

## 第6章 配列



同じデータ型のデータの集まりを配列として扱うことができます。

配列は、次のように宣言します。

**データ型 [ ] 配列名 ;**

これだけでは、変数はメモリ上のどのオブジェクトも参照していません。

**配列名 = new データ型 [要素数] ;**

new演算子を使い、指定されたデータ型を要素数分だけ格納できる配列の実体がメモリ上に確保され、その参照が「配列名」に保存されます。

このことより、配列は「参照型」です。

```
int[] myarray;  
myarray = new int[10];
```

上のようにすると、int型の配列で、要素数10の配列が準備されます。

配列の要素にアクセスするには

**配列名 [添え字]**

で行います。添え字は0から始まります。上の例ではmyarray[0], ..myarray[9]までのint型の要素があります。

これらの要素は、int型の変数と同じに扱えます。

また、次のように配列を宣言することもできます。

**データ型 [ ] 配列名 = new データ型 [要素数] ;**

配列の初期化は、配列の要素の一つずつ値を代入していてもよいのですが、宣言と同時に初期化を行うことも可能です。

**データ型 [ ] 配列名 = new データ型 [要素数] {要素1, 要素2, ...} ;**

要素数を省略して

**データ型 [ ] 配列名 = new データ型 [ ] {要素1, 要素2, ...} ;**

のように書くこともできます。

では、簡単なサンプルのプログラムを見てみましょう。

ここでは、単に配列を宣言して初期化しているだけです。

```
// array01.cs  
  
using System;  
  
class array01  
{  
    public static void Main()  
    {  
        // 要素の一つずつ値を代入  
        int[] myarray1 = new int[3];  
        myarray1[0] = 10;  
        myarray1[1] = 20;  
        myarray1[2] = 30;  
  
        // 宣言と同時に初期化  
        int[] myarray2 = new int[3]{10, 20, 30};  
  
        // 要素数を省略することも可能  
        int[] myarray3 = new int[] {10, 20, 30};  
  
        // 別な方法  
        int[] myarray4;  
        myarray4 = new int[] {10, 20, 30};  
    }  
}
```

さて、2次元以上の多次元配列も利用することができます。2次元配列は、次のように宣言します。

**データ型[,] 配列名 = new データ型[要素数1, 要素数2];**

初期化の方法も1次元配列に準じます。

配列名.Lengthでその配列のすべての要素の個数がわかります。(ArrayクラスのLengthプロパティ。後の章で解説)

配列名.Rank(これもArrayクラスのRankプロパティ)でその配列の次元がわかります。

**// array02.cs**

**using System;**

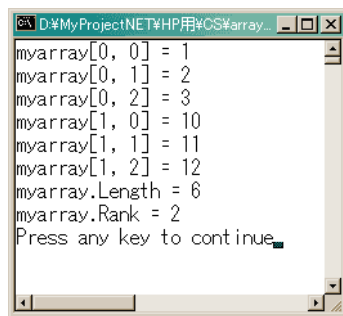
**class array02**

```
{
    public static void Main()
    {
        int[,] myarray = new int[2, 3]{{1, 2, 3}, {10, 11, 12}};

        Console.WriteLine("myarray[0, 0] = {0}", myarray[0, 0]);
        Console.WriteLine("myarray[0, 1] = {0}", myarray[0, 1]);
        Console.WriteLine("myarray[0, 2] = {0}", myarray[0, 2]);
        Console.WriteLine("myarray[1, 0] = {0}", myarray[1, 0]);
        Console.WriteLine("myarray[1, 1] = {0}", myarray[1, 1]);
        Console.WriteLine("myarray[1, 2] = {0}", myarray[1, 2]);

        Console.WriteLine("myarray.Length = {0}", myarray.Length);
        Console.WriteLine("myarray.Rank = {0}", myarray.Rank);
    }
}
```

では、結果を見てみましょう。



---

[\[C# Index\]](#) [\[総合Index\]](#) [\[Previous Chapter\]](#) [\[Next Chapter\]](#)

Update 08/Aug/2006 By Y.Kumei

当ホーム・ページの一部または全部を無断で複写、複製、転載あるいはコンピュータ等のファイルに保存することを禁じます。