Introduction to Multi-Dimensional lists

Primitive & abstract data types

- What is “Primitive”?
  - Literal meaning - old, ancient, “in-built”
  - eg int, char, float, boolean, etc.
- What is “Abstract”?
  - derived from Primitive
  - if primitive is Building blocks, then Abstract is the whole building

Python Lists

- recall Python Lists/1D arrays
- How data is stored in Computer
- Memory management
- Indexing

Conventional List, which is the one-dimensional List.

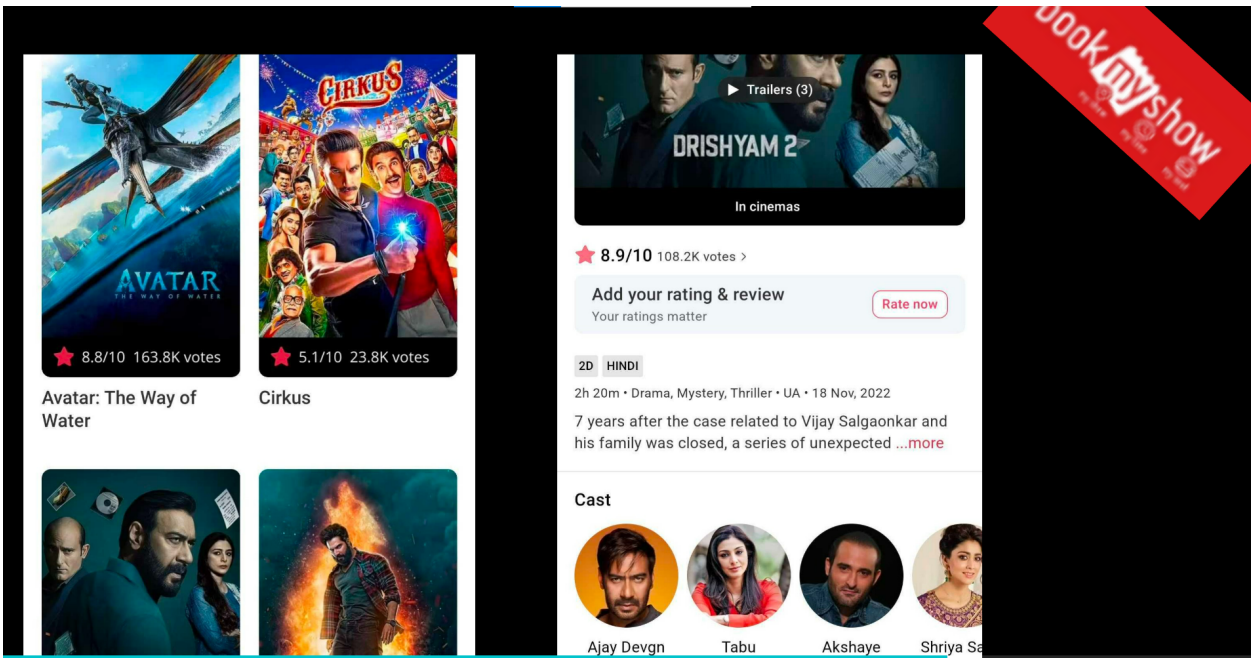
```
# example
my_list=[1,2,3,4,5];
```

- In the above example, I am storing numbers as an element of the list.
- I can also store lists inside the list.

