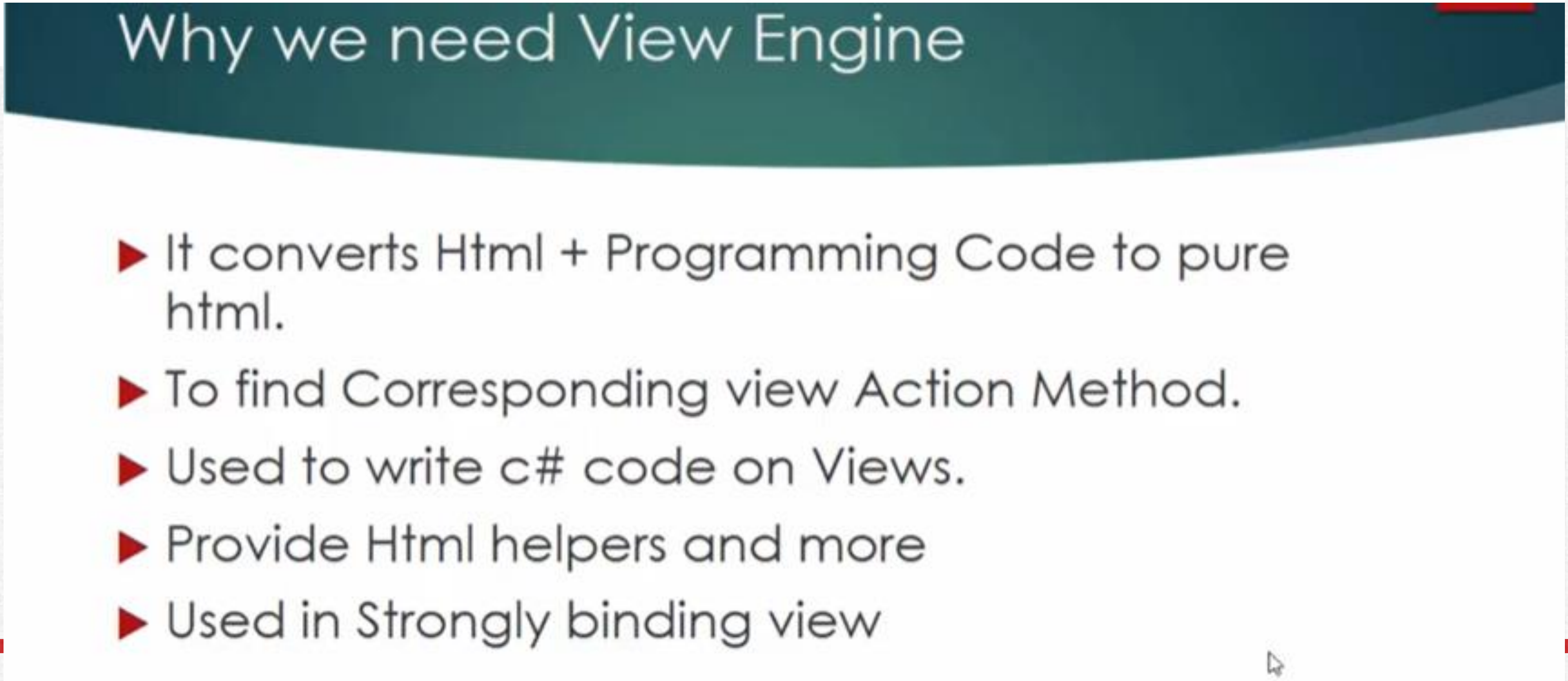
Razor View Engine in ASP.NET Core | Understanding Razor Syntax

.

