******** Custom Html Helpers Methods article ********

Hi,

This article is about custom html helpers methods in asp.net mvc.There are two kinds of custom html helpers methods in asp.net mvc those are

1. Inline Helper Method
2. External Helper Method

**Inline Helper:**

The Inline helper can be defined  and call it as shown below

```
<h2>Inline Helper</h2>

@*Inline Helper*@

@helper Print(string message)
{
    if (message.Length <= 5)
    {
        <h1 style="color: green;">Hello @message!! from inline helper.</h1>
    }
    else
    {
        <h1 style="color: red;">Hello @message!! from inline helper.</h1>
    }
}

@Print("Human")
@Print("Machine")
```

In the above image i defined an inline helper called print using @helper tag and i called it twice using by passing an arguments "Human" and "Machine".

**External Helper:**

The external helper is defined in the extension method. The extension method is used to extend the existing type with new methods, without altering the existing type. The first parameter of the extension method contains the this modifier with the extended type in our case HtmlHelper type. The return type of the external helper method should be MvcHtmlString.

```
using System.Web.Mvc;

namespace CustomHelperMethods.Helpers
{
    public static class Print
    {
        //Custom external helper
        public static MvcHtmlString PrintMessage(this HtmlHelper helper, string message)
        {
             TagBuilder tagBuilder = new TagBuilder("h1");
            if (message.Length <= 5)
            {
                tagBuilder.Attributes["style"] = "color:green";
                tagBuilder.SetInnerText(string.Format("Hello {0} !! from external helper.", message));
            }
            else
            {
                tagBuilder.Attributes["style"] = "color:red";
                tagBuilder.SetInnerText(string.Format("Hello {0} !! from external helper.", message));
            }

            return  new MvcHtmlString(tagBuilder.ToString());
        }
    }
}
```

In the above extension method i used tag builder to build the html string and sent it to the view for rendering.

**Call External Helper:**

```
@using CustomHelperMethods.Helpers

@Html.ActionLink("Inline Helper","InlineHelper") | @Html.ActionLink("External Helper","ExternalHelper")

<h2>External Helper</h2>

@Html.PrintMessage("Human")

@Html.PrintMessage("Machine")
```

In the above code to call the external helper method, I import the external helper namespace and call the external helper method with the proper arguments.

Inline Helper | External Helper

## Inline Helper

**Hello Human!! from inline helper.**

**Hello Machine!! from inline helper.**

******** END ********