

QUEZON CITY UNIVERSITY
COLLEGE OF COMPUTER STUDIES



WEEK 6

DEVELOPING VIEWS

– HTML Helpers

IPT102 – INTEGRATIVE PROGRAMMING AND TECHNOLOGIES 2

1ST SEMESTER

LEARNING OUTCOMES:

- To understand standard HTML Helpers in ASP .NET
- To identify different HTML Helpers Controls
- To learn the advantages of using Standard HTML helpers in ASP.NET

MVC

Using HTML Helpers and Tag Helpers

Standard HTML Helpers are used to render the most common type of HTML controls like Label, Textbox, Password, Text Area, Checkbox, RadioButton, Dropdown List, Listbox, Display Editor and ActionLink etc.

Html helpers are always starting with `@HTML` and It is an object of Html helper class. `@` symbol is used to access the server-side code.

Html is a property of type `HtmlHelper` included in base class of razor view `WebViewPage` (`System.Web.Mvc`) in controller.

Using HTML Helpers and Tag Helpers

An HTML Helper in ASP.NET MVC is an extension method of the HTML Helper class which is used to generate HTML content in a view. For example, if you want to generate a textbox with `id="firstname"` and `name="firstname"` then you can type all the required HTML in a view as shown below:

```
<input type="text" name="firtsname" id="firstname" />
```

But in ASP.NET MVC, you can use the following “TextBox” HTML helper method in a view to generating a text box.

```
@Html.TextBox("firstname");
```

A Web Page

← → ↻ 🔍 http://

Name Label Textbox

Phone

Email

Password Password textbox

Gender ☒ Male ☐ Female Radio Button

Country DropDown

Terms ☐ Checkbox Checkbox

Input button

The diagram shows a web browser window titled 'A Web Page'. The address bar contains 'http://'. The form contains the following elements:

- Name:** A text input field labeled 'Textbox' in red. A red label 'Label' is also shown with an arrow pointing to the 'Name' text.
- Phone:** A text input field.
- Email:** A text input field.
- Password:** A password input field (masked with asterisks) labeled 'Password textbox' in red.
- Gender:** Two radio buttons, 'Male' (selected) and 'Female', labeled 'Radio Button' in red.
- Country:** A dropdown menu showing 'India', labeled 'DropDown' in red.
- Terms:** A checkbox labeled 'Checkbox' in red.
- Submit:** A button labeled 'Input button' in red.

The following table lists the `HtmlHelper` methods and HTML control each method renders.

Extension Method	Strongly Typed Method	Html Control
<code>Html.ActionLink()</code>	NA	<code><a></code>
<code>Html.TextBox()</code>	<code>Html.TextBoxFor()</code>	<code><input type="textbox"></code>
<code>Html.TextArea()</code>	<code>Html.TextAreaFor()</code>	<code><input type="textarea"></code>
<code>Html.CheckBox()</code>	<code>Html.CheckBoxFor()</code>	<code><input type="checkbox"></code>
<code>Html.RadioButton()</code>	<code>Html.RadioButtonFor()</code>	<code><input type="radio"></code>
<code>Html.DropDownList()</code>	<code>Html.DropDownListFor()</code>	<code><select></code> <code><option></code> <code></select></code>
<code>Html.ListBox()</code>	<code>Html.ListBoxFor()</code>	multi-select list box: <code><select></code>
<code>Html.Hidden()</code>	<code>Html.HiddenFor()</code>	<code><input type="hidden"></code>
<code>Html.Password()</code>	<code>Html.PasswordFor()</code>	<code><input type="password"></code>
<code>Html.Display()</code>	<code>Html.DisplayFor()</code>	HTML text: <code>""</code>
<code>Html.Label()</code>	<code>Html.LabelFor()</code>	<code><label></code>
<code>Html.Editor()</code>	<code>Html.EditorFor()</code>	Generates Html controls based on data type of specified model property e.g. textbox for string property, numeric field for int,

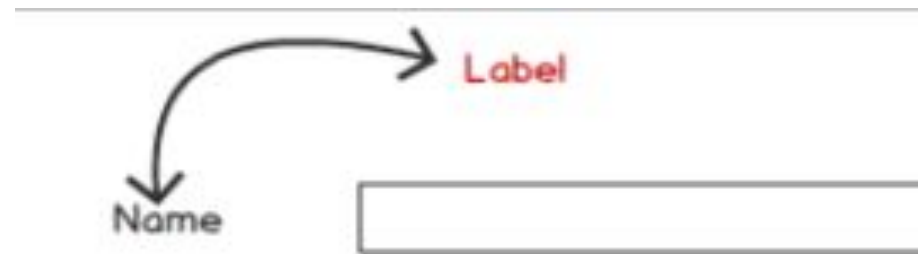
Create a Label in ASP.NET MVC

Html.Label Helper class renders an HTML label control that displays read-only text. It is a simple type method that only renders label control and it may be or not bounded by any model properties.