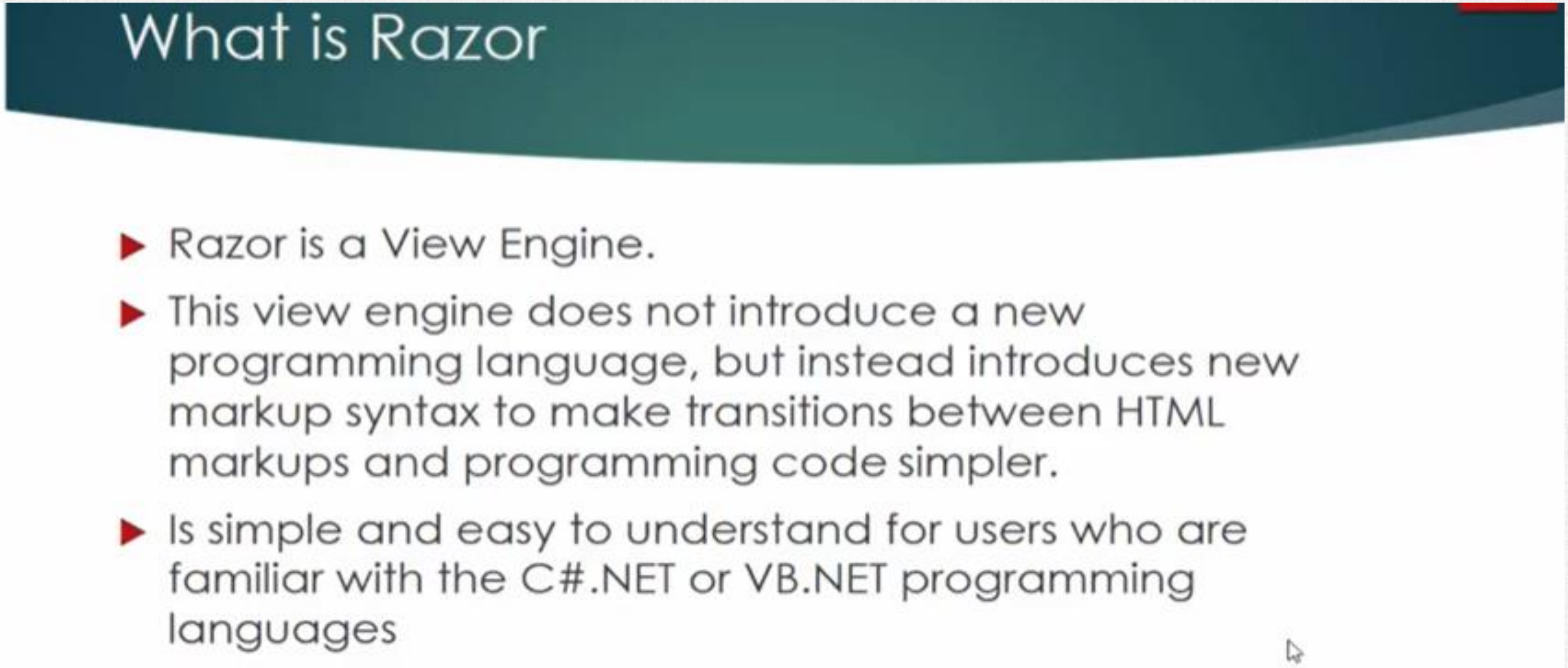
What is a View Engine

- Are part of the MVC Framework that converts the code of a view into HTML markup that a browser can understand.



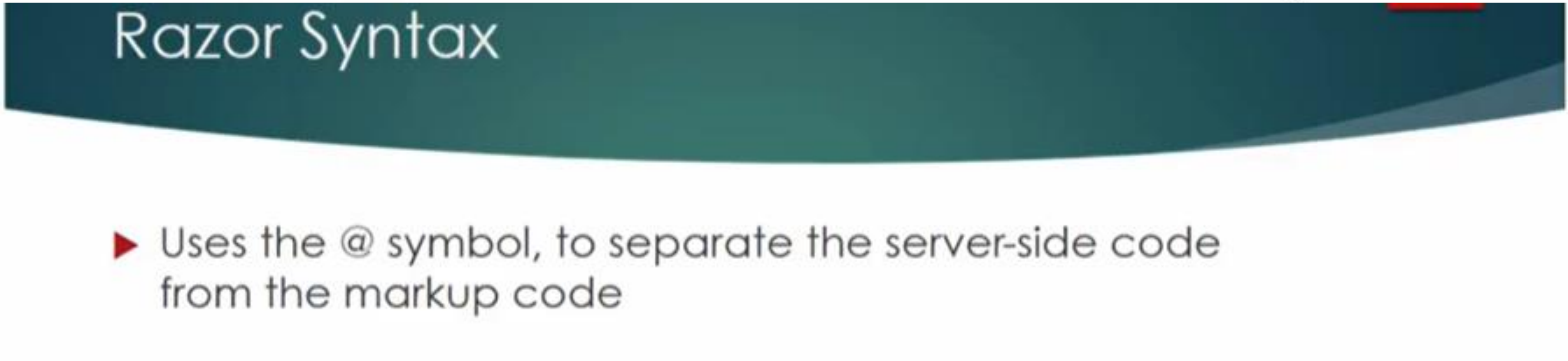
Why we need View Engine

- It converts Html + Programming Code to pure html.
- To find Corresponding view Action Method.
- Used to write c# code on Views.
- Provide Html helpers and more
- Used in Strongly binding view



