

آموزش سی شارپ (C#) قسمت ۷ : دستورات کنسول و متغیرها (نسخه چاپی)

با سلام به همراهان همیشگی انجمن تخصصی فناوری اطلاعات ایران. قسمت پنجم از سری آموزشی زبان سی شارپ را با هم شروع می کنیم. در قسمت های قبلی با ساختار اولیه برنامه های سی شارپ و محیط کنسول به صورت مقدماتی و همچنین کامپایلر زبان سی شارپ آشنا شدیم. در این قسمت قصد داریم موارد زیر را با هم بررسی کنیم:

۱. دستورات اولیه محیط Console
۲. متغیرها و شیوه تعریف و استفاده از آنها
۳. آشنایی با دستورات یا Statements
۴. شیوه نوشتن یادداشت (Comment) در کدها

دستورات اولیه محیط Console

در قسمت قبل یک پروژه ساده در محیط کنسول ایجاد کردیم که پیغامی را در خروجی نمایش می داد. به دلیل اینکه بیشتر مثال های مطالبی که در بخش مقدماتی مطرح می شوند در محیط کنسول نوشته خواهند شد، در این بخش به بررسی دستورات اولیه محیط کنسول می پردازیم و قدم به قدم این دستورات را بررسی می کنیم. دستوراتی که در این بخش مورد بررسی قرار خواهند گرفت عبارتند از:

۱. Console.WriteLine
۲. Console.WriteLine
۳. Console.ReadKey

دستور Console.WriteLine

این دستور یک پیغام را در صفحه کنسول به عنوان خروجی برای ما نمایش می دهد. ما می توانیم عبارت مورد نظر را به صورت رشته به این دستور به عنوان پارامتر ارسال کنیم تا در خروجی نمایش داده شود. برای مثال فرض کنید می خواهیم عبارت «Welcome» را در خروجی نمایش دهیم. برای اینکار دستور زیر را می نویسیم:

```
Console.WriteLine("Welcome");
```

با دستور بالا عبارت Welcome بر روی پنجره Console نمایش داده خواهد شد. رشته ها بین علامت «» قرار میگیرند. در صورتی که ما چندین دستور Console.WriteLine را پشت سر هم بنویسیم، خروجی ها پشت سر هم نمایش داده خواهند شد. برای مثال:

```
Console.WriteLine("A");  
Console.WriteLine("B");  
Console.WriteLine("C");
```

خروجی کد بالا به صورت زیر خواهد بود:

```
ABC
```

دستور Console.WriteLine

این دستور دقیقاً مشابه دستور Console.WriteLine می باشد، با این تفاوت که بعد از چاپ رشته در صفحه خروجی، مکان نما به خط بعد منتقل می شود و عبارت بعدی در خط بعد نمایش داده خواهد شد:

```
Console.WriteLine("A");
```

```
Console.WriteLine("A");  
Console.WriteLine("B");  
Console.WriteLine("C");
```

خروجی دستورات بالا به صورت زیر است:

```
A  
B  
C
```

برای درک بهتر مثالی از استفاده ترکیبی دو دستور Write و WriteLine را بررسی می کنیم:

```
Console.Write("A");  
Console.WriteLine("B");  
Console.WriteLine("C");  
Console.Write("D");  
Console.Write("E");
```

خروجی کد بالا به صورت زیر خواهد بود:

```
AB  
C  
DE
```

دستور Console.ReadKey

فرض کنید کد زیر را در محیط کنسول نوشته ایم:

```
class Program  
{  
    static void Main(string[] args)  
    {  
        Console.WriteLine("Welcome to my application!");  
    }  
}
```

با زدن کلید F5 و اجرای برنامه، پنجره کنسول لحظه ای ظاهر شده و مجدداً ناپدید می شود و برنامه به اتمام می رسد. برای مشاهده خروجی باید کاری کرد. راه حل این مشکل، استفاده از دستور Console.ReadKey می باشد. همانطور که گفتیم برنامه به صورت خط به خط اجرا می شود. زمان اجرا، زمانی که این دستور اجرا می شود، برنامه منتظر می شود تا کاربر کلیدی را فشار دهد و پس از فشار دادن کلیدی توسط کاربر ادامه برنامه اجرا خواهد شد. حال اگر کد بالا را به صورت زیر تغییر دهیم، پنجره کنسول پیش از بسته شدن، منتظر دریافت کلیدی از کاربر شده و سپس پنجره کنسول بسته شده و برنامه به اتمام می رسد:

```
static void Main(string[] args)  
{  
    Console.WriteLine("Welcome to my application!");
```

```
Console.ReadKey();  
}
```

