

Introduction of View Engine

There are two types of view engine supported by ASP.NET MVC. **The Razor and ASPX view engine View Engines are responsible for rendering the HTML from your views to the browser.** The view engine template will have different syntax for implementation.

What is @Razor ?

Razor using an @ character instead of aspx's <% %> and Razor does not require you to explicitly close the code-block, this view engine is parsed intelligently by the run-time to determine what is a presentation element and what is a code element. Razor view engine is compatible with unit testing frameworks. **This is the default view engine in MVC 3 and MVC 4. Web pages with Razor syntax have the special file extension cshtml (Razor with C#) or vbhtml (Razor with VB).**

What is <%ASPX%> ?

This is the default view engine for MVC 1 and MVC 2. The syntax for writing views with this engine is the same syntax that the ASP.NET Web Forms uses and the file extensions are also taken from ASP.NET Web Form (.aspx, .ascx, .master) .

Syntax and Uses of @Razor and <%ASPX%>

It will be easy for ASP coder to learn about ASPX View Engine, and if you are comfortable with C# language then you can easily go with Razor view engine. Let's go through how we can use Razor and or ASPX view engine in our application. Let's start reviewing some of the usages. In this article I am try to cover both syntax with each other by grouping, this help us to understand and compare the syntax also it will help to you to remember easily.

Comment in ASPX and Razor View Engine

Comments are delimited in Razor syntax by @* *@ where in ASPX comments are delimited by <%-- --%>

?

```
1  @*Your Razor Comment Here*@  
2  <%--Your Razor Comment Here--%>
```

In a C# code block the comment for Razor or ASPX will be // and /* */ comment delimiters. Following code shows comments inside a c# code block.

```
?
1
2
3
4 @{
5     //Razor
6     //Single line comment
7     /* Multi
8        line comment
9        */
10
11 <%
12 //ASPX
13 //Single line comment
14 /* Multi
15 line comment
16 */
17 %>
```

Code Block in ASPX and Razor View Engine

Writing a server side code blocks in the source html is shown below for Razor

```
?
1
2 @{
3     string strVariable = "Razor Stirng";
4     bool isRazor = true;
5     string[] arrRazor = new string[] { "MVC", "RAZOR", "AXPS"};
6 }
```

The above code declares a string variable, bool variable and a array variable in Razor and the variables are got initialized with appropriate data in it. This code block is written in .cshtml file for Razor with C# and .vbhtml file for Razor with VB (must use VB syntax). The same code is written in .aspx (ASPX C#).

```
?
1
2 <%
3     string strVariable = "Razor Stirng";
4     bool isRazor = true;
5     string[] arrRazor = new string[] { "MVC", "RAZOR", "AXPS"};
6 %>
```

Razor and ASPX with HTML Markup

Following code block shows how we can integrate the Razor Code block with html markup when working in a dynamic web page we often required mixing code and html tags.

?

```
1 <span>@Your Code</span>
```

The above code is written for Razor(C#) and the below code will achieve the same in ASPX(C#) view engine.

?

```
1 <span><%:YourCode %></span>
```

There may be some other cases , in which we can merge HTML tag with Razor or ASPX view engine.

?

```
1
2 @if (isRazor)
3 {
4     <span>ViewEngine:Razor</span>
5 }
6 else
7 {
8     <span>ViewEngine:Other</span>
9 }
```

The above code used with ASPX view engine will be.

?

```
1
2 <%if (isRazor)
3 {
4     <span>ViewEngine:Razor</span>
5 }
6 else
7 {
8     <span>ViewEngine:Other</span>
9 } %>
```

You can do more mixing of HTML tags with Razor and ASPX View Engine, Following code block shows how you can do that.

