In this video, we will discuss, **creating custom html helpers**. We will be using the example, that we worked with in [Part 47](http://csharp-video-tutorials.blogspot.com/2013/07/part-47-displaying-images-in-aspnet-mvc.html). Please watch [Part 47](http://csharp-video-tutorials.blogspot.com/2013/07/part-47-displaying-images-in-aspnet-mvc.html), before proceeding.

In [Part 47](http://csharp-video-tutorials.blogspot.com/2013/07/part-47-displaying-images-in-aspnet-mvc.html), to render the image, we used the following code. We are building the image tag, by passing values for **"src"** and **"alt"** attributes.   
<img src="@Url.Content(@Model.Photo)" alt="@Model.AlternateText" />

**Though the above code is not very complex**, it still makes sense to move this logic into it's own helper method. We don't want any complicated logic in our views. Views should be as simple as possible. Don't you think, it would be very nice, if we can render the image, using Image() html helper method as shown below. Unfortunately, MVC does not have built-in Image() html helper. So, let's build our own custom image html helper method.  
@Html.Image(Model.Photo, Model.AlternateText)  
  
**Let's take a step back and understand html heper methods**. The HTML helper method simply returns a string. To generate a textbox, we use the following code in our view.  
@Html.TextBox("TextBox Name")  
  
So, here **TextBox**() is an extension method defined in **HtmlHelper** class. In the above code, **Html** is the property of the View, which returns an instance of the **HtmlHelper**class.  
  
**Let's now add, Image() extension method, to HtmlHelper class.**  
**1.** Right click on **MVCDemo** project and add **"CustomHtmlHelpers"** folder.   
**2.** Right click on **"CustomHtmlHelpers"** folder and add **"CustomHtmlHelpers.cs"**class file.  
**3.** Copy and paste the following code. The code is commented and self-explanatory. **TagBuilder** class is in **System.Web.Mvc namespace**.  
namespace MVCDemo.CustomHtmlHelpers  
{  
public static class CustomHtmlHelpers  
{  
    public static IHtmlString Image(this HtmlHelper helper, string src, string alt)  
    {  
        // Build <img> tag  
        TagBuilder tb = new TagBuilder("img");  
        // Add "src" attribute  
        tb.Attributes.Add("src", VirtualPathUtility.ToAbsolute(src));  
        // Add "alt" attribute  
        tb.Attributes.Add("alt", alt);  
        // return MvcHtmlString. This class implements IHtmlString  
        // interface. IHtmlStrings will not be html encoded.

   return new MvcHtmlString(tb.ToString(TagRenderMode.SelfClosing));  
    }  
}  
}  
  
To use the custom Image() html helper in **Details.cshtml** view, please include the folowing using statement in Details.cshtml  
@using MVCDemo.CustomHtmlHelpers;  
  
As we intend to use this **Image**() html helper, in all our views, let's include **"MVCDemo.CustomHtmlHelpers"** namespace in **web.config**. This eliminates the need to include the namespace, in every view.  
<system.web.webPages.razor>  
  <pages pageBaseType="System.Web.Mvc.WebViewPage">  
    <namespaces>  
      <add namespace="System.Web.Mvc" />  
      <add namespace="System.Web.Mvc.Ajax" />  
      <add namespace="System.Web.Mvc.Html" />  
      <add namespace="System.Web.Routing" />  
      <add namespace="MVCDemo.CustomHtmlHelpers" />  
    </namespaces>  
  </pages>  
  <host ....../>  
</system.web.webPages.razor>  
  
If you intend to use the Image() custom html helper, only with a set of views, then, include a web.config file in the specific views folder, and then specify the namespace in it.