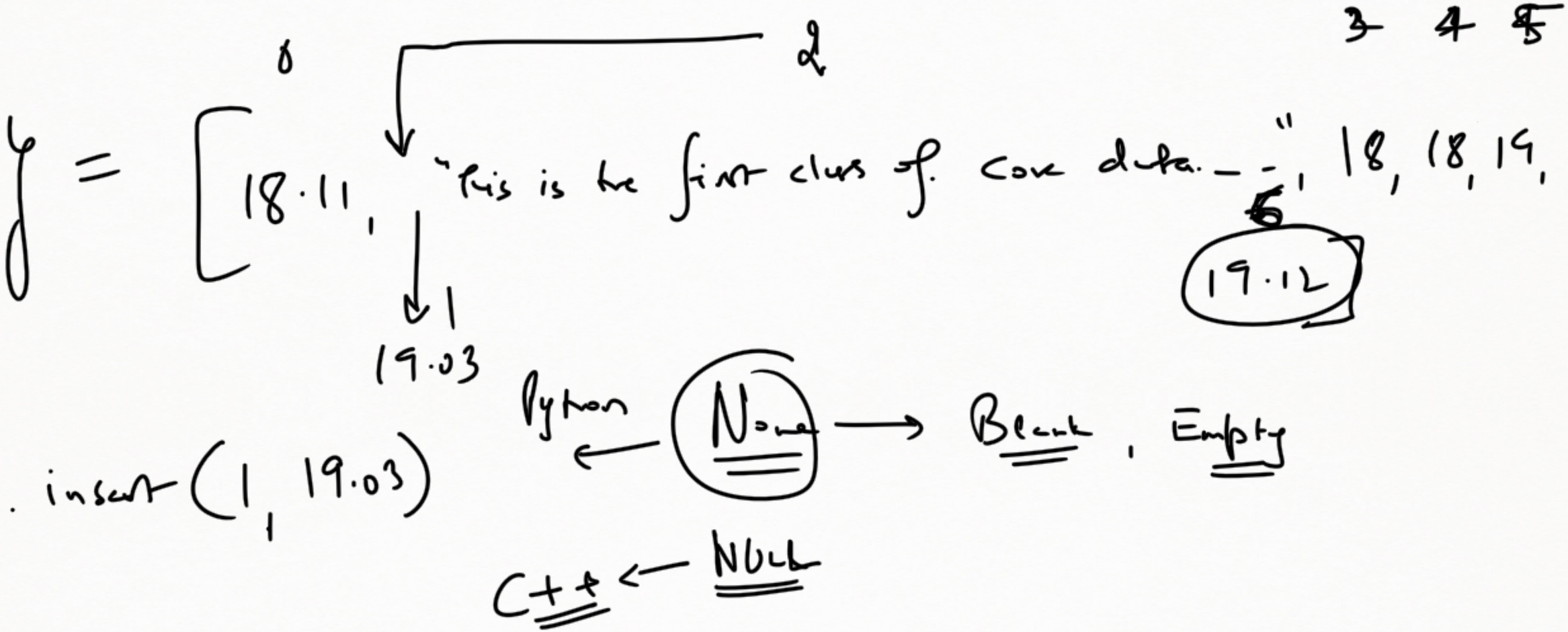