متغیرها و شیوه تعریف و استفاده از آنها

شما به عنوان برنامه نویس، هر برنامه ای که خواهید نوشت بدون شک از متغیرها داخل آن برنامه استفاده خواهید کرد. اما متغیرها چه کاربردی دارند؟ متغیرها در حقیقت خانه هایی از حافظه هستند که توسط برنامه ما برای نگهداری یک یا چند مقدار استفاده می شوند. برای مثال، شما فرض کنید انباری دارید که داخل این انبار چندین جعبه وجود دارد و داخل این جعبه ها میوه نگهداری می کنید. بر روی هر جعبه با برچسب نوع میوه های داخل جعبه را مشخص کرده اید. در یک جعبه سیب، در یک جعبه موز و در جعبه دیگر نارگیل. هر جعبه می تواند شامل یک یا چندین میوه باشد. در حقیقت انبار فضای حافظه شما، جعبه ها متغیرهای شما و میوه ها محتویات متغیرهای شما هستند. متغیرها انواع مختلفی دارند، اما برای شروع و آشنایی با شیوه تعریف و استفاده متغیرها، با نوع رشته ای و نحوه تعریف متغیرهایی از این نوع آشنا خواهیم شد. شیوه تعریف متغیرها به صورت زیر است:

```
{data-type} {variable-name} = {variable-value};
```

۱. `data-type`: نوع متغیری که می خواهیم تعریف کنیم را مشخص می کند. متغیرها می توانند رشته ای، عددی یا ... باشند.
۲. `variable-name`: نام متغیر را مشخص می کند، شما نام متغیر را هر چیزی می توانید بگذارید، اما دقت کنید که نام متغیرها می توانند ترکیبی از حروف و عدد و کاراکتر «_» باشند، و نام متغیرها نمی توانند با عدد شروع شوند. برای مثال نام های `number1` و `first_name` و `a_۱۲` برای نامگذاری صحیح بوده، اما نام های `۲number` و `first-name` صحیح نمی باشند و کامپایلر به شما پیغام خطا خواهد داد.

۳. `variable-value`: مقدار اولیه متغیر، مقدار اولیه ای که داخل متغیر قرار خواهد گرفت را مشخص می کند. نوشتن این قسمت اختیاری است، اما توجه کنید که متغیرها یا داخلشان مقداری ریخته می شود یا مقدار داخل آنها مورد استفاده قرار می گیرد. تنها متغیرهایی را می توان استفاده کرد که داخل آنها مقداری ریخته باشیم.

در این بخش و برای شروع با نوع رشته ای آشنا می شویم و متغیرها را بر این نوع تعریف می کنیم. نوع رشته ای با کلمه کلیدی `string` تعریف می شود. برای مثال فرض کنید می خواهیم متغیری با نام `website` و از نوع رشته ای تعریف کنیم و عبارت `ITPro` را داخل آن قرار دهیم. کد آن به صورت زیر خواهد بود:

```
string website = "ITPro";
```

دقت کنید که کلیه رشته باید بین علامت «"» قرار بگیرند. دستور بالا را می توان به صورت زیر نیز نوشت:

```
string website;  
website = "ITPro";
```

دستور اول متغیر مورد نظر با نام `website` را تعریف کرده و خط دوم مقدار `ITPro` را به منسوب می کند. کلاً عملیات انتساب مقادیر به متغیرها با علامت «=» انجام می شود. ابتدا نام متغیر، علامت «=» و سپس مقدار مورد نظر.

مزیت استفاده از متغیرها این است که ما متغیری را تعریف می کنیم و در قسمت های مختلف کد از آن متغیر استفاده می کنیم. برای مثال:

```
string firstName = "Hossein";  
Console.WriteLine(firstName);  
Console.WriteLine("Press any key to exit...");  
Console.ReadKey();
```

```
Console.ReadKey(),
```

۱. خط اول متغیری با نام firstName تعریف کرده و مقدار Hossein را داخل آن قرار می دهد.
۲. در خط دوم بوسیله دستور Console.WriteLine، مقدار متغیر firstName را در خروجی نمایش می دهیم.
۳. خط سوم پیامی را به کاربر مبنی بر اینکه برای خروج کلیدی را فشار دهد نمایش داده می شود.
۴. در خط پایانی یا خط چهارم، برنامه منتظر می ماند تا کاربر کلیدی را فشار دهد.

