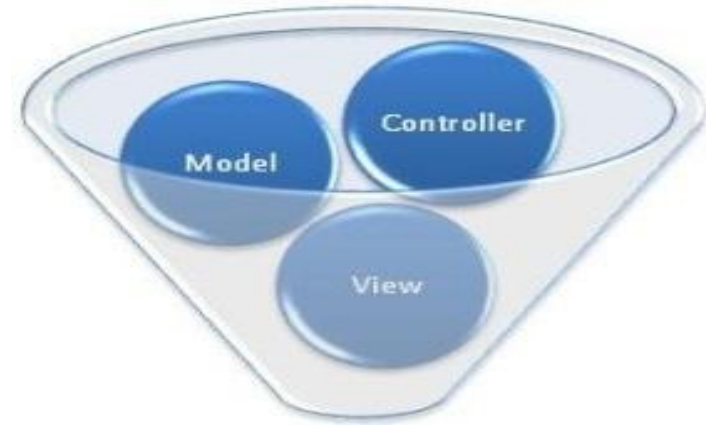


Microsoft
ASP.net MVC 5



RAZOR & HELPERS

ThS. Nguyễn Nghiệm
0913.745.789
nghiemn@fpt.edu.vn
songlong2k@gmail.com

- ☐ Razor
- ☐ MVC Helper
 - ☒ HyperLink Helpers
 - ☒ Form Helpers
 - ☒ Model Helpers
- ☐ Custom Helper

- ❑ Razor là ngôn ngữ ngắn gọn, rõ ràng và hữu ích cho phép bạn tạo ra các giao diện ứng dụng ASP.NET MVC
- ❑ Trong khối lệnh @{...} là mã C# hoặc VB.NET trộn HTML

```
<!-- Khối lệnh đơn -->
```

```
@{ var message = "Hello World"; }
```

```
<!-- Biểu thức nội tuyến -->
```

```
<p>Giá trị của message là: @message</p>
```

```
<!-- Khối nhiều dòng mã lệnh -->
```

```
@{  
    var greeting = "Welcome to our site!";  
    var weekDay = DateTime.Now.DayOfWeek;  
    var greetingMessage = greeting + " Today is: " + weekDay;  
}  
<p>Lời chào là: @greetingMessage</p>
```

Khởi mã	<pre>@{ int x = 123; string y = "because."; }</pre>
Biểu thức (đã mã hóa HTML)	<pre>@model.Message</pre>
Biểu thức (chưa mã hóa HTML)	<pre>@Html.Raw(model.Message)</pre>
Kết hợp text và HTML	<pre>@foreach (var item in items) { @item.Prop }</pre>
Trộn code và text	<pre>@if (foo) { <text>Plain Text</text> }</pre>
Trộn code và text	<pre>@if (foo) { @:Plain Text is @bar }</pre>

Khởi using	<code>@ using (Html.BeginForm()) { <input type="text" value="input here"/> }</code>
Địa chỉ email	Hi philha@example.com
Biểu thức (tường minh)	<code>ISBN@(isbnNumber)</code>
Mã hóa ký hiệu @	<code>In Razor, you use the @@foo to display the value of foo</code>
Chú thích phía server	<code>@* This is a server side multiline comment *@</code>
Trộn biểu thức và text	Hello @title. @name.

- ❑ Helper là các thành phần sinh giao diện web phù hợp buộc dữ liệu với model để duy trì thông tin trên các thành phần đó.
- ❑ Đơn giản việc viết mã sinh giao diện
- ❑ Helper được chia làm 1 số nhóm
 - ✍ Liên kết
 - ✍ Form
 - ✍ Sinh giao diện từ model
 - ✍ Kiểm lỗi

@Html.TextBox()
@Html.ActionLink()
@Html.Format()

- ❑ @Html.ActionLink() được sử dụng để **sinh liên kết**

```
@Html.ActionLink("Giới thiệu", "About" )  
<a href="/Home/About">Giới thiệu</a>
```

- ❑ @Html.ActionLink() nhận một số tham số:

- ~~✗~~ **linkText** – nhãn của liên kết
- ~~✗~~ **actionName** – tên action
- ~~✗~~ **routeValues** – tập các giá trị truyền đến action.
- ~~✗~~ **controllerName** – tên controller
- ~~✗~~ **htmlAttributes** – tập thuộc tính HTML của thẻ <a>

