Two dimensional lists (2D lists)

list of list (same type). ⇒ eg:

Integer list	Integer / float list

but

list	String

X

li [row] [column]

eg: li = [[1,2,3],[4,5,6],[7,8,9]] ⇒ 3 × 3

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

li[1][2] = 6

li[1][3] = error

li[2][0] = 11

Say,

1	2	3	li0
4	5	6	li1
11	8	9	li2

All of them are different lists.

li0 [100] li1 [200] li2 [300]

m/y reference

100
200
300

li → | 100 li0 | 200 li1 | 300 li2 |

0	1	2
1	2	3

li[0][1] ⇒ 2

Jagged list :

column size is different.

eg:

li

1	2	3	4
5	6	7	
8	9		

Size: (4) (3) (2)

li	li[0]	li[1]	li[2]

li = [[1,2,3,4], [5,6,7], [8,9]]

li[0][3] ⇒ 4

li[2][3] ⇒ error

List comprehension :-

→ multiple no. of fors are allowed

[o/p for expression condition]

↓ optional

eg: li = [1,2,3,4]

li_squared = [ele**2 for ele in li]

⇒ [1, 4, 9, 16] //

o/p → for expression → Condition

li_even_squared = [ele**2 for ele in li if ele%2==0]

= [4, 16] //

Also possible,

① [o/p for-expr1 for-expr2 ... for-exprn cond1 cond2 ... condn]

② [o/p if cond else o/p for-expr]

User i/p of 2D lists :-

Method 1:

① row & column size

```
sta = input().split()
n,m = int(sta[0]), m = int(sta[1])
```

0, 1, 2

0×0	1×0	2×0
0×1	1×1	2×1
0×2	1×2	2×2
0×3	1×3	2×3

② using list comprehension

```
li = [ [int(j) for j in input().split()] for i in
```

for each list

```
range(n)]
```

↓

row size

In case of jagged list,

```
n = int(input())
li = [ [int(j) for j in input().split()] for i in
range(n)]
```

Method 2: line1 → row & column size, line2 → array elements

① row & column size

```
sta = input().split()
n, m = int(sta[0]), int(sta[1])
```

② entire elements in 1 line

```
arrElements = input().split()
```

③ using list comprehension

```
li = [[int(arrElement[m*i+j]) for j in range(m)] for i in range(n)]
```

- `for i in range(n)` → no. of rows
- `for j in range(m)` → m elements in each row
- `m*i+j` → index position.
- `int(arrElement[...])` → integer part of each element

Method 3: Every thing in single line

```
sta = input().split()
n, m = int(sta[0]), int(sta[1])
arrElement = sta[2:]
li = [[int(arrElement[m*i+j]) for j in range(m)] for i in range(n)]
```

Printing 2D lists :

Method 1 :

```
for i in range (row_size):
    for j in range (column_size):
        print (li[row_size][column_size], end = ' ')
    print()
```

O/P :-

row 1 elements → - - -
" 2 " → - - -
:
" n " → - - -

In case of jagged list,

```
for row in li:
    for ele in row:
        print (ele), end = ' ')
    print()
```

Doesn't matter row or column size.

Method 2 : using join()

```
for row in li:
    output = ' '.join ([str(ele) for ele in row])
    print (output)
```