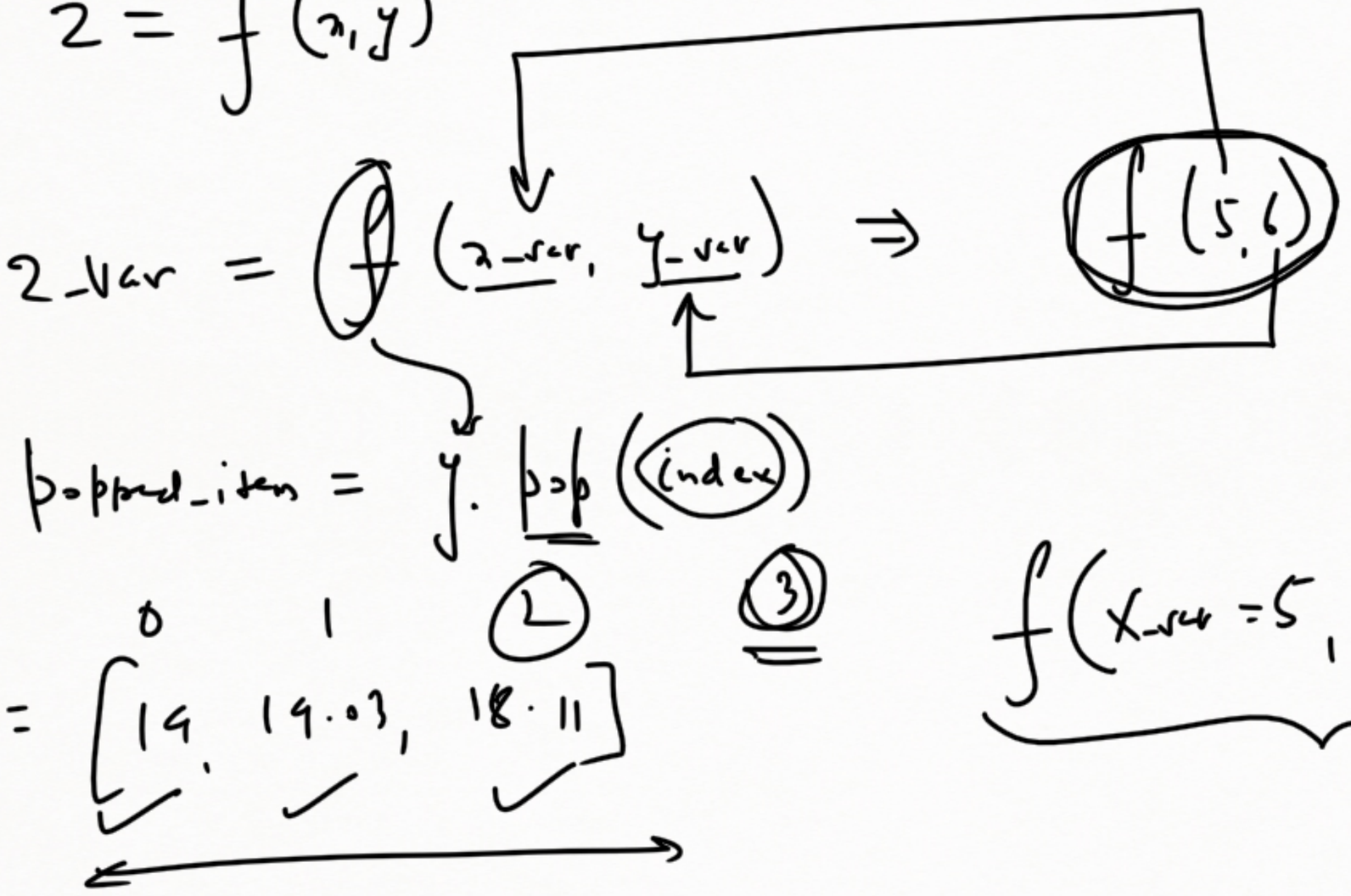