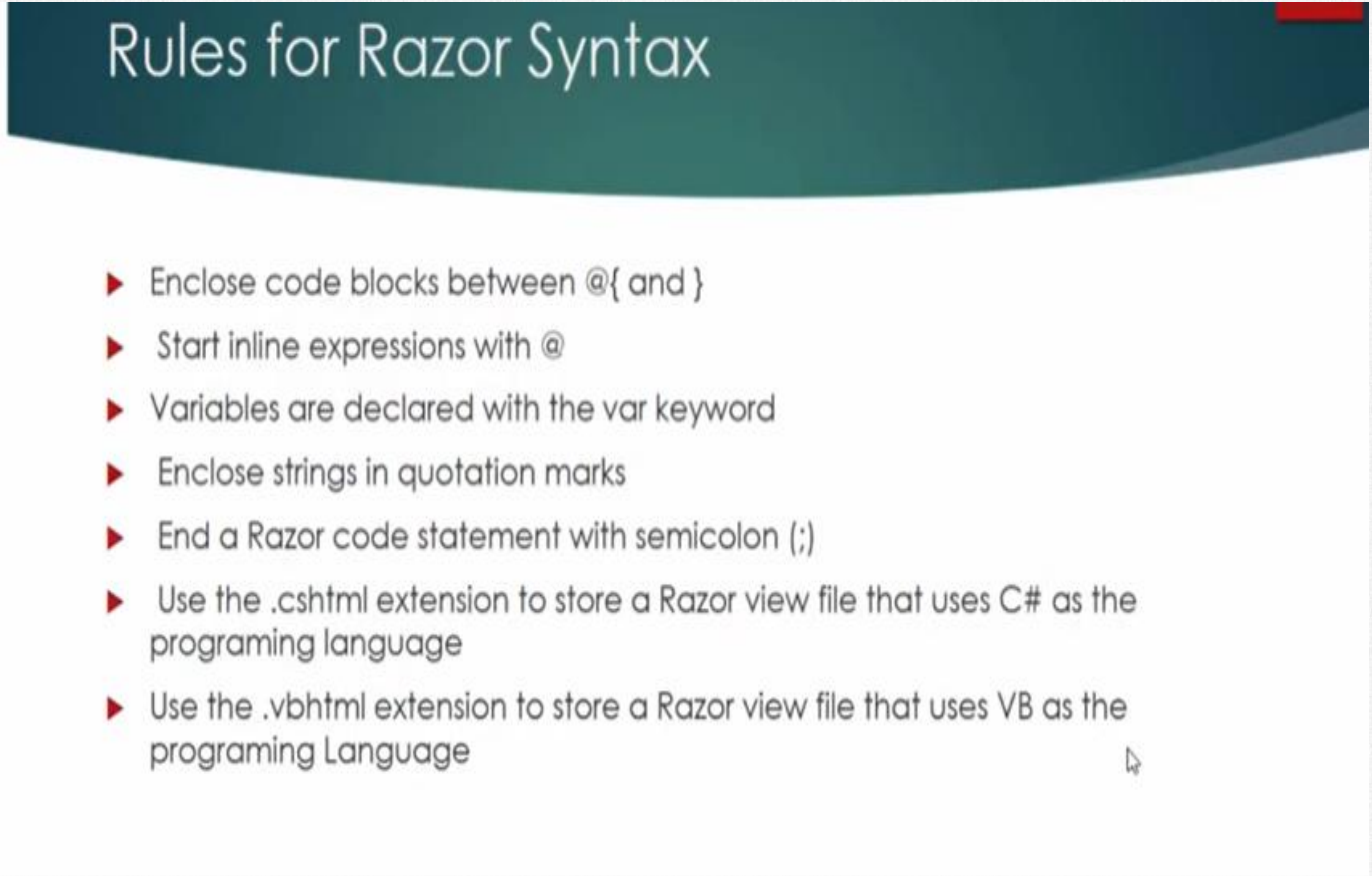
What is Razor

- Razor is a View Engine.
- This view engine does not introduce a new programming language, but instead introduces new markup syntax to make transitions between HTML markups and programming code simpler.
- Is simple and easy to understand for users who are familiar with the C#.NET or VB.NET programming languages



