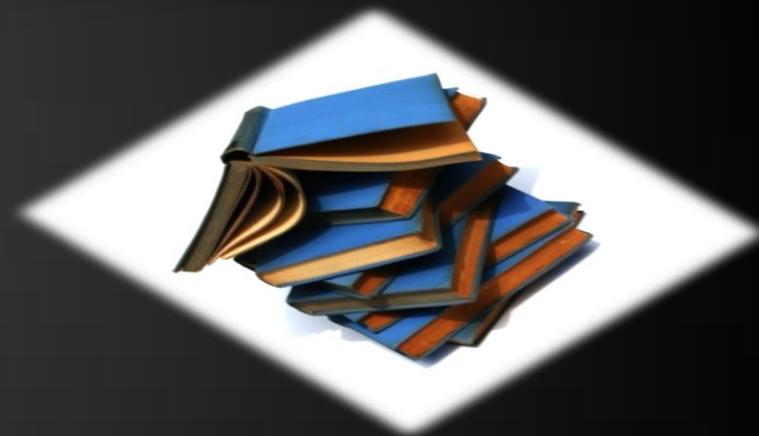


ASP.NET Web Controls and HTML Controls

ASP.NET Web Forms
Telerik Software Academy
<http://academy.telerik.com>



1. Controls Class Hierarchy in Web Forms
2. HTML Server Controls
3. Web Server Controls
 - ◆ Basic Web Controls
 - ◆ Validation Controls
 - ◆ List Controls
 - ◆ Rich Controls
4. HTML Escaping



What is ASP.NET Server Control?

- ◆ **ASP.NET server controls**
 - The simplest ASP.NET components
 - Wrap an HTML UI element, or more complex UI
 - Component-oriented programming model
 - Executed and rendered at the server side
- ◆ **Example of ASP.NET server controls:**
 - <asp:Button> → <input type="submit">
 - <asp:Label> →
 - <asp:GridView> → <table><tr><td>...

What is ASP.NET Server Control ?(2)

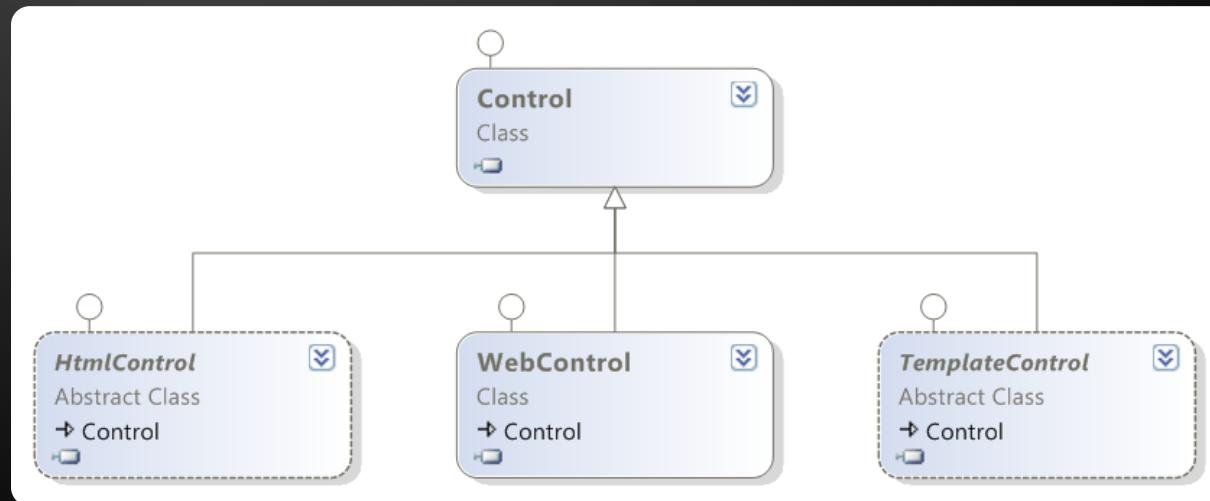
- ◆ Mandatory properties for all server controls:
 - ◆ `runat="server"`
 - ◆ `ID="..."`
- ◆ Programming model based on events
 - ◆ Each user interaction causes an event
 - ◆ Developer decides which events to handle
- ◆ Browser-specific HTML is generated
 - ◆ Controls deliver appropriate HTML depending on browser type



