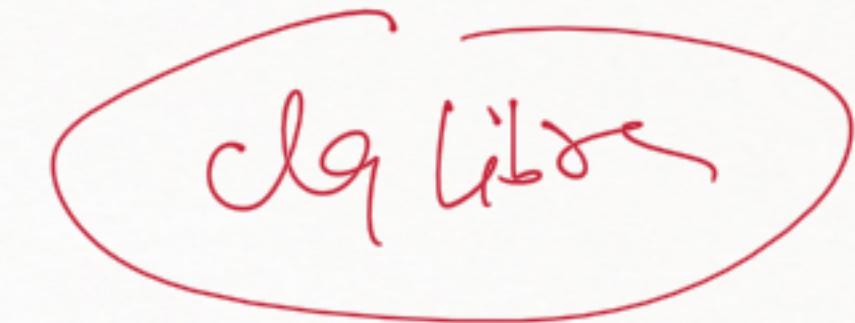
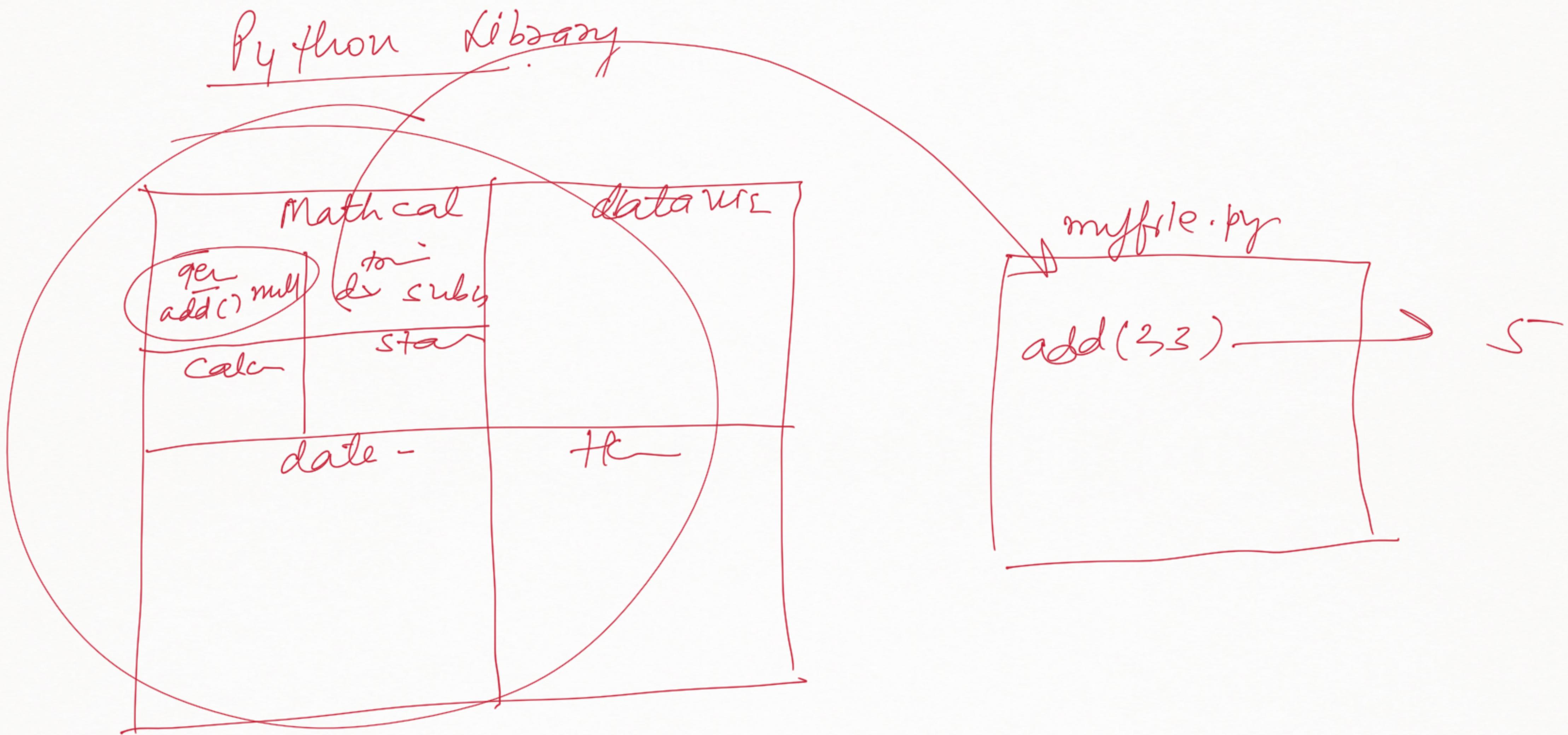


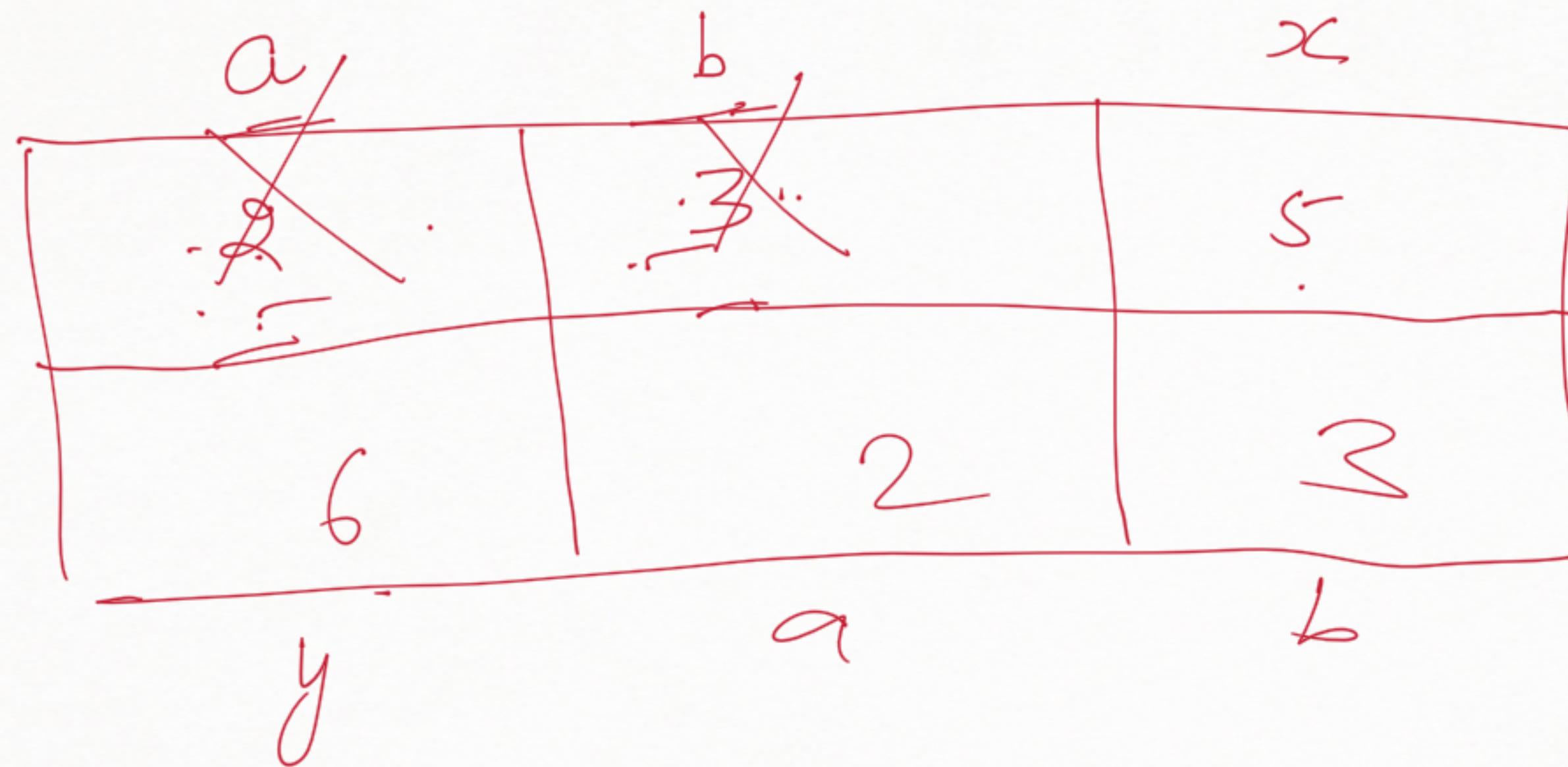
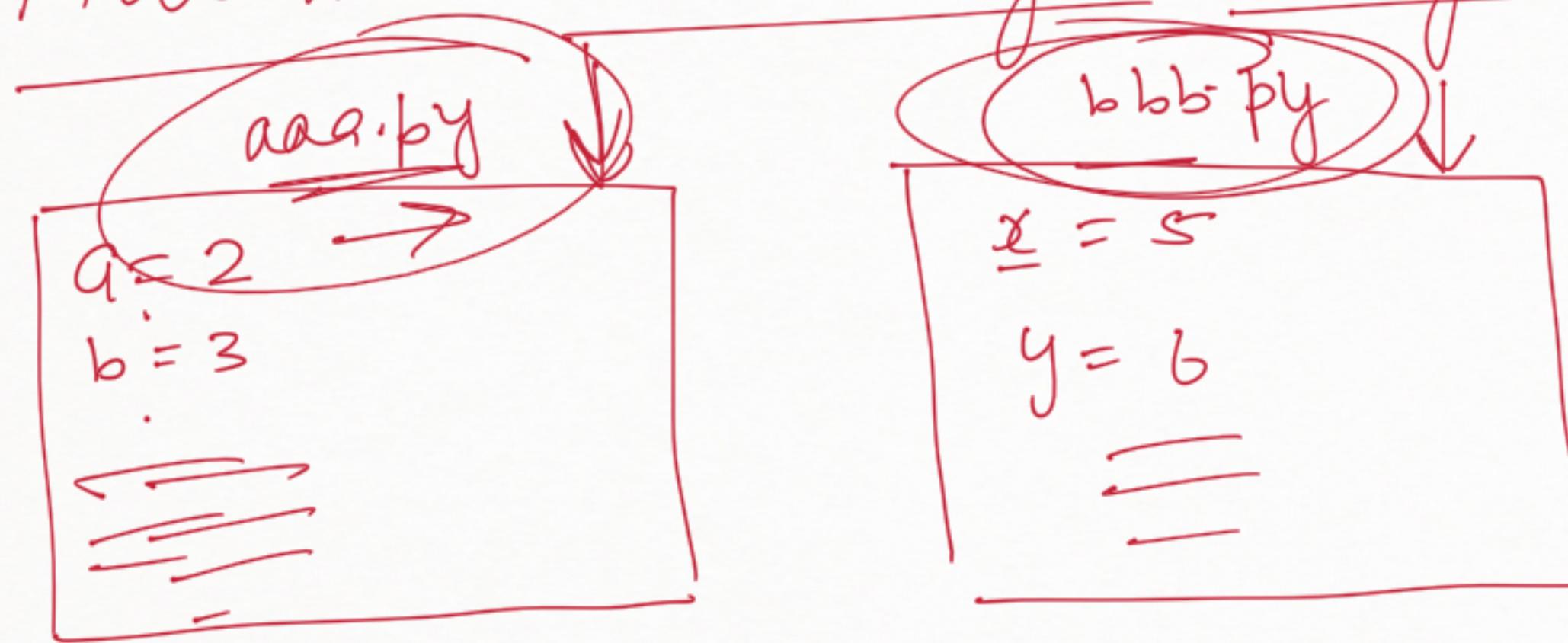
You



