

در بخش قبل از آموزش زبان برنامه نویسی سی شارپ، شما نحوه ایجاد یک پروژه جدید و آشنایی با فایل‌هایی که در هنگام ساختن پروژه ایجاد می‌گردند را فرا گرفتید. طبق آموزش جلسه پیش، نرم‌افزار Visual Studio خود را اجرا کرده و یک پروژه جدید سی شارپ بسازید.

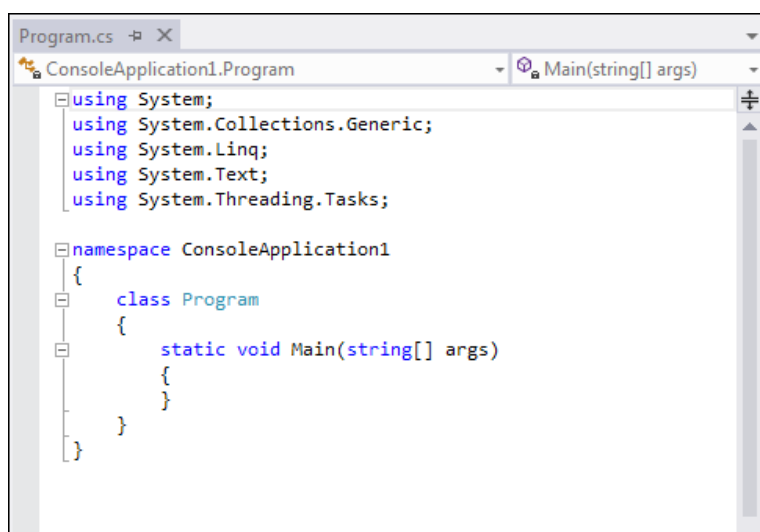
پس از آن تصور کنید که شما تغییراتی را در آن اعمال کرده‌اید و می‌خواهید که آن‌ها را ذخیره کنید. از منوی فایل، گزینه‌ی Save All را برگزینید. (و یا کلیدهای ترکیبی Ctrl+Shift+S را فشار دهید.)

هنگامی که برای بار اول روی Save All کلیک می‌کنید، پنجره‌ای برای ذخیره فایل نشان داده می‌شود در این صورت، شما می‌توانید برای پروژه خود نام موردنظر را قرار داده و در نهایت ذخیره کنید! دقت کنید که ممکن است که پنجره‌ی فوق برای شما به نمایش درنیاید، که کاملاً طبیعی می‌باشد و در هر صورت، کار شما ذخیره شده‌است. برای این‌که فایل‌ها و پوشه‌های پروژه‌ی ذخیره شده‌ی خود را ببینید، می‌توانید بسته به نوع سیستم عامل خود به فولدر Documents یا My Documents بروید و با ورود به پوشه‌ی Visual Studio 20XX که XX نشان‌دهنده‌ی ورژن نرم‌افزار Visual Studio شما می‌باشد، وارد فولدر Projects شوید و فایل‌ها و فولدرهای مربوط به پروژه‌ی خود را مشاهده کنید.

• شروع به نوشتن اولین خط کد سی شارپ

اگر در آموزش شماره ۱ به‌خاطر داشته باشید، گفتیم که یکی از راه‌های نمایش اطلاعات برنامه‌نویسی شده از طریق پنجره‌ای به نام Console می‌باشد. پنجره‌ای که شبیه به سیستم‌عامل DOS می‌باشد و شما با اجرای نرم‌افزار CMD می‌توانستید یک نمونه از آن را ببینید.

در این بخش نیز برای شروع به‌کار، ابتدا یک پروژه‌ای را ایجاد کرده و با انتخاب نوع Console Application در مراحل ساخت پروژه، صفحه‌ای همانند نیز برای شما نمایان خواهد شد که دارای خطوطی از کدها می‌باشد. این‌ها کدهایی هستند که سی شارپ به آن‌ها نیاز دارد تا این‌که برنامه شما را اجرا کند. لذا شما نباید این کدها را تغییر دهید و یا حذف کنید. (در آینده خواهید توانست بعضی از بخش‌های آن را که اضافی‌ست حذف نمایید)



```

Program.cs
ConsoleApplication1.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}

```

در ۵ سطر اول از کدها (ممکن است برای شما تعداد آن‌ها کمتر باشد!) مشاهده می‌کنید که خطوط با استفاده از کلمه‌ی کلیدی using شروع شدند. فعلاً این خط‌ها را نادیده بگیرید تا بعداً، زمانی که به یک مرحله مشخص رسیدید، بیشتر درباره‌ی آن‌ها صحبت کنیم.

