



آرایه ها در سی شارپ ، مشابه بسیاری از [انواع زبان های برنامه نویسی](#) ، یک ساختمان داده بنیادی و مهم به شمار می روند. دلیل اهمیت آرایه این است که می توان از آن برای جمع آوری و دستکاری گروه هایی از متغیرهای مرتبط بهره مند شد. در این مقاله، به بررسی کامل آرایه ها در C# می پردازیم و انواع آن را به همراه شیوه استفاده، تشریح خواهیم کرد.

### سی شارپ چیست؟

سی شارپ (C#) یک [زبان برنامه نویسی شی گرا](#) (Object Oriented)، مدرن و [Typed-Safe](#) است که توسط شرکت مایکروسافت توسعه یافته است. سی شارپ زبانی است که به طور گسترده برای توسعه [Windows Application ها](#)، بازی های ایجاد شده با [یونیتی](#) (Unity)، وب سرویس ها و موارد دیگر استفاده می شود. این زبان برنامه نویسی به دلیل استحکام و قدرتمند بودن آن، مورد توجه قرار گرفته و یکی از زبان های اصلی پشتیبانی شده در [فریمورک دات نت](#) و [.NET Core](#) است.

### آرایه ها در سی شارپ چیست؟

آرایه ها در سی شارپ ، مجموعه ای از عناصر هستند که می توان آن ها را با استفاده از اندیس گذاری (Indexing) شناسایی کرد. آرایه ها به منظور ذخیره سازی چند مقدار در یک متغیر واحد استفاده می شوند و به شما این امکان را می دهند تا داده ها را به صورت کارآمد مدیریت کرده و به آن ها دسترسی داشته باشید. نوع داده (Data Type) یک آرایه در سی شارپ ، می تواند هر دیتاتایپ معتبری (Valid) از C# باشد؛ توجه کنید که همه عناصر موجود در یک آرایه، باید از یک نوع باشند.



## کاربرد آرایه ها در سی شارپ چیست؟

آرایه ها متنوع هستند و به تعدد در برنامه نویسی به کار می روند. به صورت کلی، موارد زیر مهم ترین کاربردهای آرایه به شمار می روند:

- **ذخیره و بازیابی داده:** آرایه ها روشی ساختارمند برای ذخیره سازی و دسترسی به داده ها را فراهم می کنند.
- **بهبود عملکرد:** در مقایسه با سایر **ساختمان های داده** (Data Structures) مانند لیست، آرایه ها در پردازش حجم عظیمی از داده ها، عملکرد کارآمدتری دارند. این موضوع به طور ویژه در زمان کار با **داده های Primitive** مشهود است.
- **مدیریت چندین متغیر:** به جای اینکه چندین متغیر مجزا را به صورت تک به تک تعریف کنید، می توانید از یک آرایه برای گروه بندی موارد مشابه تحت یک متغیر واحد استفاده کنید.
- **پیاده سازی الگوریتم:** بسیاری از الگوریتم ها برای آن که بتوانند کار کنند، به آرایه ها در سی شارپ نیاز دارند. به عنوان مثال، هنگام کار با الگوریتم های مرتبط با تکنیک های مرتب سازی (Sorting) و جستجو (Search)، آرایه کاربرد خود را نشان می دهد.
- **عملیات ماتریس:** آرایه ها می توانند ماتریس ها را برای محاسبات ریاضیاتی نشان دهند.
- **ذخیره سازی بافر:** می توان از آرایه ها در سی شارپ به عنوان بافر در عملیات I/O (ورودی/خروجی) استفاده کرده و به کمک آن، به طور موقت داده ها را درحین فرآیندهای خواندن و نوشتن، نگهداری کرد.
- **پارامترها و نوع های بازگشتی:** آرایه ها به طور معمول، به عنوان پارامتر یا Return Type در متدهایی استفاده می شوند که در آن ها نیاز به ارسال یا برگرداندن چندین مقدار باشد.

- هم‌کنش‌پذیری با سایر سیستم‌ها و API ها: هنگام فراخوانی API های اکسترنال، به‌ویژه در [برنامه نویسی سیستمی](#)، که در آن چیدمان دقیق حافظه نیاز است، آرایه‌ها می‌توانند کلیدی محسوب شوند.
- شمارش (Enumeration) و تکرار (Iteration): می‌توان به راحتی آرایه‌ها را پیمایش کرد. این ویژگی، آن‌ها را برای اموری ایده‌آل می‌کند که به شمارش بر روی مجموعه آیت‌ها نیاز دارید.