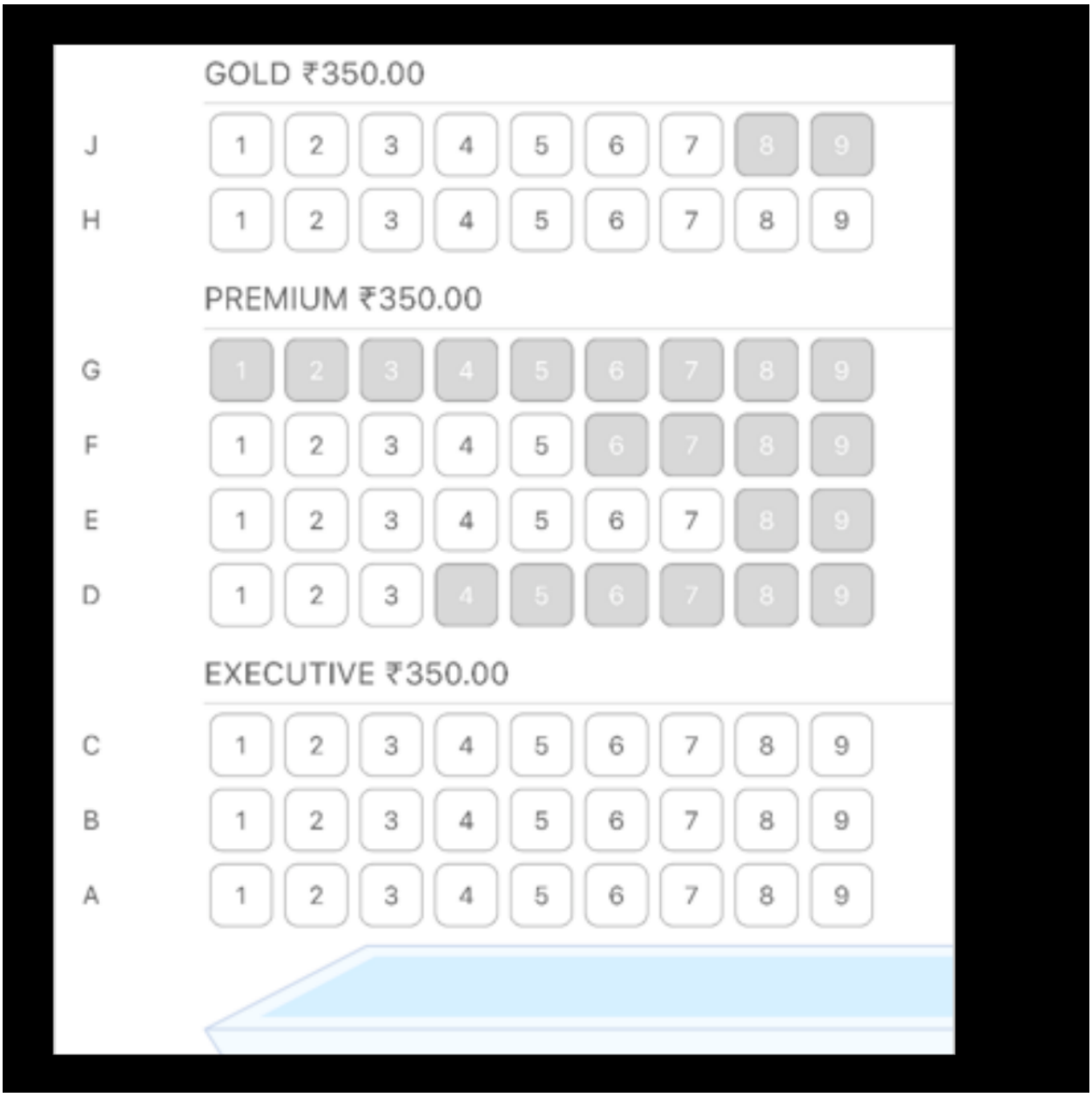
```
list1 =["A", ["M", "N"], "B"];

print(list1[0]); # A
print(list1[1]); # ["M", "N"]
```

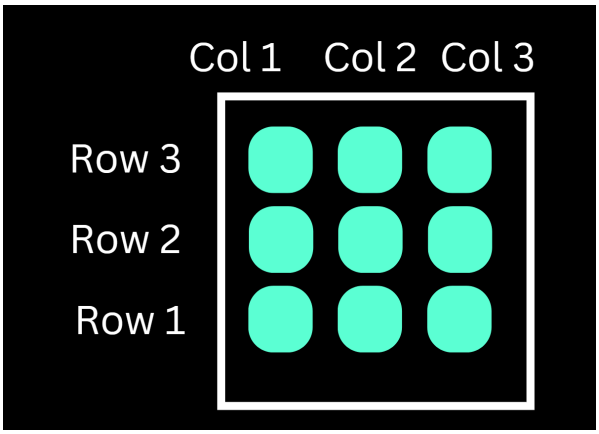
```
print(list1[2]); # B

#I am actually storing another list at index number 1 in the arr list
```

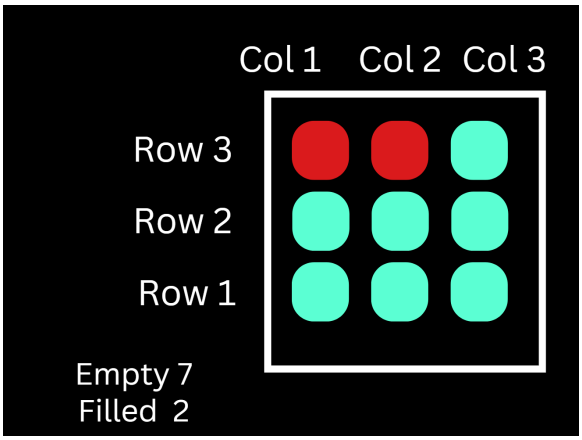
- BOOK MY SHOW app to help them visualize why I need 2D lists.
  Have you ever thought about how you go about selecting a seat at the movie theater when you go to watch the latest film with your friends and family?