در خطی که با namespace شروع شده‌است، نام برنامه‌ی شما را مشخص می‌کند. یک namespace روشی است که کدهای مربوط به هم را به یکدیگر ارتباط دهد. فعلاً به‌هیچ‌وجه نگران این خط از کدها نباشید؛ چرا که در آینده همه این‌ها را فرا خواهید گرفت.

مهم‌ترین چیزی که در این کدها می‌توان یافت، کلمه کلیدی class می‌باشد. تمامی کدهایی که در سی شارپ می‌نویسید (و خواهید نوشت)، باید عضوی از یک class باشد. این class با نام program می‌باشد. شما کلاس‌های دیگری را زمانی که می‌نویسید، می‌توانید به آن‌ها هرنامی را که دلتان بخواهد اعطا کنید؛ اما توجه داشته باشید نامی را که انتخاب می‌کنید، نباید جزو کلمات رزرو شده‌ی زبان سی شارپ باشد.

در داخل class program، کدهای زیر موجود است:

```
static void Main(string[] args)
```

```
{
```

```
}
```

این بخش از کد را متد (method) می‌نامند! نام این متد Main می‌باشد. هنگامی که شما با زبان سی‌شارپ برنامه‌ای را می‌نویسید، کامپایلر شروع به یافتن متد Main می‌کند. سی‌شارپ از متد Main به عنوان محرک و آغازگر زبان استفاده می‌کند و عدم وجود این متد، باعث می‌شود که برنامه شما اجرا نشود! هر کدی که در میان دو براکت (با پرانتز اشتباه نگیرید!) قرار گیرد، در زمان اجرای برنامه، اجرا می‌شوند.

اگر در عکس بالا دقت کنید، خواهید دید که کلماتی وجود دارند که با رنگ آبی از دیگر کلمات جدا شده‌اند. به این کلمات که با رنگ آبی در سی‌شارپ از دیگر کلمات مشخص می‌شوند، کلمات کلیدی می‌گویند. شما در جلسه‌های آموزشی آینده در رابطه با این‌ها بیشتر خواهید دانست.

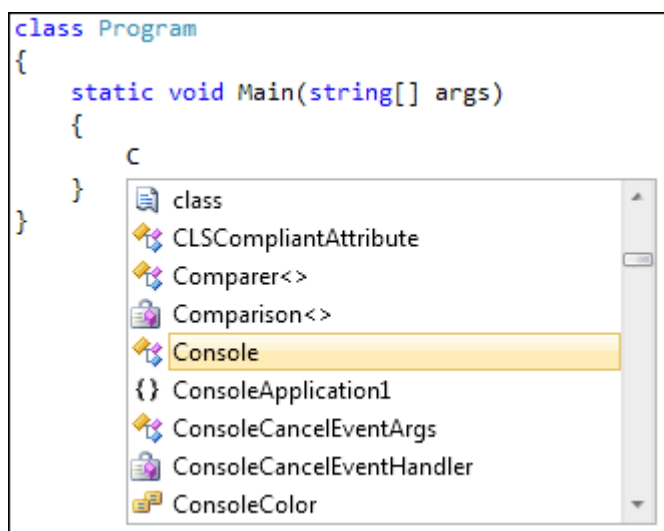
اکنون، نشانگر موس را بعد از براکت باز قرار داده و با فشردن کلید Enter، به خط بعدی رفته تا اولین خط از کد را بنویسید. با فشردن کلید Enter، بصورت اتوماتیک سرخط شما یک tab به سمت جلو خواهد رفت. این باعث می‌شود که خواندن و نوشتن کدها برای شما راحت‌تر باشد.

```
class Program
{
    static void Main(string[] args)
    {
        |
    }
}
```

اگر در کد بالا دقت کنید، متوجه خواهید شد که هر بخش از کد در داخل یک جفت براکت قرار دارد. جا انداختن هر کدام از آن‌ها، باعث خواهد شد که با یک خطا (Error) مواجه شوید. تک خطی که فعلاً خواهیم نوشت، خط زیر است:

```
Console.WriteLine("Hello C Sharp!");
```

قبل از آن‌که چیزی را تایپ کنید، ابتدا تنها حرف C را تایپ کنید. خواهید دید که یک منو به‌صورت اتوماتیک برای شما باز خواهد شد. این منو را اصطلاحاً IntelliSense می‌نامند. این منو به‌صورت هوشمندانه سعی می‌کند که کلمات کلیدی‌ای که می‌خواهید بنویسید را تشخیص دهد تا کار شما، برنامه‌نویس را راحت‌تر کند.



