

---

---

## C# Array

### Arrays in C# Programming Language

Array is a collection of many values of the same type. The array can be one or multi-dimensional array. Programmers mostly use one and two-dimensional arrays.

#### One-Dimensional Array

-Declaring one-dimension array

To declare a one-dimension array in C#, you need to use the **new** keyword as shown below.

#### Example

```
int[] i=new int[5];
```

#### Initializing array

To assign values to the array, you can write those values as below.

```
int[] i=new int[5]{1, 2, 3, 4, 5};//The array got values  
or
```

```
i[0]=1;
```

```
i[1]=2;
```

```
i[2]=3;
```

```
i[3]=4;
```

```
i[4]=5;
```

Note: The start index of the array is 0 and the end its end index is equal to its size subtracted by 1.

#### Accessing elements of an array

You can access the elements of array by specify their indexes.

**Example:**

```
//Declaring 1D array
int[] i=new int[5];

//assign values to the array
i[0]=1;
i[1]=2;
i[2]=3;
i[3]=4;
i[4]=5;

//Accessing 1D array elements
for(int j=0;j<5;j++) Console.WriteLine("{0} ",i[j]);/*accessing values from the array */
```

**-Two-dimension array**

Index	0	1	2
0	1	2	3
1	1	2	3
2	1	2	3

To create and use a two-dimension array, you will use two square brackets instead of one.

**Example:**

```
//Declaring 2D array
int[,] a=new int[3,3];

//initialize the 2D array
for (int r = 0; r < 3; r++)
```

```
for (int c = 0; c < 3; c++)  
  
    a[r, c] = c + 1;  
  
//accessing values from the array  
  
for (int r = 0; r < 3; r++)  
  
{  
  
    for (int c = 0; c < 3; c++)  
  
        Console.Write("\t{0}", a[r, c]);  
  
    Console.WriteLine();  
  
}
```

### Exercises

1. A two-dimensional array stores values in rows and columns. By using two-dimensional array, write C# program to display a table of numbers as shown below:

1	2	3	4	5
6	7		9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

#### Solution:

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.IO;  
  
  
namespace ConsoleApplication1
```

```
{
```

```
class Program
```

```
{
```

```
static void Main(string[] args)
```

```
{
```

```
    printSeries();
```

```
    Console.ReadLine();
```

```
}
```

```
public static void printSeries()
```

```
{
```

```
    int[,] tArr = new int[5, 5];
```

```
    int i, j;
```

```
    for (i = 0; i < 5; i++) //assign values to the two-dimensional array
```

```
        for (j = 0; j < 5; j++)
```

```
            {
```

```
                if (i == 0) tArr[i, j] = j + 1; //fill the first row
```

```
                else if (i > 0 && j == 0)
```

```
                    tArr[i, j] = tArr[i - 1, 4] + 1; //fetching the value of the last cell in the previous row
```

```
                else
```

```
                    tArr[i, j] = tArr[i, j - 1] + 1; //fill subsequent cells
```

```
}
```

```
for(i=0;i<5;i++){ //print the array
```

```
    for(j=0;j<5;j++)
```

```
        Console.Write("{0}\t",tArr[i,j]);
```

```
    Console.WriteLine();
```

```
}
```

```
}
```

```
}
```

```
}
```

2. In this C# exercise, you are about to write C# program to display a matrix as shown below. The diagonal of the matrix fills with 0s. The lower side fills with -1s and the upper side fills with 1s.

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

Solution:

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.IO;
```

```
namespace ConsoleApplication1
```

```
{
```

```
class Program
```

```
{
```

```
static void Main(string[] args)
```

```
{
```

```
printMatrix();
```

```
Console.ReadLine();
```

```
}
```

```
public static void printMatrix()
```

```
{
```

```
int[,] matrix = new int[5, 5];
```

```
int i, j;
```

```
for (i = 0; i < 5; i++) //assign values to the matrix
```

```
for (j = 0; j < 5; j++)
```

```
{ //if row=column=> fill the matrix with 0
```

```
if (i == j) matrix[i, j] = 0;//if row>columns=> fill matrix with -1
```

```
else if (i > j) matrix[i, j] = -1;//if row<columns=> fill matrix with 1
```

```
else matrix[i, j] = 1;
```

```
}
```

```
for (i = 0; i < 5; i++)  
{ //print the matrix  
    for (j = 0; j < 5; j++)  
        Console.Write("{0}\t", matrix[i, j]);  
    Console.WriteLine();  
}  
  
}  
  
}  
  
}
```

For more c# exercises visit <http://www.worldbestlearningcenter.com>