- In the above app, I would require a 2D list, to store the info or I can say store the data.


GOLD ₹350.00
J 1 2 3 4 5 6 7 8 9
H 1 2 3 4 5 6 7 8 9
PREMIUM ₹350.00
G 1 2 3 4 5 6 7 8 9
F 1 2 3 4 5 6 7 8 9
E 1 2 3 4 5 6 7 8 9
D 1 2 3 4 5 6 7 8 9
EXECUTIVE ₹350.00
C 1 2 3 4 5 6 7 8 9
B 1 2 3 4 5 6 7 8 9
A 1 2 3 4 5 6 7 8 9





Create a multidimensional list in the program

```
let list1=[
  [1,2,3],
  [4,5,6],
  [7,8,9]
];

# it will have rows and columns just like a matrix that we all have studied in maths

print(list1[0]); # [1,2,3]
print(list1[1]); # [4,5,6]
print(list1[2]); # [7,8,9]
```

- There is one more method of doing this, using the **Push.**

```
#if I append something in a list it will get added to the list .

items=[];

items.append(["A", "B", "C"]);
items.append(["D", "E", "F"]);
print(items)

# all the appended lists will get added to the items list.
```

- The length of the list, and the length of the elements of the list.

```
items=[
  ["Ajay", 28, "male"],
  ["Hitesh", 30, "male"]
```

```
];

print(len(items))
# 2

# what if i need to print Ajay

print(items[0][0]); # Ajay
print(items[0][1]); #  28
print(items[0][2]); #  male

# Indexing will also happen at each element of the items list, as they are also list
```

Activity Time

https://www.canva.com/design/DAFNPiVFTQU/mhIzL-eC12B8iMLz0om5Gw/view?utm_content=DAFNPiVFTQU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

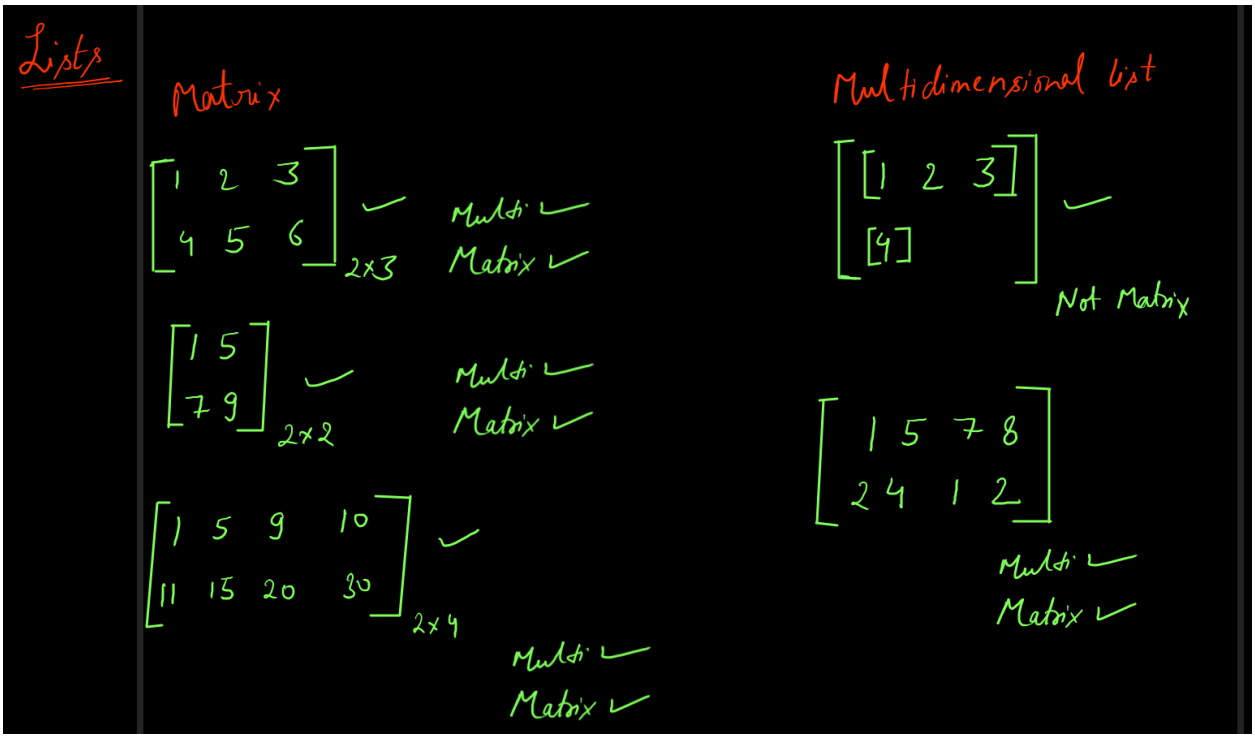
- **Print all the details of Hitesh just by the above method.**

