

زنگ سی شارپ – قسمت ششم

نوشته‌ی مسعود درویشیان  

[لینک مستقیم این مطلب در وب تارگت](#)

در [قسمت قبل](#) در مورد دیتاتایپ‌های کاراکتر و بولین صحبت شد، در این قسمت یکی از سرفصل‌های بسیار مهم سی شارپ را مورد بحث قرار می‌دهیم.

Program Control Statements

روند کنترلی و منطق برنامه، از مهم‌ترین قسمت‌های رفتار پویای یک زبان برنامه‌نویسی است. برای کنترل قسمت‌هایی از کد، سی شارپ دستورهای مخصوصی را فراهم کرده است تا برنامه‌نویس بتواند کنترل و منطق برنامه را در دست بگیرد. اکثر عبارت‌های شرطی و حلقه‌ای در C# از C و C++ می‌آیند. سه دسته‌بندی برای Program Control Statements وجود دارد:

- Selection Statements
- Iteration Statements
- Jump Statements

بخش Selection Statements شامل if و switch است و همین‌طور که از اسم آن مشخص است، در صورت داشتن شروط لازم عمل selection (انتخاب) را انجام می‌دهد. Iteration Statements شامل حلقه‌های for، while، do-while و foreach هستند و عملیات تکرار (iteration) را انجام می‌دهند. Jump Statements هم شامل break، continue، goto، return و throw است. هر کدام از این موارد مفصل توضیح داده می‌شوند. با توجه به این که اصطلاحات انگلیسی در مقالات زنگ سی شارپ بسیار هستند، به منظور دانستن هریک از اصطلاحات (اگر با زبان انگلیسی آشنایی کافی ندارید) پیشنهاد می‌شود از [Google Translate](#)، [Fast Dictionary](#) یا [Persian Dictionary](#) استفاده کنید.

The if Statement

شما می‌توانید بخشی از یک برنامه را با استفاده از دستور شرطی `if` اجرا کنید. دستور `if` در سی‌شارپ از لحاظ `syntax` با سایر زبان‌های برنامه‌نویسی مثل `C`، `C++` و جاوا مشابه است.

ساده‌ترین شکل دستور `if` به صورت زیر است:

```
if(condition) statement;
```

در این جا `condition` یک عبارت بولی (`true` یا `false`) است. اگر `condition` شامل مقدار `true` باشد، آن‌گاه `statement` اجرا می‌شود. اگر `condition` شامل مقدار `false` باشد در این صورت `statement` کنار زده شده و اجرا نمی‌شود.

به نمونه‌ی زیر دقت کنید:

```
if (10 < 11) Console.WriteLine("10 is less than 11");
```

در این مورد، تا زمانی که ۱۰ از ۱۱ کوچک‌تر باشد، `condition` برابر مقدار `true` است و دستور `Console.WriteLine()` اجرا می‌شود. نمونه‌ی زیر را ببینید:

```
if (10 < 9) Console.WriteLine("this won't be displayed");
```

در این مورد، به دلیل این که ۱۰ از ۹ کوچک‌تر نیست `condition` شامل مقدار `false` می‌شود، بنابراین دستور `Console.WriteLine()` اجرا نمی‌شود. همان‌طور که در [قسمت پنجم](#) ذکر شد عملگرهای مقایسه‌ای دو آیتم را با هم مقایسه می‌کنند، یک عبارت که شامل عملگر مقایسه‌ای است، یک مقدار بولین دارد. از این‌رو می‌توان از عملگرهای مقایسه‌ای برای عبارت شرطی (`condition`) نیز استفاده کرد:

Operator	Description	true Example	false Example
<	کوچک‌تر	3 < 8	8 < 3
>	بزرگ‌تر	4 > 2	2 > 4
==	مساوی	7 == 7	3 == 9
<=	کوچک‌تر مساوی	5 <= 5	8 <= 6
>=	بزرگ‌تر مساوی	7 >= 3	1 >= 2
!=	مخالف (نامساوی)	5 != 6	3 != 3

Comparison operators

برنامه‌ی زیر چگونگی کار با دستور if را نشان می‌دهد:

```
// Demonstrate the if.
using System;
class IfDemo
{
    static void Main()
    {
        int a, b, c;

        a = 2;
        b = 3;

        if (a < b) Console.WriteLine("a is less than b");

        // This won't display anything.
        if (a == b) Console.WriteLine("you won't see this");

        Console.WriteLine();

        c = a - b; // c contains -1

        Console.WriteLine("c contains -1");
        if (c >= 0) Console.WriteLine("c is non-negative");
        if (c < 0) Console.WriteLine("c is negative");

        Console.WriteLine();

        c = b - a; // c now contains 1
        Console.WriteLine("c contains 1");
        if (c >= 0) Console.WriteLine("c is non-negative");
        if (c < 0) Console.WriteLine("c is negative");
    }
}
```