- Extra cost → No extra cost
- Extra time → No extraction



## Automatic memory management & Garbage collector:



aaa.py ↴

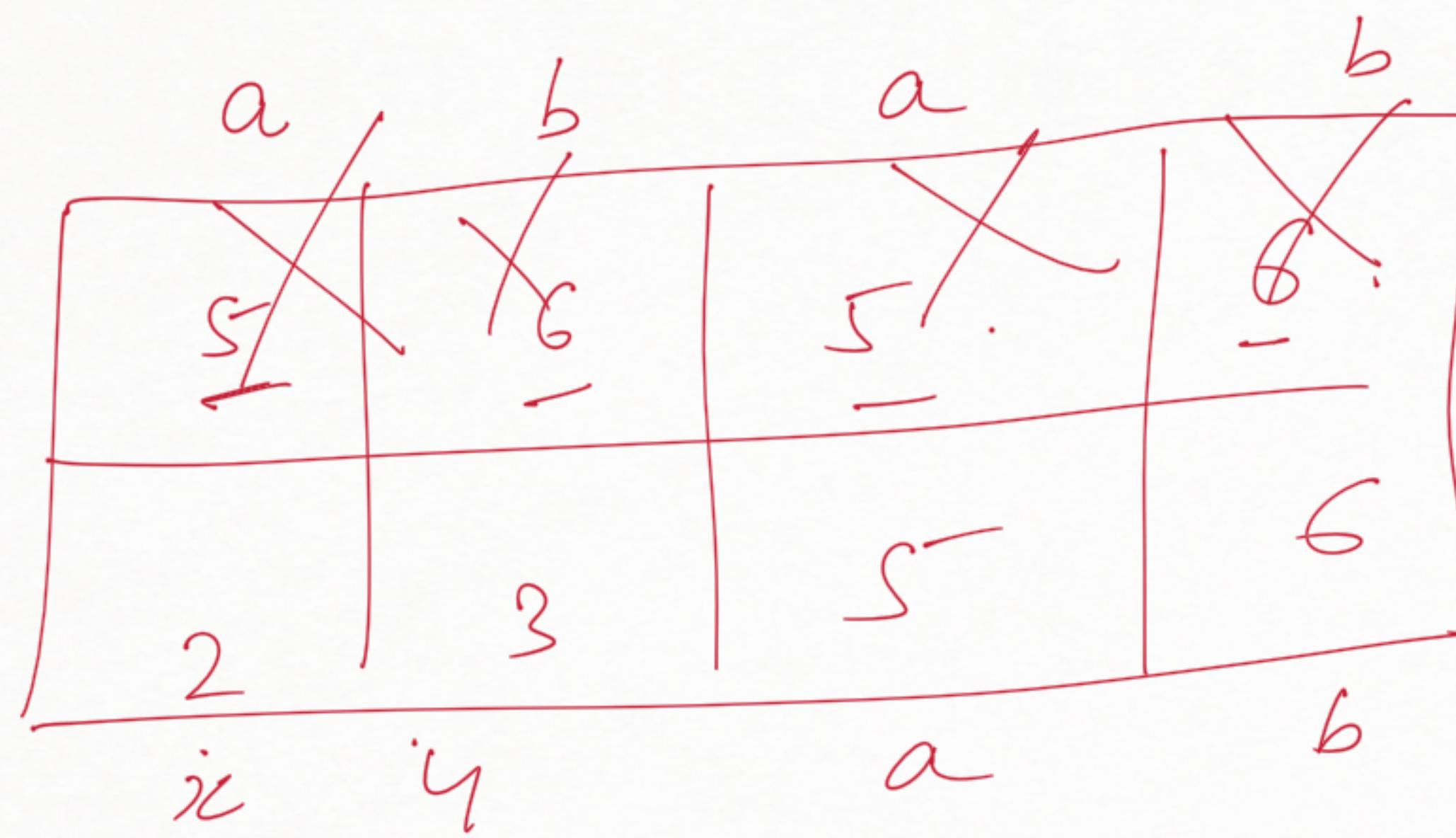
$$a = 5$$

$$b = 6$$

bbb.py ↴

$$x = 2$$

$$y = 3$$

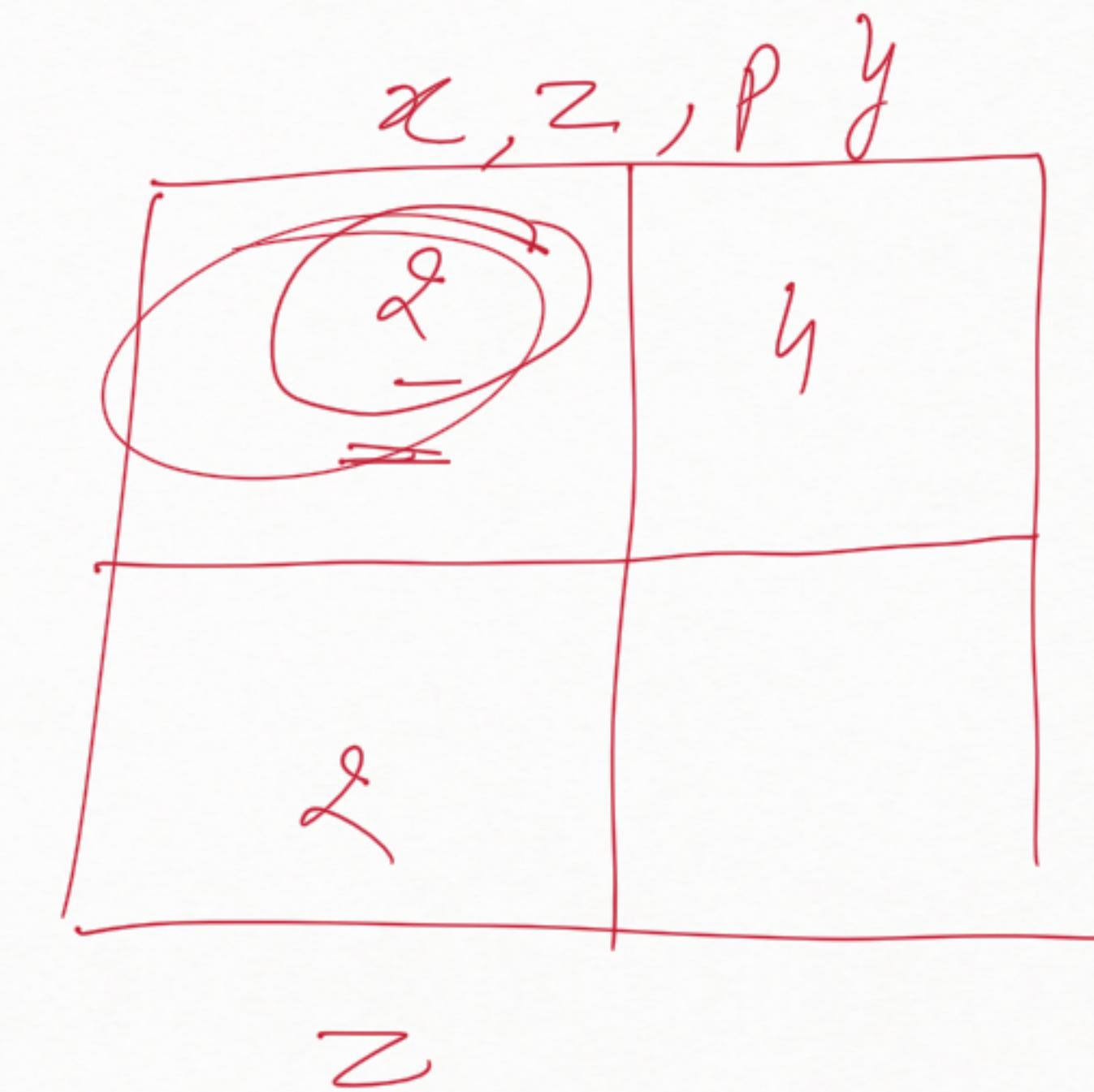


$$x = 2$$

$$y = 4$$

$$z = 2$$