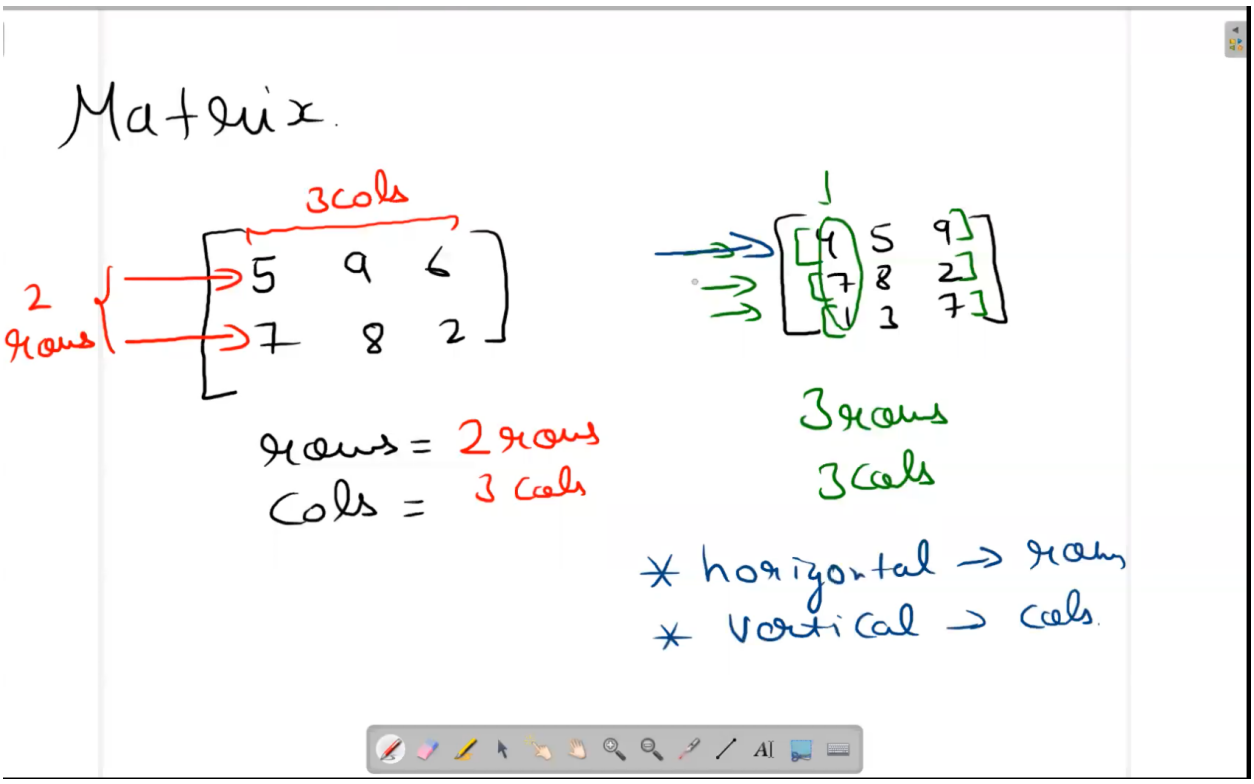
```
print(items[1][0]);   # Hitesh
print(items[1][1]);   #  30
print(items[1][2]);   #  male
```

- Show the below thing.

```
items=[
  ["Ajay", 28, "male", [1,2,3]],
  ["Hitesh", 30, "male", [4,5,6]]
];

print(items[0][3][2]); # 3
print(items[1][3][1]); # 5
print(items[1][0][2]); # t
```

