C# Concepts →

Docs / C# / Arrays

Arrays

+2

Published Nov 3, 2022 • **Updated May 15, 2024**

Contribute to Docs →

An **array** is a data structure used in C# to store a sequential collection of elements. Its size is immutable (cannot be changed after creation). The elements of an array are all of the same type, but it is possible to define a C# array that can hold elements of any type by specifying the type of the array as an object. In C#, all types directly or indirectly inherit from `Object`.

Syntax

There are several ways to create an array in C#:

```
// Create a variable of type "type[]" without initializing it:
type[] arrayName;

// Create the array variable and initialize it with an array of N items:
type[] arrayName = new type[N];

// Create the array variable and initialize it by specifying the contents:
type[] arrayName = new type[] { value1, value2, value3, ... valueN };

// Alternative way of creating the array and specifying the contents:
type[] arrayName = { value1, value2, value3, ... valueN };
```

> **Note:** Arrays in C# have a set size, meaning the number of elements they hold cannot be changed once the array has been created.

Example

Each element in an array is assigned a specific index starting at zero. To access or modify an element in the array, you refer to it by its index and operate on it accordingly.

```csharp
using System;

public class Example
{
  public static void Main(string[] args)
  {
    char[] vowels = {'a', 'e', 'i', 'o', 'u'};
    //      indexes:  0    1    2    3    4

    Console.WriteLine(vowels[0]); // Output: a

    vowels[0] = 'r';

    Console.WriteLine(vowels[0]); // Output: r
  }
}
```

In the example above, an array of `char`s was initialized with all the vowels. The first element in the array at index 0 was printed. Then, the element at index 0 was modified by assigning it a new value of `'r'`. Then, the value at index 0 was printed again.

Array Methods

Arrays in C# are objects, not just contiguous blocks of memory as in C and C++. `Array` is the base type of all arrays, and any array can use the properties and methods of the `Array` object, a few of which are listed below:

Arrays

.Clear()

Clears the contents of an array, returning each element to its default value.

.Copy()

Copies elements in an array within a certain range.

.CopyTo()

Copies the elements of an array to another array.

.Length

Returns the total number of elements in the array.

.Resize()

Updates the size of an existing array.

.Reverse()

Reverses the sequence of a subset of the elements in a one-dimensional array.

.Sort()

Arranges the elements of an array in ascending or alphabetical order.

All contributors

@THE-Spellchecker   @garanews

@StevenSwiniarski

Contribute to Docs

- [Learn more](#) about how to get involved.

- [Edit this page](#) on GitHub to fix an error or make an improvement.

- to let us know how we can improve Docs.

Learn C# on Codecademy

`Career path`

Computer Science

Looking for an introduction to the theory behind programming? Master Python while learning data structures, algorithms, and more!

Includes **6 Courses**

With **Professional Certification**

Beginner Friendly   **75** hours

`Free course`

Learn C#

Learn Microsoft's popular C# programming language, used to make websites, mobile apps, video games, VR, and more.

Beginner Friendly   **23** hours

↑ Back to top