

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ

آموزش سی شارپ - بخش ۱۹ ← حلقه‌ها - قسمت اول

• مقدمه

در بخش آموزشی قبلی، ما در رابطه با عملگرهای منطقی و مقایسه‌ای صحبتی را کردیم و تا جایی که ممکن بود از آن مثال‌هایی را طرح و حل کرده تا نحوه استفاده و کاربرهای آن‌ها را به اندازه‌ی کافی برای شما جا بیفتد.

در این بخش، موضوعی را که مطرح می‌کنیم، موضوع حلقه‌ها هست! احتمالاً خیلی از شما عزیزان (خصوصاً اگر زبان برنامه‌نویسی دیگری را بلد باشید) می‌دانید که حلقه‌ها چه هستند و البته، چقدر کاربردی هم می‌باشند!

• حلقه‌ها (Loops)

حلقه‌ها، که از این به بعد سعی می‌کنیم آن‌ها را با نام Loops صدا بزنیم، یکی دیگر از مهم‌ترین مواردی است که در برنامه‌نویسی بطور بسیار زیادی از آن استفاده می‌شود؛ سی شارپ هم از آن‌ها استثنا نشده است. اما حلقه‌ها در واقع چه هستند؟ حلقه در واقع کاری می‌کند که عملی را بارها و بارها آن را تکرار کنید تا به یک شرطی برسد که متوقف شود و دیگر آن عمل بدلیل عدم صحیح بودن شرط، انجام نگیرد.

برای مثال، شما فرض کنید می‌خواهید اعداد ۱ تا ۱۰ را با یکدیگر جمع کنید. اگر بخواهید با اطلاعاتی که تا کنون داشته‌اید، چنین کاری را بکنید، احتمالاً کاری که خواهید کرد به شکل زیر خواهد بود:

```
int answer;
```

```
answer = 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10;
```

اما می‌بینید که چنین کاری اصلاً با عقل آدمی جور در نمی‌آید؛ خصوصاً اگر تعداد ارقام بیشتر باشد؛ برای مثال ۱ تا ۹۹۹. پس برای اینکه خواسته‌ی ما برآورده شود، ما از Loops استفاده می‌کنیم.

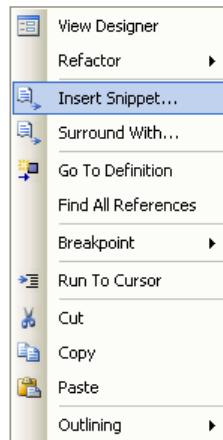
• اولین حلقه: حلقه‌ی For

اولین حلقه‌ای را که شما فرا خواهید گرفت، حلقه For است. حلقه‌های دیگری هم وجود دارند، مانند While و Do-While. اما فعلاً رایج‌ترین و پرکاربردترین حلقه، حلقه‌ی For می‌باشد.

یک پروژه‌ی جدیدی از نوع Windows Application را در Visual Studio شروع کنید. یک button به فرم خود اضافه کنید و وارد بخش کد مربوط به button ها شوید.

در میان دو براکت مربوط به button که قبلاً اضافه کردید، راست کلیک کنید و بر روی Insert Snippet کلیک کنید. همانند تصویر زیر:

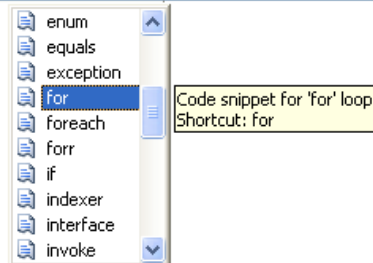
```
private void button1_Click(object sender, EventArgs e)
{
    |
}
```



وقتی که بر روی Insert Snippet کلیک می‌کنید، همانند تصویر زیر لیستی برای شما نمایش داده می‌شود:

```
private void button1_Click(object sender, EventArgs e)
{
    Insert Snippet:
}

```



کمی پایین بیایید و بر روی For دابل-کلیک کنید. یکسری کد برای شما بصورت خودکار ایجاد و درج خواهد شد:

```
private void button1_Click(object sender, EventArgs e)
{
    for (int i = 0; i < length; i++)
    {
    }
}
```

شاید اگر اولین بار شما باشد که به این کد نگاه کرده باشید، کمی برای شما پیچیده به نظر برسد. پس بگذارید قدم به قدم برای شما این کدها را توضیح دهم که ببینید که چقدر ساده است.

ساختار کلی حلقه For به شکل زیر است:

```
for (  
  
    //SOME CODE  
  
)  
{  
  
}
```

پس شما یک For می نویسید و جلوی آن دو پرانتز باز و بسته قرار می دهید. سپس با استفاده از جفت براکت باز و بسته، محدوده‌ی حلقه For را برای سی شارپ مشخص می کنید. چیزی که میان دو پرانتز For قرار می گیرد، در واقع کدی خواهد بود که به سی شارپ بگوید کدهای داخل حلقه For چند بار پشت سر هم اجرا شوند.

به صورت پیشفرض، قالب شرطی که Visual Studio برای شما در بین دو پرانتز قرار داد، به شکل زیر است:

```
int = 0; i < length; i++
```

کد بالا، در واقع از سه بخش که توسط دو Semicolon (;) از هم جدا شده اند، ساخته شده است.

