Solution Review: Print a Matrix

In this review, solution of the challenge 'Print a Matrix' from the previous lesson is provided.

We'll cover the following

Solution

```
class TwoDimArr {
    public static void main( String args[] ) {
       int n = 3;
       int[][] arr = new int[n][n];
        for (int i = 0; i < arr.length; i++) { //assign values to the arr
            for (int j = 0; j < arr.length; j++) {
                if (i == j) { //if row=column=> fill the matrix with 0
                    arr[i][j] = 0;
                } else if (i > j) { //if row>columns=> fill matrix with -1
                    arr[i][j] = -1;
                } else { //if row<columns=> fill matrix with 1
                    arr[i][j] = 1;
                }
            }
        }
        for (int i = 0; i < arr.length; i++) { //print the array
            for (int j = 0; j < arr.length; j++) {
                System.out.print(arr[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

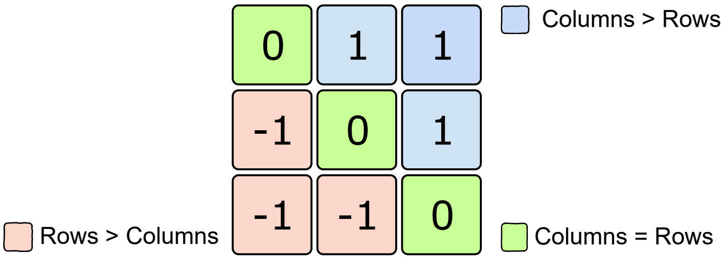
How does the above code work?

In the above solution, we have to initialize an array using nested loops having:

- All the diagonal elements as `0` which means: when the *row index:* `i` is **equal** to the *column index:* `j` we store zeros in the array.
- All the elements on the lower side of the diagonal as `-1` which means: when

- All the elements on the lower side of the diagonal as `-1` which means: when the *row index:* `i` is **greater** than the *column index:* `j` we store `-1` in the array.
- All the elements on the upper side of the diagonal as `1` which means: when the *row index:* `i` is **lesser** than the *column index:* `j` we store `1` in the array.
- After initialization, we have implemented nested loops once again to print the array.

For clearer understanding refer to the below picture:



0	1	1
-1	0	1
-1	-1	0

Let's solve another challenge related to two-dimensional arrays in the upcoming lesson.