

# УЕБ-БАЗИРАНО ПРОГРАМИРАНЕ

Павел Кюркчиев

Ас. към ПУ „Паисий Хилендарски“

<https://github.com/pkyurkchiev>

@pkyurkchiev

RAZOR СИНТАКСИС



Какво представлява Razor?

# Какво е Razor?

- Razor представлява синтаксис, който комбинира HTML маркиране със C # код, създаден за улеснение на разработчиците.
- Razor Pages и .Net MVC също използват Razor. За разлика от Blazor, те са разработени около request/response модела.
- В Blazor компонентите се използват специално за логика и композиция от страна на клиента.

# Razor keywords

- page (Requires ASP.NET Core 2.1 or later)
- namespace
- functions
- inherits
- model
- section
- helper (Not currently supported by ASP.NET Core)

# C# Razor keywords

- case
- do
- default
- for
- foreach
- if
- else
- lock
- switch
- try
- catch
- finally
- using
- while

# Reserved keywords not used by Razor

- class

# Implicit Razor expressions

- Implicit Razor expressions започват с отбелязването на @ последвана от C# програмен код.



<p>@DateTime.Now</p>

<p>@DateTime.IsLeapYear(2016)</p>

<p>@await DoSomething("hello", "world")</p>

<p>@GenericMethod<int>()</p>

## Implicit Razor expressions

# Explicit Razor expressions

- Explicit Razor expressions започват с отбелязването на @ и продължават със скоби.

<p>Last week this time:@(DateTime.Now-TimeSpan.FromDays(7))</p>

<p>Last week: @DateTime.Now - TimeSpan.FromDays(7)</p>

<p>@(GenericMethod<int>())</p>

```
@{  
    var joe = new Person("Joe", 33);  
}
```

<p>Age@(joe.Age)</p>

## Explicit Razor expressions

# Директиви (Directives)

- Директивите представляват имплицитен израз от символа @, последван от запазена дума.
  - *@code* или *@functions*
  - *@implements*
  - *@inherits*
  - *@inject*
  - *@layout*
  - *@model*
  - *@namespace*

# @code

- @code блока позволява писането на C# код в страницата.
- Псевдоним на @functions, препоръчва се използването на @code.
- Работи само в Razor components (.razor).

```
@code {  
    // C# members (fields, properties, and methods)  
}
```

@code

# @implements

- Директивата @implements имплементира интерфейс в изгледа.

@implements IDisposable

<h1>Example</h1>

```
@functions {  
    private bool _isDisposed;  
  
    ...  
  
    public void Dispose() => _isDisposed = true;  
}
```

@implements



# @inherits

- Директивата @inherits предава пълен контрол върху класа, който се наследява в изгледа.

```
----- C# class -----  
using Microsoft.AspNetCore.Mvc.Razor;  
public abstract class CustomRazorPage<TModel> : RazorPage<TModel>  
{  
    public string CustomText { get; } =  
        "Gardyloo! - A Scottish warning yelled from a window before dumping" +  
        "a slop bucket on the street below.";  
}
```

```
----- View -----  
@inherits CustomRazorPage<TModel>  
  
<div>Custom text: @CustomText</div>
```

**@inherits**

# @inject

- Директивата @inject дава възможност на Razor Page да инжектира услуга в контейнера за услуги (на DI) в изгледа.

# @layout

- Директивата @layout определя layout (шаблон на изгледа) в самия Razor Component.
- Работи само в Razor components (.razor).

# @model

- Директивата @model определя типа на модела, който се подава към изгледа или страницата.
- Работи само в MVC views и Razor Pages (.cshtml)

@model TypeNameOfModel

<div>The Login Email: @Model.Email</div>

@model

# @namespace

- Директивата @namespace дава възможност за добавяне на пространство от класове към изгледа.

@namespace Your.Namespace.Here

@namespace



# Tag helpers

- Tag helpers позволяват на server-side код да участва в създаването и визуализирането на HTML елементи в Razor файлове (изгледи).
- Tag helpers – могат да бъдат използва в ASP.NET MVC и ASP.NET Razor Pages.
- Tag helpers не може да се използва в ASP.NET Blazor

# Tag helpers

- Добавянето им към ASP.NET Core web app става във Views/\_ViewImports.cshtml

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

@addTagHelper \*, AuthoringTagHelpers

Премахване на Tag helpers, чрез добавянето на  
@removeTagHelper в Views/Folder/\_ViewImports.cshtml

@removeTagHelper

# Tag helpers в Razors

- Anchor Tag Helper
- Cache Tag Helper
- Distributed Cache Tag Helper
- Environment Tag Helper
- FormActionTagHelper
- Form Tag Helper
- Form Action Tag Helper
- Image Tag Helper
- Input Tag Helper
- Label Tag Helper
- LinkTagHelper
- ScriptTagHelper
- Partial Tag Helper
- Select Tag Helper
- Textarea Tag Helper
- Validation Message Tag Helper
- Validation Summary Tag Helper

# <form> tag helper атрибути

Attribute	Description
asp-controller	The name of the controller.
asp-action	The name of the action method.
asp-area	The name of the area.
asp-page	The name of the Razor page.
asp-page-handler	The name of the Razor page handler.
asp-route	The name of the route.
asp-route-{value}	A single URL route value. For example, asp-route-id="1234".
asp-all-route-data	All route values.
asp-fragment	The URL fragment.

```
<form asp-controller="Demo" asp-action="Register" method="post">  
    <!-- Input and Submit elements -->  
</form>
```

## Form - register page

<input asp-for="<Expression Name>">

.NET type	Input Type
Bool	type="checkbox"
String	type="text"
DateTime	type="datetime-local"
Byte	type="number"
Int	type="number"
Single, Double	type="number"

```
using System.ComponentModel.DataAnnotations;
namespace FormsTagHelper.ViewModels
{
    public class RegisterViewModel
    {
        [Required]
        [EmailAddress]
        [Display(Name = "Email Address")]
        public string Email { get; set; }

        [Required]
        [DataType(DataType.Password)]
        public string Password { get; set; }
    }
}
```

## Input - register page

@model RegisterViewModel

```
<form asp-controller="Demo" asp-action="RegisterInput" method="post">  
    Email: <input asp-for="Email" /> <br />  
    Password: <input asp-for="Password" /><br />  
    <button type="submit">Register</button>  
</form>
```

Input - register page



ВЪПРОСИ ?