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

Example:

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



Create a Textbox in ASP.NET MVC

Html.TextBox is extension method of HtmlHelper class that draws HTML TextBox and accept input from user. Html.TextBox is loosely typed method that renders plain textbox input and doesn't get bounded with any model properties.

To create a TextBox using HTML Helper Method in the ASP.NET MVC application, we need to use the TextBox Helper method. The TextBox() HTML Helper method is a loosely typed method whereas the TextBoxFor() HTML Helper method is a strongly typed method.

Example:

```
@Html.TextBox()
```

```
@Html.TextBox("txtUsername")
```

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

Output:

The image shows a user interface element consisting of a light orange rectangular container. Inside this container, on the left, is the text 'User Name' in a dark grey font. To the right of the text is a white rectangular text input field with a thin grey border.

Create a Password Helper in ASP.NET MVC

Html.Password Helper class generates password input fields in a web form that is used to ask and validate user passwords. The characters entered in the password field are masked with a special character * (asterisk) or whatever you set on it.

Example:

```
@Html.Password() = @Html.Password("Password")
```

The Password() method is a loosely typed method because It has a name parameter is a string. The name parameter can be the model class property. The value should be binding the property. So it automatically displays a value of the model property in a password and visa-versa

Output:



Password
----------	------

How to use RadioButton() Helper in ASP.NET MVC

Radio Button provides a list of options and lets you choose only one option in a panel. Radio Button is used widely in a form and carry true for selected items and false for non-selected items

RadioButton() Method Signature:

```
MvcHtmlString RadioButton(string name, object value, bool isChecked, object htmlAttributes)
```

Example: Male and Female

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

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

The `RadioButton()` method is a loosely typed method because it has a name parameter that is a string. The name parameter can be the model class property. The value should be an object value of the property.

Output:

A screenshot of a web form element. It consists of a light orange rectangular box. On the left side of the box, the text "Gender:" is displayed in a dark grey font. To the right of this text, there are two radio button options. The first option is "Male", which has a radio button with a dark grey center, indicating it is selected. The second option is "Female", which has a radio button with a light grey center, indicating it is not selected.

How to use TextArea() Helper in ASP.NET MVC

TextArea is a multiline text input control that can accept a large number of string lines. Usually, the size of TextArea is divided into rows and columns and you can set it according to your needs.

TextArea() Method Signature:

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

Example:

```
@Html.TextArea("Address")
```

The `TextArea()` method is a loosely typed method because It has a 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 textarea and visa-Versa.

Output:

The image shows a web form element. On the left, there is a label 'Address:' in a light gray font. To the right of the label is a large, empty text area input field with a light gray border and a small cursor icon at the bottom right corner. The entire form is set against a light orange background.

How to use CheckBox() Helper in ASP.NET MVC

A checkbox is a small box that lets you choose multiple options. It only carries true and false values for the checked and unchecked items. In the ASP.NET MVC Helper class, you have given `Html.CheckBox()` and `Html.CheckBoxFor()` Extension method to work with.

CheckBox() Method Signature::

```
MvcHtmlString CheckBox(string name, bool isChecked, object htmlAttributes)
```

Example:

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

The `CheckBox()` method is a loosely typed method because It has a 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.

Output:

Hobbies :	<input checked="" type="checkbox"/> Cricket <input type="checkbox"/> Dancing <input type="checkbox"/> Drawing
-----------	---

How to use DropDownList() Helper in ASP.NET MVC

DropDownList is a visual control with a down arrow that displays a list of items and forces you to choose only one item at a time. Dropdown List is not editable, it means the user cannot enter own value and is limited to choosing an item from the selection.

DropDownList() Method Signature:


```
MvcHtmlString Html.DropDownList(string name, IEnumerable<SelectListItem> selectList,  
string optionLabel, object htmlAttributes)
```

Example:

```
@Html.Label("Courses")
@{IEnumerable<String> strCourses= new List<String>
{"BSCS","BSIT","BSBA","BSIE","BSECE"}}
@Html.DropDownList("ddlCourses", new
SelectList(strCourses),"Select Course")
```

The DropDownList() method is a loosely typed method because It has a name parameter is a string. The name parameter can be model class property. The second parameter to include the list. So it automatically displays a value of the model property in a DropDownList and visa-Versa.

Output:

A screenshot of a web form. On the left, there is a text label "Courses:" followed by a white rectangular input field. To the right of this field is a dropdown menu. The dropdown menu has a white background with the text "BCA" in a dark font. A small downward-pointing arrow is visible on the right side of the dropdown menu. The entire form is enclosed in a light orange border.

A decorative background on the right side of the slide, consisting of a large teal triangle pointing downwards and a dark grey triangle pointing upwards, meeting at a diagonal line.

END OF PRESENTATION.
THANK YOU!