- ❑ Ví dụ:

```
@Html.ActionLink("Edit Record", "Edit", new {Id=3})  
<a href="/Store/Edit/3">Edit Record</a>
```

- ❑ Liên kết chứa ảnh

```
<a href="@Url.Action("Delete")">  
  </a>
```

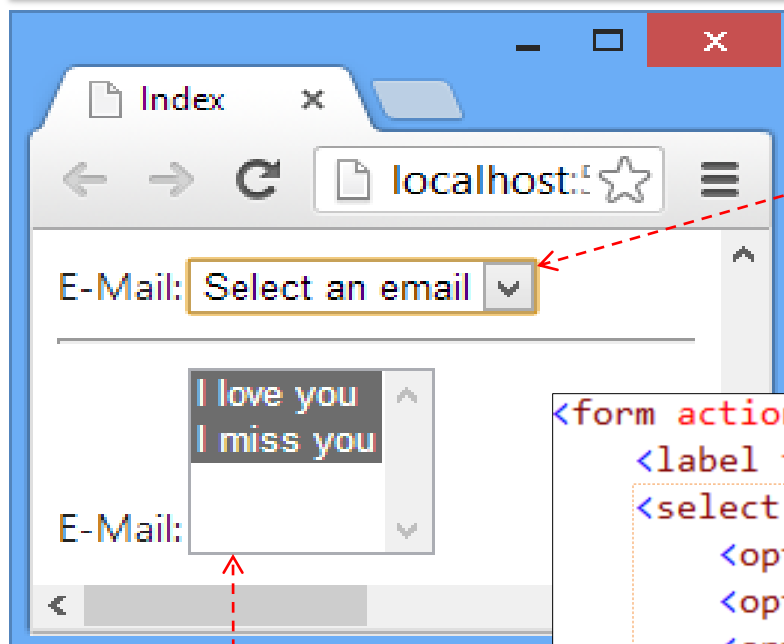
❑ Duy trì dữ liệu các trường form

Helper	HTML
@Html.BeginForm()	Sinh thẻ <form> bắt đầu
@Html.EndForm()	Sinh thẻ </form> kết thúc
@Html.CheckBox()	Sinh thẻ <input type="checkbox" >
@Html.Hidden()	Sinh thẻ <input type="hidden" >
@Html.Password()	Sinh thẻ <input type="password" >
@Html.RadioButton()	Sinh thẻ <input type="radio" >
@Html.TextArea()	Sinh thẻ <textarea></textarea>
@Html.TextBox()	Sinh thẻ <input type="text" >
@Html.DropDownList()	Sinh thẻ <select><option></select>
@Html.ListBox()	Sinh thẻ <select multiple><option></select>

Full Name	<code>@{Html.BeginForm("Action", "Controller");}</code>
<input type="text"/>	<code><div>Full Name</div></code> <code>@Html.TextBox("FullName")</code>
Password	<code><div>Password</div></code>
<input type="password"/>	<code>@Html.Password("Password")</code>
Photo	<code><div>Photo</div></code>
<input type="button" value="Chọn tệp"/> Không có tệp	<code><input name="Photo" type="file" /></code>
Married Status	<code><div>Married Status</div></code>
<input type="checkbox"/> Single	<code><label>@Html.CheckBox("Status") Single</label></code>
Gender	<code><div>Gender</div></code>
<input type="radio"/> Male <input type="radio"/> Female	<code><label>@Html.RadioButton("Gender", true) Male</label></code> <code><label>@Html.RadioButton("Gender", false) Female</label></code>
Description	<code><div>Description</div></code>
<input type="text"/>	<code>@Html.TextArea("Description")</code>
	<code>@Html.Hidden("Active")</code>
	<code><hr /></code>
<input type="button" value="Submit"/>	<code><input type="submit" value="Submit" /></code> <code>@{Html.EndForm();}</code>

```
List<Mail> Mails = new List<Mail>{  
    new Mail {  
        To = "sender1@gmail.com",  
        Subject = "I love you"  
    },  
    new Mail {  
        To = "sender2@gmail.com",  
        Subject = "I miss you"  
    }  
};  
ViewBag.Mails = new SelectList(Mails, "To", "Subject");
```

```
@using (Html.BeginForm()){  
    @Html.Label("Mails", "E-Mail:");  
    @Html.DropDownList("Mails", "Select an email")  
    <hr />  
    @Html.Label("Mails", "E-Mail:");  
    @Html.ListBox("Mails")  
}
```



