

Keeping Your Razor Code Clean With HTML Helpers



Alex Wolf

www.alexwolfthoughts.com



Utilizing HTML Helpers

HTML Helper Examples

Helper	Benefit
<code>Html.ActionLink()</code>	Build anchor tags with correct URLs
<code>Html.TextBoxFor()</code>	Strong typing for Model Properties to simplify binding
<code>Html.EditorForModel()</code>	Provides templating for Models

Inline Razor Helpers

```
@helper DisplayList(List<string> data)
{
    <ol>
        @foreach (var item in data)
        {
            <li>@item</li>
        }
    </ol>
}
```

Pros

Easy to create, simple syntax
Reduce markup

Cons

Not reusable, tied to one view
Adding logic can be awkward

HTML Helpers Using Extension Methods

```
public static MvcHtmlString DisplayList(this HtmlHelper html,
List<string> data)
{
    var ulBuilder = new TagBuilder("ul");
    foreach(var item in data)
    {
        var liBuilder = new TagBuilder("li");
        liBuilder.SetInnerText(item);
        ulBuilder.InnerHtml += liBuilder.ToString();
    }
    return new MvcHtmlString(ulBuilder.ToString());
}
```

HTML Helper Examples

Partial View/Child Action	Helper Method
Section of more complex HTML	Single or closely related HTML elements (template)
Higher level, model related logic that may involve controller action method	Simpler, low level logic for displaying elements
Reduce Code Duplication	

Summary



- MVC provides multiple options for writing cleaner Razor code
- Inline Razor Helpers use friendly syntax and reduce code, but are not very reusable
- Custom HTML Helpers can include C# logic and be reused across multiple views
- HTML Helper code should be mostly limited to display logic