$$p = 2$$



$\rightarrow a = \underline{2}$

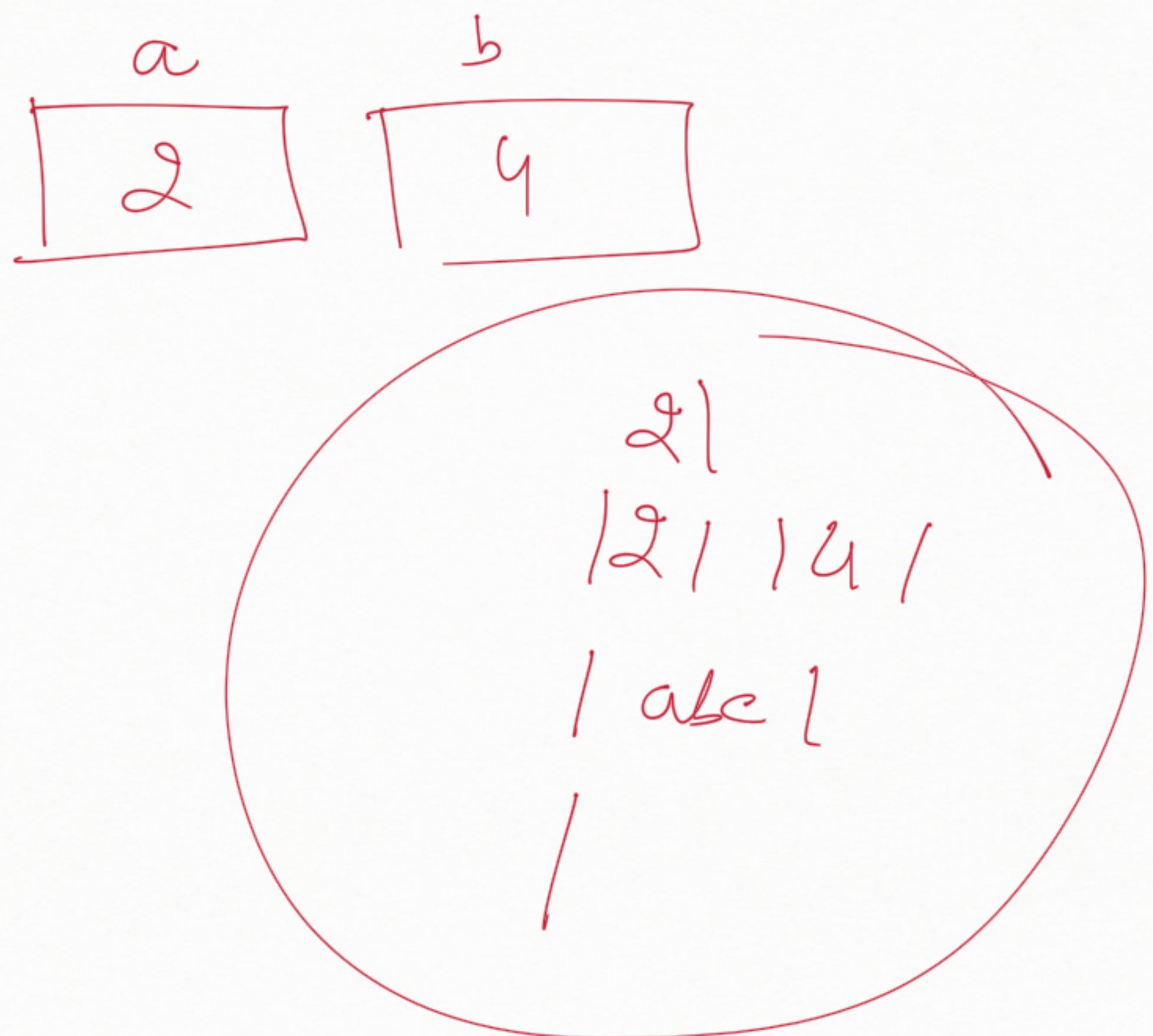
$\rightarrow b = \underline{4}$

$\rightarrow \text{print}(\underline{a})$

$\rightarrow \text{print}(a, \underline{b})$

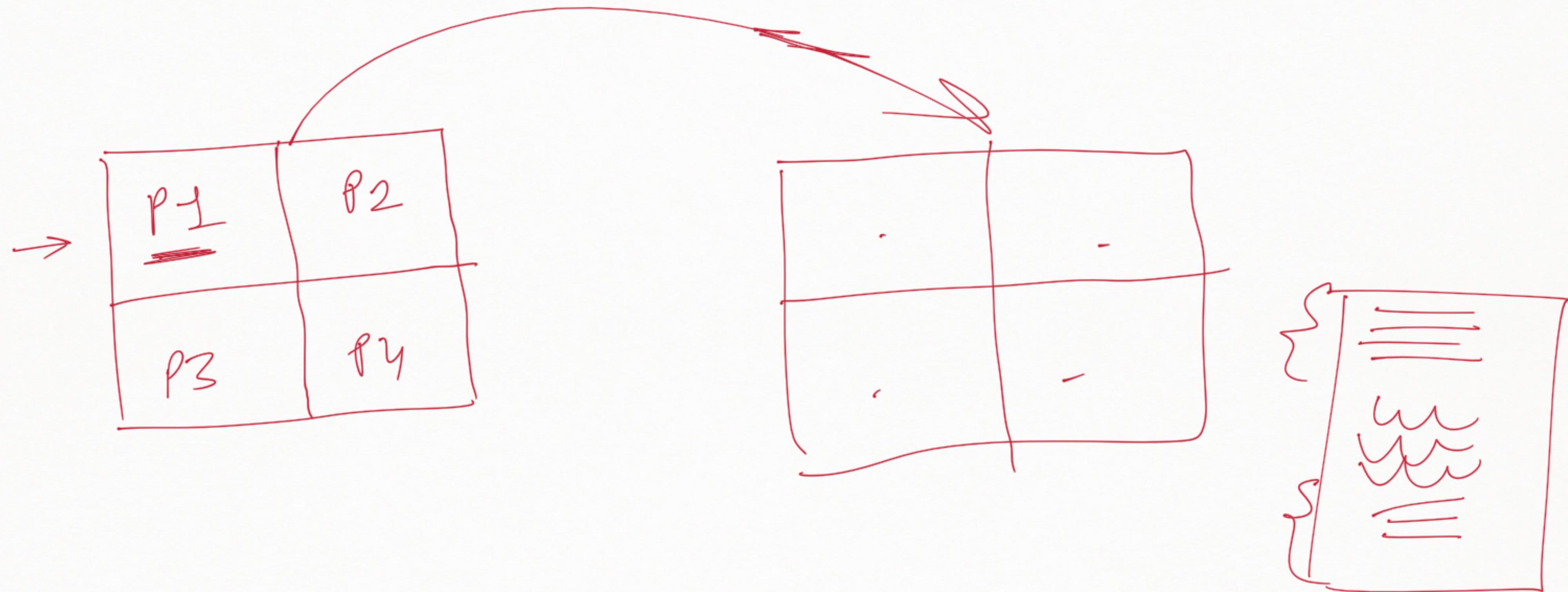
$\rightarrow \text{print}('abc' \underline{c})$

