TutorialKart

C# Array

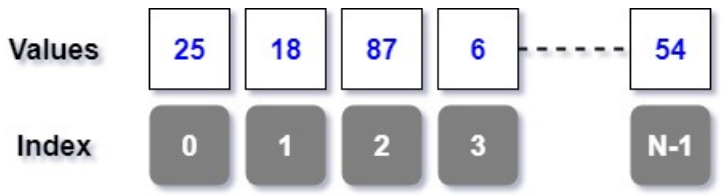
C# Array

C# Array is a sequential collection of same datatype values which can be accessed by index.

Any element in the array can be accessed using the array name and index of the element.

The index of an array starts with `0` and increments by `1` for the subsequent elements.

Consider the following figure which depicts an integer array with values and index.



If name of the array is myArray, you can access the values using index as shown below.

- `myArray[1]` gives `18`
- `myArray[0]` gives `25`
- `myArray[3]` gives `6`

Declare C# Array

To declare an array in C#, use the following syntax.

```
datatype[] arrayName;
```

where

- `datatype` is the datatype of elements that will be stored in the array.
- `arrayName` is the name by which we reference this array.

Examples

```
int[] numbers;
 string[] names;
```

Define C# Array

During the declaration, we only declared that `arrayname` would be used for for an array to store elements of specified datatype.

But, in the definition, we are actually defining the size of the array.

```
arrayName = new datatype[size];
```

where

- `new` is keyword.
- `datatype` is the datatype of elements stored in this array.
- `size` is the number of elements that can be stored in this array.

Both declaration and definition of an array can be done in a single statement as shown below.

```
datatype[] arrayName = new datatype[size];
```

Examples

```
int[] numbers = new int[10];
 string[] names = new string[6];
```

Initialize C# Array

You can initialize an array with values using curly braces and comma as shown below.

```
int[] numbers = {25, 18, 87, 6, 41, 54};
 string[] names = {"Aby", "Skye", "Mack"};
```

Example C# Array

In the following example, we declare and initialize an integer array and print some of its elements using index.

```
using System;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            int[] numbers = {25, 18, 87, 6, 41, 54};
```

```
            Console.WriteLine("numbers[1] : "+numbers[1]);
            Console.WriteLine("numbers[3] : "+numbers[3]);
        }
    }
}
```

Output

```
numbers[1] : 18
numbers[3] : 6
```

Example – C# String Array

In the following example, we will define and initialize a string array, and print some of its values using index.

Program.cs

```
using System;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            string[] names = {"Aby", "Skye", "Mack", "Phil", "May"};
            Console.WriteLine("names[1] : "+names[1]);
            Console.WriteLine("names[3] : "+names[3]);
        }
    }
}
```

Output

```
names[1] : Skye
names[3] : Phil
```

C# Array Length

You can get the length of the array using array.Length property.

Program.cs

```
using System;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            string[] names = {"Aby", "Skye", "Mack", "Phil", "May"};
            int len = names.Length;
            Console.WriteLine("Length of array : "+len);
        }
```

```
    }
}
```

Output

```
Length of array : 5
```

Example – C# Print Array using for loop

In the following example, we will use C# for loop to print elements of array.

Program.cs

```
using System;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            string[] names = {"Aby", "Skye", "Mack", "Phil", "May"};
            for(int index=0;index<names.Length;index++){
                Console.WriteLine(names[index]);
            }
        }
    }
}
```

Output

```
Aby
Skye
Mack
Phil
May
```

Example – C# Print Array using For loop

In the following example, we will use C# for loop to print elements of array.

Program.cs

```
using System;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            string[] names = {"Aby", "Skye", "Mack", "Phil", "May"};
            for(int index=0;index<names.Length;index++){
                Console.WriteLine(names[index]);
            }
        }
```

```
    }
}
```

Output

```
Aby
Skye
Mack
Phil
May
```

Example – C# Print Array using While loop

In the following example, we will use C# while loop to print elements of array.

Program.cs

```
using System;

namespace CSharpExamples {
    class Program {
        static void Main(string[] args) {
            string[] names = {"Aby", "Skye", "Mack", "Phil", "May"};
            int index=0;
            while(index<names.Length){
                Console.WriteLine(names[index]);
                index++;
            }
        }
    }
}
```

Output

```
Aby
Skye
Mack
Phil
May
```

Example – C# Print Array using foreach

In the following example, we will use C# foreach to print elements of array without index.

Program.cs

```
using System;

namespace CSharpExamples {
```

```
    class Program {
        static void Main(string[] args) {
            string[] names = {"Aby", "Skye", "Mack", "Phil", "May"};
            foreach(string name in names){
                Console.WriteLine(name);
            }
        }
    }
}
```

Output

```
Aby
Skye
Mack
Phil
May
```

Conclusion

In this C# Tutorial, we have learned to declare, define and initialize an array, access the elements using index, and iterate through elements using looping statement.

- C# IOException
- C# FieldAccessException

C# Collections

- C# List
- C# SortedList
- C# HashSet
- C# SortedSet
- C# Stack
- C# Queue
- C# LinkedList
- C# Dicionary
- C# SortedDictionary

C# Errors [Solved]

- C# Error: Class does not contain a constructor that takes arguments