Razor Syntax

Razor is one of the view engines supported in ASP.NET MVC. Razor allows you to write a mix of HTML and server-side code using C# or Visual Basic. Razor view with visual basic syntax has .vbhtml file extension and C# syntax has .cshtml file extension.

Razor syntax has the following Characteristics:

* **Compact**: Razor syntax is compact, enabling you to minimize the number of characters and keystrokes required to write code.
* **Easy to Learn**: Razor syntax is easy to learn where you can use your familiar language C# or Visual Basic.
* **Intellisense**: Razor syntax supports statement completion within Visual Studio.

Inline expression

Start with @ symbol to write server-side C# or VB code with HTML code. For example, write @Variable\_Name to display the value of a server-side variable, e.g., DateTime.Now returns the current date and time. So, write @DateTime.Now to display the current date and time, as shown below. A single line expression does not require a semicolon at the end of the expression.

C# Razor Syntax

<h1>Razor syntax demo</h1>

<h2>@DateTime.Now.ToShortDateString()</h2>

Output:

**Razor syntax demo**

08-09-2014

Multi-statement Code block

You can write multiple lines of server-side code enclosed in braces @{ ... }. Each line must ends with a semicolon the same as C#.

Example: Server side Code in Razor Syntax

 Copy

@{

var date = DateTime.Now.ToShortDateString();

var message = "Hello World";

}

<h2>Today's date is: @date </h2>

<h3>@message</h3>

Output:

Today's date is: 08-09-2014

Hello World!

Display Text from Code Block

Use @: or <text>/<text> to display texts within code block.

Example: Display Text in Razor Syntax

@{

var date = DateTime.Now.ToShortDateString();

string message = "Hello World!";

@:Today's date is: @date <br />

@message

}

Output:

Today's date is: 08-09-2014

Hello World!

Display text using <text> within a code block, as shown below.

Example: Text in Razor Syntax

@{

var date = DateTime.Now.ToShortDateString();

string message = "Hello World!";

<text>Today's date is:</text> @date <br />

@message

}

Output:

Today's date is: 08-09-2014

Hello World!

if-else condition

Write if-else condition starting with @ symbol. The if-else code block must be enclosed in braces { }, even for a single statement.

Example: if else in Razor

 Copy

@if(DateTime.IsLeapYear(DateTime.Now.Year) )

{

@DateTime.Now.Year @:is a leap year.

}

else {

@DateTime.Now.Year @:is not a leap year.

}

Output:

2014 is not a leap year.

for loop

Example: for loop in Razor

 Copy

@for (int i = 0; i < 5; i++) {

@i.ToString() <br />

}

Output:

0

1

2

3

4

Model

Use @model to use model object anywhere in the view.

Example: Use Model in Razor

 Copy

@model Student

<h2>Student Detail:</h2>

<ul>

<li>Student Id: @Model.StudentId</li>

<li>Student Name: @Model.StudentName</li>

<li>Age: @Model.Age</li>

</ul>

Output:

**Student Detail:**

- Student Id: 1

- Student Name: John

- Age: 18

Declare Variables

Declare a variable in a code block enclosed in brackets and then use those variables inside HTML with @ symbol.

Example: Variable in Razor

 Copy

@{

string str = "";

if(1 > 0)

{

str = "Hello World!";

}

}

<p>@str</p>

Output:

Hello World!

Razor View engine is a markup syntax which helps us to write HTML and server-side code in web pages using C# or VB.NET. It is server-side markup language however it is not at all a programming language.

Razor is a templating engine and ASP.NET MVC has implemented a view engine which allows us to use Razor inside of an MVC application to produce HTML. However, Razor does not have any ties with ASP.NET MVC.

Now, Razor Syntax is compact which minimizes the characters to be used, however it is also easy to learn.

Some of Razor Syntax Rules for C# are given below.

* It must be always enclosed in @{ ... }
* Semicolon “;” must be used to ending statements
* Files have .cshtml extension.
* Variables are declared with var keyword
* Inline expressions (variables and functions) start with @
* C# code is case sensitive

Now, let’s see how to write Razor code:

**Variables**

// Using the var keyword:

**var** greeting = "Welcome to Razor";

**var** counter = 200;

**var** day = DateTime.Today;

// Using data types:

string greeting = "Welcome to Razor ";

**int** counter = 200;

DateTime day = DateTime.Today;

Now, in the above example, we have declared variables using var keyword as well as by their types i.e. string, int, DateTime. You can go for any of them. However, it uses int, float, decimal, bool and string data types.

**Conditions**

**If statement**

It starts with code block and its condition is written in parenthesis. And the code which needs to be executed once condition gets satisfied is written inside braces.

Let’s understand with the below example.