$\rightarrow$

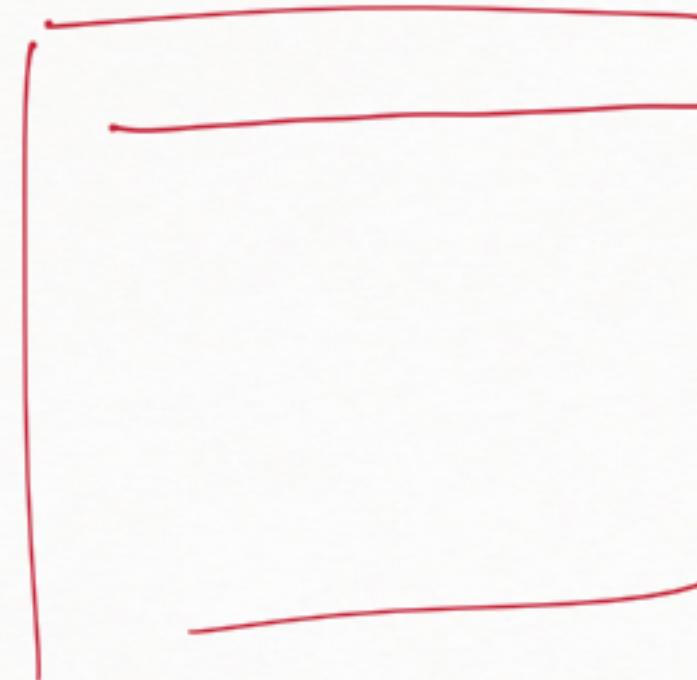


Scf = ''' end = '\u20ac'

Comments in Python:-

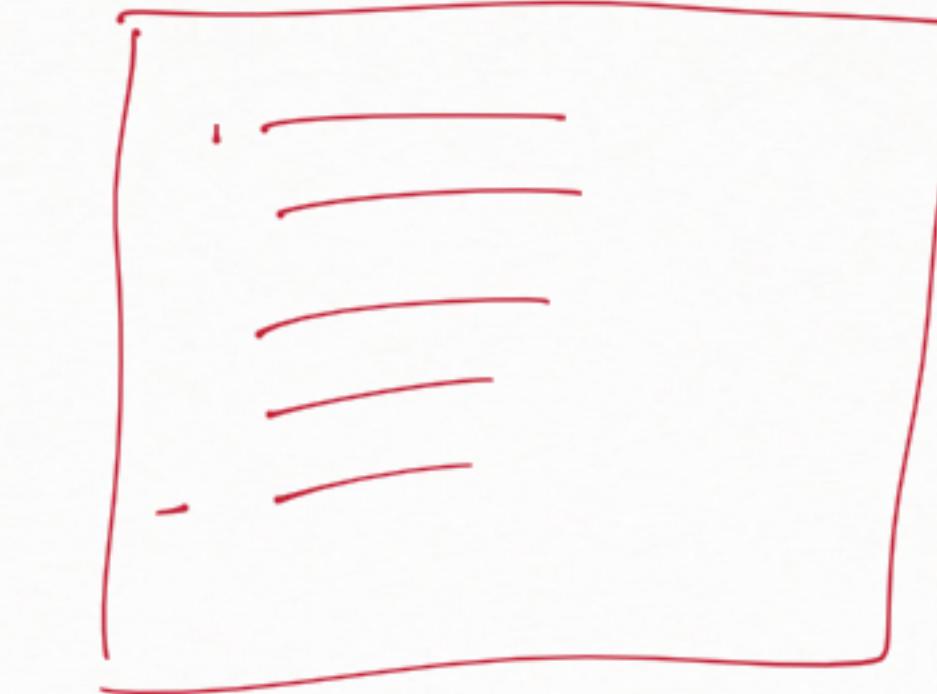


#



Single line

Comment



Multi line comment

. " " " = " "

" " "



" " "

## Data type in Python:

### Number

int  $\rightarrow$  -n to +n

float  $\rightarrow$  -n.n to +n.n

complex  $\rightarrow$   $2+9j$   
 $\rightarrow$   
real imag

$j \rightarrow \sqrt{-1}$ .

String :

'2'  
"2"  
'''2'''  
""2""

collection of charact -

'abc' → strg

'a' → str

|    |    |    |
|----|----|----|
| -3 | -2 | -1 |
| a  | b  | c  |
| 0  | 1  | 2  |

$x : 'abc'$

$x \leftarrow$

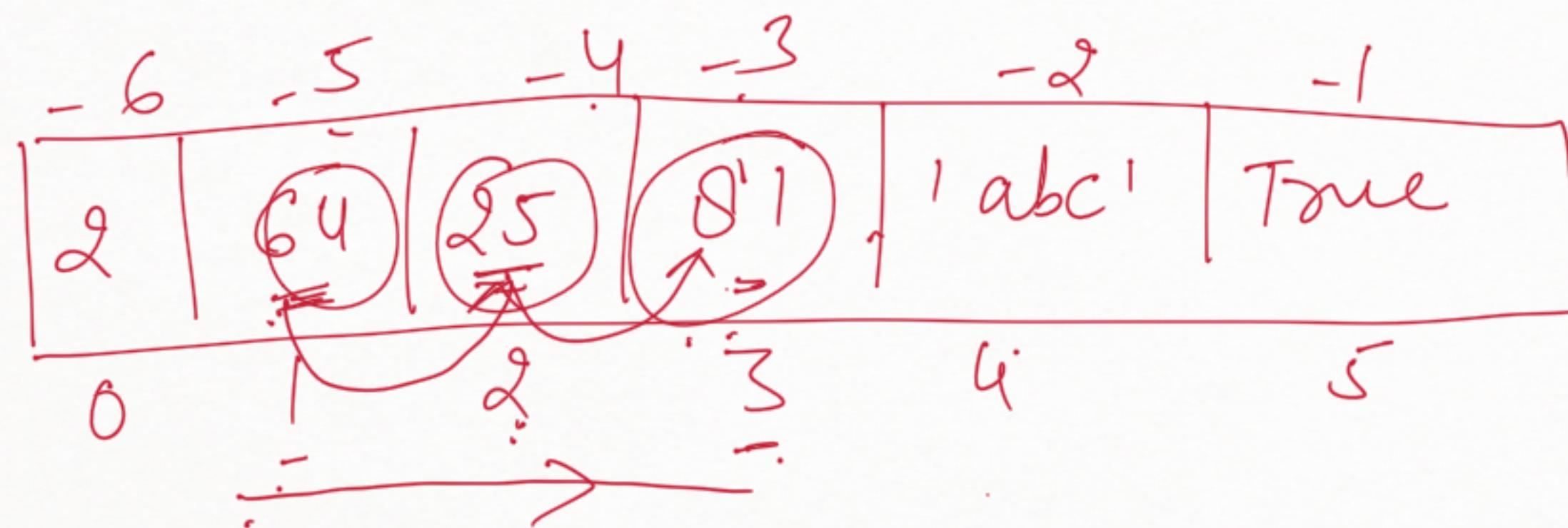
$x[0]$  or  $x[-3]$

→ 'a'

$x[-2]$  or  $x[1] \rightarrow b,$

$x[-1]$  or  $x[2] \rightarrow 'c'$

$a = [2, 64, 25, 81, 'abc', \text{True}]$



$\ln[\underline{\text{start}} : \underline{\text{stop}} : \underline{\text{step}}]$