Controls – Class Hierarchy

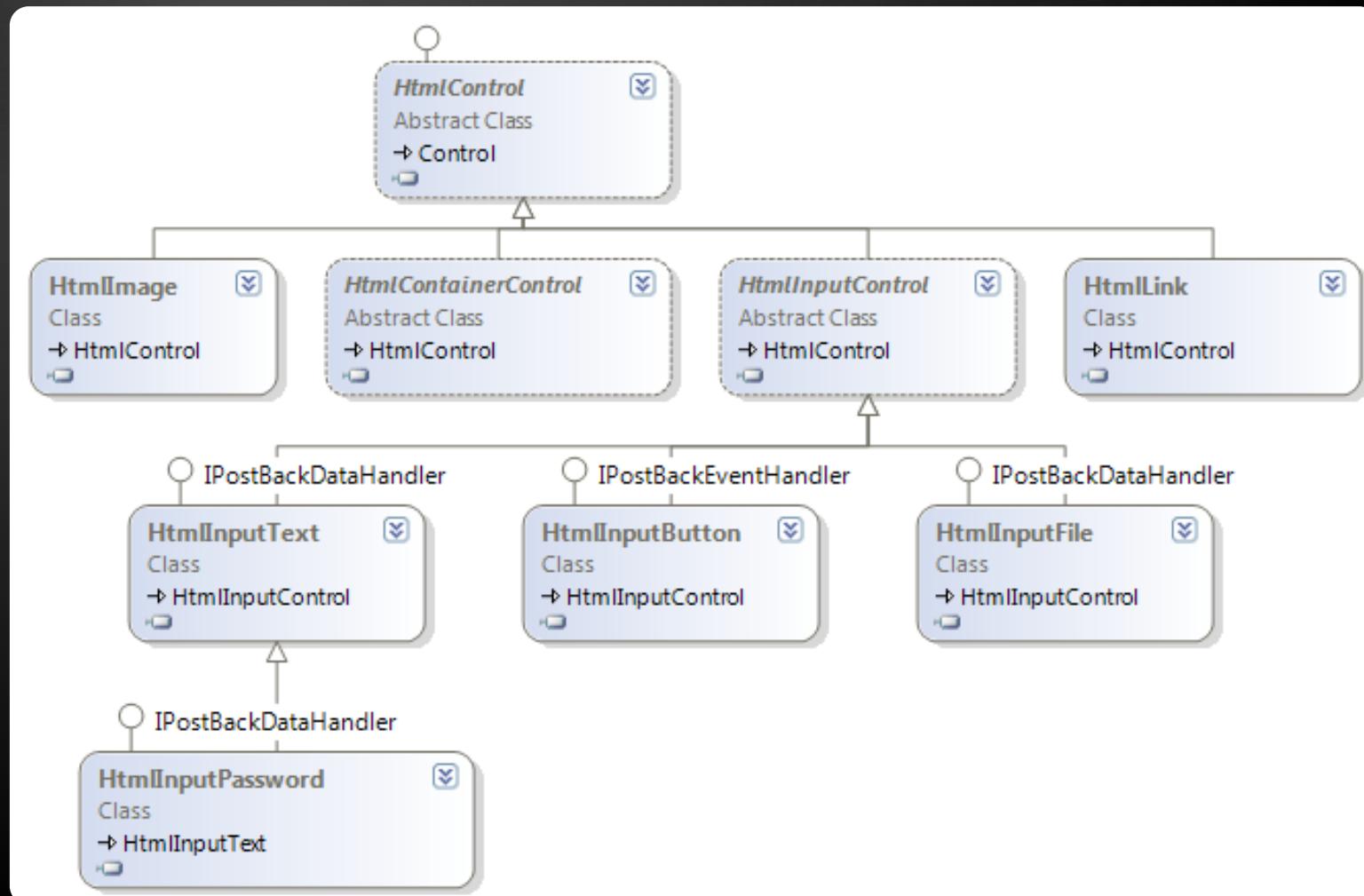
Controls – Class Hierarchy

- ◆ **System.Web.UI.Control**
 - ◆ **Base class for all controls**
 - ◆ **Properties – ID, Page, Context, ViewState, ClientIDMode, Controls , Parent, Visible**
 - ◆ **Methods – Render(HtmlTextWriter writer)**