Html.DropDownList

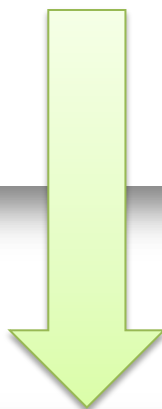
```
<form action="/" method="post">
  <label for="Mails">E-Mail:</label>
  <select id="Mails" name="Mails">
    <option value="">Select an email</option>
    <option value="sender1@gmail.com">I love you</option>
    <option value="sender2@gmail.com">I miss you</option>
  </select>

  <hr />

  <label for="Mails">E-Mail:</label>
  <select id="Mails" multiple="multiple" name="Mails">
    <option value="sender1@gmail.com">I love you</option>
    <option value="sender2@gmail.com">I miss you</option>
  </select>
</form>
```

Html.ListBox

```
@using (Html.BeginForm("Register", "Member")) {  
    ... nội dung form ...  
}
```



```
<form action="/Member/Register" method="post">  
    ... nội dung ...  
</form>
```

Helper	Mô tả
@Html.FormatValue (value, format)	Định dạng một giá trị số, chuỗi hoặc thời gian
@String.Format(format, value1, value2...)	Định dạng nhiều giá trị hỗn hợp
@Html.Raw (html)	Giải mã chuỗi đã mã hóa HTML

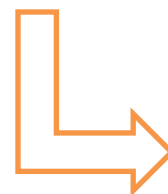
- Số bình thường: 12345.8765
- Phân nhóm: 12,345.877
- Tiền tệ: \$12,345.88
- Phần trăm: 72.00 %

- Ngày bình thường: 5/27/2014 9:26:09 PM
- Định dạng D: Tuesday, May 27, 2014
- Định dạng ISO: 2014-05-27
- Định dạng English: 05/27/2014
- Định dạng 24 giờ: 21:26:09
- Định dạng 12 giờ: 09:26:09 PM

- Có mã hóa HTML: Hello
- Không mã hóa HTML : **Hello**

Ký hiệu	Mô tả
{0:C}	Currency – tiền tệ theo ngôn ngữ
{0:P}	Percent – số phần trăm
{0:#,###.##0}	Number – số phân nhóm và 3 số lẻ

```
@{
    var number1 = 12345.8765;
    var number2 = 0.72;
}
<ul>
    <li>Số bình thường: @number1</li>
    <li>Phân nhóm: @Html.FormatValue(number1, "{0:#,###.##0}")</li>
    <li>Tiền tệ: @Html.FormatValue(number1, "{0:c}")</li>
    <li>Phần trăm: @Html.FormatValue(number2, "{0:p}")</li>
</ul>
```



- Số bình thường: 12345.8765
- Phân nhóm: 12,345.877
- Tiền tệ: \$12,345.88
- Phần trăm: 72.00 %

Ký hiệu	Mô tả
{0:D}	Date – theo ngôn ngữ được chọn
{0:MMMM-dd-yyyy hh:mm:ss tt}	<ul style="list-style-type: none"> ✓ M,MM,MMM,MMMM: tháng 1, 2 ký tự số, 3 ký tự viết tắt, tên tháng đầy đủ ✓ d,dd: ngày 1, 2 ký tự ✓ yy,yyyy: năm 2, 4 ký tự số ✓ H, HH, h, hh: 1,2 ký tự giờ 24 hoặc 12 giờ mỗi ngày ✓ m,mm: 1,2 ký tự số phút ✓ s,ss: 1,2 ký tự số giây ✓ tt: 2 ký tự sáng/chiều

