

# Razor of MVC 3

## Quick Reference

Based on scottgu's blog

Think Tan

2011-4-22

# 语法 -- @

```
<h1>Razor Example</h1>
```

```
<h3>
```

```
    Hello @name, the year is @DateTime.Now.Year
```

```
</h3>
```

```
<p>
```

```
    Checkout <a href="/Products/Details/@productId">this product</a>
```

```
</p>
```

对象

```
First Product: @products[0].ProductName costs: @products[0].UnitPrice
```

数组/属性

```
The @products.Count() products cost @products.Sum(p=>p.UnitPrice) dollars.
```

函数

```

```

显式定义代码块

# 语法 – 循环 & 嵌套HTML

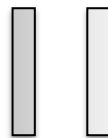
```
<ul id="products">  
    @foreach(var p in products) {  
        <li>@p.Name ($@p.Price)</li>  
    }  
</ul>
```

# 语法 – if语句

```
@if(products.Count == 0) {  
    <p>Sorry - no products in this category</p>  
} else {  
    <p>We have a products for you!</p>  
}
```

# 语法 - 多行语句

```
@{  
    int number = 1;  
    string message = "Number is" + number;  
}  
  
<p>Your Message: @message</p>
```



```
<p>Your Message: @("Number is: " + number)
```

# 嵌套内容识别- 隐式地通过标签

```
@if (DateTime.Now.Year == 2010) {  
    <span>  
        if year is 2010 then print this <br/>  
        multi-line text block and  
        the date: @DateTime.Now  
    </span>  
}
```

## 嵌套内容识别 – 显式通过@:

```
<ul>
  @foreach (var p in products) {
    <li>
      @p.ProductName

      @if (p.UnitsInStock == 0) {
        @: (Out of stock!)
      }
      else if (p.UnitsInStock < 4) {
        @: (Only @p.UnitsInStock left!)
      }
    </li>
  }
</ul>
```

# 嵌套内容识别 - 显式通过 <text>

```
@if (p.UnitsInStock == 0) {  
    <text>  
        This is a multi-line block of content.  
        The text tag wraps us and will be removed by the  
        Razor parser. We can still have code  
        nuggets too: @DateTime.Now  
        Isn't that neat?  
    </text>  
}
```



# 布局（母版页） – layout.cshtml

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple Site</title>
  </head>
  <body>

    <div id="header">
      <a href="/">Home</a>
      <a href="/About">About</a>
    </div>

    <div id="body">
      @RenderBody()
    </div>

  </body>
</html>
```

# 布局 – content.cshtml

```
@{  
    LayoutPage = "SiteLayout.cshtml";  
}
```

```
<h1>About This Site</h1>
```

```
<p>
```

This is some content that will make up the "about" page of our web-site. We'll use this in conjunction with a layout template. The content you are seeing here comes from the Home.cshtml file.

```
</p>
```

```
<p>
```

And obviously I can have code in here too. Here is the current date/time: @DateTime.Now

```
</p>
```

# 布局 - 添加多个内容区域(1)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Simple Site</title>
  </head>
  <body>

    <div id="header">
      <a href="/">Home</a>
      <a href="/About">About</a>
    </div>

    <div id="left-menu">
      @RenderSection("menu", optional:true)
    </div>

    <div id="body">
      @RenderBody()
    </div>

    <div id="footer">
      @RenderSection("footer", optional:true)
    </div>

  </body>
</html>
```