0
2
3 4 5
y = [18.11, "This is the first class of core data.....", 18, 18, 19,
19.03
19.12
y.insert(1, 19.03)
Python
None
Blank, Empty
C++
NULL

$z = f(x, y)$

In Place sorting



DICTIONARY

CORE DATA STRUCTURES —

1. LIST
2. DICTIONARY
3. TUPLE
4. SET

→ What was so wrong with the list?

Advantages and Disadvantages

A way to store data in memory

RAM

i = 50, 30, 4...

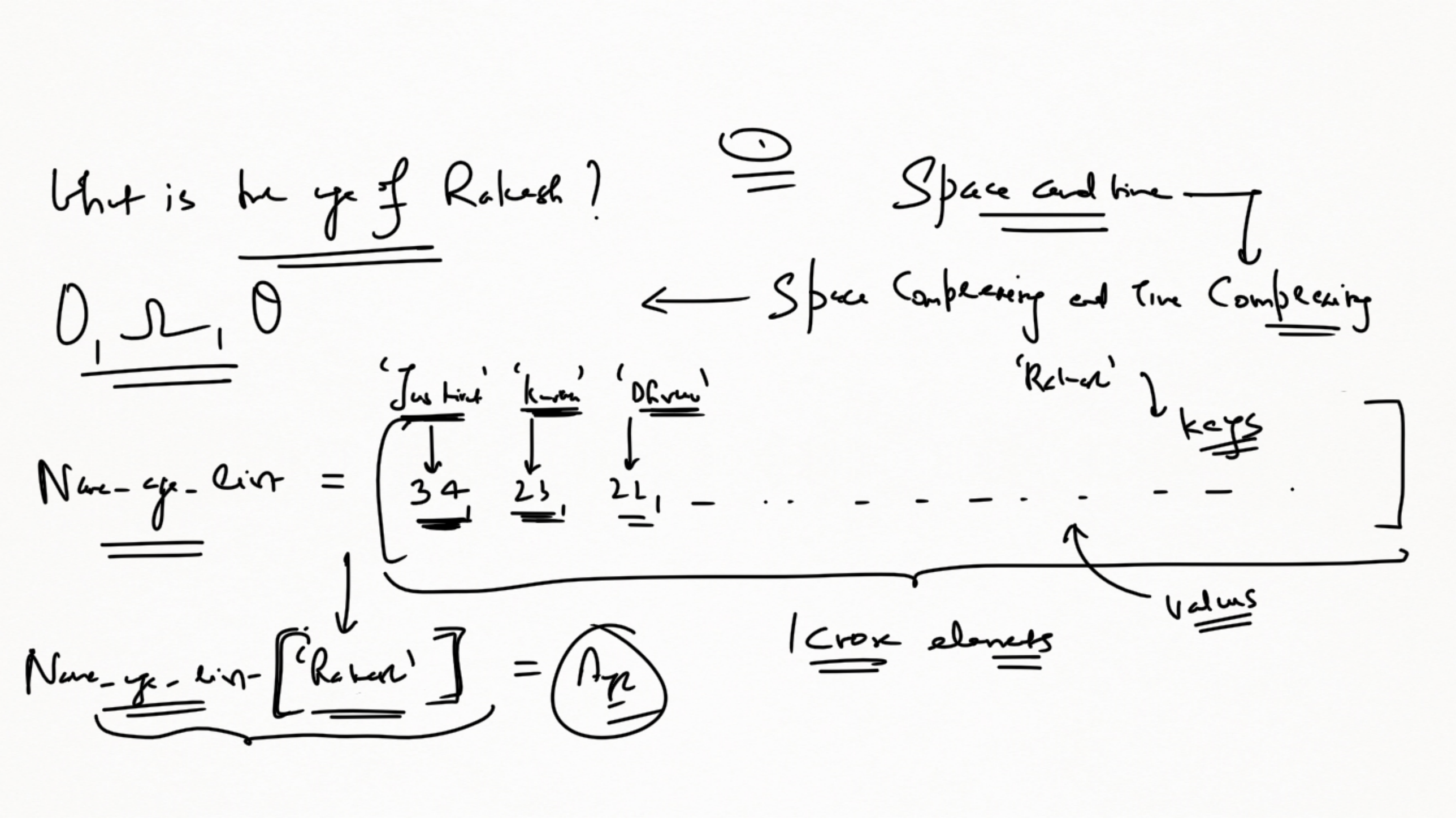
Name = ['Jaslicat', 'Karan', 'Dhruv', ...]

1 row elements

Age[i]

Age =
0	1	2
34,	23,	44, ...

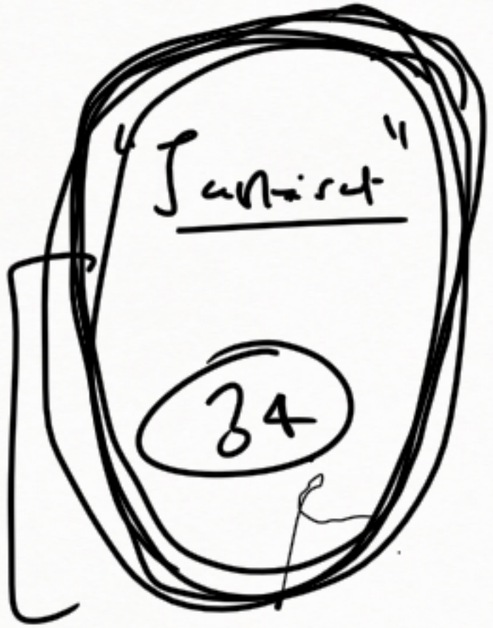
1 row elements


What is the age of Rakesh ?
Space and time
Space Complexity and Time Complexity
O, Ω, Θ
'Jashwin'
'Karan'
'Dhruv'
'Rakesh'
keys
Name_age_list =
34
23
21
values
Name_age_list['Rakesh'] =
Age
Index elements

DICTONARY ⟶ [ ] × Subscript ⟵ [0]

⟶ { } ⌣

SET



"Kannada" "Dhruva" . . .

23 22 ____ .

