

ویدیو هشتم

در ویدیو هشتم در ابتدا نحوه دانلود ویژوال استادیو را در سایت مایکروسافت یاد میگیریم

سپس ساخت یک پروژه جدید C# را یاد میگیریم. پس از باز کردن ویژوال استادیو، روی گزینه Create new project کلیک میکنیم. در قسمت بعد از بین لیستی از انواع پروژه ها روی گزینه Console App (Core) کلیک کنید و روی دکمه Next بزنید.

در قسمت بعدی نهایتاً نام پروژه را انتخاب میکنیم و روی Create کلیک میکنیم تا پروژه ساخته شود.

وقتی پروژه باز میشه با چند خط کد روبرو میشیم که نیاز نیست در ابتدا به آنها توجه کنیم

اما کدی که ما باید بنویسیم بایستی از خط ۹ م برنامه، جایی که `Console.WriteLine("Hello World!");` نوشته شده است، شروع شود.

اگر روی دکمه های ترکیبی `Ctrl+F5` کلیک کنیم برنامه اجرا می شود و میبینیم که کلمه Hello World! چاپ میشود.

اما بعد از ساخت پروژه جدید یاد میگیریم که چطور با شرط ها در زبان C# کار کنیم. شرط ها یا همان if ها یک شرطی رو درون خود دارن که در صورت درست بودن آن شرط، کدی را اجرا خواهند کرد.

مثل $\text{if } (a + b > 10)$

اگر a به علاوه b بزرگتر از 10 بود. این یک شرط ساده است.

همچنین میتوانیم شرط خود را داخل یک متغیر از نوع bool بریزیم و آن متغیر را داخل if بگذاریم.

دستور else در صورت غلط بودن شرط اجرا میشود. در مثال بالا اگر $a + b$ بزرگتر از 10 نباشند قسمت else اجرا می شود.

در بعضی از پروژه ها نیاز است که برقرار بود دو شرط را بررسی کنیم و عملی انجام دهیم، در این صورت کافیست دو شرط را درون if بنویسیم و سپس بین آنها از && استفاده کنیم

اگر بخواهیم بگوییم از بین دو شرط یکی هم درست باشد، کد را اجرا کند باید از || بین آن دو شرط استفاده کنیم.

ویدیو نهم

در این ویدیو در مورد ساختار یک پروژه سی شارپ یاد میگیریم. وقتی یک برنامه جدید را آغاز میکنیم در ابتدا از دستور using استفاده شده است. که این دستور در فراخوانی کتابخانه ها کاربرد دارد. برای مثال در این پروژه کتابخانه system فراخوانی شده است، که دارای کلاس های مختلفی می باشد.

اما چون system یک کتابخانه پیش فرض هست. نیازی نیست که برای کلاس های آن اول از اسم system استفاده کنیم. برای مثال Console یک کلاس از کلاس های کتابخانه system هست. پس برای استفاده از آن نیازی به نوشتن نام system در ابتدای آن نیست.

Console.WriteLine()

دستور بعدی namespace هست که بایستی برای گروه بندی کلاس های برنامه از آن استفاده کنیم. اگر مجموعه زیادی از کلاس ها را در برنامه داشته باشیم می توانیم با namespace ها آنها را گروه بندی کنیم. کلاس program که کلاس اصلی برنامه هست که درون namespace پیش فرض برنامه است نوشته شده.

مثل

```
namespace ConsoleApp1
```

```
{
```

```
    Class Program
```

```
    {
```

```
    }
```

```
}
```

همچنین تابع Main() تابع اصلی برنامه است که برنامه از این تابع اجرا می شود و درون کلاس program وجود دارد.

اما نوع این تابع void است و به این معناست که این تابع هیچ خروجی ندارد. تابع static به این معناست که این تابع ثابت و در هنگام اجرای برنامه تغییر نمیکند.

و اما داخل پرانتز ورودی تابع مشخص شده است. که از نوع رشته است. یعنی میتوان یک رشته را از خارج این تابع به آن فرستاد و در داخل تابع از آن استفاده کرد.

مثل

```
Static void Main(String[] args)
```

```
{
```

```
}
```

ویدیو دهم

در ویدیو دهم چگونگی کار با حلقه ها را می آموزیم. یکی از دستورات حلقه while است.

While به اینصورت عمل می کند که یک شرط را درون پرانتز میگیرد و در صورت برقراری آن کد مربوطه را اجرا میکند و سپس دوباره شرط را بررسی میکند. آنقدر اینکار تکرار میشود تا دیگر شرط ما برقرار یا درست نباشد و سپس از حلقه بیرون می آید.

مثل

```
While(counter < 10)
```

```
{  
    Console.WriteLine($"The count is {counter}");  
    Counter++;  
}
```

در اینجا شرط درون while کوچکتر بودن متغیر counter از 10 است.

پس تا وقتی که counter کوچکتر از 10 باشد. کد درون آن اجرا میشود. و در هر بار اجرا یک عدد به counter اضافه خواهد شد. (counter++)

با فرض اینکه مقدار counter در ابتدای برنامه 0 است. در صورت اجرای این برنامه چنین خروجی چاپ می شود.

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

حلقه do while هم همانند while است با این تفاوت که do while برعکس عمل میکند و ابتدا کد خود را اجرا میکند و سپس برای دفعات بعدی شرط را بررسی میکند، در صورت درست بودن شرط، حلقه ادامه پیدا می کند.

حلقه آخر for نام دارد و شرط آن کمی متفاوت است. به اینصورت عمل می کند که ابتدا یک شمارنده برای حلقه تعریف می کنیم و سپس بازه اجرای حلقه را برای for تعیین می کنیم و در آخر تعیین می کنیم که شمارنده در طول اجرا حلقه چگونه تغییر کند.

مثل

```
For(int index = 0; index < 10; index++)
```

Index متغیر شمارنده ماست که در ابتدا تعریف شده است. سپس تعیین شده که بازه شما تا کجا باشد. در این جا تا وقتی که index کوچکتر از 10 باشد شرط برقرار است. و در مرحله سوم هر بار که حلقه اجرا میشود، یک عدد به مقدار index اضافه می شود.

ویدیو یازدهم

در این ویدیو یک سوال برنامه نویسی را حل می کنیم. سوال این است که اعداد قابل تقسیم بر ۳ که در بازه ۱ تا ۲۰ وجود دارند را با هم جمع کنیم و در خروجی چاپ کنیم.

در اینجا ما به یک حلقه و یک if نیاز داریم، ابتدا حلقه for را با یک شمارنده به نام i تعریف می کنیم با مقدار ۱ و بازه آن را تا ۲۰ مشخص میکنیم و هر بار یکی به شمارنده اضافه می کنیم.

در قسمت کد های حلقه یک if ایجاد می کنیم و در قسمت شرط آن اینگونه می نویسیم $\text{if}(i \% 3 == 0)$

به این معنا که اگر i ما باقیمانده اش بر 3 برابر 0 بود (بر 3 بخش پذیر بود)

i را با متغیر sum که از پیش تعریف شده جمع کن.

```
int sum = 0;
```

```
for (int i = 1; i <= 20; i++)
```

```
{
```

```
    if(i % 3 == 0)
```

```
    {
```

```
        sum = sum + i;
```

```
    }
```

```
}
```

```
Console.WriteLine($"The sum is {sum}");
```

در خط آخر کد متغیر sum که مجموع تمام اعداد بخش پذیر بر 3 کوچکتر از 20 را چاپ می کند.