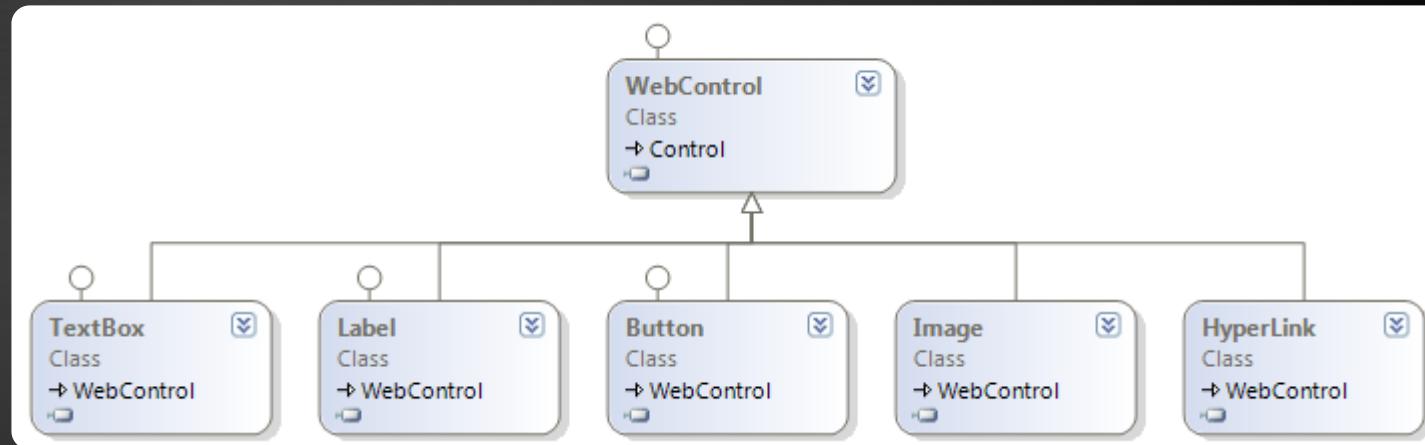
Controls – Class Hierarchy (2)

◆ System.Web.UI.HtmlControls.HtmlControl

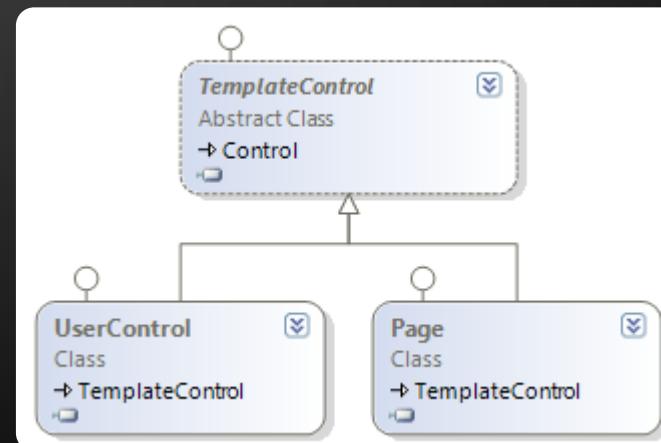


Controls – Class Hierarchy (3)