@{**var** price=60;}

<html>

   <body>

      @**if** (price>50)

       {

          <p>The price is  greater than 50.</p>

       }

   </body>

</html>

In the above example, we have declared var price has a value of 60 and we are using if statement which is validating price. If the price is greater than 50 then the code written in side braces gets executed.

**If – Else statement**

It starts with code block and its condition is written in parenthesis. And code which needs to be executed once the condition gets satisfied is written inside braces and if it does not gets satisfied then code written inside else block gets executed.

Let’s understand with the below example.

1. @{**var** price=60;}
2. <html>
3. <body>
4. @**if** (price>50)
5. {
6. <p>The price is  greater than 50.</p>
7. }
9. **else**
10. {
11. <p>The price is less than 50.</p>
12. }
14. </body>
15. </html>

In the above example, we have declared var price having a value of 60 and we are using if statement which is validating price. If the price is greater than 50 then the code written inside braces gets executed else the price is less than 50 and it will get executed.

**Switch statement :**

It is used to check multiple individual conditions.

@{

string month = String.Format("{0:MMMM}", DateTime.Now);

**var** message = "";

}

<html>

  <body>

  @ {

**switch** (month)

        {

**case** "January":

                message = "Starting of Year.";

**break**;

**case** "June":

                message = "Middle of Year";

**break**;

**case** "December":

                message = "End of Year";

**break**;

        }

        @message;

  </body>

</html>

**Loops**

Loops are used for executing the  same statement multiple times.

**For Loops**

If you know how many times you want to run the loop, you can use a For loop.

1. <html>
2. <body>
3. @**for**(**var** i = 0; i < 10; i++)
4. {<p>No @i</p>}
5. </body>
6. </html>

**For Each Loops**

Foreach loop is used when you are working with collections or an array.

1. <html>
2. <body>
3. <ul> @foreach (**var** x **in** Request.ServerVariables) {
4. <li>@x</li>} </ul>
5. </body>
6. </html>

**Model**

If you want to use model in, then you must declare the model in the view;  for this you must use @model shown as below.

***@****model modelname*

Let's understand in the below example.

1. @model Employee
2. <h2>Employee Detail:</h2>
3. <ul>
4. <li>Student Id: @Model.EmployeeId</li>
5. <li>Student Name: @Model. EmployeeName</li>
6. <li>Age: @Model.Age</li>
7. </ul>

Now, in the above example, we have used Employee model which is created in the Model folder by declaring @model Employee.

And by using @Model, we are using its properties.

## ****Razer View Syntax in MVC****

In this article, we will discuss razor view syntax in the ASP.NET MVC application.

Use @ symbol to switch between C# code and HTML.

@for (int i = 1; i <= 10; i++)

{

**<b>**@i**</b>**

}

**Output:**  
**1 2 3 4 5 6 7 8 9 10**

Use @{ } to define a code block. If we want to define some variables and perform calculations, then use code block. The following code block defines 2 variables and computes the sum of the first 10 even and odd numbers.

@{

int SumOfEvenNumbers = 0;

int SumOfOddNumbers = 0;

for (int i = 1; i <= 10; i++)

{

if (i % 2 == 0)

{

SumOfEvenNumbers = SumOfEvenNumbers + i;

}

else

{

SumOfOddNumbers = SumOfOddNumbers + i;

}

}

}

**<h3>**Sum of Even Numbers = @SumOfEvenNumbers**</h3>**

**<h3>**Sum of Odd Numbers = @SumOfOddNumbers**</h3>**

**Output:**  
**Sum of Even Numbers = 30**

**Sum of Odd Numbers = 25**

**Use <text> element or @: to switch between c# code and literal text**

@for (int i = 1; i <= 10; i++)

{

**<b>**@i**</b>**

if (i % 2 == 0)

{

**<text>** - Even **</text>**

}

else

{

**<text>** - Odd **</text>**

}

**<br** **/>**

}

###### **The above program can be re-written using @: as shown below.**

@for (int i = 1; i <= 10; i++)

{

**<b>**@i**</b>**

if (i % 2 == 0)

{

@: - Even

}

else

{

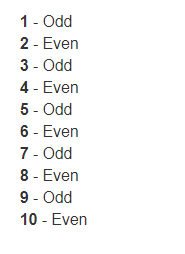
@: - Odd

}

**<br** **/>**

}

**Output:**



###### **Use @\* \*@ to comment in razor views**

**@\*This is a comment in razor views\*@**

**The transition between c# expressions and literal text**

@{

int day = 31;

int month = 12;

int year = 2013;

}

Date is @day-@month-@year

**Output:**  
Date is 31-12-2013

The @ symbol is used as a code delimiter in razor views. However, the razor is smart enough to recognize the format of internet email address and not to treat the @ symbol as a code delimiter.

This is my email address<br />

<b>info@dotnettutorials.net</b>  
Use @ symbol to escape @

I will meet you @@ office

**Output:**  
I will meet you @ office