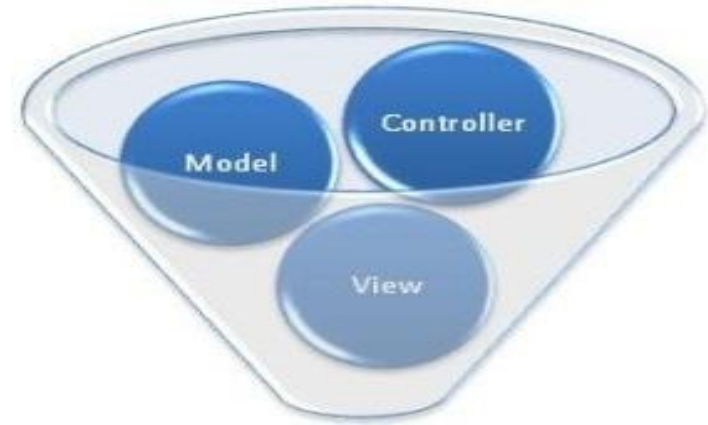
- Ngày bình thường: 5/27/2014 9:26:09 PM
- Định dạng D: Tuesday, May 27, 2014
- Định dạng ISO: 2014-05-27
- Định dạng English: 05/27/2014
- Định dạng 24 giờ: 21:26:09
- Định dạng 12 giờ: 09:26:09 PM

```
@{
    var now = DateTime.Now;
}
<ul>
    <li>Ngày bình thường: @now</li>
    <li>Định dạng D: @Html.FormatValue(now, "{0:D}")</li>
    <li>Định dạng ISO: @Html.FormatValue(now, "{0:yyyy-MM-dd}")</li>
    <li>Định dạng English: @Html.FormatValue(now, "{0:MM/dd/yyyy}")</li>
    <li>Định dạng 24 giờ: @Html.FormatValue(now, "{0:HH:mm:ss}")</li>
    <li>Định dạng 12 giờ: @Html.FormatValue(now, "{0:hh:mm:ss tt}")</li>
</ul>
```

- Có mã hóa HTML: `Hello`
- Không mã hóa HTML : **Hello**

```
@{  
    var chuoi = "<strong>Hello</strong>";  
}  
<ul>  
    <li>Có mã hóa HTML: @chuoi</li>  
    <li>Không mã hóa HTML : @Html.Raw(chuoi)</li>  
</ul>
```


Microsoft
ASP.net MVC 5



SINH GIAO DIỆN THEO MODEL

ThS. Nguyễn Nghiệm
0913.745.789
nghiemn@fpt.edu.vn
songlong2k@gmail.com

- ❑ Dựa vào các đặc điểm của thuộc tính trong lớp model để sinh ra giao diện người dùng.
 - ✍ Sinh các control tường minh
 - ✍ Sinh các control ngầm định

```
public class Student
{
    [DisplayName("Mã sinh viên")]
    public String Id { get; set; }
    [DisplayName("Mật khẩu")]
    public String Password { get; set; }
    [DisplayName("Họ và tên")]
    public String FullName { get; set; }
    [DisplayName("Giới tính")]
    public bool Gender { get; set; }
    [DisplayName("Ngày sinh")]
    public DateTime Birthday { get; set; }
    [DisplayName("Ghi chú")]
    public String Notes { get; set; }
}
```



Đăng ký thành viên

Mã sinh viên

Mật khẩu

Họ và tên

Giới tính ☒ Nam ☐ Nữ

Ngày sinh

Ghi chú

❑ Chỉ định loại control đối với các thuộc tính

Helper	Mô tả
@Html.TextBoxFor (m => m.Id)	<input type="text" name="Id" id="Id">
@Html.PasswordFor (m => m.Pwd)	<input type="password" name="Pwd" id="Pwd">
@Html.TextAreaFor (m => m.Notes)	<textarea name="Notes" id="Notes"></textarea>
@Html.CheckBoxFor (m => m.Status)	<input type="checkbox" name="Status" id="Status">
@Html.RadioButtonFor (m => m.Gender)	<input type="radio" name="Gender" id="Gender">
@Html.HiddenFor (m=> m.Name)	<input type="hidden" name="Name" id="Name">
@Html.DropDownListFor (m=> m.Blood)	<select id="Blood" name="Blood">...</select>
@Html.ListBoxFor (m=> m.Jobs)	<select id="Jobs" name="Jobs" multiple>...</select>
@Html.LabelFor (m=> m.Name)	<label for="Name"> Name </label>

@model Mvc5CodeDemo.Models.Student

<h2>Đăng ký thành viên</h2>

@using (Html.BeginForm())