- ◆ **System.Web.UI.WebControls.WebControl**



- ◆ **System.Web.UI.TemplateControl**



HTML Server Controls



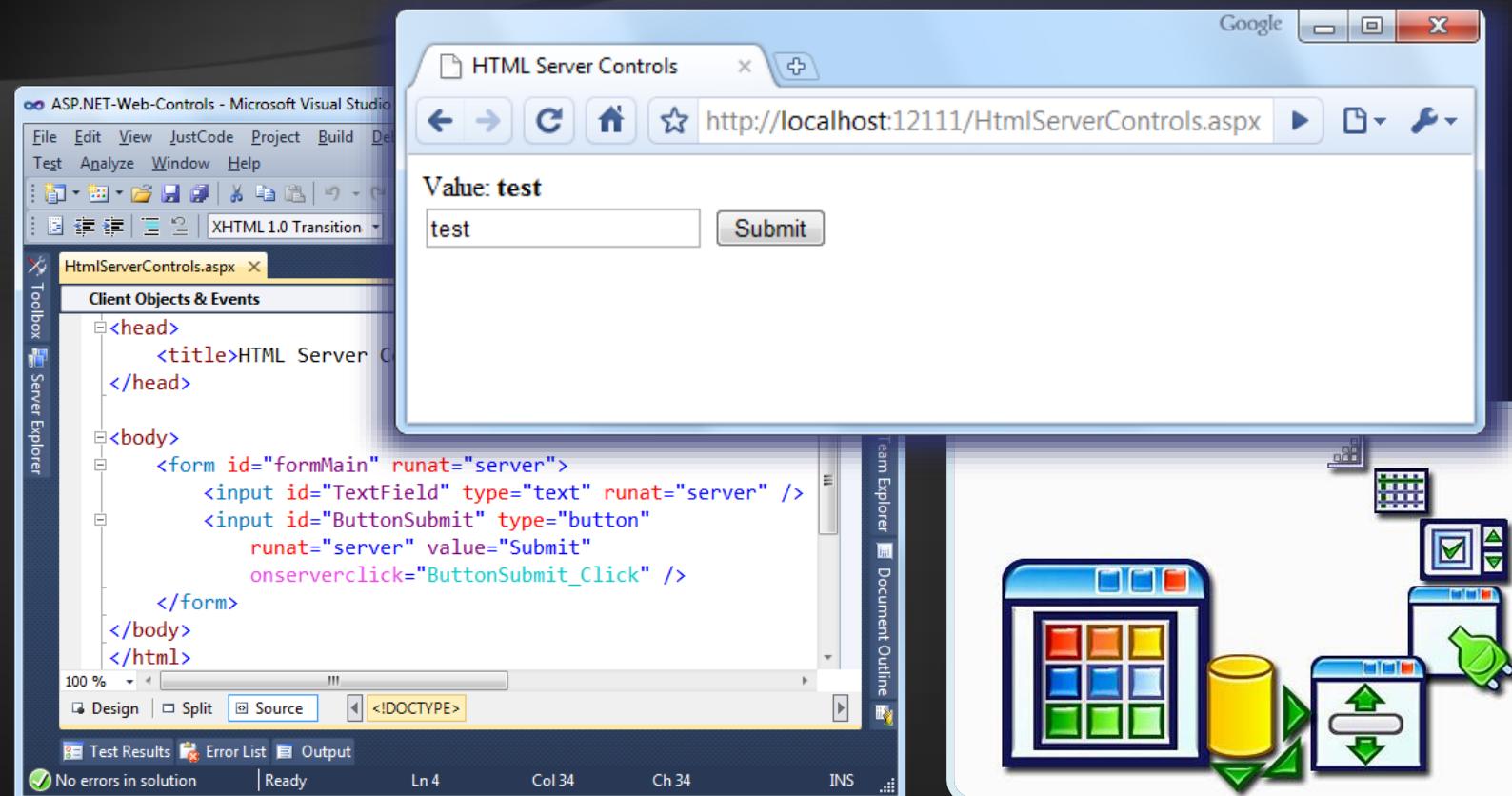
HTML Server Controls

- ◆ HTML server controls are very simple extension of Control class
- ◆ Look like traditional HTML
 - Defined by `runat="server"`
 - Simple HTML seems like text on the server
 - If an HTML element is converted to HTML server control, a server side object is associated with it
- ◆ Valid only inside a Web form tag:

```
<form runat="server">...</form>
```

HTML Server Control – Example

```
<%@ Page Language="C#" %>
<script language="c#" runat="server">
    void ButtonSubmit_Click(Object sender, EventArgs e) {
        Response.Write("Value:<b>" + TextField.Value + "</b>");
    }
</script>
<html>
<head><title>HTML Server Controls</title></head>
<body>
    <form id="formMain" runat="server">
        <input id="TextField" type="text" runat="server" />
        <input id="ButtonSubmit" type="button"
            runat="server" value="Submit"
            onserverclick="ButtonSubmit_Click" />
    </form>
</body>
</html>
```



HTML Server Controls

Live Demo

HTML Server Control Classes

- ◆ **HtmlForm** – <form>...</form>
- ◆ **HtmlInputText** – <input type="text">
- ◆ **HtmlButton** – <input type="button" />
- ◆ **HtmlAnchor** – ...
- ◆ **HtmlSelect** – <input type="select">
- ◆ **HtmlTable, HtmlTableCell, HtmlTableRow**
– <table><tr><td>...</td></tr></table>
- ◆ **HtmlImage** –
- ◆ ...

◆ **HtmlGenericControl**

- ◆ Used for all other HTML elements
- ◆ <p>
- ◆ <div>
- ◆
- ◆ <meta>
- ◆ <body>
- ◆ ...

