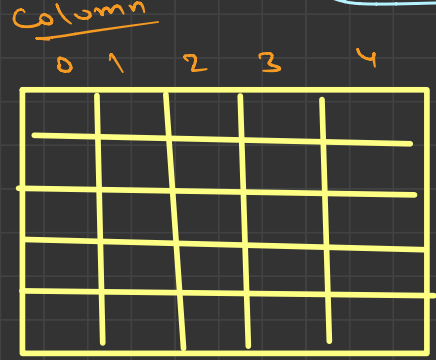
2-D array ⇒ 2-Dimensional Array

Matrix format



arr[row][column]

arr[4][4]

4 × 4 = 16 elements

Creation

2D ⇒ row → 5, Col → 10 > `int arr[5][10];`

Initialization → atleast we have to tell col size

```
int arr[2][3] = { {10,20,30},
                  {50,60,70} };
```

	0	1	2
0	10	20	30
1	50	60	70

Or

```
{ {1,2,3},{5,6,7} }
```

```
int arr[ ][ ]   } error throw
int arr[2][ ]   }
int arr[ ][2]   } correct
int arr[3][2]   }
```

↳ reason → formula also contain col

How to access?

row \ col	0	1	2
0	(0,0) 10	(0,1) 20	(0,2) 30
1	(1,0) 40	(1,1) 50	(1,2) 60
2	(2,0) 70	(2,1) 80	(2,2) 90

arr[0][0] = 10
arr[0][1] = 20
arr[0][2] = 30
arr[1][0] = 40
arr[1][1] = 50
arr[1][2] = 60
arr[2][0] = 70
arr[2][1] = 80
arr[2][2] = 90

In memory 2-D array store has linear array

10	20	30	40	50	60	70	80	90
0	1	2	3	4	5	6	7	8

formula

`Col * i + j` → to find index of element in memory

arr[1][1] => 3 * 1 + 1 => 4 index => 50