۱. از چه شماره‌ای این حلقه شروع به کار کند؟

۲. چندبار می خواهید که کد اجرا شود؟

۳. مقدار اولیه را پس از اتمام هر حلقه، چطور باید بروز شود؟

این سه بخش، تعریفی از سه بخش بالاست. به عکس‌های زیر دقت کنید. اولین بخش به صورت زیر است:

```
int i = 0; i < length; i++
```

بخش دوم نیز به شکل زیر است:

```
int i = 0; i < length; i++
```

و بخش سوم نیز:

```
int i = 0; i < length; i++
```

بخش اول (شماره ۱) به سی شارپ می گوید که حلقه من از چه شماره ای باید شروع شود:

```
int i = 0;
```

کاری که در بخش اول می کنید، تعریف یک متغیری است که معمولاً آن‌ها را با نام‌های i ، j ، k و ... نامگذاری می کنند. البته هر نامی را می توانید انتخاب کنید؛ به شرطی که از قوانین نامگذاری متغیرها پیروی کند. البته دقت کنید که می توانید متغیر مربوط به حلقه را بیرون از حلقه تعریف کنید. به شکل زیر:

```
int i;
```

```
for (i = 0; i < length; i++)
```

```
{
```

```
}
```

سپس، بعد از تعریف متغیر، در همان جا آن را مقدار دهی می کنیم. مقداری که به آن به صورت پیشفرض داده شده، عدد صفر است.

بخش دوم کد (شماره ۲)، نشان دهنده شرطی است که به سی شارپ می گوید که این حلقه چند بار باید اجرا شود! به صورت پیشفرض کد زیر قرار داده شده است:

```
i < length;
```

اگر مبحث جلسه قبل را به خاطر داشته باشید، این کد می گوید «زمانی که مقدار i از مقدار $length$ کوچک تر باشد.» دقت کنید که جای $length$ باید عدد یا متغیر عددی قرار گیرد؛ پس $length$ یک کلمه کلیدی نیست که خواهید بر اساس آن تصمیمی بگیرید. مثلاً کدمان را به شکل زیر تغییر می دهیم:

```
int length = 101;
```

```
for (int i = 0; i < length; i++)
```

```
{
```

```
}
```

دقت کنید که لازم نیست حتماً نام متغیری که بر اساس آن شرط تعیین می شود، $length$ باشد. نام آن می تواند به دلخواه شما تعیین شود.

بخش سوم کد (شماره ۳)، به سی شارپ می گوید که پس از هر بار که کد داخل حلقه (در صورت برقرار بودن شرط در مرحله قبل) اجرا می شود، چه اتفاقی باید بیفتد. ما در کد بالا اینطور نوشتیم که پس از هرباری که حلقه اجرا می شود، یک مقدار به مقدار متغیر i اضافه کند ($i++$ یعنی اینکه عددی را که i در خود ذخیره می کند را با عدد ۱ جمع کند).

سی شارپ تا زمانی که بخش دوم کد (شماره ۲) شرطش درست باشد، حلقه را در حال اجرا قرار می دهد و پس از پایان هر بار اجرا، یک مقدار به i اضافه می کند.

الان وقت آن است که به صورت کاربردی، استفاده از حلقه و مفهوم آن را متوجه شوید. کدهایتان را دقیقاً به شکل زیر در آورید:

```
private void button1_Click(object sender, EventArgs e)
{
    int answer = 0;

    for (int i = 1; i < 101; i++)
    {
        answer = answer + i;
    }

    MessageBox.Show(answer.ToString());
}
```

کدی که می خواهیم در هر بار در Loop اجرا شود، $answer = answer + i$ است.

ما در کد بالا، در واقع می خواهیم اعداد ۱ تا ۱۰۰ را با یکدیگر جمع کنیم. پس اگر کدهای بالا را اجرا کنیم، دور اولی که حلقه شروع به کار می کند، مقادیر زیر را دارد:

```
answer = answer + i;
0 = 0 + 1
```

پس از اتمام دور اول، دور دوم شروع می شود. این بار مقادیر به شکل زیر است:

```
answer = answer + i;
1 = 1 + 2
```

دور سوم نیز بدین شکل است:

```
answer = answer + i;
3 = 3 + 3
```

و دور چهارم:

`answer = answer + i;`

`6 = 6 + 4`

دقت کنید که چگونه مقدار `i` در هر بار اجرا، تغییر می‌کند. این افزایش مقدار `i` ناشی از بخش سوم کد (شماره ۳) می‌باشد.

کدهایتان را اجرا کنید و بر روی `button` کلیک کنید. مقداری را که برای شما باید نمایش دهد، عدد `۵۰۵۰` است؛ که جمع اعداد از `۱` تا `۱۰۰` می‌باشد.

در بخش بعدی، بیشتر در رابطه با مقدار آغازین حلقه و مقدار پایانی آن صحبت خواهیم کرد.

نویسنده: سید جعفر موسوی (Jeff Mosawy)

منبع مقاله: مجله گویا آی تی

لینک ثابت مقاله (دسترسی آنلاین): <http://www.gooyait.com/1391/11/29/c-sharp-tutorial-part-19.html>

با آرزوی موفقیت – GOOYAIT.COM