## 布局 - 添加多个内容区域(2)

```
<h1>About This Site</h1>

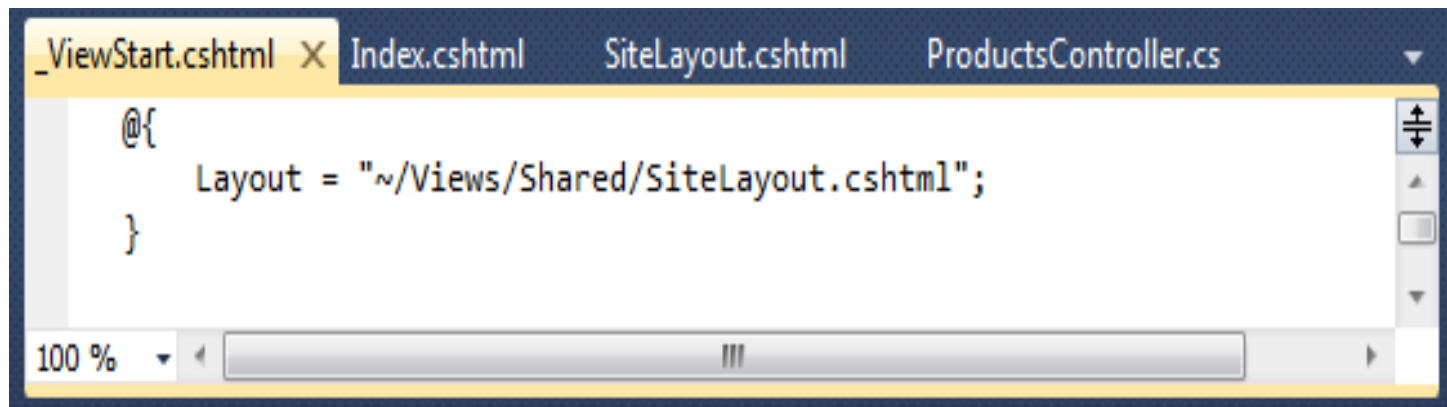
<p>
  This is some content that will make up the "about"
  page of our web-site. We'll use this in conjunction
  with a layout template. The content you are seeing here
  comes from the Home.cshtml file.
</p>
<p>
  And obviously I can have code in here too. Here is the
  current date/time: @DateTime.Now
</p>

@section menu {
  <ul id="sub-menu">
    <li>About Item 1</li>
    <li>About Item 2</li>
  </ul>
}

@section footer {
  <p>This is my custom footer for Home</p>
}
```

# 布局 – 使用\_ViewStart.cshtml设定默认值

- \_ViewStart.cshtml在View最前面执行



```
@{  
    Layout = "~/Views/Shared/SiteLayout.cshtml";  
}
```

# Helpers

- **HtmlHelper**
  - 用于在View中生成HTML控件
  - [http://msdn.microsoft.com/zh-cn/library/system.web.mvc.htmlhelper\(v=VS.98\).aspx](http://msdn.microsoft.com/zh-cn/library/system.web.mvc.htmlhelper(v=VS.98).aspx)
- **AjaxHelper**
  - 在View中生成Ajax式样的HTML.
  - [http://msdn.microsoft.com/zh-cn/library/system.web.mvc.ajaxhelper\(v=VS.98\).aspx](http://msdn.microsoft.com/zh-cn/library/system.web.mvc.ajaxhelper(v=VS.98).aspx)
- **UrlHelper**
  - 用于基于Controller/Action等参数生成链接字符串
  - [http://msdn.microsoft.com/zh-cn/library/system.web.mvc.urlhelper\(v=VS.98\).aspx](http://msdn.microsoft.com/zh-cn/library/system.web.mvc.urlhelper(v=VS.98).aspx)
- **Expression**
  - 通过表达式获得Model名字
  - [http://msdn.microsoft.com/zh-cn/library/system.web.mvc.expressionhelper\(v=VS.98\).aspx](http://msdn.microsoft.com/zh-cn/library/system.web.mvc.expressionhelper(v=VS.98).aspx)

# Helps -- HtmlHelper

```
<fieldset>
  <legend>Edit Product</legend>

  <div>
    @Html.LabelFor(m => m.ProductID)
  </div>
  <div>
    @Html.TextBoxFor(m => m.ProductID)
    @Html.ValidationMessageFor(m => m.ProductID)
  </div>
</fieldset>
```

# Helpers – Helper方法定义与重用

```
@helper ProductListing(List<Product> products) {  
    <ul id="products">  
        @foreach(var p in products) {  
            <li>@p.Name ($@p.Price)</li>  
        }  
    </ul>  
}
```

```
<body>  
    <h1>Here are My Products</h1>  
    <div>  
        @ProductListing(Model.Products)  
    </div>  
</body>
```

