Two Dimensional Arrays

In this lesson, an introduction of another type of Arrays known as two-dimensional Array is provided.

We'll cover the following

- What are two-dimensional arrays?
  - Declaration
  - Instantiation
  - Graphical Representation
  - Example program

In the previous lesson, we got familiar with the basic concepts of Arrays in Java. All we have been learning till now was about **linear arrays**.

Another type of arrays is a **two dimensional array**, let's discuss briefly about it.

What are two-dimensional arrays?

Unlike linear arrays, two-dimensional arrays are just like an *m* x *n* matrix with **m** number of *rows* and **n** number of *columns*.

Declaration

The declaration of *2-d arrays* is as follows:

```
Datatype[m][n] name;
```

The elements in a two-dimensional array are arranged in the form of rows and columns.

Instantiation

The instantiation of a character array with 2 rows and 2 columns will be as:

```
int[][] twoDimArray = new int[2][2];
```

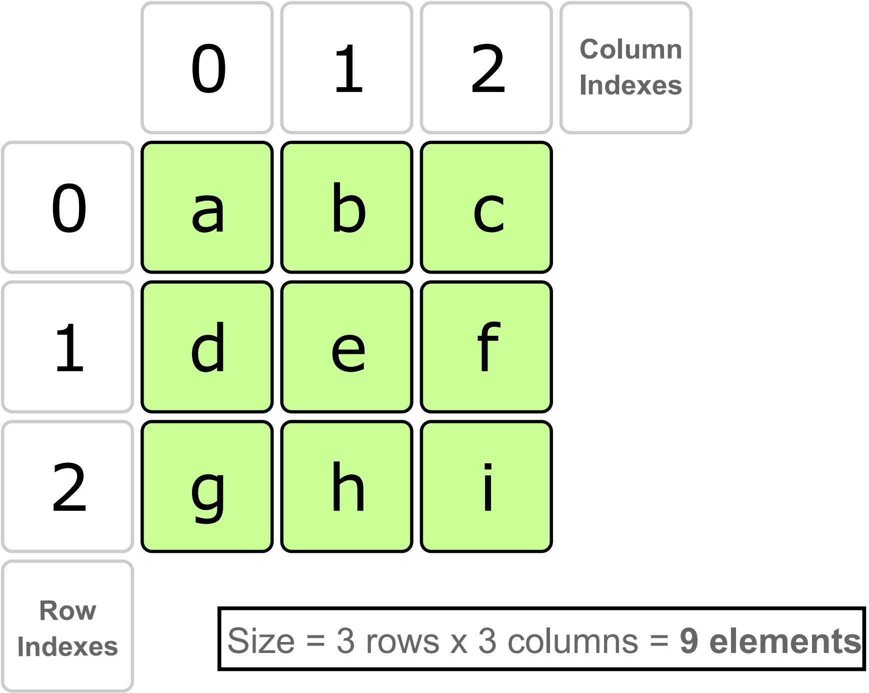
> What should be the index range of two-dimensional arrays?

Hide Hint

The index range of a two-dimensional array of size m x n will be: row indexes: 0 to m-1 column indexes: 0 to n-1

Graphical Representation

Graphically we can portray the 2D array as:



	0	1	2	Column Indexes
0	a	b	c	
1	d	e	f	
2	g	h	i	
Row Indexes				

Size = 3 rows x 3 columns = **9 elements**

Two-dimensional Arrays

Each element in a 2-d array is indexed by means of row and column index using square bracket notation. To access the `b` in the above array we will use the following notation:

```
arrayName[0][1]
```

Example program

Let’s initialize this two-dimensional character array with some characters and get them printed to the console:

```
class HelloWorld {
    public static void main(String args[]) {

        char[][] twoDimArray = new char[3][3]; //instantiating a character array of size 3*3 = 9 e

        twoDimArray[0][0] = 'a'; //stores a at row:0, column:0 position
        twoDimArray[0][1] = 'b'; //stores b at row:0, column:1 position
        twoDimArray[0][2] = 'c'; //stores c at row:0, column:2 position
        twoDimArray[1][0] = 'd'; //stores d at row:1, column:0 position
        twoDimArray[1][1] = 'e'; //stores e at row:1, column:1 position
        twoDimArray[1][2] = 'f'; //stores f at row:1, column:2 position
        twoDimArray[2][0] = 'g'; //stores g at row:2, column:0 position
        twoDimArray[2][1] = 'h'; //stores h at row:2, column:1 position
        twoDimArray[2][2] = 'i'; //stores i at row:2, column:2 position

        //Printing out the stored values

        System.out.print(twoDimArray[0][0] + "  " + twoDimArray[0][1] + "  " + twoDimArray[0][2] +
        System.out.print(twoDimArray[1][0] + "  " + twoDimArray[1][1] + "  " + twoDimArray[1][2] +
        System.out.print(twoDimArray[2][0] + "  " + twoDimArray[2][1] + "  " + twoDimArray[2][2]);
    }
}
```

> **Note:** The dimensions of an `array` is not limited to just `2`. An `array` can be of `n` number of dimensions depending on the problem.

This was pretty much about the Arrays, now let’s test our understanding through solving some coding challenges.