خروجی تولید شده توسط این برنامه:

```
a is less than b

c contains -1
c is negative

c contains 1
c is non-negative
```

همان‌طور که مشاهده می‌کنید با توجه به مقادیر متغیرها محاسباتی انجام می‌شود و سپس با توجه به مقدار بولینی که در قسمت **condition** قرار می‌گیرد، **statement** ها اجرا می‌شود.

نکته: برای کوتاه‌تر شدن خط کد در این برنامه، **namespace** های اضافی را حذف کردیم و تنها از فضای نام **system** که برای مثال ما ضروری است استفاده کردیم. همچنین **args [] string** را از متد **Main()** حذف

کردیم. این‌ها به صورت پیش‌فرض توسط ویژوال استدیو نوشته می‌شوند و ممکن است در مواقعی که به آن‌ها نیازی نیست، حذف‌شان کنیم.

فرم کلی دستور **if** بدین شکل است:

```
if(condition) statement;  
else statement;
```

این در صورتی است که شما برای قسمت‌های **if** یا **else** تنها یک **statement** دارید و نیازی نیست از بلوک (براکت باز و بسته) استفاده کنید. عبارت **else** اختیاری است. همچنین قسمت‌های **if** و **else** می‌توانند در یک بلوک قرار بگیرند. اگر برای هر کدام، یک **statement** داشته باشیم استفاده از بلوک اختیاری است، اما اگر برای هر کدام، بیشتر از دو **statement** داشته باشیم استفاده از بلوک اجباری است و فرم کلی آن به شکل زیر است:

```
if(condition)  
{  
    statement sequence  
}  
else  
{  
    statement sequence  
}
```

در این‌جا منظور از **statement sequence** دنباله‌ای از **statement** ها است. اگر **condition** شامل مقدار **true** باشد، آن‌گاه بلوک **if** اجرا می‌شود در غیر این صورت (اگر از عبارت **else** استفاده کرده باشیم) بلوک **else** اجرا می‌شود. هیچ‌گاه هر دو آن‌ها با هم اجرا نخواهند شد. در قسمت **condition** همیشه باید مقدار بولین تولید شود.

به نمونه‌ی زیر توجه کنید:

```
using System;  
public class MyClass  
{  
    public static void Main()  
    {  
        int num1 = 6;  
        int num2 = 23;  
  
        int result = num1 + num2;  
  
        if (result > 25)  
        {  
            Console.WriteLine("Result is more than 25");  
        }  
    }  
}
```

```

        else
        {
            Console.WriteLine("Result is less then 25");
        }
    }
}

```

در این مثال، مقدار متغیر num1 و num2 باهم جمع می‌شوند و نتیجه‌ی آن‌ها در متغیر result ذخیره می‌شود که نهایتاً result شامل مقدار ۲۹ می‌شود. همان طور که می‌بینید مقدار result با ۲۵ مقایسه شده است و می‌دانیم که پاسخ true است. به دلیل این که مقدار condition برابر با true شده است، بلوک if اجرا می‌شود و بلوک else نادیده گرفته شده و هیچ گاه اجرا نخواهد شد.

به نمونه‌ی بعدی زیر توجه کنید:

```

using System;
public class ifDemo
{
    public static void Main()
    {
        if (2 + 2 == 5)
            Console.WriteLine("Does not compute"); // this won't be displayed;
        else
            Console.WriteLine("This is False!");
    }
}

```

در این مثال، در قسمت condition $2 + 2$ می‌شود که حاصل آن ۴ است، سپس ۴ با ۵ توسط عملگر مقایسه‌ای == مقایسه می‌شود و همان‌طور که می‌دانید ۴ و ۵ باهم برابر نیستند، بنابراین مقدار condition برابر با false می‌شود. با این حساب بلوک دستور if اجرا نمی‌شود و بلوک دستور else اجرا خواهد شد.

نکته: همین‌طور که می‌بینید، هم if و هم else، به دلیل این که یک statement بیشتر ندارند، استفاده از بلوک (براکت باز و بسته) اختیاری است.

کلیه حقوق مادی و معنوی برای وبسایت [وب‌تارگت](#) محفوظ است.

استفاده از این مطلب در سایر وبسایت‌ها و نشریات چاپی تنها با ذکر و درج لینک منبع مجاز است.