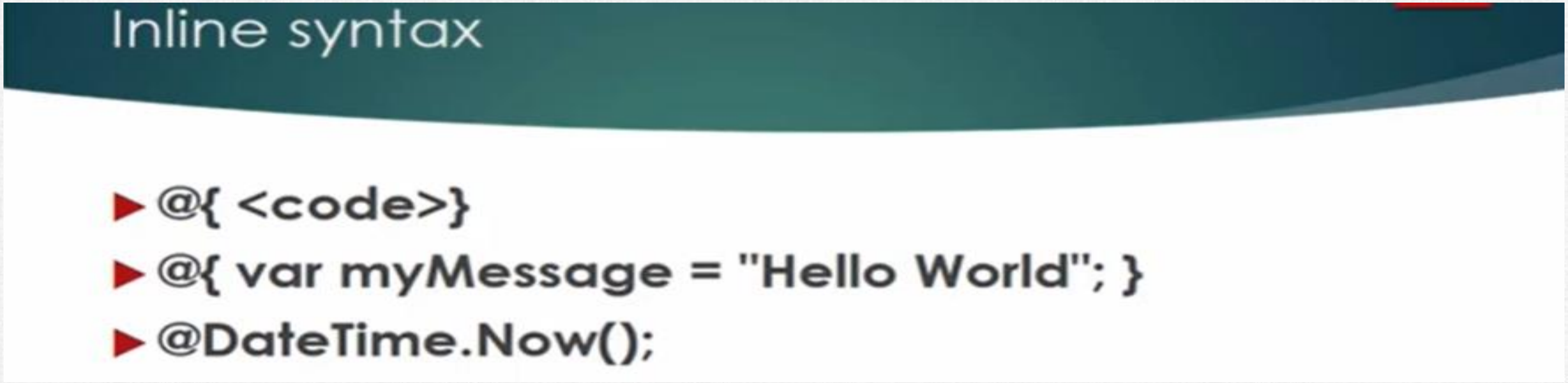
Razor Syntax

- Uses the @ symbol, to separate the server-side code from the markup code



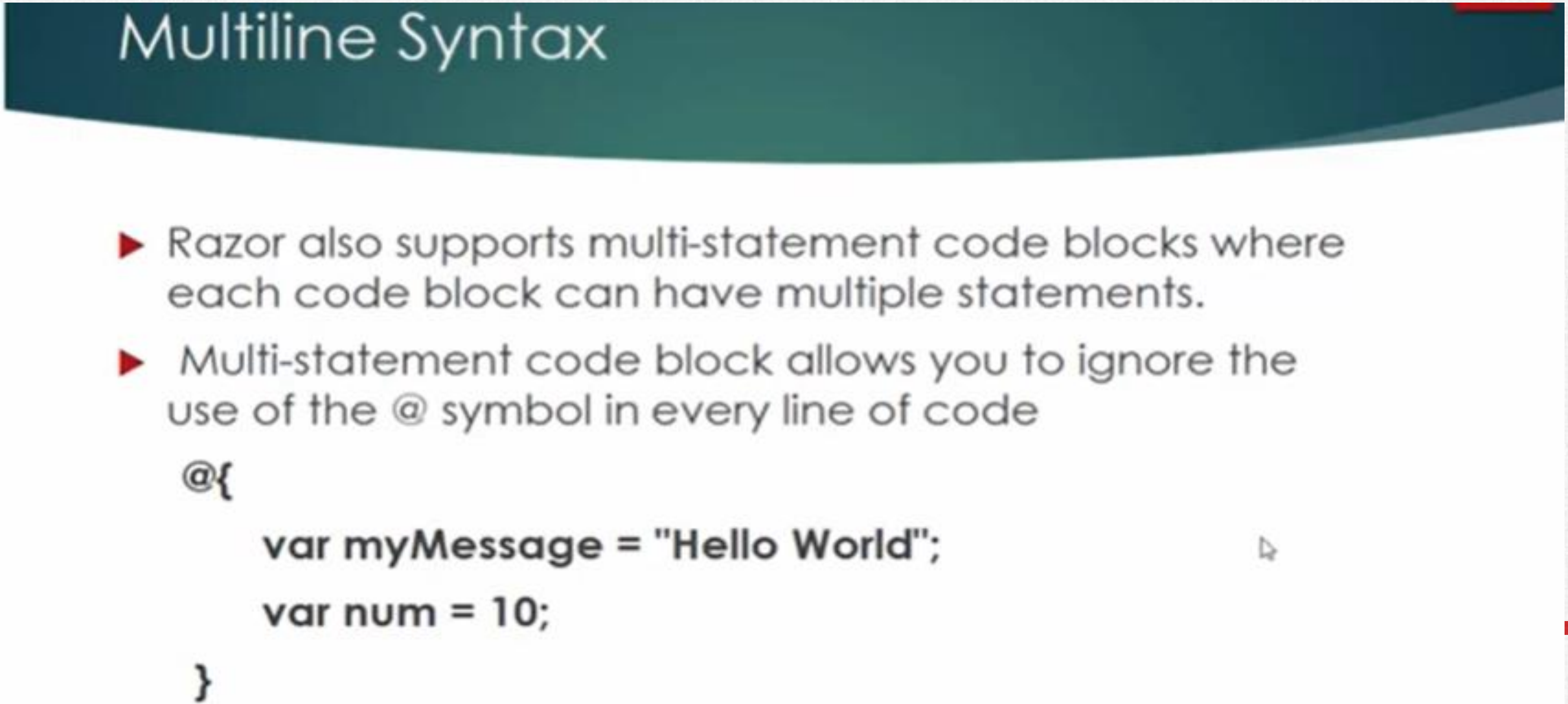
Rules for Razor Syntax

- Enclose code blocks between @{ and }
- Start inline expressions with @
- Variables are declared with the var keyword
- Enclose strings in quotation marks
- End a Razor code statement with semicolon (;)
- Use the .cshtml extension to store a Razor view file that uses C# as the programing language
- Use the .vbhtml extension to store a Razor view file that uses VB as the programing Language



