

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

آموزش سی شارپ - بخش ۳۹ ← حلقه در آرایه‌ها

• مقدمه

در بخش پیشین از آموزش زبان برنامه‌نویسی سی شارپ، با موضوع و مفهوم آرایه‌ها تا حدی آشنا شدید. آرایه‌ها همراه با ساختار و مفهوم خود، با حلقه‌ها همراه است. یعنی در واقع شما با ایجاد یک حلقه، می‌توانید در کل آرایه چرخه بزنید و در نقطه‌ی خاصی نیز توقف داشته باشید.

در این بخش، ما کمی با مثال در رابطه با این موضوع صحبت خواهیم کرد؛ پس با ما همراه باشید.

• حلقه در آرایه‌ها

اگر پروژه‌ی از پیش ساخته‌ای از قبل ندارید، یک پروژه‌ی جدیدی را آغاز کنید و در در form ایجاد شده، یک button و یک listBox از بخش toolbox اضافه کنید. بر روی button دابل-کلیک کنید تا پنجره مربوط به کدنویسی باز شود. برای شروع، کد زیر را اضافه کنید تا ابتدا اگر مواردی در listBox موجود باشد، آن‌ها را محو کند:

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();
}
```

برای دومین و سومین خط از کدنویسی، موارد زیر را اضافه کنید:

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();

    int[] lottery_numbers;
    lottery_numbers = new int[4];
}
```

حال برای هر یک از بخش‌های موجود در آرایه، یک مقداردهی اولیه انجام دهید:

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();

    int[] lottery_numbers;
    lottery_numbers = new int[4];

    lottery_numbers[0] = 1;
    lottery_numbers[1] = 2;
    lottery_numbers[2] = 3;
    lottery_numbers[3] = 4;
}
```

برای نمایش تک‌تک مقادیر موجود در آرایه در `listbox`، شما اینکار را خواهید کرد:

```
listBox1.Items.Add(lottery_numbers[0]);
```

```
listBox1.Items.Add(lottery_numbers[1]);
```

```
listBox1.Items.Add(lottery_numbers[2]);
```

```
listBox1.Items.Add(lottery_numbers[3]);
```

و برای اینکه مقادیر را از آرایه‌ها دریافت کنیم، ما فقط نام آرایه و محل مورد نظر خودمان را می‌نویسیم تا مقدار مورد نظر برگردانده شود. هماهنگی زیر:

```
lottery_numbers[0];
```

این کارها برای نمایش مقادیر داخل آرایه در `listbox` کفایت می‌کند؛ دقت کنید که کدهایتان می‌بایست مشابه زیر شده

باشد:

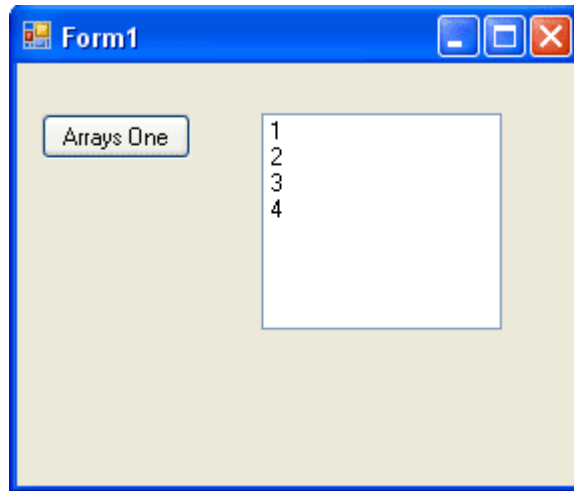
```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();

    int[] lottery_numbers;
    lottery_numbers = new int[4];

    lottery_numbers[0] = 1;
    lottery_numbers[1] = 2;
    lottery_numbers[2] = 3;
    lottery_numbers[3] = 4;

    listBox1.Items.Add(lottery_numbers[0]);
    listBox1.Items.Add(lottery_numbers[1]);
    listBox1.Items.Add(lottery_numbers[2]);
    listBox1.Items.Add(lottery_numbers[3]);
}
```

اکنون کافیت که یکبار نرم افزار خود را اجرا کرده تا نتیجه را مشاهده کنید:



خوب؛ برنامه خود را ببینید. اگر دقت کنید، متوجه خواهید شد که سایز آرایه ما بسیار کوچک است و تنها ۴ شماره را در خود ذخیره می کند؛ فرض کنید اگر آرایه‌ی ما ۱۰۰ تایی باشد چه اتفاقی رخ خواهد داد؟ پس می بایست ما با استفاده از یک حلقه، بتوانیم (برای مثال) ۱۰۰ بار یک آرایه را پیمایش کنیم تا تمامی مقادیر داخل آن را بدست بیاوریم و در listBox نمایش دهیم.

به کد زیر دقت کنید:

```
for (int i = 0; i != (lottery_numbers.Length); i++)  
{  
    listBox1.Items.Add(lottery_number[i]);  
}
```

کد بالا بدین شکل است؛ چون آرایه‌ها از مکان ۰ (صفر) شروع می شوند، پس ابتدای حلقه را صفر در نظر می گیریم ($i = 0$)

(0)

در قدم بعدی، می بایست به اندازی طول (length) آرایه، در آن پیمایش کنیم؛ پس کد زیر بدین معنی است که تا زمانی که به انتهای آرایه نرسیدیم، حلقه را ادامه بده:

```
i != (lottery_numbers.Length);
```

و در نهایت، در هر بار با استفاده از کد $i++$ ، یک خانه در آرایه جلوتر می رویم. کد مربوط به بدنه نیز کاملاً مشخص می باشد و نیازی به توضیح ندارد.

حال کدهای قبلی خودتان را پاک کنید و سعی کنید کدتان شبیه به کد زیر شود:

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();

    int[] lottery_numbers;
    lottery_numbers = new int[4];

    lottery_numbers[0] = 1;
    lottery_numbers[1] = 2;
    lottery_numbers[2] = 3;
    lottery_numbers[3] = 4;

    for (int i = 0; i != (lottery_numbers.Length); i++)
    {
        listBox1.Items.Add(lottery_numbers[i]);
    }
}
```

حال برنامه خود را اجرا کنید و مجدداً نتیجه را ببینید؛ متوجه خواهید شد که برنامه شما اینبار با کدهای خیلی کمتر، به درستی کار می‌کند. به عکس‌های زیر نیز دقت کنید تا نحوه کار حلقه را کمی بهتر متوجه شوید:

```
private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();

    int[] lottery_numbers;
    lottery_numbers = new int[4];

    lottery_numbers[0] = 1;
    lottery_numbers[1] = 2;
    lottery_numbers[2] = 3;
    lottery_numbers[3] = 4;

    for (int i = 0; i != (lottery_numbers.Length); i++)
    {
        listBox1.Items.Add(lottery_numbers[i]);
    }
}
```

Assign a value of zero to the loop variable called *i*

```

private void button1_Click(object sender, EventArgs e)
{
    listBox1.Items.Clear();

    int[] lottery_numbers;
    lottery_numbers = new int[4];

    lottery_numbers[0] = 1;
    lottery_numbers[1] = 2;
    lottery_numbers[2] = 3;
    lottery_numbers[3] = 4;

    for (int i = 0; i != (lottery_numbers.Length); i++)
    {
        listBox1.Items.Add(lottery_numbers[i]);
    }
}

```

Use the value in the **i** variable as the index number of the array

Loop	Array
i = 0	<u>lottery_numbers[0]</u>
i = 1	<u>lottery_numbers[1]</u>
i = 2	<u>lottery_numbers[2]</u>
i = 3	<u>lottery_numbers[3]</u>

شما می‌توانید با استفاده از خود حلقه، حتی نسبت به مقداردهی آرایه نیز اقدام کنید؛ انجام این کار را به عهده شما قرار می‌دهیم، تنها یک نمونه کد در اینجا برای شما قرار می‌دهیم:

```

for (int i = 0; i != (lottery_numbers.Length); i++)
{
    lottery_numbers[i] = i + 1;
    listBox1.Items.Add(lottery_numbers[i]);
}

```

لینک ثابت مقاله (دسترسی آنلاین): <http://www.gooyait.com/1392/04/17/c-sharp-tutorial-part-39.html>

GOOYAIT.COM – با آرزوی موفقیت