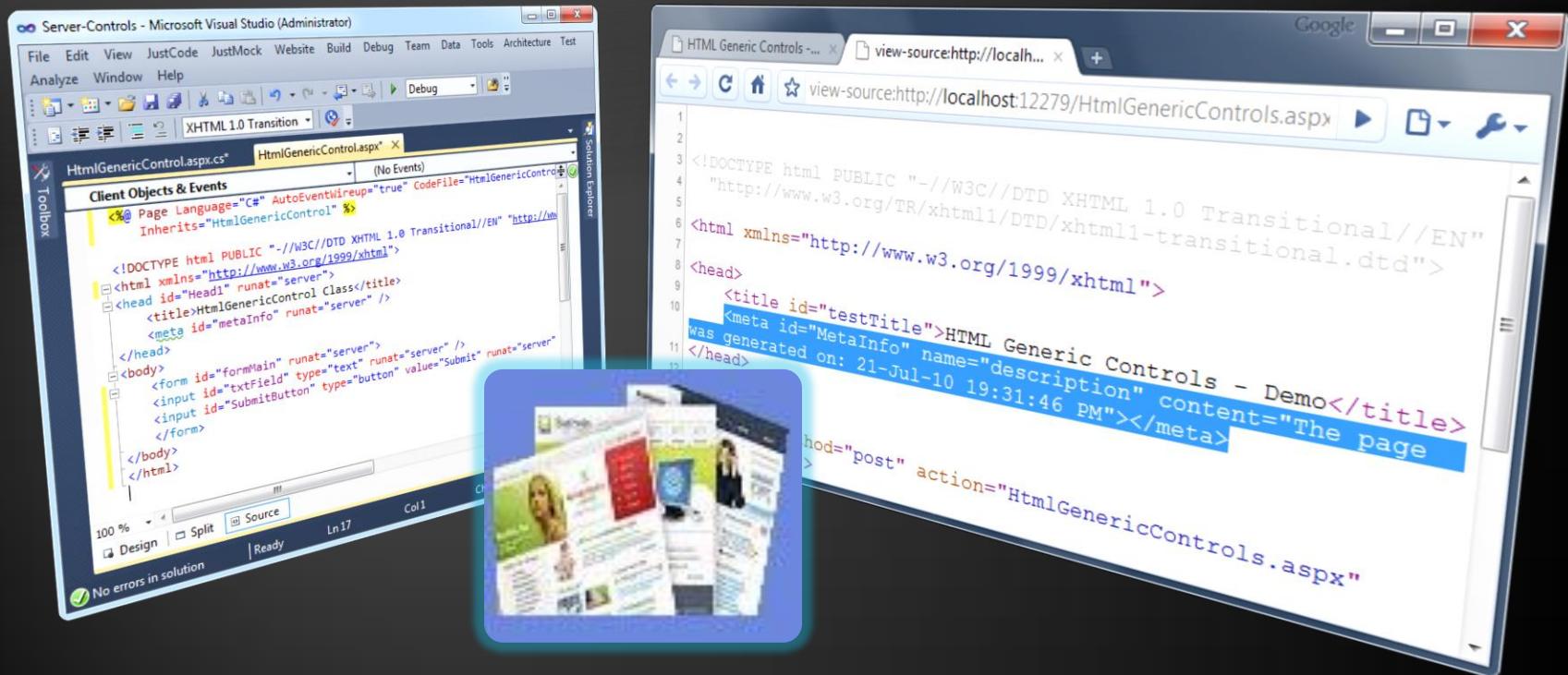


HtmlGenericControl – Example

```
<%@ Page Language="C#" %>

<script runat="server">
    void Page_Load(Object sender, EventArgs e) {
        this.MetaInfo.Attributes["name"] = "description";
        this.MetaInfo.Attributes["content"] = "The page was
            generated on: " + DateTime.Now.ToString();
    }
</script>

<html>
<head>
    <meta id="MetaInfo" runat="server" />
</head>
<body>
    <form id="formMain" runat="server">...</form>
</body>
</html>
```