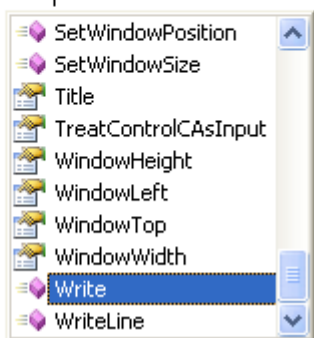
```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine('');
    }
}
```

وقتی که منوی بالا برای شما باز می‌شود، در کنار نام Console، آیکن خاصی وجود دارد. هر کلمه‌ی کلیدی‌ای با این آیکن، نشان‌گر یک Class است. الان با فشردن کلید Enter، بگذارید که Visual Studio بصورت اتوماتیک برای شما کلمه Console را بنویسید.

```
class Program
{
    static void Main(string[] args)
    {
        Console|
    }
}
```

اکنون یک نقطه قرار دهید؛ منوی IntelliSense مجدداً ظاهر می‌شود. شما می‌توانید با کلیدهای بالا/پایین در این منو جابجا شوید. اما شما به آرامی و حرف به حرف شروع به تایپ حروف کلمه‌ی Write کنید؛ خواهید دید که با حرفی که جلو بروید، IntelliSense دقتش بالاتر رفته و میزان تشخیص کلمه‌ی مورد نظر بالاتر می‌رود. حال شما به دنبال کلمه WriteLine بگردید. (و یا آن را به‌صورت کامل تایپ کنید)

```
class Program
{
    static void Main(string[] args)
    {
        Console.|
    }
}
```



حال یک پرانتز باز کنید؛ تصویری مشابه تصویر زیر برای شما باز خواهد شد.

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine(|
    }
}
```

1 of 19 void Console.WriteLine ()
Writes the current line terminator to the standard output stream.

WriteLine، یکی دیگر از متدهای موجود در زبان سی‌شارپ می‌باشد. جعبه زرد رنگی که برای شما نمایش داده می‌شود، نشان‌دهنده‌ی این است که ۱۹ ورژن مختلف از این متد در سی‌شارپ وجود دارد. شما فعلاً خط زیر را وارد کنید:

“Hello C Sharp!”

فراموش نکنید که دو علامت “Double Quotes” (نقل قول) را در ابتدا و انتهای Hello C Sharp! قرار دهید. قرار دادن Double Quotes به سی‌شارپ این فرمان را می‌دهد که شما یک متن می‌خواهید؛ نه یک کد! بعد از وارد کدهای بالا، می‌بایست که شکل و شمایل کدهای شما شبیه به زیر شده باشد!

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello C Sharp!")|
    }
}
```

حال یک پرانتز بسته در انتهای کد خود قرار داده؛ اگر دقت کنید در انتهای پرانتز بسته، یک خط کوچک قرمز رنگی برای شما نمایش داده می‌شود. هر زمانی که شما به یک خط قرمز رنگی در سی‌شارپ مواجه شدید، بدانید که یک جای کار ایراد دارد (معمولاً یک چیزی را فراموش کرده‌اید و یا کلمه‌ای را به اشتباه تایپ کرده‌اید).

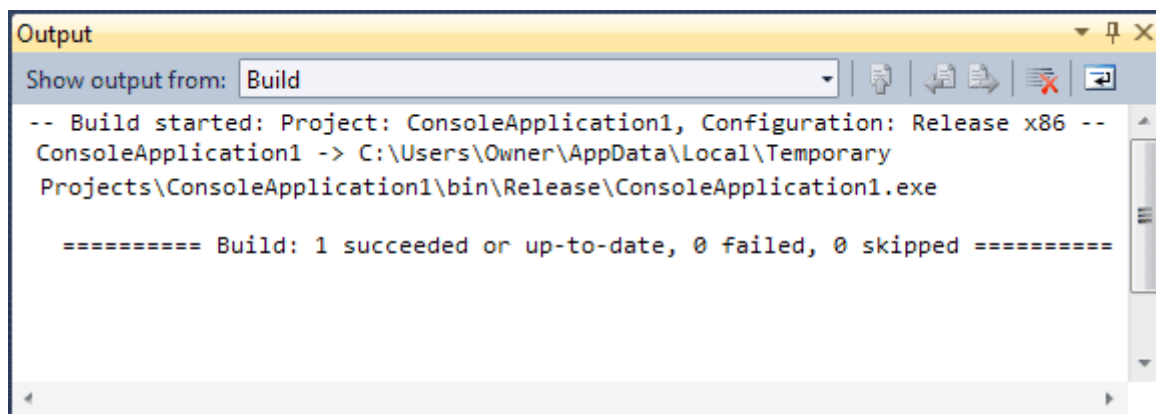
```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello C Sharp!");
    }
}
```

قوانینی که در زبان برنامه‌نویسی سی‌شارپ وجود دارد (که شبیه آن در زبان های ++C و PHP نیز دیده می‌شود) انتهای هر کد می‌بایست یک "Semicolon" یعنی ";" قرار دهید. با قرار دادن آن در انتهای کد خود، خط قرمز رنگ ناپدید می‌شود. الان زمان آن رسیده که کد خود را اجرا کنید ...

