

What is a Standard HTML helper in Asp.NET MVC5?

- Standard HTML helpers are used to render the most common type of HTML controls like Label, TextBox, Password, TextArea, CheckBox, RadioButton, DropDownList, ListBox, Display, Editor and ActionLink etc.
- HTML helpers always start with @HTML. They are an object of Html helper class. @ symbol used to access the server-side code. The extension method of the HTML helper class has several overloaded methods. We can use it as per our requirement.
- HTML is a property of type HtmlHelper included in the base class of razor view WebViewPage. TextBox() or TextBoxFor()... are extension methods included in HtmlHelper class.
- The HtmlHelper class generates HTML elements. For example, @Html.ActionLink("Add Employee", "Create", "Employee") would generate the anchor tag Add Employee.

List of Standard HTML Helpers In ASP.NET MVC 5

- @Html.TextBox
- @Html.Password
- @Html.TextArea
- @Html.CheckBox
- @Html.RadioButton
- @Html.DropDownList
- @Html.ListBox
- @Html.Hidden
- @Html.Display
- @Html.Editor
- @Html.ActionLink
- @Html.BeginForm
- @Html.Label

@Html.TextBox()

TextBox() helper method is a loosely typed extension method that is used to create textbox(<input type="text">) element in razor view.

@Html.TextBox() method inside the "System.Web.Mvc.Html" namespace. The return type is "MvcHtmlString".

TextBox() Method Signature

```
MvcHtmlString Html.TextBox(string name, string value, object htmlAttribute)
```

The TextBox() method is a loosely typed method because it has a name parameter that is a string. It automatically displays the value of the model property in a textbox and vice-versa. The name parameter can be a model class property. The value should be an object value of the property. The third parameter is that the HTML attribute can provide CSS or Id to the textbox. It has several overload methods we can use as per our requirement.

How to use TextBox() in Razor view

```
@Html.TextBox("txtUserName", "", new { @class = "form-control" })
```

Html Result

```
<input class="form-control" id="txtUserName" name="txtUserName" type="text">
```

Output

User Name	<input type="text"/>
-----------	----------------------

How to set default value to TextBox() in Razor view

```
@Html.TextBox("txtUserName", "Test Value", new { @class = "form-control" })
```

@Html.Password()

The HTML helper class is used to generate HTML elements. The Password() helper method is a loosely typed extension method that is used to create textbox(<input type="password">) element in razor view.

@Html.Password() method inside the "System.Web.Mvc.Html" namespace. The return type is "MvcHtmlString".

Password() Method Signature

```
MvcHtmlString Html.Password(string name, string value, object htmlAttributes)
```

The Password() method is a loosely typed method because its name parameter is a string. The name parameter can be a model class property. The value should be an object value of the property. So it automatically displays a value of the model property in a password and visa-versa. The third parameter is that the HTML attribute can provide CSS or Id to the Password. It has several overload methods we can use as per our requirement.

How to use Password() Helper in Razor view

```
@Html.Password("Password", "", new { @class = "form-control" })
```

Html Result

```
<input class="form-control" id="Password" name="Password" type="password">
```

Output

Password	<input type="password"/>
----------	--------------------------

@Html.TextArea()

TextArea() helper method is loosely typed extension method used to create a TextArea(<textarea cols="20" id="Address" name="Address" rows="2"> </textarea>) element in razor view.

@Html.TextArea() method inside the "System.Web.Mvc.Html" namespace. The return type is "MvcHtmlString".

TextArea() Method Signature

```
MvcHtmlString Html.TextArea(string name, string value, object htmlAttrib
```

The TextArea() method is a loosely typed method because its name parameter is a string. The name parameter can be model class property. The value should be an object value of the property. It automatically displays a value of the model property in a textarea and visa-versa. The third parameter is that an HTML attribute can provide CSS, Id to the TextArea. It has several overload methods we can use as per our requirement.