$a[-5 : -2 : 1]$

$[64, 25, 81]$

$$2 - 1 = (1,)$$

$$-4 + 5 = 1$$

$$3 - 2 = (1,)$$

$a[1 : 4 : 1]$

$a[-5 : 4 : 1]$

$a[1 : -2 : 1]$

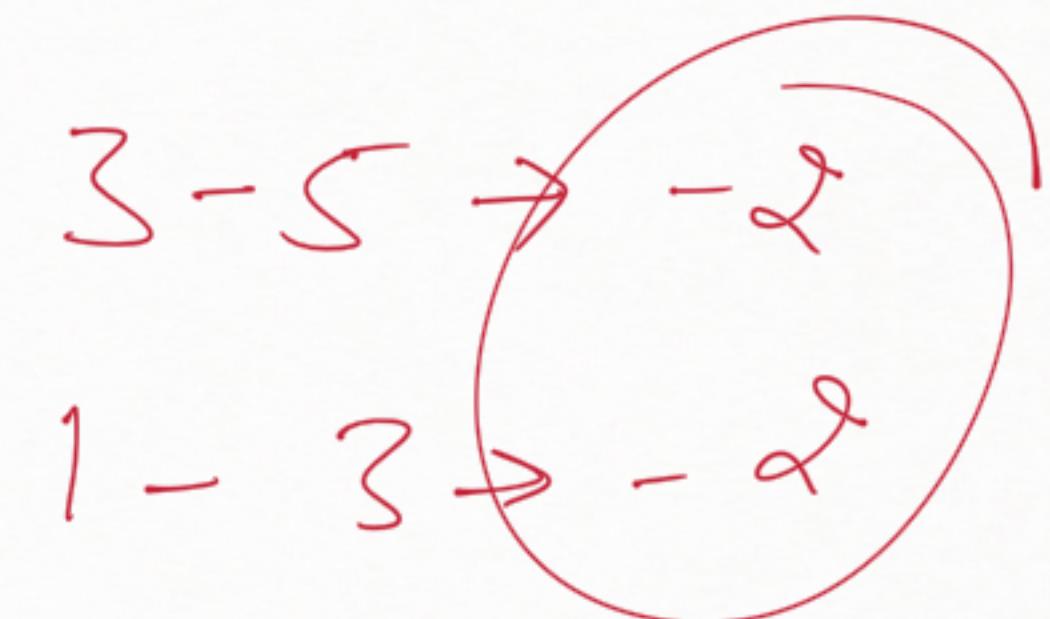
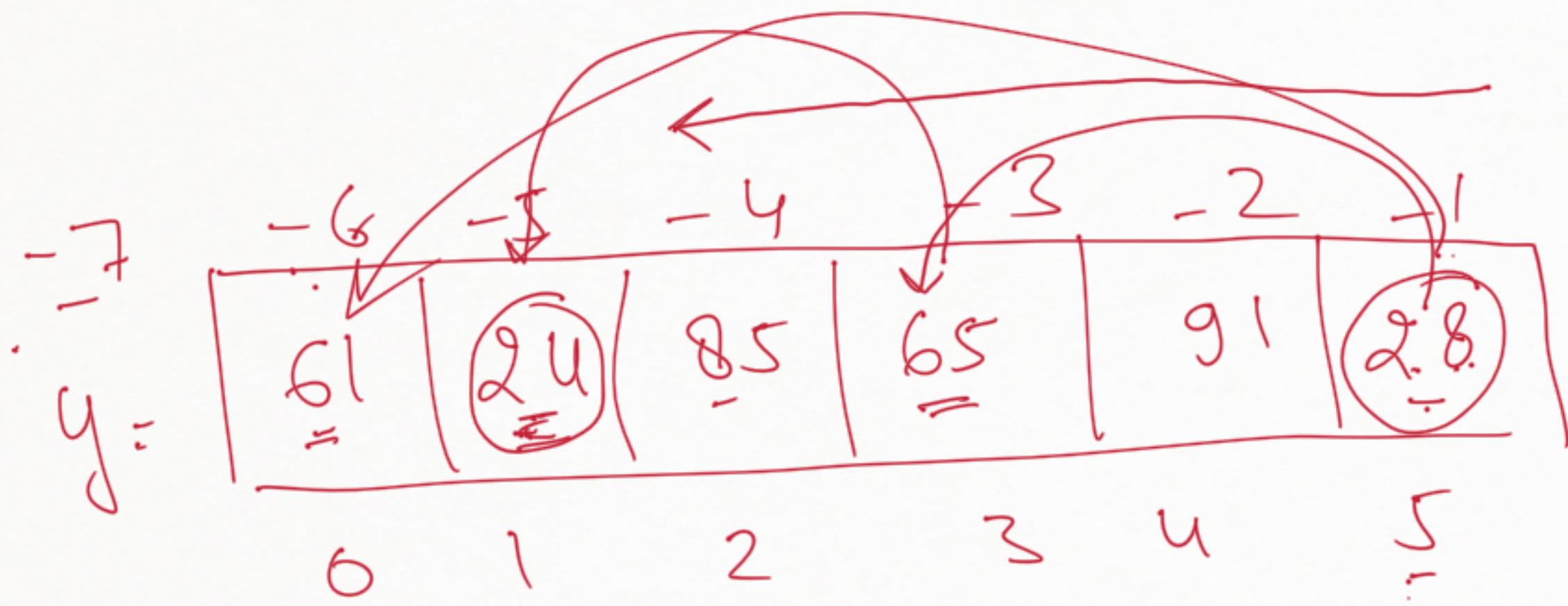
| $-5$   | $4$  | $2$  | $-2$ | $-1$ |
|--------|------|------|------|------|
| $24$   | $81$ | $61$ | $92$ | $85$ |
| $\div$ |      |      |      | $4$  |

$$4 - 0 = \textcircled{4}$$

$[24, 85]$

$x[0:5:4]$

$x[-5:5:4]$



$$[28, 65, 24]$$

$$y[5:-6:-2]$$

$$y[-1:0:-2]$$

$$[28, 61]$$

$$y[5:-7:-5]$$

$$y[-1:-7:-5]$$

$$0 - 5 \rightarrow -5$$

Start

M:N:L

M

M

M

M:

M: : L

M

O

O

O

O

Stop

N-1

N-1

len(a) - 1

"

N-1

N-1

len(a)-1

"

Step

L

+1

+1

L

L

+1

L

+1

$$a = \boxed{2 \mid 8 \mid 45 \mid 61 \mid 29} \quad \begin{matrix} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 1 & 2 & 3 & 4 \end{matrix}$$

$$a[1:5:2] \rightarrow [8, 61]$$

$$a[2:5] \rightarrow$$

list-name [ start : stop : step ]

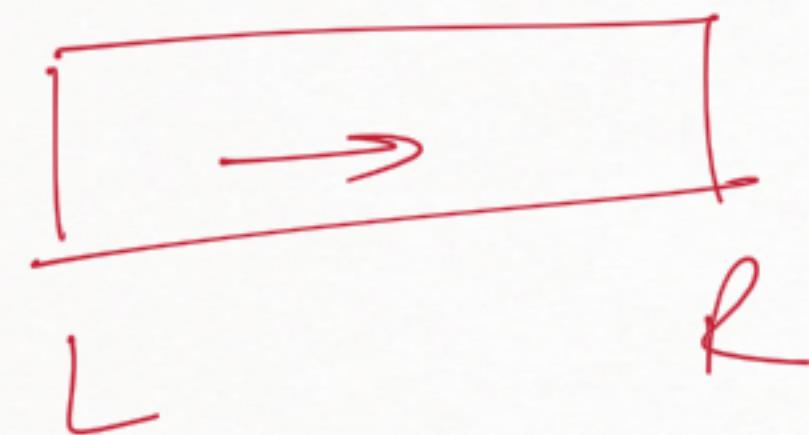
a =

| -6 | -5 | -4 | -3 | -2 | -1 |
|----|----|----|----|----|----|
| 2  | 8  | 9  | 27 | 67 | 25 |
| 0  | 1  | 2  | 3  | 4  | 5  |

[2, 8, 9]

a[0:3:1]

Slicing  $L \rightarrow R$



Start  $\rightarrow 0$

Stop  $\rightarrow -1$

Step  $\rightarrow +1$

Slicing  $R \rightarrow L$



start  $\rightarrow -1$

stop  $\rightarrow 0$

step  $\rightarrow +1$

$$a = \left[ \begin{array}{cccccc|c} 2 & -6 & -5 & -4 & -3 & -2 & -1 \\ 2 & | & 9 & 61 & 28 & 49 & 62 \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \right] \rightarrow$$

$$a[1:6:3] \rightarrow [9, 49]$$

$$a[1:6] \rightarrow [9, 61, 28, 49, 62] \\ \rightarrow$$

$$a[1:] \rightarrow [9, 61, 28, 49, 62, 40]$$

if step → ±  
L to R

—  
R to L

| $a =$   |
|---|
| $\begin{array}{ccccccc} -7 & -6 & -4 & -3 & -2 & -1 \\ \hline 6 & 81 & -6 & 49 & 82 & 83 & 45 \\ \hline 10 & 1 & 2 & 3 & 4 & 5 & 6 \end{array}$ |
| $a[::] \rightarrow [4, 81, -6, 49, 82, 83, 45]$   |
| $a[6:-2:-1] \Rightarrow [45, 83, 82, 49, -6]$   |
| $a[5:] \Rightarrow []$  |
| $a[5:] \rightarrow [83, 45]$  |