- **Matrix Vs. Multidimensional** list**.**

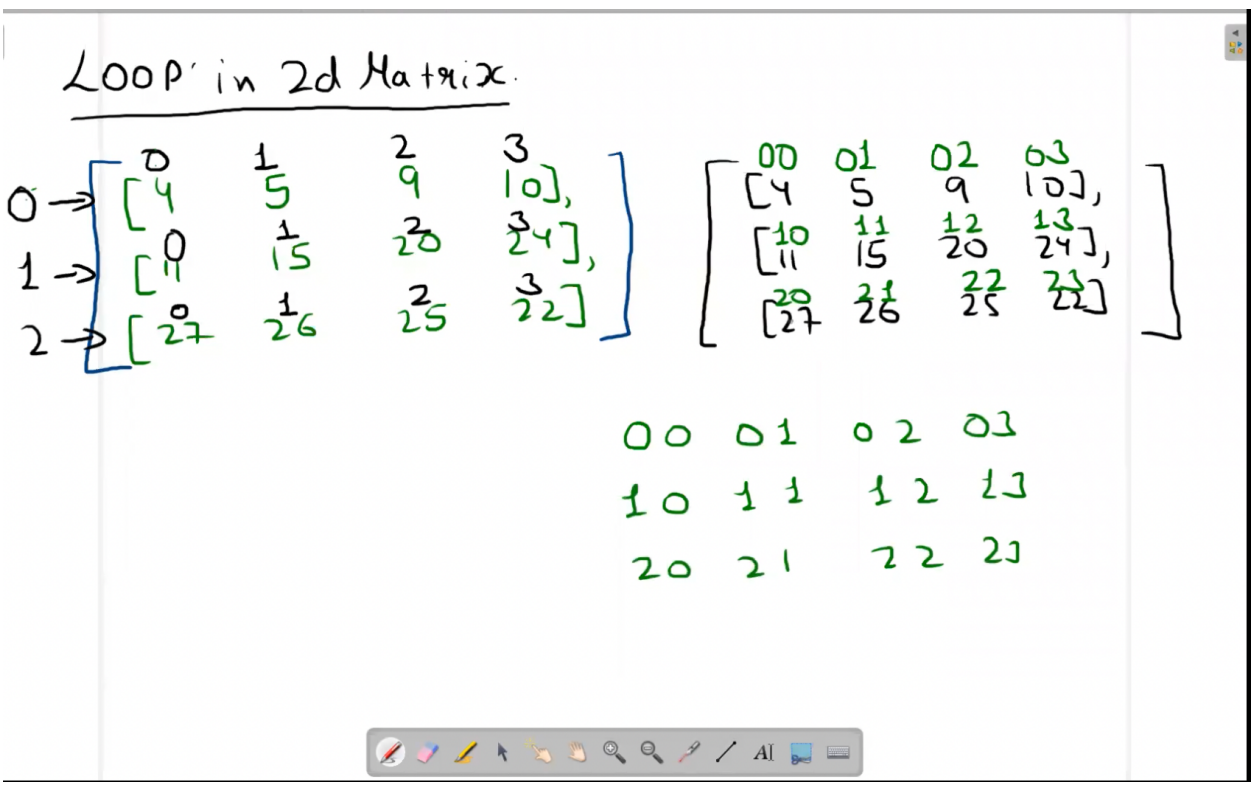

Lists
Matrix
Multidimensional list
[1 2 3]
[4 5 6]
2x3
Multi
Matrix
[1 5]
[7 9]
2x2
Multi
Matrix
[1 5 9 10]
[11 15 20 30]
2x4
Multi
Matrix
[[1 2 3]
[4]]
Not Matrix
[1 5 7 8]
[2 4 1 2]
Multi
Matrix


Matrix.
3cols
2 rows
[5 9 6]
[7 8 2]
rows = 2 rows
Cols = 3 cols
[[4 5 9]
[7 8 2]
[1 3 7]]
3rows
3cols
* horizontal → rows
* vertical → cols.

- The concept of **rows** and **columns.**
- **Rows = Length of the** list**.**
- **Column = Length of the element in the** list**.**