Inline syntax

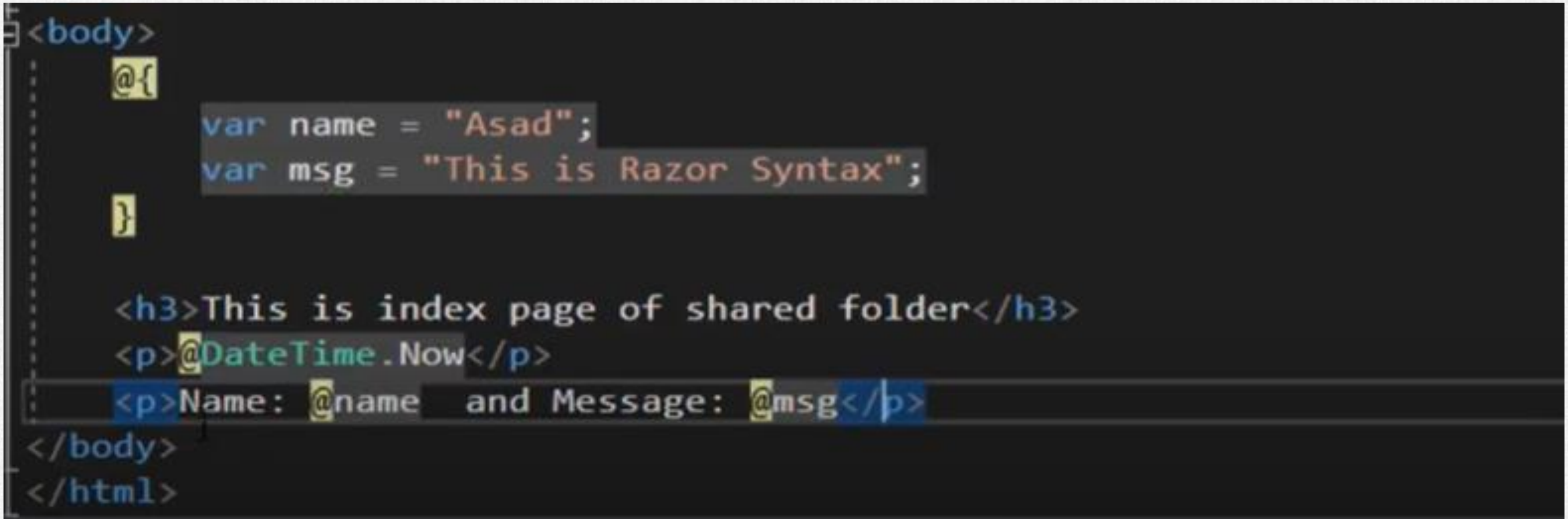
- `@{ <code>}`
- `@{ var myMessage = "Hello World"; }`
- `@DateTime.Now();`



