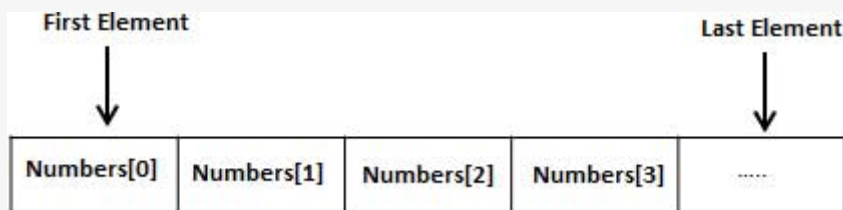


C# 数组 (Array)

数组是一个存储相同类型元素的固定大小的顺序集合。数组是用来存储数据的集合，通常认为数组是一个同一类型变量的集合。

声明数组变量并不是声明 number0、number1、...、number99 一个个单独的变量，而是声明一个就像 numbers 这样的变量，然后使用 numbers[0]、numbers[1]、...、numbers[99] 来表示一个个单独的变量。数组中某个指定的元素是通过索引来访问的。

所有的数组都是由连续的内存位置组成的。最低的地址对应第一个元素，最高的地址对应最后一个元素。



声明数组

在 C# 中声明一个数组，您可以使用下面的语法：

```
datatype[] arrayName;
```

其中，

- *datatype* 用于指定被存储在数组中的元素的类型。
- *[]* 指定数组的秩（维度）。秩指定数组的大小。
- *arrayName* 指定数组的名称。

例如：

```
double[] balance;
```

初始化数组

声明一个数组不会在内存中初始化数组。当初始化数组变量时，您可以赋值给数组。

数组是一个引用类型，所以您需要使用 **new** 关键字来创建数组的实例。

例如：

```
double[] balance = new double[10];
```

赋值给数组

您可以通过使用索引号赋值给一个单独的数组元素，比如：

```
double[] balance = new double[10];  
balance[0] = 4500.0;
```

您可以在声明数组的同时给数组赋值，比如：

```
double[] balance = { 2340.0, 4523.69, 3421.0};
```

您也可以创建并初始化一个数组，比如：

```
int [] marks = new int[5] { 99, 98, 92, 97, 95};
```

在上述情况下，你也可以省略数组的大小，比如：

```
int [] marks = new int[] { 99, 98, 92, 97, 95};
```

您也可以赋值一个数组变量到另一个目标数组变量中。在这种情况下，目标和源会指向相同的内存位置：

```
int [] marks = new int[] { 99, 98, 92, 97, 95};  
int[] score = marks;
```

当您创建一个数组时，C# 编译器会根据数组类型隐式初始化每个数组元素为一个默认值。例如，int 数组的所有元素都会被初始化为 0。

访问数组元素

元素是通过带索引的数组名称来访问的。这是通过把元素的索引放置在数组名称后的方括号中来实现的。例如：

```
double salary = balance[9];
```

下面是一个实例，使用上面提到的三个概念，即声明、赋值、访问数组：

实例

```
using System;  
namespace ArrayApplication  
{  
    class MyArray  
    {  
        static void Main(string[] args)  
        {  
            int [] n = new int[10]; /* n 是一个带有 10 个整数的数组 */  
        }  
    }  
}
```

```
int i,j;

/* 初始化数组 n 中的元素 */
for ( i = 0; i < 10; i++ )
{
    n[ i ] = i + 100;
}

/* 输出每个数组元素的值 */
for ( j = 0; j < 10; j++ )
{
    Console.WriteLine("Element[{0}] = {1}", j, n[j]);
}
Console.ReadKey();
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
Element[0] = 100
Element[1] = 101
Element[2] = 102
Element[3] = 103
Element[4] = 104
Element[5] = 105
Element[6] = 106
Element[7] = 107
Element[8] = 108
Element[9] = 109
```

使用 *foreach* 循环

在前面的实例中，我们使用一个 `for` 循环来访问每个数组元素。您也可以使用一个 **`foreach`** 语句来遍历数组。

实例

```
using System;

namespace ArrayApplication
{
    class MyArray
    {
        static void Main(string[] args)
        {
            int [] n = new int[10]; /* n 是一个带有 10 个整数的数组 */

            /* 初始化数组 n 中的元素 */
            for ( int i = 0; i < 10; i++ )
            {
```

```
        n[i] = i + 100;
    }

    /* 输出每个数组元素的值 */
    foreach (int j in n )
    {
        int i = j-100;
        Console.WriteLine("Element[{0}] = {1}", i, j);
    }
    Console.ReadKey();
}
}
```

当上面的代码被编译和执行时，它会产生下列结果：

```
Element[0] = 100
Element[1] = 101
Element[2] = 102
Element[3] = 103
Element[4] = 104
Element[5] = 105
Element[6] = 106
Element[7] = 107
Element[8] = 108
Element[9] = 109
```

C# 数组细节

在 C# 中，数组是非常重要的，且需要了解更多的细节。下面列出了 C# 程序员必须清楚的一些与数组相关的重要概念：

概念	描述
多维数组	C# 支持多维数组。多维数组最简单的形式是二维数组。
交错数组	C# 支持交错数组，即数组的数组。
传递数组给函数	您可以通过指定不带索引的数组名称来给函数传递一个指向数组的指针。
参数数组	这通常用于传递未知数量的参数给函数。
Array 类	在 System 命名空间中定义，是所有数组的基类，并提供了各种用于数组的属性和方法。

 点我分享笔记