$89$

$a[:, :-1] \rightarrow [45, 83, 82, 49, -6, 81, 4]$

$a[2:89:-2] \rightarrow [-26, 82, 45]$

$a[-24:-2] \rightarrow [45, 82, -26, 4]$

## Adding elements

append() → add one element at last of the list

insert() → add one element at your desired location

extend() → add more than one element at last of the list  
,

## Removing elements

pop() → remove one element by using index from list

remove() → remove one element ————— value from list

del → remove one or more than one

clear() → remove all —————

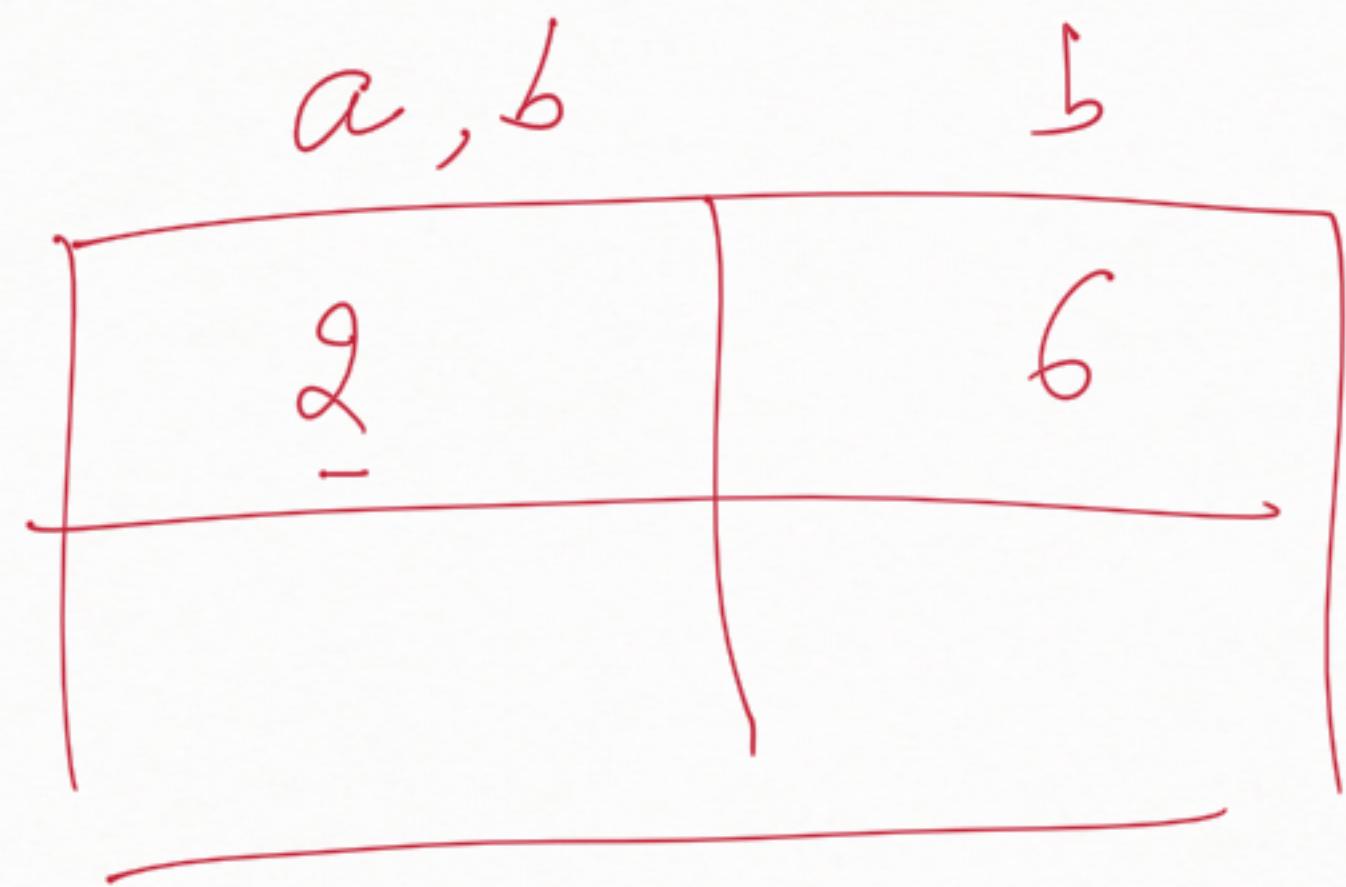
## other methods

- {
  - index() → accoring index value
  - count() → occur of any el
  - reverse() → give the element
  - sort() → sorting into ascend or descend <sup>ord</sup>
  - copy()

$$a = 2$$

$$b = 6$$

$$\begin{matrix} c \\ - \end{matrix} = 2$$



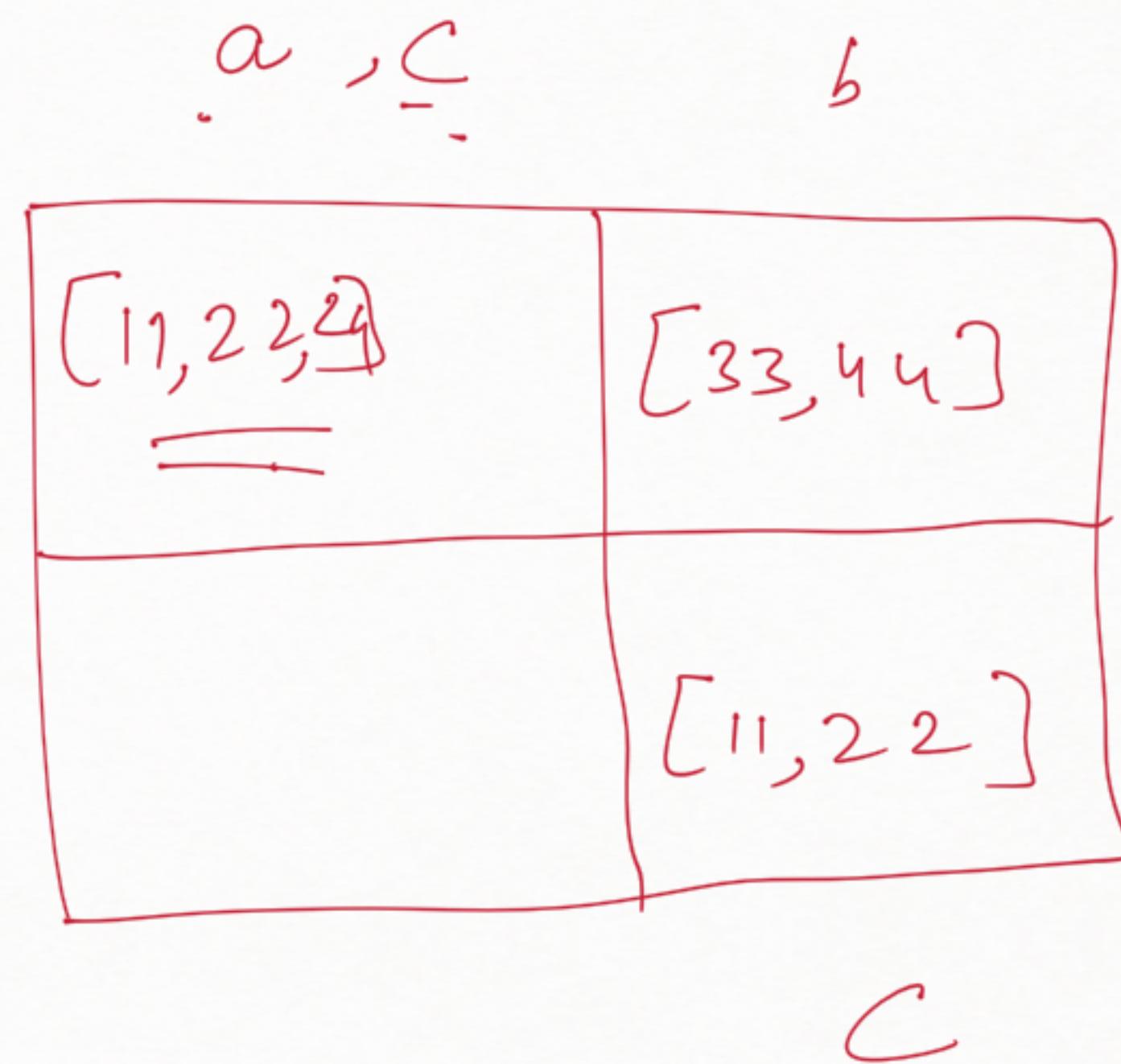
$a = [11, 22]$

$b = [33, 44]$

$c = a$

$a.append(22)$

$c = a.copy()$



Tuple : Collection of heterogeneous type of data,  
indexed, ordered, immutable (unchangeable).

$a[-1][1]$

$$a = (2, 2.5, \text{True}, 3+9j, [2, 5, 6])$$

$a[0:3:1]$

|    |     |      |        |                           |
|----|-----|------|--------|---------------------------|
| -5 | -4  | -3   | -2     | -1                        |
| 2  | 2.5 | True | $3+9j$ | $[2, 5, 6]$               |
| 0  | 1   | 2    | 3      | $\frac{-4}{-3} \mid 2 -1$ |

$a[-1][::2] \rightarrow [2, 5]$

|    |    |    |    |
|----|----|----|----|
| -4 | -3 | -2 | -1 |
| 2  | 5  | 6  |    |
| 0  | 1  | 2  |    |

$a[0] \rightarrow 2$

$a[-5] \rightarrow 2$

$a[-3] \rightarrow a[2]$

$a[3].real \rightarrow 3.0$

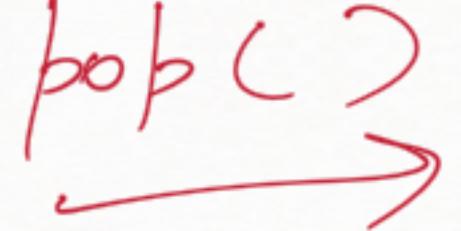
$a[-2].imag \rightarrow 9.0$

Dictionary :- collection of heterogeneous type of  
data, unindexed, ordered, mutable...