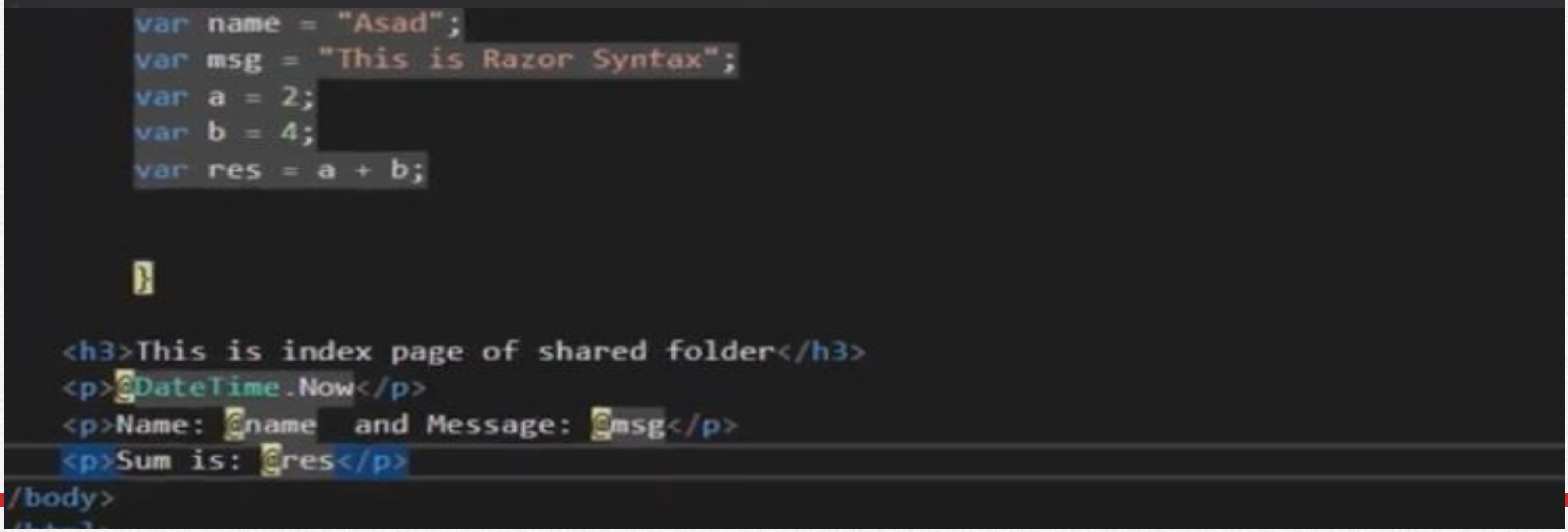
Multiline Syntax

- Razor also supports multi-statement code blocks where each code block can have multiple statements.
- Multi-statement code block allows you to ignore the use of the @ symbol in every line of code

```
@{
     var myMessage = "Hello World";
     var num = 10;
 }
```

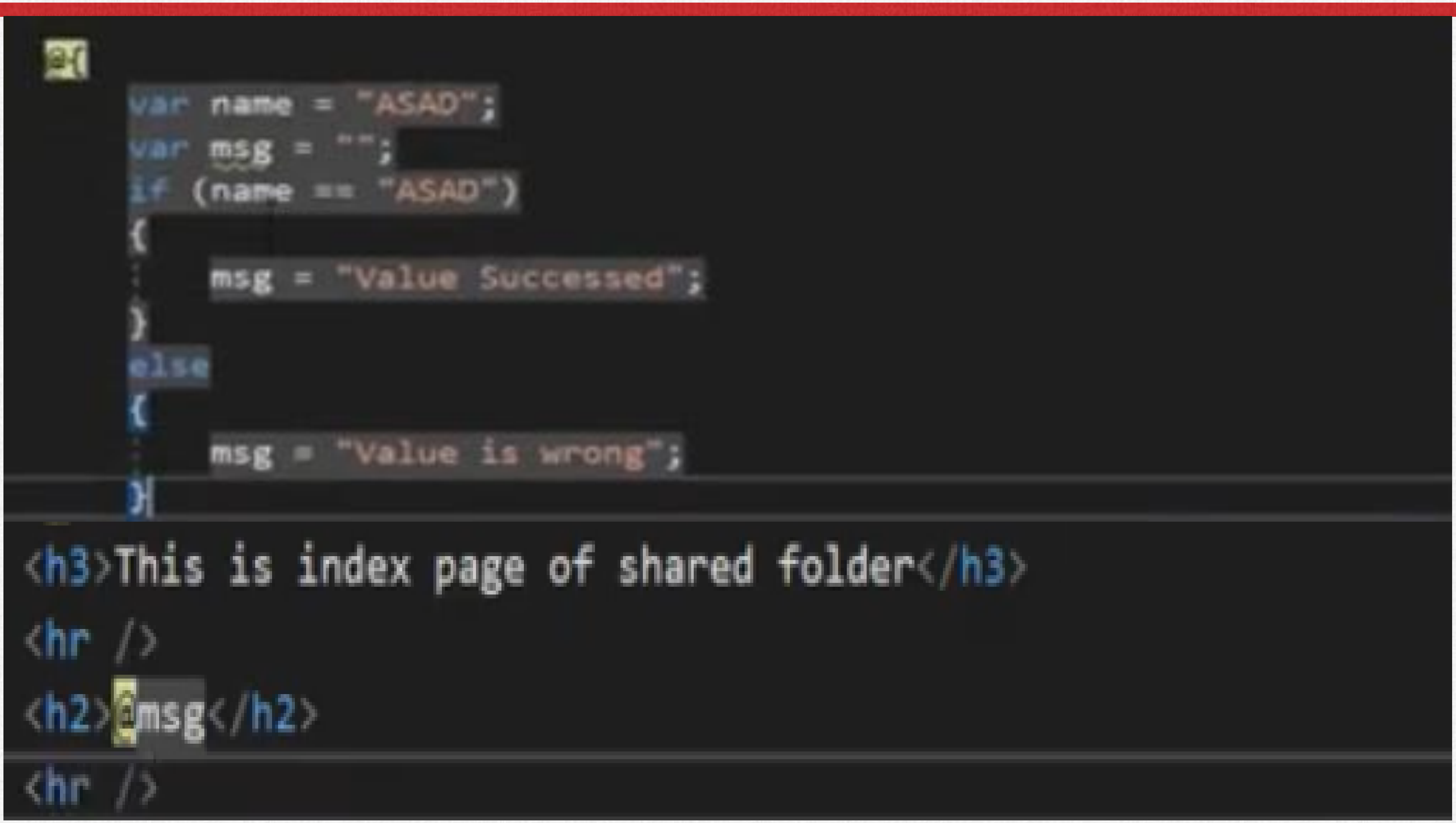


```
<body>
    @{
        var name = "Asad";
        var msg = "This is Razor Syntax";
    }

    <h3>This is index page of shared folder</h3>
    <p>@DateTime.Now</p>
    <p>Name: @name  and Message: @msg</p>
</body>
</html>
```



