



THE MULTI-TALENTED RAZOR

*In support of lauded appearance in MVC3 as “View Engine”, Razor
makes appearance in DotNetNuke playing the part of “Scripting
Language”*

INTRODUCING...RAZOR

- ❖ View Engine? Scripting language? Templating language?
- ❖ Gu Says: “HTML generation ... code-focused templating”
- ❖ Clean syntax, fewer characters
- ❖ Code is “first class”

SYNTAX

❖ ASPX

```
<p>Hello Ian, the year is <%= DateTime.Now.Year %></p>
```

❖ Razor

```
<p>Hello Ian, the year is @DateTime.Now.Year</p>
```

SYNTAX

❖ ASPX

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

❖ Razor

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


HELPERS

```
@helper SayHello(string yadda)
{
    <p>Text: @yadda</p>
}

@SayHello("Hello, world!")
```

ASP.NET MVC 3

- ❖ Ships with choice of ASPX or Razor as view engine
- ❖ Displaying and working with data (your model)
- ❖ Cool: Automatic HTML Encoding

ASP.NET MVC 3 is an Open Source MVC Framework on .NET

Version 3 Released January 2011

ASP.NET MVC 3 DEMO

LogOn.cshtml x

Client Objects & Events

```
@model RazorViewEngine.Models.LogOnModel

@{
    ViewBag.Title = "Log On";
}

<h2>Log On</h2>
<p>
    Please enter your username and password. @Html.ActionLink("Register", "Register", "Account")
</p>

<script src="@Url.Content("~/Scripts/jquery.validate.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")" type="text/javascript"></script>

@Html.ValidationSummary(true, "Login was unsuccessful. Please correct the errors.")

@using (Html.BeginForm()) {
    <div>
```




WEB MATRIX

- ❖ Think “PHP Story”
- ❖ “Entry level” application creation
- ❖ “Web Pages” .cshtml
- ❖ In this world...
 - Razor scripting allows dynamic features

WEB MATRIX DEMO

```
@{
    Page.Title = "Home";

    var db = Database.Open("bakery");
    var products = db.Query("SELECT * FROM PRODUCTS").ToList();
    var featured = products[new Random().Next(products.Count)];
}

<h1>Welcome to Fourth Coffee!</h1>

<div id="featuredProduct">
    
        <div id="productInfo">
            <h2>@featured.Name</h2>
            <p class="price">$@string.Format("{0:f}", featured.Price)</p>
            <p class="description">@featured.Description</p>
        </div>
```

DOTNETNUKE

- ❖ Enable light weight but powerful development
- ❖ Not **the** new approach, but **a** new approach
- ❖ Razor host module
- ❖ Similar to web matrix style development
 - Less the nice IDE
 - Add DNN Context (API)

RAZOR HOST MODULE

▼ Edit Script

🔍 Select Script:

_TabInfo.cshtml

+ Add New Script File

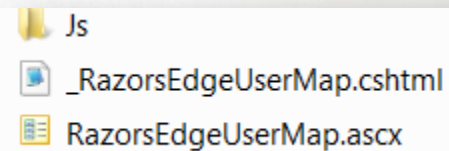
Source File - ~/DesktopModules/RazorModules/RazorHost/Scripts/_TabInfo.cshtml

Script Source:

```
<h2>Razor Page Info Module Example</h2>
<div>The current page is:</div>
<ul>
  <li>Name: @Dnn.Tab.TabName</li>
  <li>Id: @Dnn.Tab.TabID</li>
  <li>Path: @Dnn.Tab.TabPath</li>
</ul>
```


A RAZOR MODULE

- ❖ MyModule.ascx
 - Inherits from RazorModuleBase
- ❖ _MyModule.cshtml
 - Contains html/js/razor
- ❖ Host -> Module Definitions
 - Create new module -> **From Control**



USER MAP DEMO

```
@using DotNetNuke.Security.Roles
@using DotNetNuke.Entities.Users
@using DotNetNuke.Entities.Profile

@functions {
    public static string GetProfileProperty(UserProfile profile, string propertyName)
    {
        ProfilePropertyDefinition objProperty = profile.GetProperty(propertyName);
        if (objProperty == null || string.IsNullOrEmpty(objProperty.PropertyValue)) return string.Empty;
        return objProperty.PropertyValue;
    }
}
```