**What are Tag Helpers**  
  
Tag Helpers are server side components. They are processed on the server to create and render HTML elements in Razor files. Tag Helpers are similar to HTML helpers. There are many built-in Tag Helpers for common tasks such as generating links, creating forms, loading assets etc.  
  
  
**Importing built-in Tag Helpers**  
  
To make the built-in tag helpers available for all the views in our entire application, import the tag helpers using \_ViewImports.cshtml file. To import tag helpers we use @addTagHelper directive.

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

The wildcard **\*** indicates that we want to import all the tag helpers  
Microsoft.AspNetCore.Mvc.TagHelpers is the assemly that contains the built-in tag helpers  
  
**Generating Links using Tag Helpers**  
  
Let's say we want to view a specific employee details. So we want to generate the following hyperlink. The number 5 is the ID of the employee whose details we want to view.  
  
/home/details/5  
  
There are several ways we could do this in a razor view  
  
**Option 1 :** Manually generating the links

@foreach (var employee in Model)

{

    <a href="/home/details/@employee.Id">View</a>

}

**Option 2 :** Using HTML helpers

@Html.ActionLink("View", "details", new { id = employee.Id })

generates an anchor element

<a href="/hom/details/5">View</a>

@Url.Action("details", "home", new { id = employee.Id })

generates a string  
  
/hom/details/5

**Option 3 :**Using Tag Helpers

<**a** **asp-controller**="home" **asp-action**="details"

**asp-route-id**="@employee.Id">View</**a**>

generates

<a href="/Home/details/5">View</a>

**Anchor Tag Helper**  
  
The Anchor Tag Helper enhances the standard HTML anchor (<a ... ></a>) tag by adding new attributes such as   
**asp-controller**  
**asp-action**  
**asp-route-{value}**  
  
The rendered anchor element's href attribute value is determined by the values of these **asp-** attributes.   
  
As the names imply **asp-controller** specifies the controller name and **asp-action** specifies the action name to include in the generated href attribute value. **asp-route-{value}** attribute is used to include route data in the generated href attribute value. {value} can be replaced with the route parameters such as id for example.

<**a** **asp-controller**="home" **asp-action**="details"

**asp-route-id**="@employee.Id">View</**a**>

generates

<a href="/Home/details/5">View</a>

As you can see from the code below, manually generating links is much easier than using HTML Helpers or Tag Helpers.

<a href="/home/details/@employee.Id">View</a>

The obvious question that comes to our mind is, why should we use **HTML helpers** or **Tag Helpers**over manually generating these links.

Why Tag Helpers

we will discuss **why should we use tag helpers** instead of writing the same HTML by hand. Let's understand the advantage of using tag helpers with an example.  
  
  
Let's say we want to view a specific employee details. So we want to generate the following hyperlink. The number 5 is the ID of the employee whose details we want to view.  
  
/home/details/5  
  
  
We could manually generate this as shown below

<a href="/home/details/@employee.Id">View</a>

**OR**  
  
Use the anchor <a> tag helper as shown below

<**a** **asp-controller**="home" **asp-action**="details" **asp-route-id**="@employee.Id">View</**a**>

**Advantage of using Tag helpers**  
  
Tag helpers generate links based on the application routing templates. This means if we later change routing templates, the links generated by tag helpers will automatically reflect those changes made to the routing templates. The generated links just work.   
  
Where as if we have hard-coded the URL paths manually, we would have to change the code in lot of places when the application routing templates change.  
  
Let's understand this with an example  
  
The following is the route template our application is using

app.UseMvc(routes =>  
{  
    routes.MapRoute("default", "{controller=Home}/{action=Index}/{id?}");  
});

The following code is **not using tag helpers**. We are manually generating the link by hard-coding the URL paths.

<a href="/home/details/@employee.Id">View</a>

The following code is using **<a>** anchor tag helper. 

<**a** **asp-controller**="home" **asp-action**="details"

**asp-route-id**="@employee.Id">View</**a**>

Notice here, we are not hard coding the URL paths. We are only specifying the **controller and action names**and the route parameters along with their values. When tag helpers are executed on the server they look at the route templates and generate the correct URLs automatically.  
  
Both the above techniques generate the correct URL path (/home/details/5) and it works with the current route template ({controller=Home}/{action=Index}/{id?})  
  
Now let's say we have changed the routing template to the following. Notice in the URL we have the string literal "pragim"

app.UseMvc(routes =>

{

    routes.MapRoute("default", "pragim/{controller=Home}/{action=Index}/{id?}");

});

The code that's using the anchor tag helper generates the following correct link

<a href="/pragim/home/details/1">View</a>

Where as the code that's not using the **anchor tag helper**continues to generate the following link. Notice the URL path "/pragim" is missing.

<a href="/home/details/1">View</a>

We also have tag helpers that generate forms. When this form is posted back to the server, the posted values are automatically handled and the associated validation messages are displayed. Without these tag helpers we would have to write a lot of custom code to achieve the same.