## انواع آرایه ها در سی شارپ

درک انواع مختلف آرایه ها در سی شارپ به شما کمک می‌کند تا به طور مؤثر از ساختار مناسب استفاده کنید. در این بخش، به معرفی نوع‌های اصلی آرایه ها در C# می‌پردازیم.

- **آرایه های تک بعدی (Single-Dimensional):** آرایه های تک بعدی یا یک بعدی، ساده‌ترین نوع آرایه ها در سی شارپ هستند که به هر عنصر از طریق یک اندیس واحد می‌توان دسترسی پیدا کرد. این نوع آرایه‌ها نشان‌دهنده مجموعه‌ای Linear از عناصر هستند.
- **آرایه های چندبعدی (Multi-Dimensional):** این آرایه ها در سی شارپ تحت عنوان [Rectangular Arrays](#) نیز شناخته می‌شوند و دارای دو یا چند بعد هستند که از آن‌ها به منظور نمایش یک جدول یا شبکه‌ای از مقادیر استفاده می‌شود. این آرایه‌ها به طور ویژه در سناریوهایی مفید به شمار می‌روند که نیاز به ذخیره‌سازی داده‌ها در قالبی ساختاریافته مانند ماتریس است.
- **آرایه‌های دنداندار (Jagged Arrays):** این نوع، آرایه‌هایی از آرایه‌ها هستند؛ این یعنی که هر عنصر در آرایه اصلی، خود یک آرایه است. نکته جالب در مورد آرایه‌های Jagged این است که آرایه‌های درونی هر یک می‌توانند طول متفاوتی داشته باشند. به همین دلیل، آرایه‌های دنداندار در سناریوهای خاص، انعطاف‌پذیرتر از آرایه‌های چندبعدی هستند.

## نحوه کار با آرایه ها در سی شارپ

برای آن که شما بتوانید درک صحیحی از چگونگی کار با آرایه ها به دست آورید، توجه به اقدامات این بخش از مطلب ضروری است. توجه کنید که در این بخش، هدف بررسی نوع تک بعدی از آرایه ها در سی شارپ است.

## اعلان آرایه (Array Declaration)

برای استفاده از آرایه ها در سی شارپ، ابتدا باید آن‌ها را اعلان کنید. یک مثال ساده از نحوه [اعلان یک آرایه](#) در C# آورده شده است:

```
int[] myArray;
```

## تعیین عناصر و مقداردهی آرایه در سی شارپ

قطعه کد مذکور، یک آرایه به نام myArray اعلان می‌کند که اعداد صحیح را در خود ذخیره خواهد کرد. برای مقداردهی اولیه آن، مشخص کنید که چه تعداد عناصری را نگهداری خواهد کرد:

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

اکنون MyArray یک آرایه است که امکان نگهداری ۵ عدد صحیح را دارد. درحقیقت، با استفاده از کلمه کلیدی new و تعیین تعداد یا همان سائز آرایه، مقداردهی آن آغاز می‌شود.

در زمان اعلان، می‌توان آرایه ها در سی شارپ را به چندین روش مقداردهی اولیه (Initialization) کرد:

```
int[] myArray = new int[5]; // An array of five integers, initialized to zero
```

اگر مقداردهی مشابه فوق باشد، مقدار اولیه عناصر آرایه صفر خواهد بود. اما اگر مقداردهی اولیه به شکل زیر انجام شود، آنگاه شما یک آرایه را همراه با مقادیر اولیه مشخصی اعلان کرده‌اید:

```
int[] myArray = {1, 2, 3, 4, 5}; // An array declared with initial values
```

نسخه طولانی‌تر آن به شکل زیر است:

```
int[] myArray = new int[] { 1, 2, 3, 4, 5 };
```

### دسترسی به عناصر آرایه

برای دسترسی به آرایه ها در سی شارپ باید از طریق اندیس یا همان ایندکس آن‌ها اقدام کرد که از عدد ۰ شروع می‌شود. به‌منظور دسترسی یا تغییر اولین عنصر آرایه myArray، به‌صورت زیر عمل کنید:

```
int firstValue = myArray[0]; // Gets the first element
myArray[0] = 10; // Sets the first element to 10
```