در صورت نوشتن کد بالا در متد Main خروجی زیر را در پنجره کنسول مشاهده خواهید کرد:

```
Hossein  
Press any key to exit...
```

خوب، تا اینجا کار با شیوه اولیه تعریف متغیرها، عملیات انتساب مقدار و استفاده از متغیرها آشنا شدید. در قسمت بعدی به بررسی انواع داده های عددی و رشته ای خواهیم پرداخت.

دستورات زبان سی شارپ

دستورات، مجموعه ای از کلمه ها هستند که در کنار هم کاری را انجام می دهند. برای مثال، دستور تعریف متغیر یا دستور نمایش خروجی در صفحه کنسول. هر دستور در زبان سی شارپ باید با علامت ; به اتمام برسد. در صورت عدم قرار دادن این علامت در انتهای دستورات زبان سی شارپ، با پیغام خطا مواجه خواهید شد. به مجموعه ای از دستورات که در پشت هم نوشته شده و به صورت خط به خط اجرا می شوند روند کنترل (flow of control) یا روند اجرا (flow of execution) می گویند. مثال زیر چند دستور را نشان می دهد:

```
string name;  
name = "Hossein"  
Console.WriteLine(name);
```

دستور اول تعریف متغیر، دستور دوم مقدار دهی متغیر و دستور سوم نمایش خروجی بر روی صفحه می باشد. دستورها انواع مختلفی دارند که در بخش های بعدی با انواع دستورات آشنا خواهیم شد.

نوشتن یادداشت یا Comment برای کدها

خیلی وقت ها نیاز است که برای برخی قسمت های کدی که در حال نوشتن آن هستیم یادداشت قرار دهیم. یادداشت قطعه ای از کد است که توسط کامپایلر، کامپایل نمی شود و با این ترتیب اجرا نیز نخواهد شد. برای یادداشت گذاری در برنامه های سی شارپ دو روش وجود دارد:

۱. یادداشت های تک خطی
۲. یادداشت های چند خطی

یادداشت تک خطی

برای نوشتن این یادداشت ها، در ابتدای خط از علامت // استفاده می کنیم. مثال:

```
// C# tutorial  
// by Hossein Ahmadi  
// ITPro.ir  
string message = "Welcome!";  
Console.WriteLine(message);
```

یادداشت های چند خطی

نوشتن این یادداشت ها به صورت زیر می باشد:

```
/*  
C# tutorial  
by Hossein Ahmadi  
ITPro.ir  
*/  
string message = "Welcome!";  
Console.WriteLine(message);
```

در حقیقت، هر متنی که بین `/**` و `*/` قرار بگیرد به عنوان یادداشت در نظر گرفته خواهد شد.

نویسنده : حسین احمدی

منبع : [جزیره برنامه نویسی وب سایت توسینسو](#)

هرگونه نشر و کپی برداری بدون ذکر منبع و نام نویسنده دارای اشکال اخلاقی است

پیمان بهرامی

int نوع عددیه، چرا بهش رشته دادی؟

حسین احمدی

مرسی از تذکرتون. اصلاح شد :

سلمان بدرقه

سلام مهندس جان

ببخشد قسمت پنج بخش دیگه هم داره چون نوشته بخش اول و دیگه خبری از بخش های دیگه نیست - رفتی بخش ششم؟

با تشکر بدرقه

ImDope

سلام من اولین پروژه کنسول که

```
Console.WriteLine("Hello C#");
```

رو کامپایل که میکنم . پروژه زود بسته میشه . فرم نمایش داده نمیشه / مشکل از کجاست ؟

این هم کد کامل

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;
```

```
using System.IO;\nusing System.Threading.Tasks;\n\nnamespace ConsoleApplication1\n{\n    class Program\n    {\n        static void Main(string[] args)\n        {\n            Console.WriteLine("Hello C#");\n        }\n    }\n}
```

مطلب اصلی