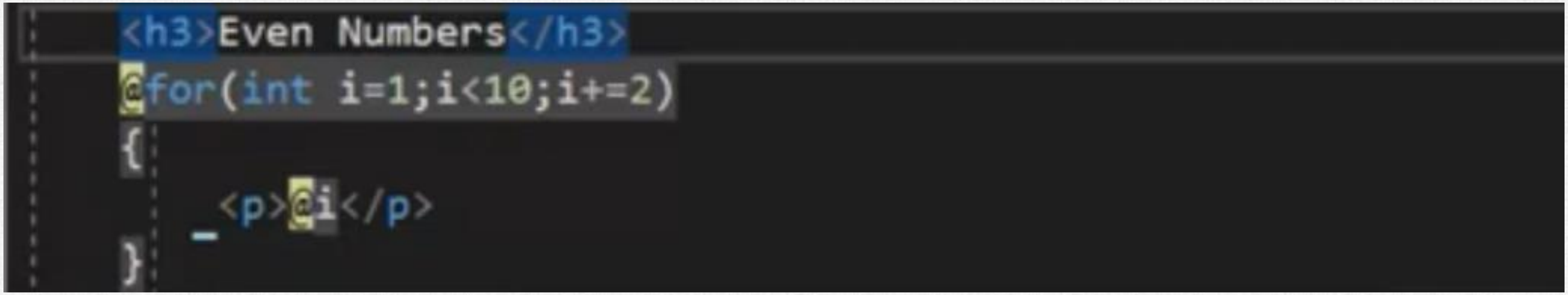
```
        var name = "Asad";
        var msg = "This is Razor Syntax";
        var a = 2;
        var b = 4;
        var res = a + b;


    }

  <h3>This is index page of shared folder</h3>
  <p>@DateTime.Now</p>
  <p>Name: @name  and Message: @msg</p>
  <p>Sum is: @res</p>
/body>
```

If-Else and Loops in Razor Syntax ASP.NET Core

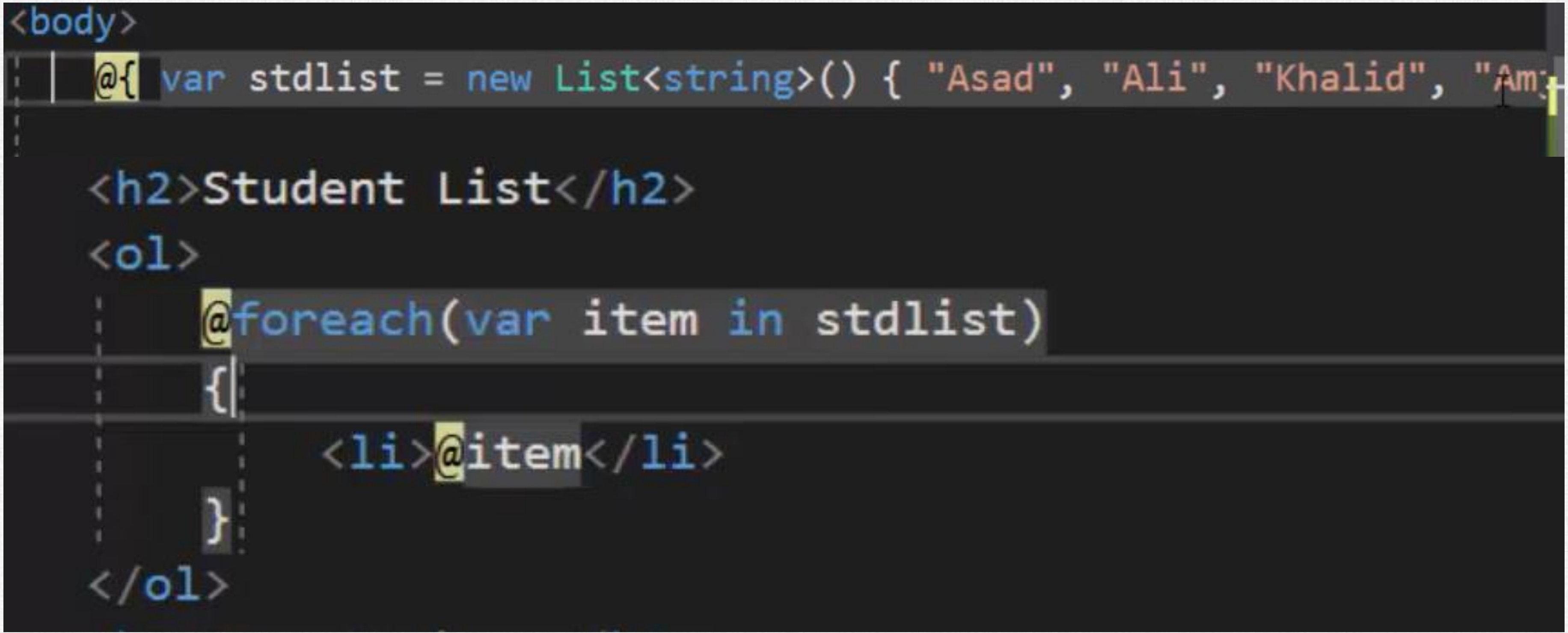


```
@{
    var name = "ASAD";
    var msg = "";
    if (name == "ASAD")
    {
        msg = "Value Successed";
    }
    else
    {
        msg = "Value is wrong";
    }
<h3>This is index page of shared folder</h3>
<hr />
<h2>@msg</h2>
<hr />
```