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Hello C Sharp!");
    }
}
```

قبل از آنکه برنامه‌ی خود را اجرا کنید، باید چندکار را پیش از آن انجام دهید:

- از منوی بالا بر روی View کلیک کنید، بر روی Output کلیک کنید. (اگر که گزینه Output را در منوی View ندید، با رفتن به گزینه Tools و سپس با رفتن به Settings و بعد از آن Expert Settings ، می‌بایست که Output را ببینید.)
- از منوی Build بر روی Build Express کلیک کنید.
- شما می‌بایست که یک گزارشی شبیه به گزارش پایین ببینید.



آخرین خط می‌بایست که متن زیر را ببینید:

Build: 1 succeeded or up-to-date, 0 failed, 0 skipped

این خط نشان‌دهنده‌ی این است که همه چیز به‌صورت درست انجام شده و برنامه آماده کامپایل شدن می‌باشد.

حال مراحل زیر را تست کنید:

- Semicolon-ای که در انتهای کد خود اضافه کردید را حذف کنید.
- از منوی بالا بر روی Build کلیک کرده و سپس بر روی Build Solution کلیک کنید.

این بار خواهید دید که در گزارش زیر خطاهای زیر را مشاهده خواهید کرد:

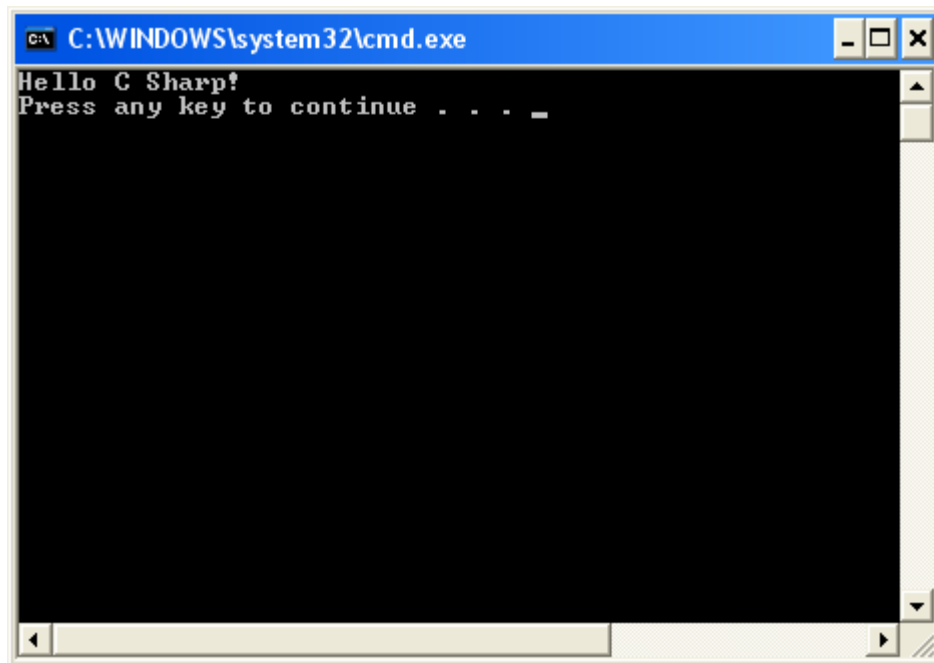
Compile complete – ۱ errors, 0 warnings

Build: 0 succeeded or up-to-date, 1 failed, 0 skipped

این خط به شما این پیغام را می‌دهد که برنامه نتوانست ساخته شود چون یک خطایی در کدهای شما وجود دارد. حال دوباره همان Semicolon را برگردانید.

از منوی بالا بر روی Debug کلیک کنید؛ در منوی باز شده بر روی Start Debugging کلیک کرده و آن را انتخاب کنید.

پنجره Console بر شما پدیدار خواهد شد؛ زود هم ناپدید می‌شود! فعلاً برای جلوگیری از این مشکل، از منوی Debug، گزینه Start Without Debugging را انتخاب کنید. پنجره‌ی زیر را خواهید دید.



برنامه‌ی شما اجرا شد! جمله‌ی Hello C Sharp را خواهید دید!

از جلسه بعد، بر روی Windows Application کار خواهیم کرد.

اگر سوالی داشتید، از طریق فرم نظرات سوال‌های خود را بپرسید.