2D
Arrays

2D Arrays are sort of like a Table

```
int[][] m = new int[2][3];
```

- `m` has 2 rows and 3 columns

	0	1	2
0			
1			

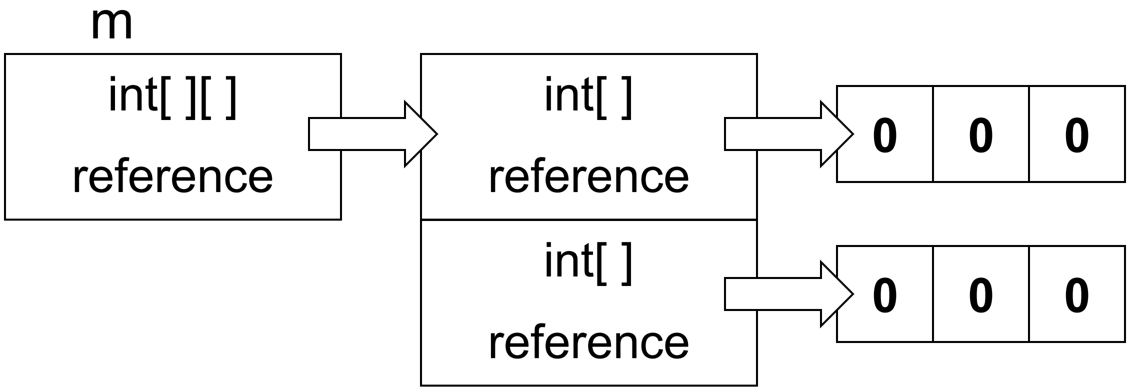
2D Arrays are Arrays of Arrays

```
int[][] m = new int[2][3];
```

- Creates an array of length 2.
- Each entry is an array of 3 `int` variables.
- `m[i][j]` is an `int`
- `m[i]` is an `int[]`

2D Arrays are Arrays of Arrays

```
int[][] m = new int[2][3];
```



`m[i][j]` is an `int` `m[i]` is an `int[]`

2D Array Dimensions

- Instance Variables (public final int)

- Rows –
  `m.length`

- Columns –
  `m[0].length`

- `m[rowIndex].length`

Note: For the AP Exam, you can assume that all 2D Arrays are rectangular. In other words, all the rows are equal length.

2D Array Declaration

- `int[][] m;`   // No array, only a reference to an array

- `int[][] m = new int[0][0];`   // Silly, can’t change size.

- `int[][] m = new int[2][3];`

-
  ```
  int[][] m = { {4,4,4},
                {4,4,4} };
  ```

2D Array Default Initialization

- `type[][] m = new type[4][5];`

- Initialized to zero or the moral equivalent of zero
  - 0 for `int`
  - 0.0 for `double`
  - `false` for `boolean`
  - `null` for objects

2D Array Accessing

- `m[4][3] = 5;`
- `int n = m[i][j];`

- `int[] a = m[3];`
- `m[2] = a;`

- Zero indexed

Row-Major Traversal (Indexed For Loop)

```
int[][] m = {{1, 2}, {3, 4}, {5, 6}, {7, 8}};

for (int r = 0; r < m.length; r++)
{
   for (int c = 0; c < m[0].length; c++)
     System.out.print(m[r][c] + "\t");
   System.out.println();
}
```

Row-Major Traversal
(Indexed For Loop)

m is {{1, 2}, {3, 4}, {5, 6}, {7, 8}};

1	2
3	4
5	6
7	8

Output

Row-Major Traversal (For-Each Loop)

```
int[][] m = {{1, 2}, {3, 4}, {5, 6}, {7, 8}};

for (int[] row : m)
{
   for (int num : row)
     System.out.print(num + "\t");
   System.out.println();
}
```

Enhanced for Statement
(For-Each Loop)

m is {{1, 2}, {3, 4}, {5, 6}, {7, 8}};

Output

```
1    2
3    4
5    6
7    8
```

Column-Major Traversal

```
int[][] m = {{1, 2}, {3, 4}, {5, 6}, {7, 8}};

for (int c = 0; c < m[0].length; c++)
{
   for (int r = 0; r < m.length; r++)
     System.out.print(m[r][c] + "\t");
   System.out.println();
}
```

Column-Major Traversals

m is {{1, 2}, {3, 4}, {5, 6}, {7, 8}};

```
1       3   5   7   Output
2   4   6   8
```