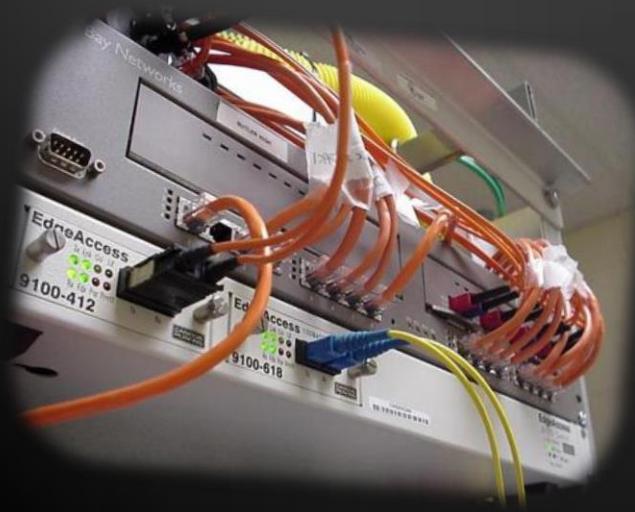


HTML Generic Controls

Live Demo

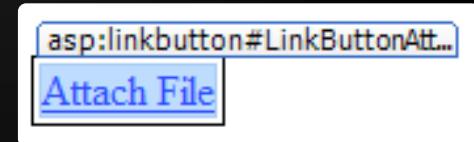
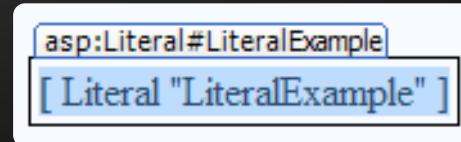
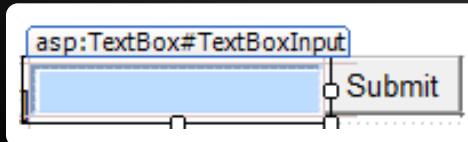


Web Server Controls



- ◆ Web server controls are server UI controls that abstract the common HTML elements
 - ◆ Have own lifecycle and functionality
- ◆ Built-in with .NET Framework
 - ◆ Located in `System.Web.UI.WebControls` namespace
 - ◆ Inherit from the `WebControl` class
- ◆ The rendered HTML tags are quite different from the design-time markup

- ◆ Rich functionality
- ◆ Type-safe programming capabilities
- ◆ Automatic Web browser detection
- ◆ AutoPostBack
 - ◆ Submit when the focus is lost
- ◆ Support for themes



Web Server Controls – Syntax

`tag_prefix`
determines unique
namespace for the
web control

Attributes are
properties of the
Web control

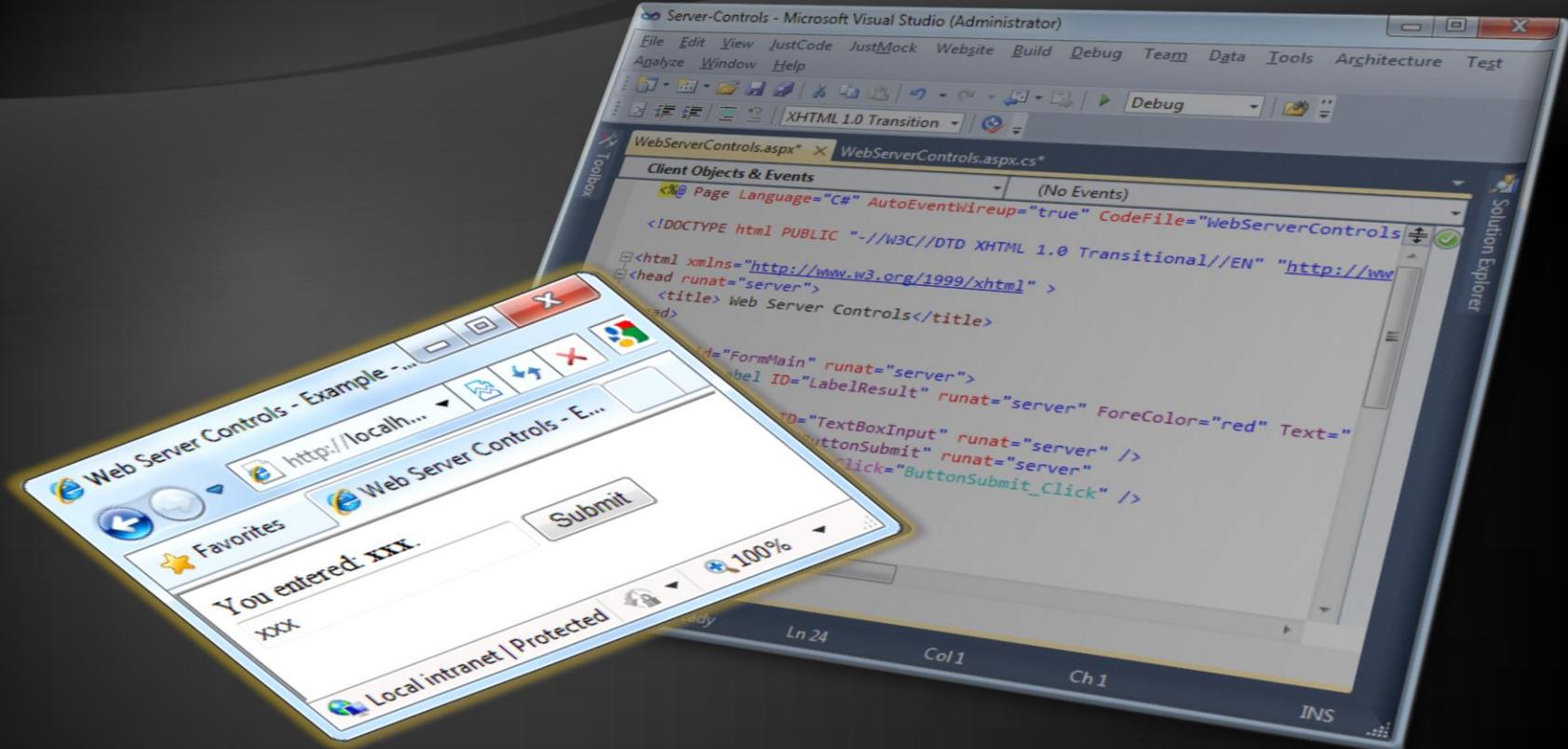
```
<tag_prefix:controlname attributes runat="server"/>
```