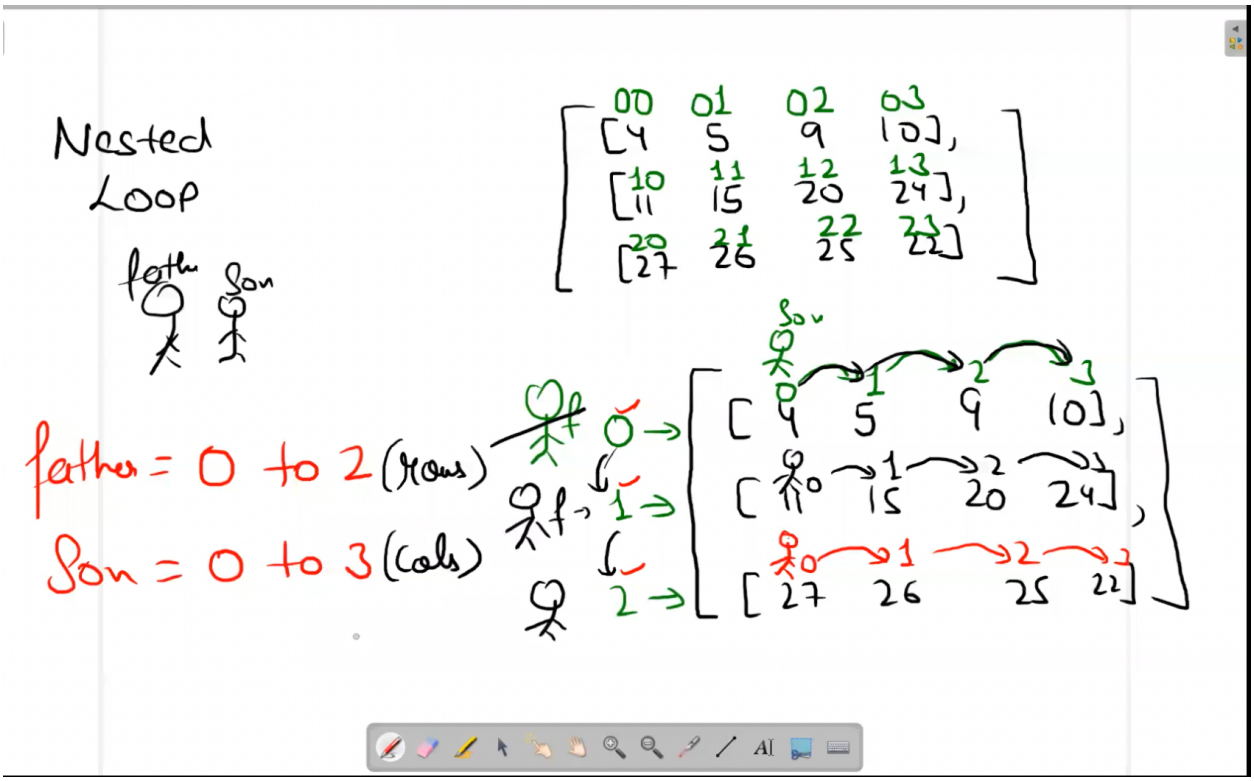
```
items=[
  [1,2,3],
  [4,5,6],
  [7,8,9]
];

rows= len(items)
cols=   len(items[0])
print(rows)
print(cols)
```

Loop in a 2D Matrix



- A nested loop to print all the elements.

- how can we use a nested loop for it?

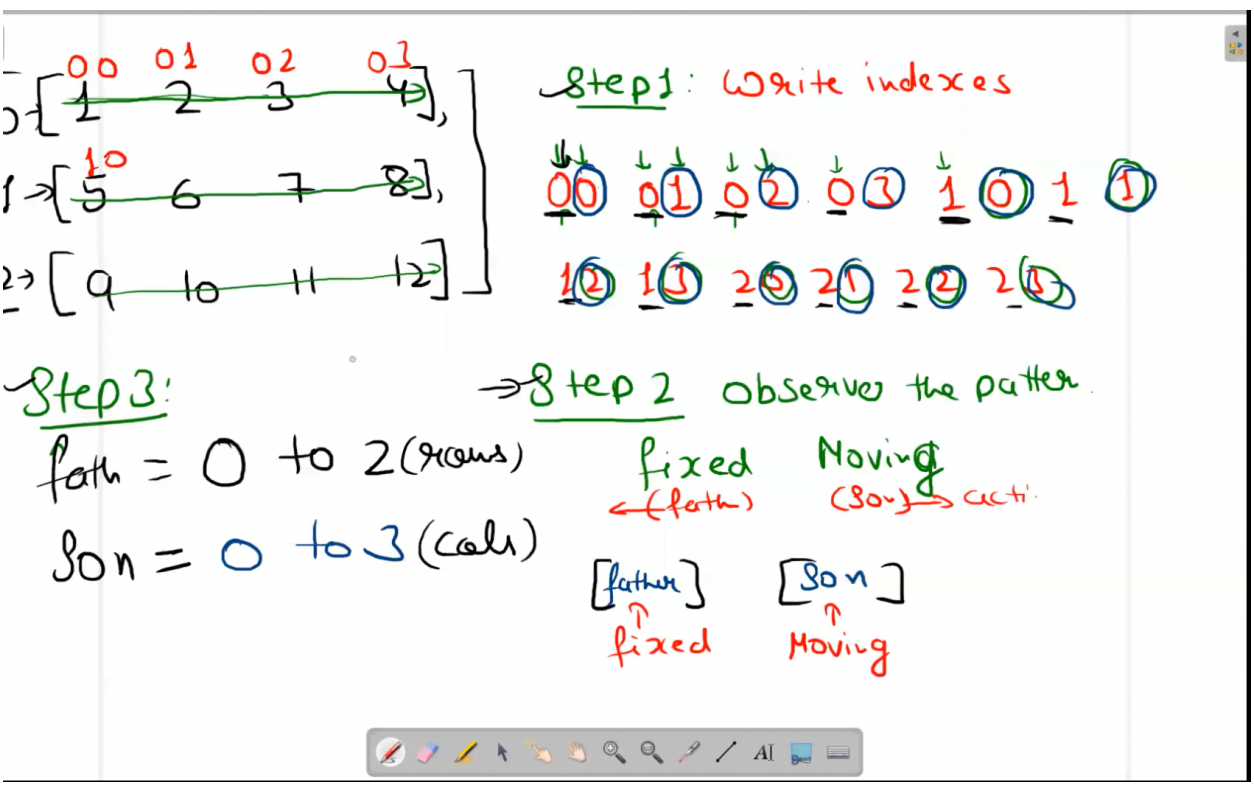



```
items=[
  [1,2,3],
  [4,5,6],
  [7,8,9]
];

rows= len(items)
cols= len(items[0])

for i in range(rows):
  for j in range(cols):
      print(items[i][j]);


# This will print all the elements of the matrix
```