### پیمایش (Iteration) آرایه ها

برای انجام عملیات روی تک‌به‌تک عناصر موجود در آرایه ، می‌توانید از **حلقه** (Loop) کمک بگیرید. حلقه For یکی از روش‌های رایج برای پیمایش آرایه ها است. در چنین شرایطی، یک حلقه for به‌صورت زیر نیاز است:

```
for (int i = 0; i < myArray.Length; i++)
{
    Console.WriteLine(myArray[i]);
}
```

این حلقه for به‌طور مکرر اجرا شده و تا زمانی که به انتهای آرایه برسد، مقدار هر عنصر در آرایه را در خروجی چاپ می‌کند. علاوه‌بر For، شما می‌توانید مشابه زیر، از حلقه foreach برای پیمایش مستقیم تک تک عناصر آرایه ها در سی شارپ استفاده کنید:

```
foreach (int value in myArray)
{
    Console.WriteLine(value);
}
```



### نحوه استفاده از آرایه چندبعدی در سی شارپ

استفاده مؤثر از آرایه های چندبعدی در سی شارپ ، مستلزم درک نحوه اعلان، مقداردهی اولیه و دسترسی به عناصر آنها است. آرایه های چندبعدی به طور ویژه برای اپلیکیشن هایی مناسب هستند که به ساختاری شبیه ماتریس نیاز دارند. به عنوان مثال، می توان از آنها در مدیریت داده ها در جداول، شبکه ای از داده ها یا محاسبات ریاضی بهره مند شد. در ادامه، نکاتی برای استفاده از آرایه های چندبعدی ، به طور خاص با تمرکز بر آرایه های دو بعدی، شرح داده خواهند شد.

### اعلان آرایه دو بعدی در C#

برای اعلان کردن یک آرایه دو بعدی در C# ، نوع داده هایی که در آن قرار خواهند گرفت را تعیین کنید و سپس از دو مجموعه علامت [ ] استفاده کنید که هرکدام شامل یک کاما ( , ) برای نشان دادن بُعد هستند.

```
int[,] matrix;
```

### مقداردهی اولیه آرایه های دو بعدی

در سی شارپ می توانید یک آرایه دو بعدی را در زمان اعلام آن، مقداردهی اولیه کنید. این روش زمانی مناسب است که از قبل، ابعاد دقیق داده ها را می دانید. نحوه انجام این کار به صورت زیر است:

```
int[,] matrix = new int[3, 4];
```

در این جا یک ماتریس ۳ در ۴ از اعداد صحیح ایجاد کردیم. توجه کنید که اگر آرایه را به روش فوق مقداردهی اولیه می‌کنید، عناصر به‌طور خودکار با مقادیر پیش‌فرض (معمولاً ۰ برای اعداد صحیح) پر خواهند شد.

راه دیگر این است که شما آرایه دوبعدی خود را با مقادیر مدنظر مقداردهی اولیه کنید. برای درک بهتر، به آرایه زیر توجه کنید:

```
int[, ] matrix = {
    { ۱, ۲, ۳, ۴ },
    { ۵, ۶, ۷, ۸ },
    { ۹, ۱۰, ۱۱, ۱۲ }
};
```

این کد یک ماتریس ۳ در ۴ را با مقادیر ۱ تا ۱۲ مقداردهی اولیه می‌کند. هر { } درونی، نماینده یک سطر از ماتریس است.

### دسترسی و ویرایش آرایه دوبعدی در سی شارپ

در یک آرایه دوبعدی در C#، دسترسی به عناصر با تعیین اندیس سطر (Row Index) و اندیس ستون (Column Index) آن‌ها (به‌صورت جداگانه) انجام می‌شود. به‌عنوان مثال، برای دسترسی به عنصر دوم، از سطر یک آرایه از این طریق اقدام کنید:

```
int element = matrix[0, 1];
```

برای ویرایش مقدار عنصر دوم از سطر یک آرایه و تغییر آن به عدد ۲۰، کد زیر نیاز است:

```
matrix[0, 1] = 20;
```

### پیمایش (Iteration) یک آرایه دو بعدی

برای انجام عملیات روی تک تک عناصر موجود در یک آرایه دوبعدی، می‌توان از [حلقه های تودرتو](#) (Nested Loops) بهره برد. حلقه بیرونی روی سطرهای آرایه و حلقه درونی روی ستون‌های آن Iterate خواهد شد.

```
for (int i = 0; i < matrix.GetLength(0); i++) {
    for (int j = 0; j < matrix.GetLength(1); j++) {
        Console.Write(matrix[i, j] + " ");
    }
    Console.WriteLine();
}
```

GetLength(dimension) متدی است که برای به‌دست آوردن تعداد عناصر در یک بعد خاص از یک آرایه چندبعدی به کار می‌رود.