The name of
the control

Mandatory
attribute
`runat="server"`

Web Server Control – Example

```
<form id="formMain" runat="server">
    <asp:Label ID="LabelResult" runat="server"
        Text="" Visible="false" />
    <asp:TextBox ID="TextBoxInput" runat="server" />
    <asp:Button ID="ButtonSubmit" runat="server"
        Text="Submit" OnClick="ButtonSubmit_Click" />
</form>
...
protected void ButtonSubmit_Click(
    object sender, EventArgs e)
{
    this.LabelResult.Text =
        "You entered: " + this.TextBoxInput.Text;
    this.LabelResult.Visible = true;
}
```

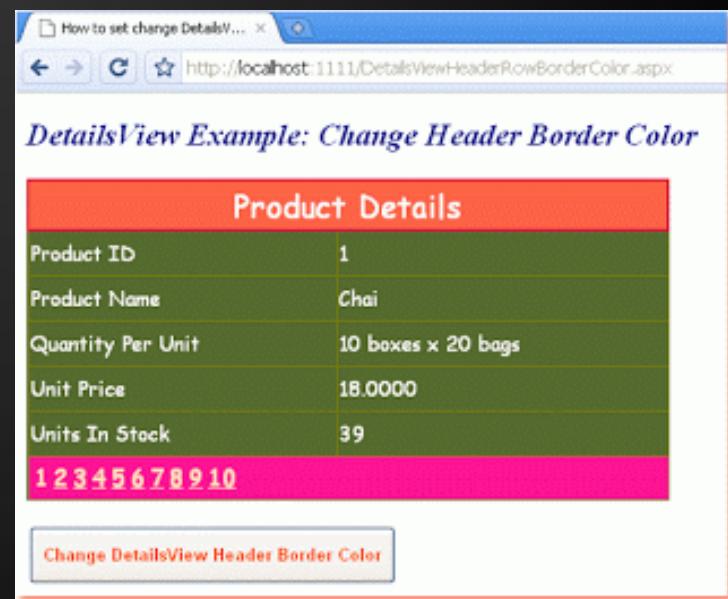


Web Server Controls

Live Demo

System.Web.UI. WebControls.WebControl

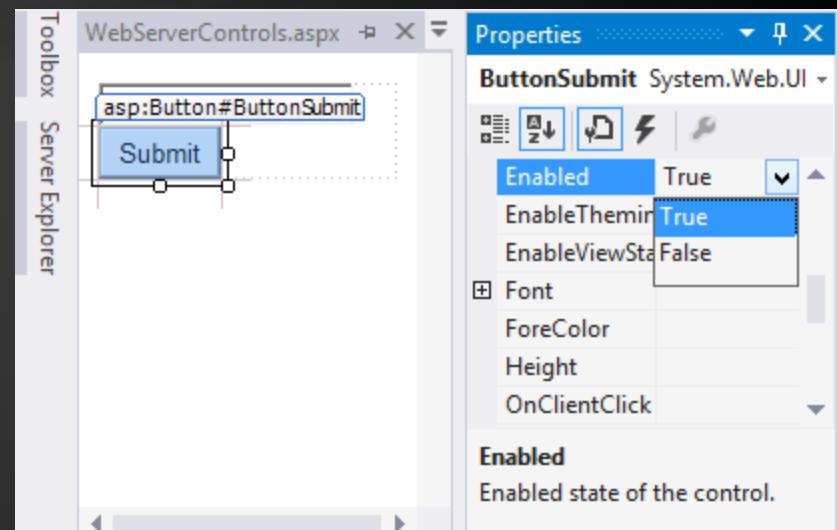
- ◆ The **WebControl** class defines properties, events and methods for all Web controls
- ◆ Control the appearance
 - ◆ **BackColor**
 - ◆ **ForeColor**
 - ◆ **BorderWidth**
 - ◆ **BorderStyle**
 - ◆ **BorderColor**



