Two-Dimensional Arrays
Two-Dimensional Arrays

- A *one-dimensional array* stores a list of elements
- A *two-dimensional array* can be thought of as a table of elements, with rows and columns

<table>
<thead>
<tr>
<th>one dimension</th>
<th>two dimensions</th>
</tr>
</thead>
</table>

![Diagram of one-dimensional and two-dimensional arrays](image-url)
Two-Dimensional Arrays

- In Java, a two-dimensional array is an *array of arrays*

```
int[][] matrix = new int[12][50];
```

- A two-dimensional array is declared by specifying the size of each dimension separately:
# Two-Dimensional Arrays

- **Declaration:**
  ```java
  int[][] matrix = new int[12][50];
  ```

- **Referencing a single element:**
  ```java
  value = matrix[3][6];
  ```

- The array stored in one row can be specified using one index

<table>
<thead>
<tr>
<th>Expression</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>matrix</td>
<td>int[][]</td>
<td>2D array of integers, or array of integer arrays</td>
</tr>
<tr>
<td>matrix[5]</td>
<td>int[]</td>
<td>array of integers</td>
</tr>
<tr>
<td>matrix[5][12]</td>
<td>int</td>
<td>integer</td>
</tr>
</tbody>
</table>
Looping Through a 2D Array

```java
int M = 10, N = 5;
double[][] a = new double[M][N];
for (int i = 0; i < M; i++)
    for (int j = 0; j < N; j++)
        a[i][j] = 0;
```

```java
int M = 10, N = 5;
double[][] a = new double[M][N];
for (int i = 0; i < a.length; i++)
    for (int j = 0; j < a[i].length; j++)
        a[i][j] = 0;
```
Ragged 2D Arrays

```java
int M = 9;
double[][] a = new double[M][];
for (int i = 0; i < M; i++) {
    a[i] = new double[M-i];
    for (int j = 0; j < a[i].length; j++)
        a[i][j] = 0.0;
}
```
Ragged 2D Arrays

```java
int scores[][] = {
    {44, 55, 66, 77},
    {36},
    {87, 97},
    {68, 78, 88}
};
```