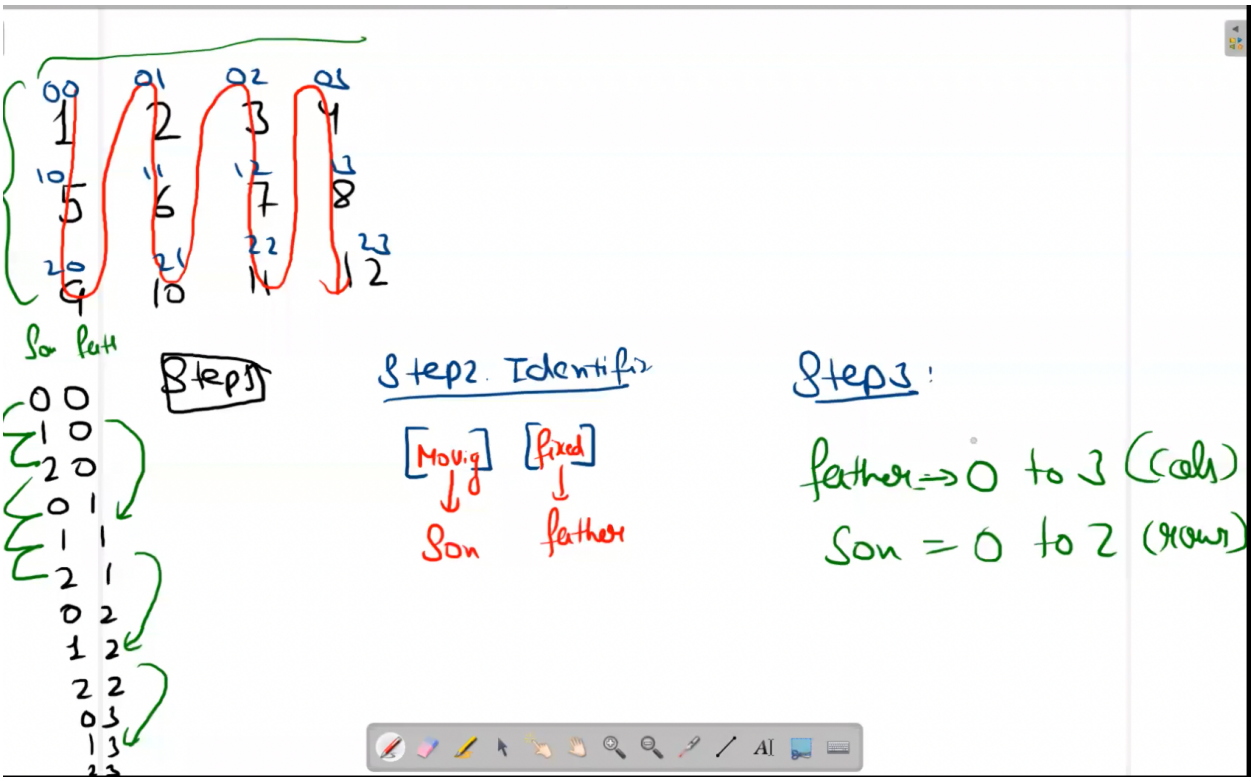
- How can I print it in matrix form?

```
Take Jhola
items=[
  [1,2,3],
  [4,5,6],
  [7,8,9]
];

rows= len(items)
cols = len(items[0])

for i in range(rows):
  jhola="";
  for j in range(cols):
    jhola=jhola+str(items[i][j]);


  print(jhola)

#This will print the elements in matrix form
```