## نحوه استفاده از آرایه Jagged در سی شارپ

آرایه های Jagged ، یا اصطلاحاً دنداندار، درحقیقت آرایه‌هایی از آرایه‌های دیگر هستند. این ساختار به شما اجازه می‌دهد داده‌هایی را ذخیره‌سازی کنید که ماهیت‌های غیریکسانی دارند.

### اعلان آرایه Jagged

برای اعلان یک آرایه ناهمگن یا همان دنداندار، لازم است تعیین شود که این آرایه، آرایه‌ای از آرایه‌ها است. در اینجا Syntax مربوط به اعلان یک آرایه دنداندار بررسی می‌شود که امکان نگهداری آرایه‌هایی متشکل از اعداد صحیح را دارد:

```
int[][] jaggedArray;
```

### مقداردهی اولیه آرایه دنداندار در سی شارپ

برخلاف آرایه های چندبعدی ، در آرایه دنداندار هر زیرآرایه می‌تواند اندازه متفاوتی داشته و باید به‌صورت جداگانه مقداردهی اولیه شود. به کد زیر توجه کنید تا بهتر مفهوم آن را درک کنید:

```
jaggedArray = new int[3][]; // Jagged array with 3 rows

jaggedArray[0] = new int[5]; // First row has 5 columns
jaggedArray[1] = new int[3]; // Second row has 3 columns
jaggedArray[2] = new int[4]; // Third row has 4 columns
```

همچنین می‌توانید آرایه دنداندار را مشابه زیر و به‌صورت مستقیم مقداردهی اولیه کنید:

```
jaggedArray = new int[][]
{
    new int[] { 1, 2, 3, 4, 5 },
    new int[] { 2, 3, 4 },
    new int[] { 5, 6, 7, 8 }
};
```

### دسترسی به عناصر آرایه دنداندار

برای دسترسی یا تغییر عناصر در یک آرایه Jagged، باید از اندیس‌های سطر و ستون به‌صورت ترکیبی استفاده کنید. با این دیدگاه، کد زیر را بررسی کنید:

```
int firstElement = jaggedArray[0][0]; // Accesses the first element of the
first row
jaggedArray[1][2] = 10; // Sets the third element of the second row to 10
```

## پیمایش (Iteration) روی آرایه دنداندار در سی شارپ

برای پیمایش روی تک تک عناصر موجود در این نوع از آرایه ها در سی شارپ ، می‌توان از حلقه‌های تودرتو کمک گرفت. حلقه بیرونی روی تمام زیرآرایه‌های (سطرهای) آرایه و حلقه درونی روی عناصر هر زیرآرایه (ستون‌ها) پیمایش می‌کند. توجه کنید که ممکن است طول هر زیرآرایه با دیگری متفاوت باشد. کد زیر به شما کمک می‌کند تا این مباحث را بهتر متوجه شوید:

```
for (int i = 0; i < jaggedArray.Length; i++)
{
    for (int j = 0; j < jaggedArray[i].Length; j++)
    {
        Console.WriteLine(jaggedArray[i][j] + " ");
    }
    Console.WriteLine();
}
```

## کلاس آرایه

**کلاس Array** به‌عنوان کلاس پایه برای تمامی آرایه ها در سی شارپ عمل می‌کند. این کلاس انتزاعی (Abstract) است؛ به این معنی که نمی‌توانید به‌طور مستقیم Instance خاصی از آن ایجاد کنید، اما می‌توانید از **متدهای استاتیک** (Static Methods) آن برای انجام عملیات مختلف روی هر آرایه بهره‌مند شوید. این متدها در اموری مانند مرتب‌سازی، جستجو و معکوس‌کردن عناصر درون آرایه‌ها مورد استفاده قرار می‌گیرند.