{

<table><tr>

<td>@Html.LabelFor(m => m.Id)</td>

<td>@Html.TextBoxFor(m => m.Id)</td>

</tr><tr>

<td>@Html.LabelFor(m => m.Password)</td>

<td>@Html.PasswordFor(m => m.Password)</td>

</tr><tr>

<td>@Html.LabelFor(m => m.FullName)</td>

<td>@Html.TextBoxFor(m => m.FullName)</td>

</tr><tr>

<td>@Html.LabelFor(m => m.Gender)</td>

<td>

<label>@Html.RadioButtonFor(m => m.Gender, true) Nam</label>

<label>@Html.RadioButtonFor(m => m.Gender, false) Nữ</label>

</td>

</tr><tr>

<td>@Html.LabelFor(m => m.Birthday)</td>

<td>@Html.TextBoxFor(m => m.Birthday)</td>

</tr><tr>

<td>@Html.LabelFor(m => m.Notes)</td>

<td>@Html.TextAreaFor(m => m.Notes)</td>

</tr><tr>

<td> </td>

<td><input type="submit" value="Register" /></td>

</tr></table>

}

Sinh <label for="Id">Mã
sinh viên</label>

Kiểu của Model

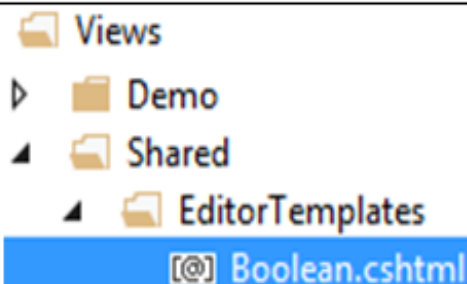
Sinh <input type="text" name="Id"
id="Id"> từ thuộc tính Id của Model

- ❑ Tự sinh loại control phù hợp với đặc điểm của thuộc tính của lớp model.

Helper	Mô tả
<code>Html.EditorFor(m=>m.Property)</code>	Sinh 1 control cho 1 thuộc tính.
<code>Html.EditorForModel()</code>	Sinh toàn form theo các thuộc tính của Model
<code>Html.Editor(object)</code>	Sinh toàn form theo các thuộc tính của Object đặt trong ViewBag

Bổ sung thêm template để hiển thị giới tính dạng RadioButtonList bằng cách thêm Boolean.cshtml vào thư mục Views/Shared/EditorTemplates

```
@model Boolean
<label>@Html.RadioButton("Gender", true, @Model == true)
Nam</label>
<label>@Html.RadioButton("Gender", false, @Model == false)
Nữ</label>
```




```
public class Student
{
    [DisplayName("Mã sinh viên")]
    public String Id { get; set; }
    [DisplayName("Mật khẩu"), DataType(DataType.Password)]
    public String Password { get; set; }
    [DisplayName("Họ và tên")]
    public String FullName { get; set; }
    [DisplayName("Giới tính")]
    public bool Gender { get; set; }
    [DisplayName("Ngày sinh")]
    public DateTime Birthday { get; set; }
    [DisplayName("Ghi chú"), DataType(DataType.MultilineText)]
    public String Notes { get; set; }
}
```

```
@model Mvc5CodeDemo.Models.Student
@using (Html.BeginForm())
{
    @Html.EditorForModel()
    <input type="submit" value="Register" />
}
```

Mã sinh viên

Mật khẩu

Họ và tên

Giới tính

☒ Nam ☐ Nữ

Ngày sinh

Ghi chú

Register

Data Type	Mô tả
DataType.CreditCard	Chỉ cho phép nhập số thẻ tín dụng
DataType.Currency	Hiển thị và tiếp nhận dạng tiền tệ theo địa phương được chọn
DataType.Date	Hiển thị và tiếp nhận dạng ngày theo địa phương được chọn
DataType.DateTime	Hiển thị và tiếp nhận dạng ngày và giờ theo địa phương được chọn
DataType.Duration	Sinh slider trên thiết bị di động
DataType.EmailAddress	Chỉ cho phép nhập email
DataType.Html	Cho phép nhập mã html
DataType.ImageUrl	Chỉ cho phép nhập địa chỉ ảnh
DataType.MultilineText	Sinh <textarea>
DataType.Password	Sinh <input type="password">
DataType.PhoneNumber	Sinh phần tử nhập số điện thoại trên thiết bị di động
DataType.PostalCode	Chỉ chấp nhận dạng postal code
DataType.Text	Nhập văn bản thông thường
DataType.Time	Ô nhập thời gian
DataType.Upload	Ô nhập upload file
DataType.Url	Chỉ chấp nhận địa chỉ tài nguyên