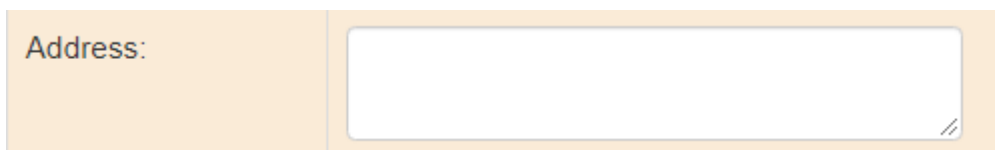
How to use TextArea() Helper in Razor view

```
@Html.TextArea("Address", " ", new { @class = "form-control", id="IdAddress"
```

Html Result

```
<textarea class="form-control"
cols="20"
id="IdAddress"
name="Address"
rows="2">
</textarea>
```

Output

The screenshot shows a rendered HTML element. On the left, the text "Address:" is displayed in a dark grey font. To its right is a white rectangular text input field with a light grey border and a small cursor icon at the bottom right corner. The entire element is set against a light orange background.

@Html.CheckBox():

This HTML helper class is used to generate HTML elements. The CheckBox() helper method is a loosely typed extension method which is used to create CheckBox(<input type="checkbox" >) element in razor view.

@Html.CheckBox() method inside the "System.Web.Mvc.Html" namespace. The return type is "MvcHtmlString".

CheckBox() Method Signature

```
MvcHtmlString CheckBox(string name, bool isChecked, object htmlAttribute
```

The CheckBox() method is a loosely typed method because its name parameter is a string. The name parameter can be model class property. The value should be an object value of the property, so it automatically displays a value of the model property in a CheckBox and visa-versa. The third parameter is an HTML attribute that can add CSS and the Id to the CheckBox. It has several overload methods we can use as per our requirement.

How to use CheckBox() Helper in Razor view

```
@Html.CheckBox("Dancing")
```

How to set CheckBox() default as checked

```
@Html.CheckBox("Cricket", true)
```

Html Result

```
<input checked="checked" id="Cricket" name="Cricket" type="checkbox" value="Cricket" />  
<input id="Dancing" name="Dancing" type="checkbox" value="Dancing" />
```

Output

Hobbies : ☒ Cricket ☐ Dancing ☐ Drawing

@Html.RadioButton()

This HTML helper class is used to generate HTML elements. The RadioButton() helper method is a loosely typed extension method which is used to create RadioButton(<input type="radio" >) element in razor view.

@Html.RadioButton() method inside the "System.Web.Mvc.Html" namespace. And return type is "MvcHtmlString".

RadioButton() Method Signature

```
MvcHtmlString RadioButton(string name, object value, bool isChecked, obj
```

The RadioButton() method is a loosely typed method because its name parameter is a string. The name parameter can be model class property. The value should be an object value of the property so that it automatically displays a value of the model property in a RadioButton and visa-versa. The third parameter is an HTML attribute that can add CSS and Id to the RadioButton. It has several overload methods we can use as per our requirement.

How to use RadioButton() Helper in Razor view

Example: Male and Female Gender

```
@Html.RadioButton("Gender", "Male", true, new { id = "male" }) Male
@Html.RadioButton("Gender", "Female", false, new { id = "female" }) Female
```

How to set RadioButton() default as checked

```
@Html.RadioButton("Gender", "Male", true, new { id = "male" }) Male
```

Html Result

```
<input checked="checked" id="male" name="Gender" type="radio" value="Male" />
<input id="female" name="Gender" type="radio" value="Female" />
```

Output

Gender: ☒ Male ☐ Female

@Html.DropDownList()

The DropDownList() helper method is a loosely typed extension method used to create a DropDownList(<select id="" name="">) element in razor view with a specified name, list items and HTML attributes.

@Html.DropDownList() method inside the "System.Web.Mvc.Html" namespace. The return type is "MvcHtmlString".

DropDownList() Method Signature

```
MvcHtmlString Html.DropDownList(string name, IEnumerable<SelectListItem> items)
```

The DropDownList() method is a loosely typed method because it has a name parameter that is a string. The name parameter can be a model class property. The second parameter included is the list. It automatically displays the value of the model property in a DropDownList and visa-Versa. The third parameter is an HTML attribute that can add CSS, Id to the DropDownList. It has several overload methods we can use as per our requirement.

