



Syntax Razor

José Antônio da Cunha

Sintaxe Razor

Nesta apresentação será mostrada a sintaxe usada pelo Razor, o view engine usado nas aplicações ASP .NET MVC3+ e no Web Matrix.

A sintaxe **Razor** embute em uma mesma página web código que roda no servidor, marcações HTML, seletores CSS e código de linguagens de script, como JavaScript.

O caractere **@** é usado para declarar blocos de instruções, retornar o resultado de expressões, invocar métodos, acessar propriedades, declarar variáveis, criar uma instância de uma classe, etc.

Sintaxe Razor

A sintaxe **Razor** é extremamente simples. Por exemplo, para retornar a data atual, usamos a linha a seguir:

```
<p>@DateTime.Now</p>
```

As instruções são iniciadas com o caractere @ e cercada por chaves:

```
@{  
    string nome = "José Antônio da Cunha";  
    var data = DateTime.Now;  
}
```

Sintaxe Razor

Variáveis – você pode declarar variáveis normalmente definindo o tipo de dados:

```
@{  
    string nome = "José Antônio da Cunha";  
    string[] cores = new String[3];  
    int x = 10;  
    int y = 50;  
    int resultado = x + y;  
}  
<!DOCTYPE html>  
<html>  
    <body>  
        <div>  
            <p>Total: @resultado</p>  
            <p>Multiplicação: @(x*y)</p>  
        </div>  
    </body>  
</html>
```

Sintaxe Razor

Ou usando a palavra-chave **var**:

```
@{  
    var nome = "José Antônio da Cunha";  
    var cores = new string[3];  
    var x = 10;  
    var y = 50;  
    var resultado = x + y;  
}
```

Sintaxe Razor

Comentários – no Razor, comentários começam com `@*` e termina com `*@`. Isso se aplica a comentários em múltiplas linhas ou em uma linha.

`@{`

`@*Este comentário tem uma única linha*@`

`@*`

Comentário dividido em várias linhas

`*@`

`}`

Se preferir, use os caracteres de comentários do C#:

`@{`

`@*Este comentário tem uma única linha*@`

`/*`

Comentário dividido em várias linhas

`*/`

`}`

Sintaxe Razor

Instrução if – a instrução if retorna verdadeiro ou falso com base em um teste específico:

```
@{
    int ano = DateTime.Now.Year;
    if ( ano > 2012) {
        <p>Ano: @DateTime.Now.Year</p>
    }
    else if ( ano < 2012) {
        <p>Ano: 2011</p>
    }
    else {
        <p>Ano: indefinido.</p>
    }
}
```

Sintaxe Razor

Instrução switch – a instrução switch testa se determinado valor combina com uma instrução **case**. Se retorna verdadeiro, o código da instrução **case** é executado.

```
@{  
    int mes = DateTime.Now.Month;  
    string mensagem = "";  
    switch (mes) {  
        case 1:  
            mensagem = "Janeiro";  
            break;  
        case 2:  
            mensagem = "Fevereiro";  
            break;  
        case 3:  
            mensagem = "Março";  
            break;  
        case 4:  
            mensagem = "Abril";  
            break;  
        case 5:  
            mensagem = "Maio";  
            break;  
    }  
}
```

```
<!DOCTYPE html>  
<html>  
    <body>  
        <div>  
            <p>  
                @mensagem  
            </p>  
        </div>  
    </body>  
</html>
```


Sintaxe Razor

Instrução for – a instrução for é caracterizada como loop contador, ou seja, você percorre os extremos conhecidos:

```
@{  
    for (int i = 0; i < 10; i++) {  
        <p>Número: @i </p>  
    }  
}
```

Sintaxe Razor

Instrução foreach – a instrução foreach é usada para percorrer itens de uma coleção:

```
@{  
    string[] cidades = new string[3];  
    cidades[0] = "Natal";  
    cidades[1] = "Caicó";  
    cidades[2] = "Acarí";  
  
    <ul>  
        @{ foreach (var cidade in cidades) {  
            <li>@cidade</li>  
        }  
    </ul>  
}
```

Sintaxe Razor

Propriedades e métodos

Com o caractere @ acessamos propriedades:

```
@{  
    @Request.UserLanguage[0] <br />  
    @Request.Url <br q>  
    @Request.UserAgent  
}
```

E invocamos métodos:

```
@{Html.RenderAction ("ListaProdutos");  
  
@{  
    @Server.HtmlEncode("<b>tags HTML desativadas. </b>");  
}
```

Sintaxe Razor

Manipulando exceções – para manipular exceções com Razor, basta incluir um bloco try e catch. Exemplo:

```
@{  
    try {  
        var fs = File.Open(@"c:\teste1.txt", FileMode.Open);  
    }  
    catch (FileNotFoundException) {  
        <p>Arquivo não encontrado.</p>;  
    }  
    catch (Exception) {  
        <p>Erro detectado.</p>  
    }  
}
```

Sintaxe Razor

Conversões – a sintaxe Razor possui diversos métodos que nos permitem forçar uma conversão explícita de um tipo de dados para outro.

Tabela 1 Mostra os métodos usados na conversão e verificação de dados.

Método	Descrição
AsInt(), IsInt()	AsInt() converte uma string para um inteiro. IsInt() verifica se é um tipo int válido.
AsBool(), IsBool()	AsBool() converte string para booleano (true, false). IsBool() verifica se é um tipo Bool
AsFloat(), IsFloat()	AsFloat() converte uma string para um tipo flutuante. IsFloat() verifica se é um tipo flutuante válido.
AsDecimal(), IsDecimal()	AsDecimal() converte uma string para um tipo decimal. IsDecimal() verifica se é um tipo decimal válido.
AsDateTime(), IsDateTime()	AsDateTime() converte uma string para um tipo data e hora. IsDateTime() verifica se é um tipo DateTime válido.
ToString()	Converte qualquer tipo de dado em um tipo string.

Sintaxe Razor

Exemplo de conversão de tipo:

```
@{  
    string numero = "12345";  
    int i = 0;  
  
    if (!numero.IsInt()) {  
        i = numero.AsInt();  
    }  
    <p>@i</p>  
}
```

Sintaxe Razor

Operadores – os operadores do C# podem ser usados normalmente com a sintaxe Razor.

Exemplo:

```
@{  
    int x = 12;  
    int y = 5;  
    int resultado = 0;  
    if (x >= 10 && y < 7) {  
        resultado = y + y;  
    }  
    else {  
        resultado = y - y;  
    }  
}  
...  
<div>  
    <p>@resultado</p>  
</div>  
...
```

Sintaxe Razor

Href – o método Href converte endereços relativos de uma imagem, arquivo CSS, páginas web para um endereço que o navegador entende. Exemplo:

```
  
<link rel="stylesheet" type="text/css" href="@Href("~/Content/Site.css")" />
```

O mecanismo de exibição ASPX usa o método Resolver. Exemplo:

```
<%:Html.Image("idImag", ResolveUrl("~/Content/imagem.jpg"), "Descrição") %>
```


Sintaxe Razor

Associando classes – a sintaxe **Razor** usa a palavra-chave *model* para associar classes a um view:

```
@model IEnumerable<AppMeuProjeto.Models.Clientes>
```

Para acessar membros da classe Clientes é simples. Exemplo:

```
<ul>  
  @{ foreach (var item in Model)  
    {  
      <li>  
        @item.Nome;  
      </li>  
    }  
  }  
</ul>
```