Key : Value , Key Unique, -- , { }

```
Student = { name : 'Manoj',  
            age : 12,  
            sub : ['maths', 'chem', 'Python']  
        }
```

remove :

{ clear ()  
del  
popitem ()  
pop 

Nested dictionary :-

dictionary inside dictionary

Set :- Collection of immutable - like of data ,  
unique , mutable , { } , unordered , indexed

$$a = \{ 8, 2.5, \text{True}, 2+3j, (2,5,6) \}$$

$$a1 = \{ 2, 5, 6, 2, 8, 1 \} \quad \text{error X}$$

$$\{ 2, 5, 6, 8, 1 \}$$

Union

$$S_1 = \{1, 2, 3, 4\}$$

intersection

$$S_2 = \{3, 4, 5, 6\}$$

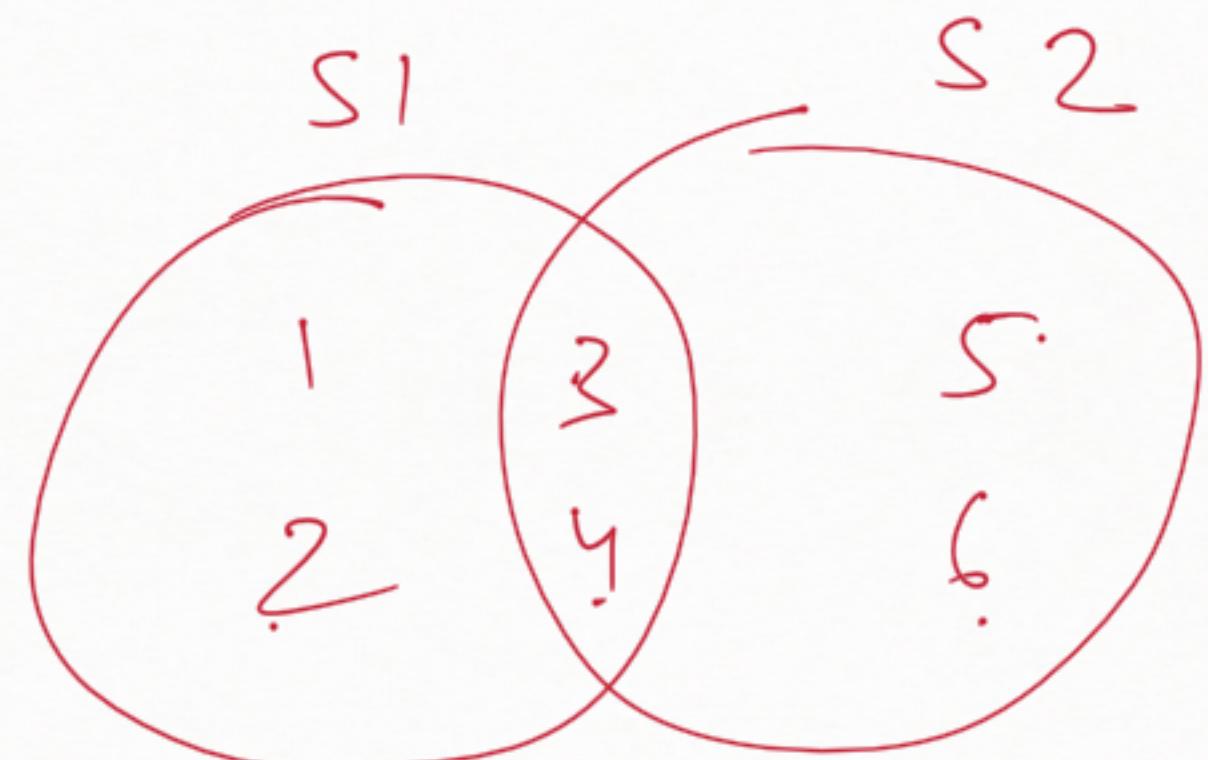
difference

symmetric

$$S_1 - S_2 = \{1, 2\}$$

$$S_2 - S_1 = \{5, 6\}$$

$$S_1 \cap S_2 = \{3, 4\}$$



$$S_1 \cup S_2$$

$$\{ \cancel{1}, \cancel{2}, \cancel{3}, 4, \cancel{3}, 4, 5, 6 \}$$

$$\{ \cancel{1}, \cancel{2}, \cancel{3}, 4, 5, 6 \}$$

adding element in set :

add → 1 element

update → more than 1 el.

del el.

discard()

remove()

pop()

del

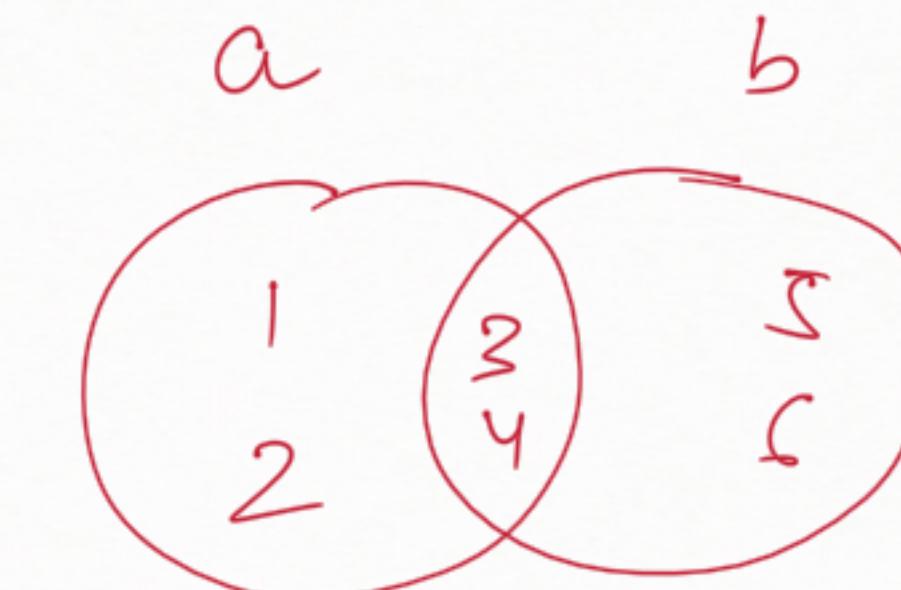
$$a = \{1, 2, 3, 4\}$$

$$b = \{3, 4, 5, 6\}$$

$$a.\text{difference}(b) = \{1, 2\}$$

$$a = \{1, 2, 3, 4\}$$

$$b = \{3, 4, 5, 6\}$$



$$\begin{aligned} a.\text{diff\_update}(b) &= \{1, 2\} \\ \uparrow & \end{aligned}$$

$$a = \{1, 2\}$$

$$b = \{3, 4, 5, 6\}$$

Boolean :- True , False  
                  |            0

logical operator :- and, or, not, xor -----

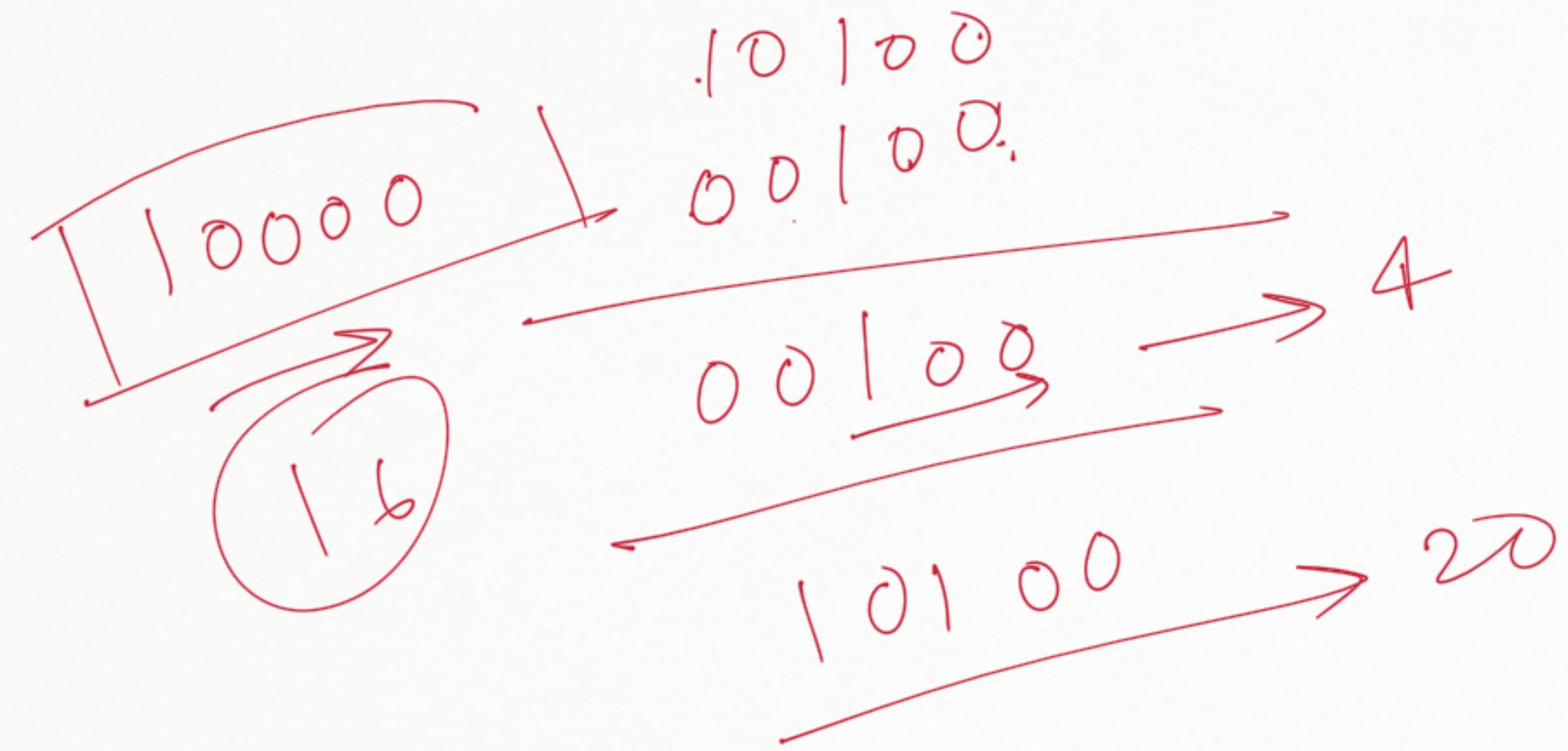
$i_1$   $i_2$  and  $o$   $xor$   
0 0 0 0 0  
0 1 0 1 1  
1 0 0 1 1  
1 1 1 1 0