How to use DropDownList() Helper in Razor view

```
@{
    IEnumerable<string> strList = new List<string> { "BCA", "BCS", "MCA", "M
```

```
}  
@Html.DropDownList("ddlCourse", new SelectList(strList, strList.FirstOrD
```

Html Result

```
<select id="ddlCourse" name="ddlCourse">  
<option value="">--Select Course----</option>  
<option selected="selected">BCA</option>  
<option>BCS</option>  
<option>MCA</option>  
<option>MCS</option>  
</select>
```

Output

Courses:	<input type="text" value="BCA"/>
----------	----------------------------------

@Html.ListBox()

This HTML helper class is used to generate HTML elements. ListBox() helper method is a loosely typed extension method which is used to create ListBox(<select id="" multiple="multiple" name="">) element in razor view with specified name, list items and HTML attributes.

@Html.ListBox() method inside the "System.Web.Mvc.Html" namespace. The return type is "MvcHtmlString".

ListBox() Method Signature

```
MvcHtmlString Html.ListBox(string name, IEnumerable<SelectListItem> sele
```

The ListBox() method is a loosely typed method because it has a string name parameter. The name parameter can be a model class property. The value should be an object value of the property. Therefore, it automatically displays a value of the model property in a ListBox and visa-versa. The third parameter is an HTML attribute that can add CSS, Id to the ListBox. It has several overload methods we can use as per our requirement.

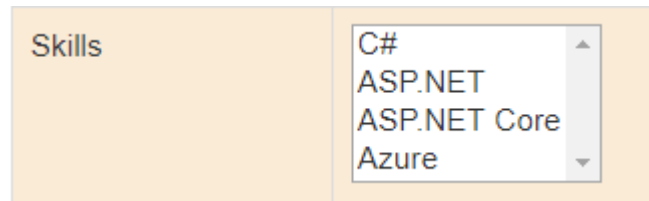
How to use ListBox() Helper in Razor view

```
@Html.ListBox("Select Skills",new List<SelectListItem> {  
    new SelectListItem{Text= "C#",Value="1"},  
    new SelectListItem{ Text="ASP.NET",Value="2" },  
    new SelectListItem{ Text="ASP.NET Core",Value="3" },  
    new SelectListItem{ Text="Azure",Value="4" }  
})
```

Html Result

```
<select id="Select_Skills" multiple="multiple" name="Select Skills">
<option value="1">C#</option>
<option value="2">ASP.NET</option>
<option value="3">ASP.NET Core</option>
<option value="4">Azure</option>
</select>
```

Output



@Html.Label()

This HTML helper class is used to generate HTML elements. The Label() helper method is a loosely typed extension method used to create Label(<label>) element in razor view.

@Html.Label() method inside the "System.Web.Mvc.Html" namespace. The return type is "MvcHtmlString".

Label() Method Signature

```
MvcHtmlString Html.Label(string name, string value, object htmlAttribute
```

The Label() method is a loosely typed method because its name parameter is a string. The name parameter can be a model class property. The value should be an object value of the property. Therefore, it automatically displays the value of the model property in a Label and visa-versa. The third parameter is an HTML attribute which can provide CSS and the Id to the Password. It has several overload methods that we can use as per our requirement.

How to use Label() Helper in Razor view

```
@Html.Label("User Name ")
```

Html Result

```
<label for="User_Name_">User Name </label>
```

@Html.ActionLink()

Html helper class is used to generate HTML elements. The ActionLink() helper method is loosely typed extension method which is used to create a Label(element in razor view.

@Html.ActionLink() method inside the "System.Web.Mvc.Html" namespace. The return type is "MvcHtmlString".