❑ **@Html.DisplayNameFor** (m=> m.**Property**)

✎ Hiển thị tên của thuộc tính Property

❑ **@Html.DisplayFor** (m=>m.**Property**)

✎ Hiển thị giá trị cho thuộc tính Property

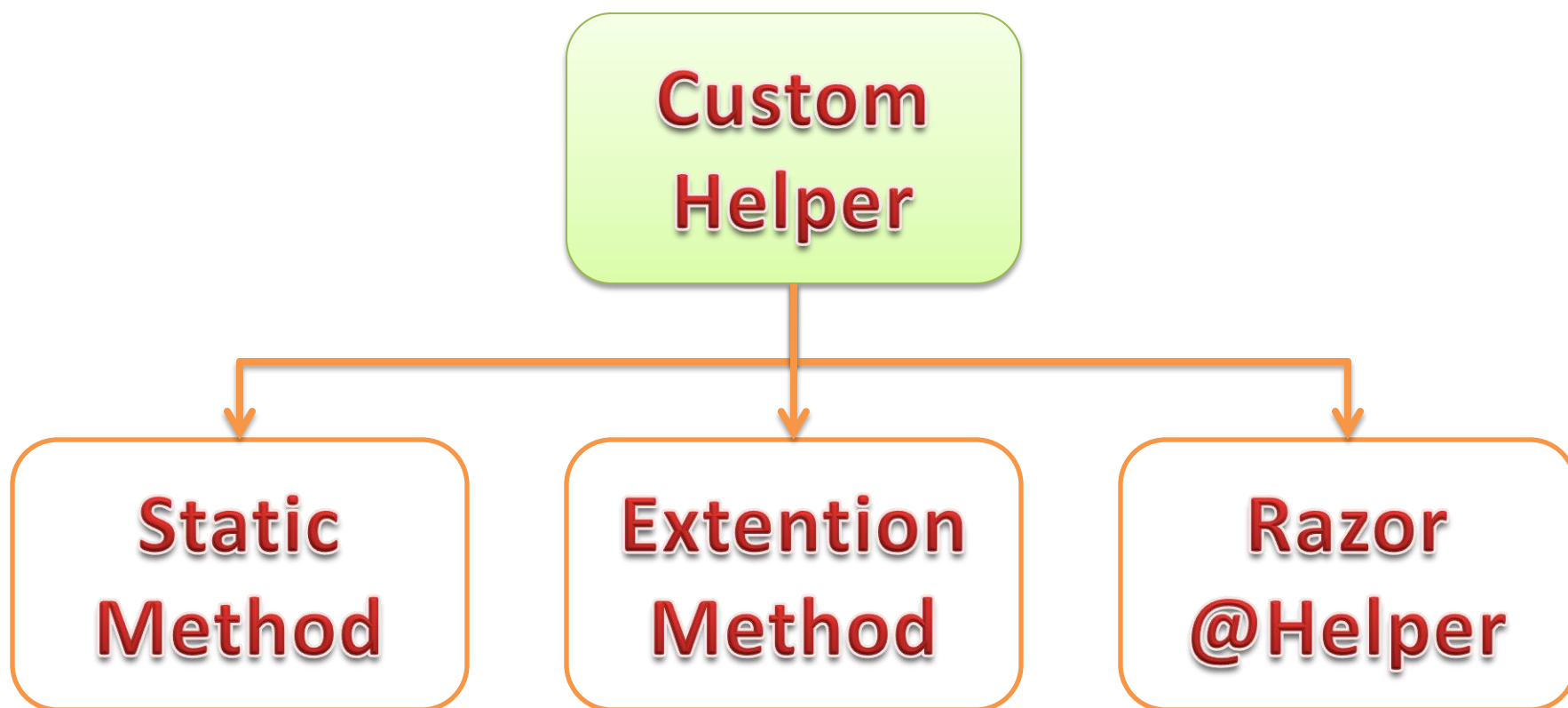
❑ **@Html.DisplayForModel** ()

✎ Hiển thị giá trị của tất cả các thuộc tính

❑ **@Html.Display** ("Mail")