not 0  $\rightarrow$  1  
1  $\rightarrow$  0

$a = 20$

$b = 4$

$a \text{ and } b \Rightarrow 4$



and → logical operators

$$20 \rightarrow 10100$$
$$4 \rightarrow 100$$

$$\begin{array}{r} 2 | 4 \\ 2 | 2 \end{array}$$

$$\begin{array}{r} 2 | 20 \\ 2 | 10 \\ 2 | 5 \\ 2 | 2 \\ 2 | 1 \\ 1 \end{array}$$

$a = 2$

$b = 4$

$a+b \rightarrow 6$

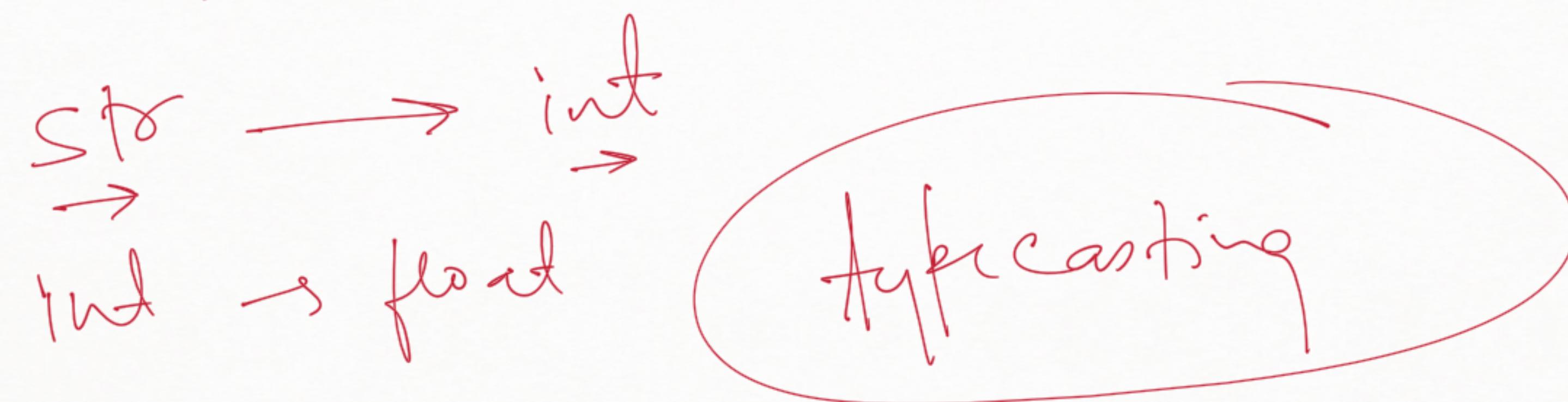
input( )



string

str  $\rightarrow$  num

process of converting one data-type into  
another type of data -



a = 'a'

$a: 'a' \rightarrow \text{int} \quad \text{str}$

`int(a)`

## Control Statement:

if

if - else

if - elif - else

nested if

  ↳

if True / false  
condition :

{  
==  
=  
=



if - elif - else :

if <sup>T/f</sup> condite :

{ == }

→ elif <sup>T/f</sup> condition :

{ == }

→ elif <sup>T/f</sup> condite :

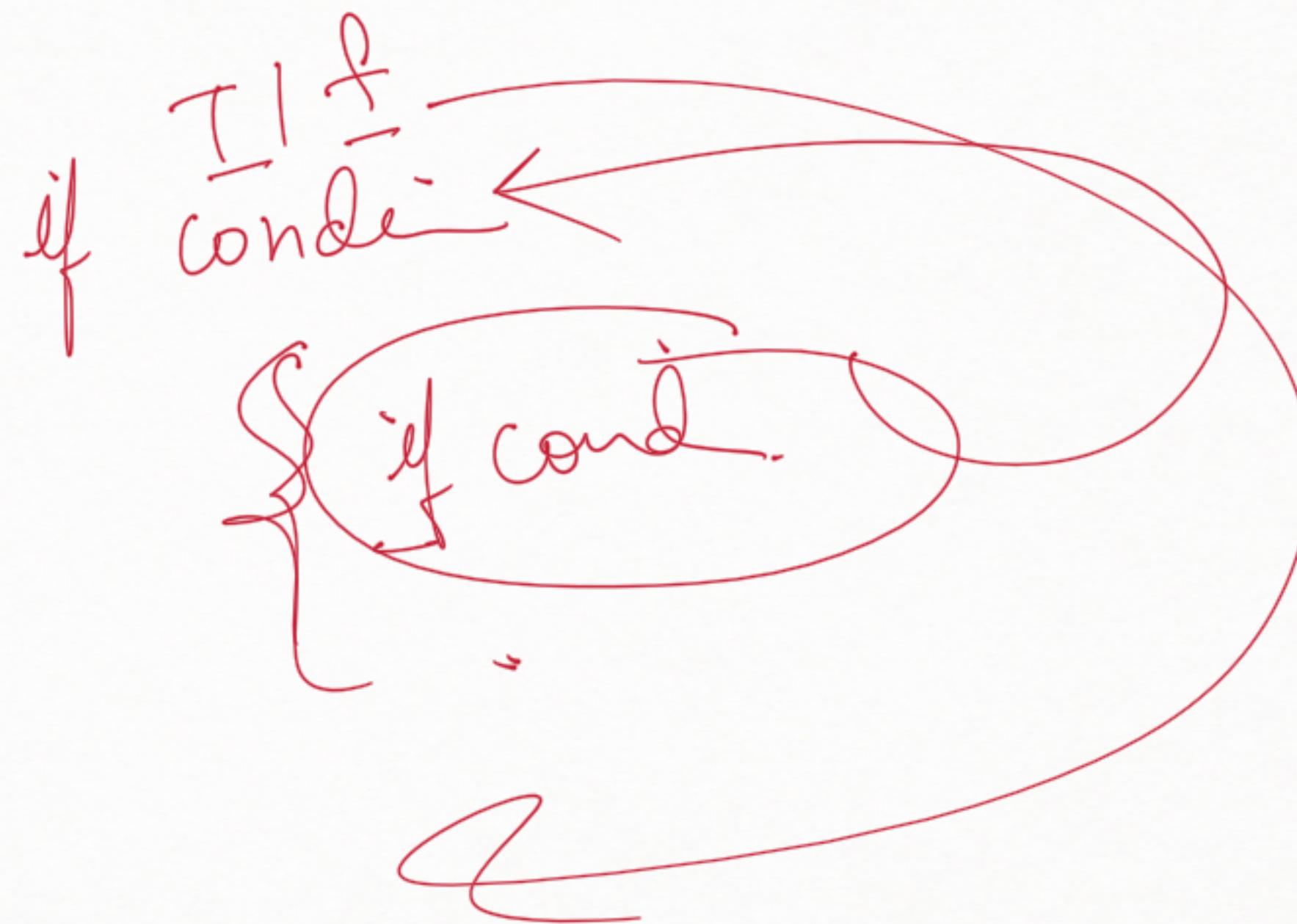
{ == }

→ else :



Nested if.

if inside if



$a, b, c = 2, 5, 9$

if ( $a \geq b$  and  $a \geq c$ ):

print('a is greater than b and c')  
→

$2 \geq 5$  and  $2 \geq 9$

f and f → f

Project - I

Quiz Game