ActionLink() Method Signature

```
MvcHtmlString ActionLink(string linkText, string actionName, object route)
```

How to use ActionLink() Helper in Razor view

```
@Html.Label("User Name ")
```

Html Result

```
<a href="/Home/About">Go to About</a>
```

Output

Student Registration Form:

[Go to About](#)

@Html.BeginForm():

This HTML helper class is used to generate HTML elements. The BeginForm() helper method is loosely typed extension method which is used to create a Form (<form action="/" method="post">) element in razor view.

@Html.BeginForm() method is inside the "System.Web.Mvc.Html" namespace. The return type is "MvcHtmlString".

BeginForm() Method Signature

```
MvcHtmlString ActionLink(string linkText, string actionName, object route)
```

How to use BeginForm() Helper in Razor view

```
@using (Html.BeginForm("Index", "Home", FormMethod.Post))  
  
{  
    // Code here..  
}
```

Html Result

```
<form action="/" method="post">  
</form>
```


How to use Standard HTML helpers on View?

We will create a student registration form using Standard Html Helper in MVC 5.

Add the following code in index.cshtml page

```
<div class="container">
    <h3>Student Registration Form: </h3>
    @Html.ActionLink("Go to About", "About", new { })
    @Html.Hidden("StudentId", "100")
    @using (Html.BeginForm("Index", "Home", FormMethod.Post))
    {
        <div>
            <table border="1" class="table table-bordered" style="background-color: #f2f2f2">
                <tbody>
                    <tr>
                        <td>
                            @Html.Label("User Name ")
                        </td>
                        <td>
                            @Html.TextBox("txtUserName", "", new { @class = "form-control" })
                        </td>
                    </tr>
                    <tr>
                        <td>
                            @Html.Label("Password", "Password")
                        </td>
                        <td>
                            @Html.Password("Password", "", new { @class = "form-control" })
                        </td>
                    </tr>
                    <tr>
                        <td>
                            Gender:
                        </td>
                        <td>
                            @Html.RadioButton("Gender", "Male", true, new { @class = "form-control" })
                            @Html.RadioButton("Gender", "Female", false, new { @class = "form-control" })
                        </td>
                    </tr>
                    <tr>
                        <td>
                            Address:
                        </td>
                        <td>
                            @Html.TextArea("Address", "", new { @class = "form-control" })
                        </td>
                    </tr>
                </tbody>
            </table>
        </div>
    }
</div>
```

```

        <td>
            @Html.TextArea("Address", " ", new { @class
        </td>
    </tr>
    <tr>
        <td>
            Hobbies :
        </td>
        <td>
            @Html.CheckBox("Cricket", true) Cricket
            @Html.CheckBox("Dancing") Dancing
            @Html.CheckBox("Drawing") Drawing

        </td>
    </tr>

    <tr>
        <td>
            Courses:
            @{
                IEnumerable<string> strList = new List<S
            }
        </td>
        <td>
            @Html.DropDownList("ddlCourse", new SelectLi
        </td>
    </tr>
    <tr>
        <td>Skills</td>
        <td>
            @Html.ListBox("Select Skills",new List<Sele
            new SelectListItem{Text= "C#",Value="1"},
            new SelectListItem{ Text="ASP.NET",Value="2"
            new SelectListItem{ Text="ASP.NET Core",Value
            new SelectListItem{ Text="Azure",Value="4" }
            })
        </td>

    </tr>
    <tr>
        <td>

        </td>
    </tr>

```

```

        <td>
            <input type="submit" class="btn-primary" />
        </td>
    </tr>
</tbody>
</table>
</div>
}
</div>

```

Output

Student Registration Form:

[Go to About](#)

User Name	<input type="text" value="shri"/>
Password	<input type="password" value="...."/>
Gender:	<input checked="" type="radio"/> Male <input type="radio"/> Female
Address:	<input type="text" value="Pune"/>
Hobbies :	<input checked="" type="checkbox"/> Cricket <input checked="" type="checkbox"/> Dancing <input type="checkbox"/> Drawing
Cources:	<input type="text" value="MCA"/>
Skills	<div> C# ASP.NET ASP.NET Core Azure </div>
	<input type="submit" value="Submit"/>