**View Engine**

Before stating the description of View Engine. We discuss one thing that is suppose we create any web form related to HTML TAGS. That web page will be treated as static but suppose we want to add some c# or any other programming language codes in between HTML so it is not possible. So View Engines provides the facility to mix HTML and Programming Codes.

So in Asp.net we have the two options regarding the View Engine

1. **ASPX View Engine or Web Form View Engine (Old)**
2. **Razor View Engine. (Latest)**

**Note: Some third party View Engine (like Spark , Nhaml, Brail, SharpTiles, Hasic and many more) support is also available in ASP.NET we give the preference to default Veiw Engine of ASP.NET**

**Difference between ASPX View Engine and Razor View Engine**

|  |  |  |
| --- | --- | --- |
| **s.no** | **Razor View Engine** | **ASPX or Web Form View Engine** |
| 1 | It is advance view engine that was introduce with MVC3 | It is the default View Engine for Asp.net MVC and it is included from the beginning. |
| 2 | The name space for the Razor View Engine is **System.Web.Razor** | The name space for the Aspx View Engine is **System.Web.Mvc.WebFormViewEngine** |
| 3 | The extension of Razor View Engine is **.cshtml (Razor with c#)** | The extension of Aspx View Engine is **.aspx** |
| 4 | Razor has new and advance syntax which is easy to understand and usable and provide very clear view and reduce typing. | Comparatively ASPX View Engine Code is complicated |
| 5 | Razor use **@** symbol to make the code  **For Eg @{**  **int a=2,b=4,c=0;**  **c=a+b;**  **}** | Aspx View Engine use **<% and %>**  **For Eg <%:**  **int a=2,b=4,c=0;**  **c=a+b;**  **%>** |
| 6 | Razor does not support design mode in visual studio mean we cannot see the design without running the application. | While Aspx support the design mode in visual studio we can see the design without running application. |
| 7 | Razor View is little bit slow in comparison of Aspx View | Comparatively faster |
| 8 | Razor View Engine prevent **Cross Site Scripting Attack**s means it encodes the script or html tags before rendering to view | Aspx View Engine does not prevent |
| 9 | Razor Engine Support Test Driven Deployment | Aspx doesn’t support Test Driven Deployment |
| 10 | The Razor View Engine is compatible with a unit testing framework | Not very much compatible with the unit testing |
| 11 | Razor also supports the concept of layout pages | Aspx support the concept of Master Page |

**Advantages of Razor View Engine**

1. It is easy to use and implement
2. Razor helps us to minimize the coding and provide us a fast and fluid coding work flow.
3. Powerful built-in validation of markup that helps us to avoid unwanted runtime exceptions due to errors in the view.
4. The code looks clean.
5. Razor does not require any special tool to write markup. We can also write our markup code with any old plain text editor like Notepad.
6. The @model directive provides a cleaner and more concise way to define a strongly typed model
7. Improves readability of the markup and code

**Razor Code Syntax**

1. **Razor Expression**

**These are single Line C# Statement**

**@Statements**

1. **Razor Block**

**These are multiline Statement**

**@**

**{**

**Statements**

**}**

**Note: So due to these Razor engine is capable to identify which is HTML code and What are c# codes**

|  |  |
| --- | --- |
|  |  |