Jagged Arrays

This lesson will tell you about the jagged arrays in C# and how they differ from the multidimensional arrays.

Jagged arrays are arrays that instead of primitive types, contain arrays (or other collections).

> It's like an **array of arrays** - each array element contains another array

They are similar to multidimensional arrays, but have a slight difference - as *multidimensional arrays* are limited to a **fixed number of rows and columns**, with *jagged arrays*, **every row can have a different number of columns**.

Declaring a Jagged Array

For example, declaring a jagged array with **8** rows:

```
int[][] a = new int[8][];
```

The second `[]` is initialized without a number. To initialize the sub arrays, you would need to do that separately:

```
using System;

public class MainClass {

   public static void Main(String [] args)
```

```
    {
        int[][] a = new int[8][];
        for (int i = 0; i < a.Length; i++)

        {
            a[i] = new int[10];
        }
    }
}
```

Getting/Setting Values

Now, getting one of the *subarrays* is easy. Let's print all the numbers of the **3rd row** of `a`:

```
using System;

public class MainClass {

    public static void Main(String [] args)
    {
        int[][] a = new int[8][];
        for (int i = 0; i < a.Length; i++)
        {
            a[i] = new int[10];
        }
        for (int i = 0; i < a[2].Length; i++)
        {
            Console.WriteLine(a[2][i]);
        }
    }
}
```

To *get* a specific value, follow this syntax:

```
a[<row_number>][<column_number>]
```

Similarly, to set a specific value:

```
a[<row_number>][<column_number>] = <value>
```

Note:

> It’s always recommended to use *jagged arrays* (**arrays of arrays**) rather than *multidimensional arrays* (**matrices**). It’s faster and safer to use.

So far, you pretty much got the idea of how arrays work in C# - now let’s have a quick quiz on arrays!