- Tell them the stepwise approach of printing the matrix.



- Print in zig zag manner.



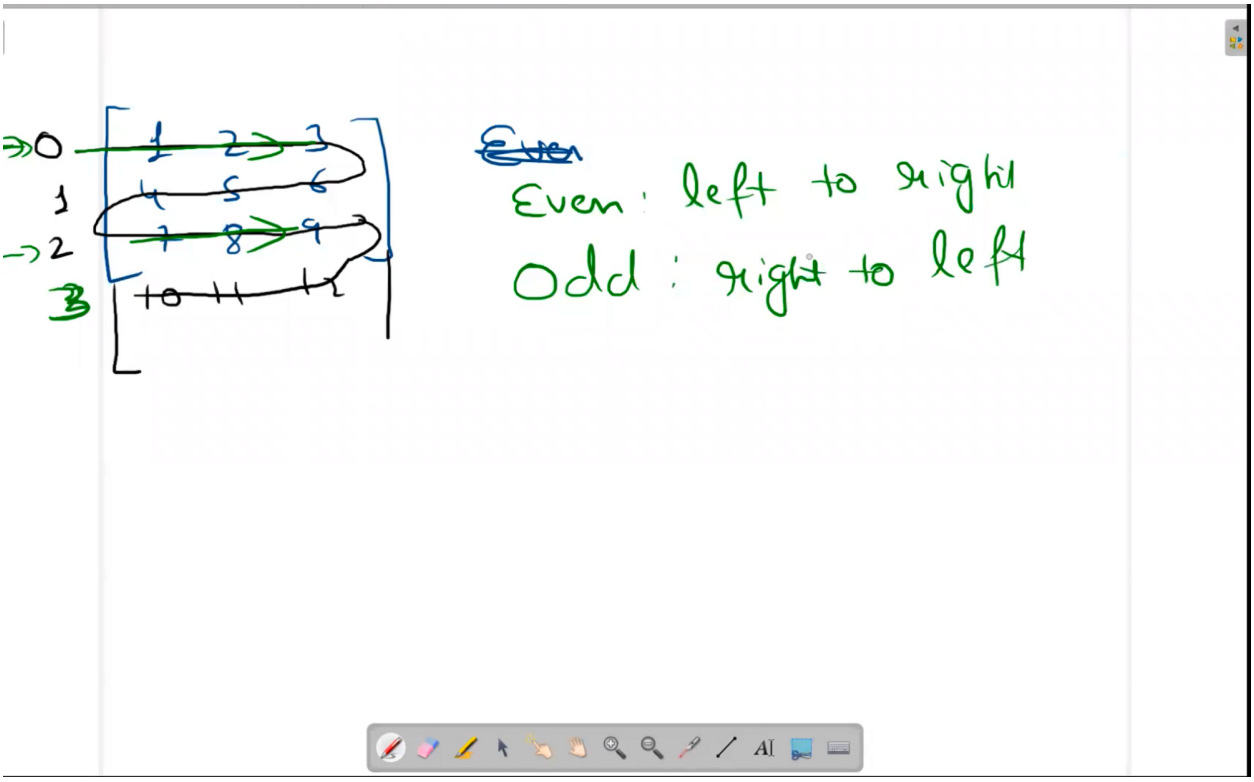

```
items=[
  [1,2,3],
  [4,5,6],
  [7,8,9]
];

rows= len(items)
cols = len(items[0])

for i in range(rows):
  jhola="";
  for j in range(cols):
    jhola=jhola+str(items[i][j])+" ";
```

IW Assignment

- **Problem 1: Given a square matrix, print both diagonals.**

```
# Left Diagonal = when i is equal to j
# Right Diagonal = sum of i and j is equal to rows
```

- **Problem 2: Print the matrix in snake pattern**



Happy Coding!