?

```
1 Your Sample Html Code or Text @RazorCode (@AnotherRazorCode)
2 Your Sample Html Code or Text <%: ASPXCode %> (<%:AnotherASPXCode %>)
```

When the above code will rendered with the value of RazorCode and AnotherRazorCode variable is replaced as we do in server side coding. Similarly you can assign dynamic HTML attribute for any HTML tags. Let's understand by an example.

```
?  
1 <input type="checkbox" checked="@isRazor">  
2 <input type="checkbox" checked="<%=isASPX%>">
```

Conditional Statement in ASPX and Razor View Engine

Below is a sample Conditional statement for Razor (C#).

```
?  
1  
2 @if(isRazor)  
3 {  
4     <p>View Engine:Razor</p>  
5 } else {  
6     <p> View Engine:Other</p>  
7 }
```

Below is a sample Conditional statement for ASPX (C#).

```
?  
1  
2 <%= if (isASPX)  
3 { %>  
4     <p> View Engine:ASPX </p>  
5 <%= }  
6 else  
7 { %>  
8     <p> View Engine:Other </p>  
9 <%= } %>
```

The above usage gives us a very easy approach for rendering things conditionally.

Looping Statement in ASPX and Razor View Engine

The below statement in Razor view engine is used for rendering the ListItem .

```
?  
1  
2 <ul>  
3     @foreach (var item in ListItem)  
4     {  
5         <li>@item</li>  
6     }  
7 </ul>
```

Razor view engine is smart enough to identify the tag as an html tag and the @item as a server side value. The will be rendered multiple times with the list of products.

?

```
1
2 <ul>
3     <%foreach (var item in ListItem)
4         { %>
5         <li><%=item%></li>
6         <%} %>
7 </ul>
```

Calling Server Side in Method in ASPX and Razor View Engine

To understand to call let's assume that we have a static method named ViewEngineMethod in a class and called inside a tag using the appropriate view engines syntax.

?

```
1 <p>@{RazorASPXViewEngine.Models.SampleUtility.ViewEngineMethod();}</p>
```

This code will be rendering with the output returned from the ViewEngineMethod()

?

```
1 <p><%=RazorASPXViewEngine.Models.SampleUtility.ViewEngineMethod();%></p>
```

Creating a Delegate in Method in ASPX and Razor View Engine

The delegates Razor can be define as templates build by C# and HTML. Here we have used Func instead of using Func. and we have used the 'item' to get the value inside the Func<>.

?

```
1
2
3
4 @{
5     Func<string[], object=""> SomeList =
6     @<ul>
7         @foreach (string p in item)
8         {
9             <li>@p</li>
10        }
11    </ul>;
12 }
13
14 Some listing from delegate-
15 @SomeList(ListItem);
16
17 </string[],>
```

Creating a method using ASPX and Razor View Engine

The @functions block in Razor helps you to define utility functions directly in the view.

?

```
1
2 @functions{
3     public string myMethod()
4     {
5         return "My Method";
6     }
7     Name:@myMethod();
```

We should create the function only when the function is specific to the view and can't be done using other conditional statement. By using @function we can separate view specific logic from controller.

Escape Character in ASPX and Razor View Engine

To escape an @ symbol just add @@ and it will render a single @ symbol. Razor language parser is very smart to know the @ symbol is used for a static content or dynamic content. Let's take an example to understand this.

?

```
1 @@@ViewEngine
2 @@ViewEngine
3 example@DotNet-Tutorial.com
```

The first line will print @C# whereas the second line will print @@ViewEngine. The third one will be parsed intelligently by the parser and will print the email address example@DotNet-Tutorial.com. The second line printed @@ViewEngine because of the escape symbol the variable will not be considered/evaluated.

Summary: Razor view engine is widely used because of the lightweight syntax, readability, maintainability, TDD, and more. Razor uses @ symbol whereas ASPX uses <% %>. File extensions are different for Razor (C#-cshtml, VB-vbhtml) and ASPX(aspx, ascx, master). Namespace for Razor is System.Web.Razor and for ASPX it's System.Web.Mvc.WebFromViewEngine. Razor syntax is very clean and easy to learn than ASPX view engine.