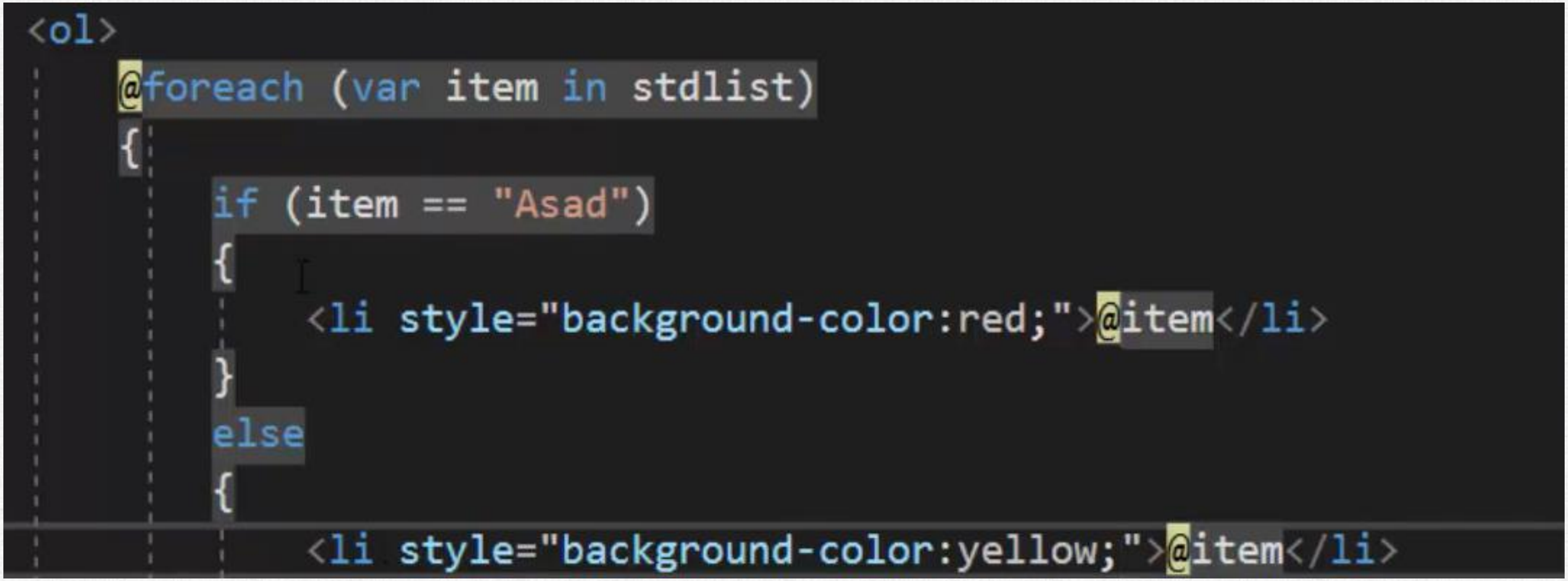
Loops:-



```
    <h3>Even Numbers</h3>
    @for(int i=1;i<10;i+=2)
    {
        <p>@i</p>
    }
```



```
<body>
    @{ var stdlist = new List<string>() { "Asad", "Ali", "Khalid", "Am

    <h2>Student List</h2>
    <ol>
        @foreach(var item in stdlist)
        {
            <li>@item</li>
        }
    </ol>
```



```
<ol>
    @foreach (var item in stdlist)
    {
        if (item == "Asad")
        {
            <li style="background-color:red;">@item</li>
        }
        else
        {
            <li style="background-color:yellow;">@item</li>
```

Thanks