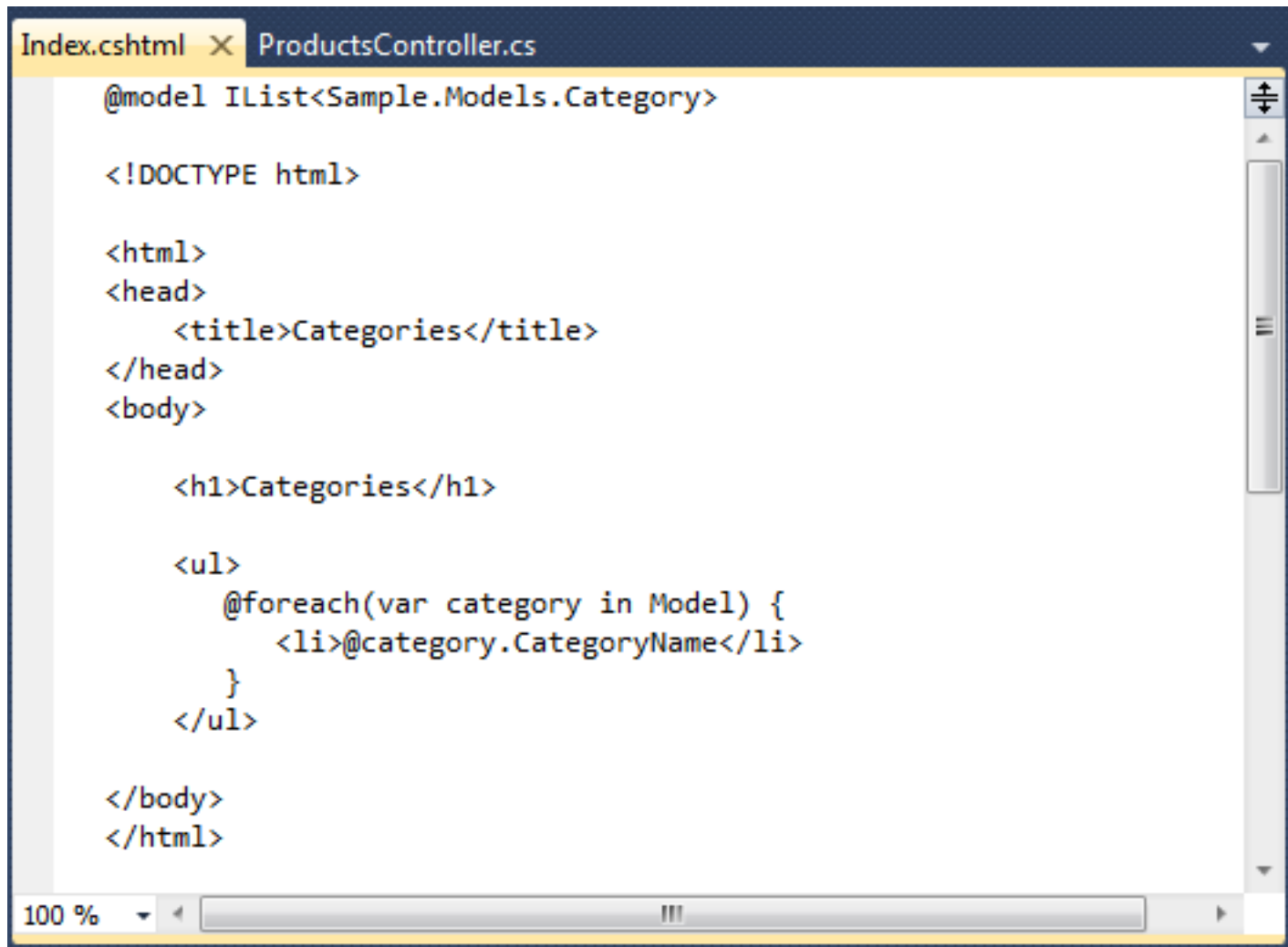


# 使用内联模板传参

```
<h1>Small Bakery Products</h1>
```

```
@Grid.Render(  
  data: Model.Products  
  tableStyle: "grid",  
  headerStyle: "head",  
  alternatingRowStyle: "alt",  
  columns: Grid.Columns(  
    Grid.Column("Name", "Product", style: "product"),  
    Grid.Column("Description", format:@<i>@item.Description</i>),  
    Grid.Column("Price", format:@<span>$@item.Price</span>)  
  )  
)
```

# 强类型View -- @model



```
Index.cshtml X ProductsController.cs

@model IList<Sample.Models.Category>

<!DOCTYPE html>

<html>
<head>
    <title>Categories</title>
</head>
<body>

    <h1>Categories</h1>

    <ul>
        @foreach(var category in Model) {
            <li>@category.CategoryName</li>
        }
    </ul>

</body>
</html>
```

注释@\* \*@