✎ Hiển thị giá trị của tất cả các thuộc tính của đối tượng trong ViewData hay ViewBag

- ❑ Bên cạnh các helper dựng sẵn, bạn có thể tạo cho mình các helper có mục đích cho riêng mình.
- ❑ Trong MVC có 3 phương pháp tạo helper tùy biến



Static method

```
public class XString
{
    public static String Truncate(String input, int length)
    {
        if (input.Length <= length)
        {
            return input;
        }
        else
        {
            return input.Substring(0, length) + "...";
        }
    }
}
```

View

```
@{
    var input = @"Thật sự rất đơn giản với phương pháp này, bạn
                  chỉ cần định nghĩa phương thức tĩnh là có thể sử
                  dụng được như bạn từng thấy với String.Format.";
}

@XString.Truncate(input, 10);
```

Kết quả: Thật sự rất đơn giản với ...

❑ Phương thức tĩnh

✎ Sử dụng: `@String.Format("{0:F}", Model.Price)`

❑ Phương thức mở rộng của HtmlHelper

✎ Sử dụng: `@Html.Submit(String label)`

❑ Chỉ thị **@helper** để định nghĩa trực tiếp trên View

✎ Sử dụng: `@Truncate(title, 20)`

Qui ước

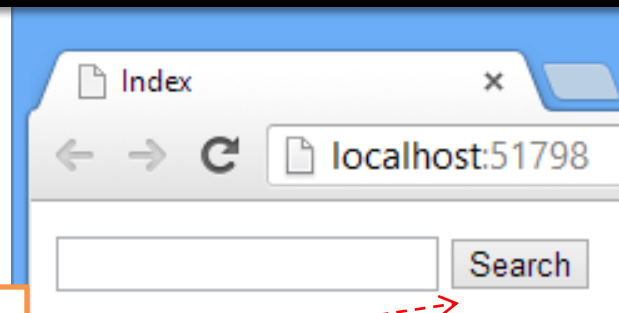
Lớp được mở rộng

```
public static class MyHelpers
{
    public static MvcHtmlString Submit(this HtmlHelper helper, string label)
    {
        TagBuilder tag = new TagBuilder("input");
        tag.MergeAttribute("type", "submit");
        tag.MergeAttribute("value", label);
        return MvcHtmlString.Create(tag.ToString(TagRenderMode.SelfClosing));
    }
}
```

```
@using (Html.BeginForm()){
    @Html.TextBox("txtSearch")
    @Html.Submit("Search")
}
```

Sử dụng helper
mới định nghĩa

Control sinh ra



```
public static class XHtmlHelper
{
    public static MvcHtmlString Submit(this HtmlHelper helper, String label,
                                       String name = null, object htmlAttributes = null)
    {
        var tag = new TagBuilder("input");
        tag.Attributes["type"] = "submit";
        tag.Attributes["value"] = label;
        if (name != null)
        {
            tag.Attributes["name"] = name;
        }
        if (htmlAttributes != null)
        {
            var attributes = htmlAttributes.GetType().GetProperties();
            foreach (var a in attributes)
            {
                tag.Attributes[a.Name] = a.GetValue(htmlAttributes).ToString();
            }
        }
        return MvcHtmlString.Create(tag.ToString(TagRenderMode.SelfClosing));
    }
}
```

Helper/HTML xuất ra

```
@Html.Submit("Save")
```

Sinh mã HTML: `<input type="submit" value="Save" />`

```
@Html.Submit("Save", "Command")
```

Sinh mã HTML: `<input name="Command" type="submit" value="Save" />`

```
@Html.Submit("Save", "Command", new { @class = "btn", id="save" })
```

Sinh mã HTML: `<input class="btn" id="save" name="Command" type="submit" value="Save" />`

```
@helper Truncate(String input, int length)
{
    if(input.Length < length)
    {
        @input
    }
    else
    {
        @input.Substring(0, length)<text>...</text>
    }
}
```

```
@foreach (var item in Model) {
    <tr>
        <td>
            @Truncate(item.Artist.Name, 25)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Genre.Name)
        </td>
        <td>
            @Truncate(item.Title, 25)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Price)
        </td>
    </tr>
}
```

- ☐ Base 64, MD5, Ascii
- ☐ XMail
- ☐ RadioButtonList, Image, File