## متدهای اصلی کلاس Array

در این بخش، به معرفی تعدادی از پرکاربردترین متدهای کلاس Array در C# می‌پردازیم:

- **Length**: این متد، تعداد کل عناصر موجود در تمام ابعاد یک آرایه را بازمی‌گرداند و برای به‌دست آوردن تعداد عناصر در یک آرایه و پیمایش روی آن بسیار کاربردی است.
- **Sort**: این متد با استفاده از **الگوریتم QuickSort** ، تمامی عناصر آرایه تک بعدی را مرتب‌سازی می‌کند.
- **Reverse**: با استفاده از آن، می‌توانید ترتیب عناصر آرایه تک بعدی را معکوس کنید.

در ادامه مطلب، سایر امور مربوط به Data Manipulation از طریق آرایه بررسی می‌شوند.





### نحوه جستجو در آرایه سی شارپ

برای جستجوی یک عنصر خاص در یک آرایه، متد IndexOf از کلاس Array را استفاده کنید. این متد اولین موقعیت یا همان ایندکسی را پیدا می‌کند که عنصر موردنظر در آن قرار دارد. به مثال زیر توجه کنید:

```
int[] myArray = { 1, 2, 3, 4, 5 };
int index = Array.IndexOf(myArray, 3);
Console.WriteLine("Index of 3: " + index); // Outputs the index of the
number 3 in the array
```

### نمایش عناصر آرایه در سی شارپ

برای نمایش کل عناصر موجود در یک آرایه در سی شارپ، از حلقه‌های for یا foreach استفاده کنید. نحوه انجام این کار با حلقه foreach به صورت زیر است:

```
foreach (int element in myArray)
{
    Console.WriteLine(element);
}
```

### مرتب سازی آرایه ها در سی شارپ

شما می‌توانید از دستور Array.Sort به هدف مرتب‌سازی صعودی یک آرایه در سی شارپ بهره‌مند شوید. این دستور، آرایه اصلی را تغییر می‌دهد:

```
Array.Sort(myArray);
Console.WriteLine("Sorted Array: ");
foreach (int item in myArray)
{
    Console.WriteLine(item);
}
```

اگر بخواهید آرایه را به صورت نزولی مرتب کنید، لازم است پس از مرتب‌سازی آرایه، آن را مشابه زیر معکوس کنید:

```
Array.Sort(myArray);
Array.Reverse(myArray); // Reverses the sorted array to make it
descending
Console.WriteLine("Sorted in descending order: ");
foreach (int item in myArray)
{
    Console.WriteLine(item);
}
```

### حذف عناصر تکراری آرایه در سی شارپ

برای حذف عناصر تکراری از یک آرایه در سی شارپ، کارآمدترین رویکرد، به‌کارگیری یک `HashSet<T>` است. مشابه زیر، این ساختار داده به‌طور اتوماتیک موارد تکراری را حذف می‌کند و تنها به آیتم‌های منحصر به فرد (Unique) اجازه می‌دهد تا در آن قرار داشته باشند:

```
int[] myArray = { 1, 2, 2, 3, 4, 4, 5 };
HashSet<int> uniqueItems = new HashSet<int>(myArray);
int[] uniqueArray = uniqueItems.ToArray();

Console.WriteLine("Array without duplicates: ");
foreach (int item in uniqueArray)
{
    Console.WriteLine(item);
}
```

### سخن پایانی: آرایه ها در سی شارپ

در این مقاله، به بررسی انواع آرایه ها در سی شارپ پرداختیم و نحوه کار با آن‌ها را به‌همراه قطعه کد بیان کردیم. علاوه‌براین، کلاس `Array` و تعدادی از متدهای آن معرفی شدند. در مجموع، آرایه‌ها بخش جدایی‌ناپذیر برنامه‌نویسی هستند و به‌عنوان ابزارهای قدرتمندی عمل می‌کنند که مدیریت داده‌ها را در توسعه نرم‌افزار تسهیل می‌بخشند. شما می‌توانید برای دریافت منابع آموزشی تکمیلی مربوط به برنامه‌نویسی با سی شارپ، درخواست خود را در کامنت‌ها اعلام کنید تا ما به آن رسیدگی کنیم.