]

LIST ⟶ . append ([])

. append ({})

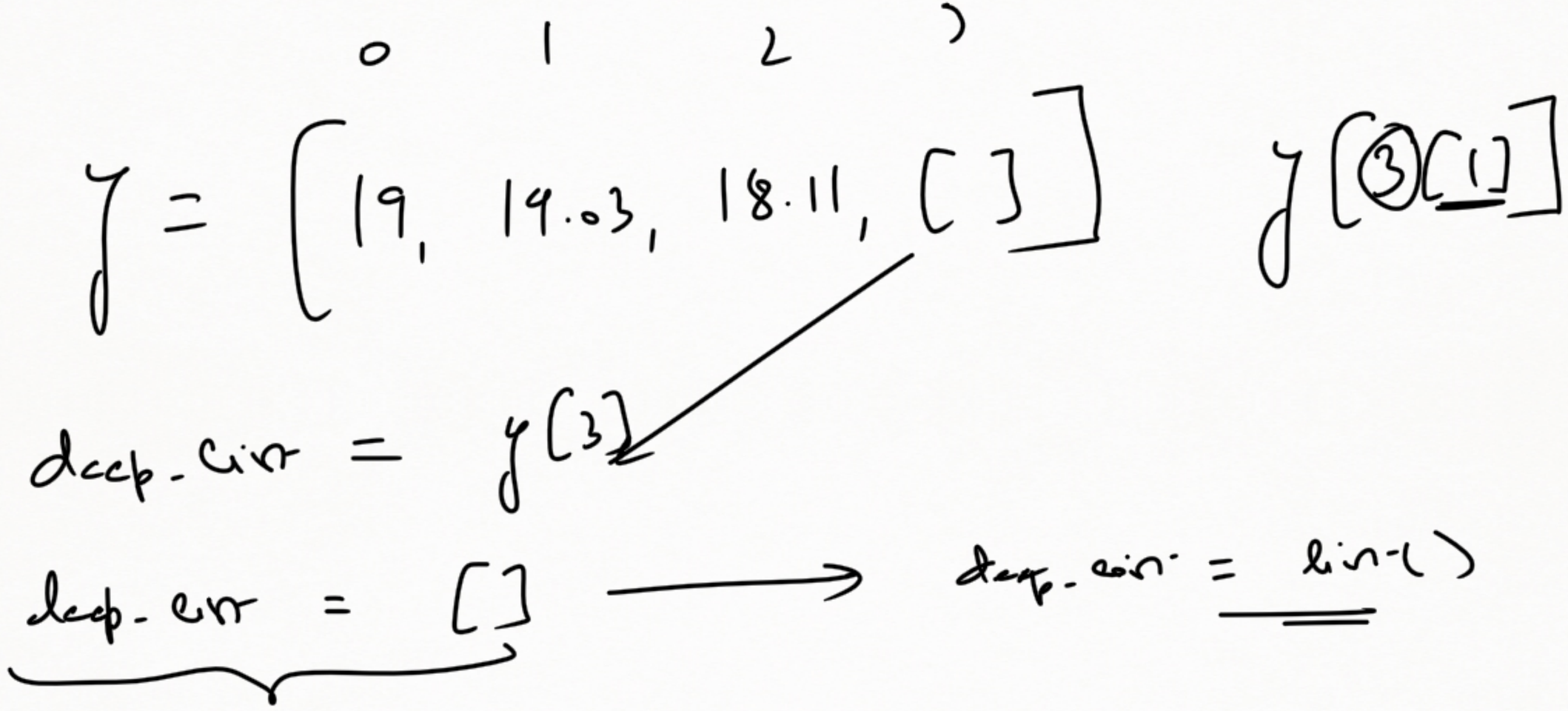
. append (())

. append (set())

name_age_pair.keys()

[(⟷), (), ()]

key value pairs


0 1 2 3
y = [19, 19.03, 18.11, [ ]]
y[3][1]
deep_cir = y[3]
deep_cir = [] ⟶ deep_cir = list()

y[3] ⟶ y[3][1]

inner_lin = y[3]

inner_lin[1]

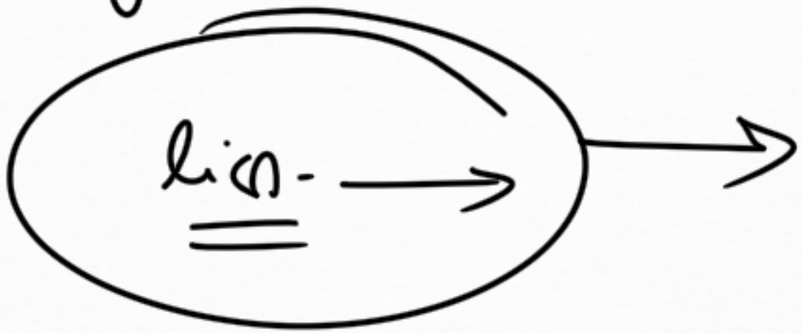
↓

y[3][1]

{NESTED DATA STRUCTURES}

Set in python is actually very similar to Set of Set theory in.

Mathematics.



→ Memory consumption is less in comparison to dictionary, Indexing speed is less

→ CV

→ NLP → Vocabulary

Memory complexity is more but time complexity is less