System.Web.UI. WebControls.WebControl (2)

- ◆ Control the behavior

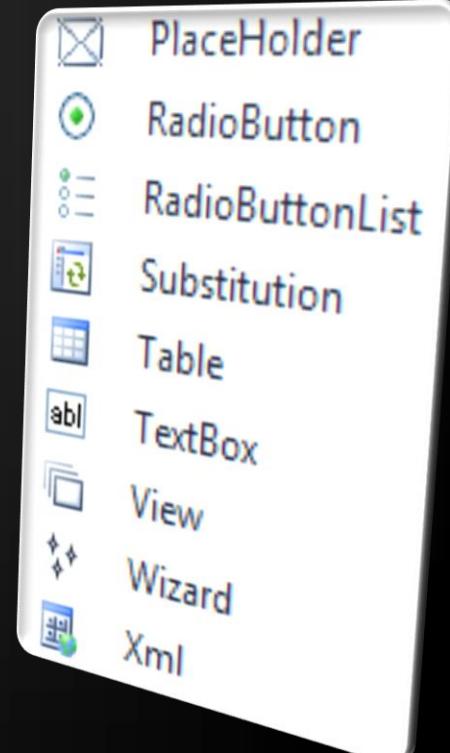
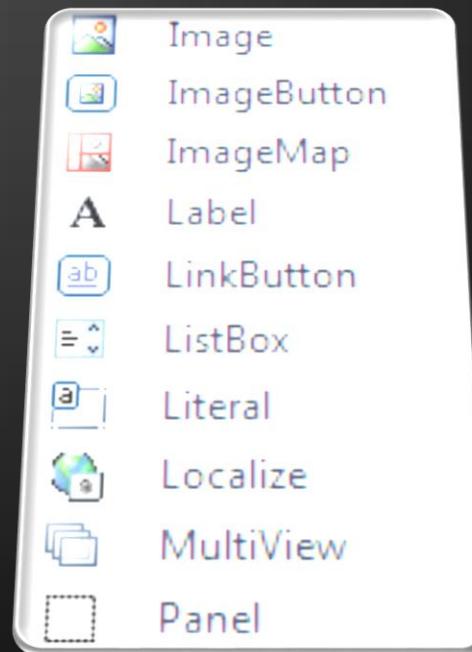
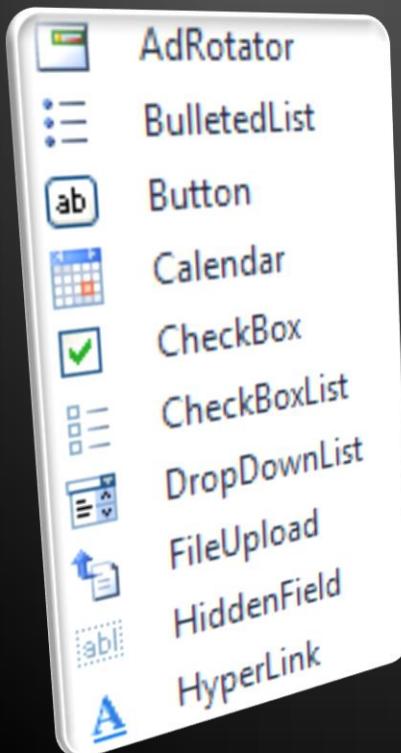
- ◆ Enabled
- ◆ Visible
- ◆ TabIndex
- ◆ ToolTip
- ◆ ...



- ◆ Not all controls support all these properties
- ◆ See the documentation for details

Web Server Controls

Basic Web Controls

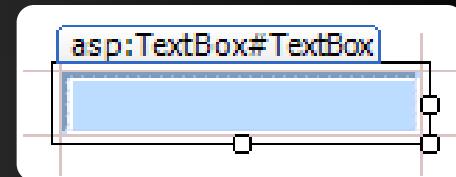


Basic Web Controls ⇔ HTML

<asp:button>	<input type="submit">
<asp:checkbox>	<input type="checkbox">
<asp:hyperlink>	
<asp:image>	
<asp:imagebutton>	<input type="image">
<asp:linkButton>	
<asp:label>	...
<asp:listbox>	<select size="5"></select>
<asp:panel>	<div>...</div>
<asp:radiobutton>	<input type="radio">
<asp:table>	<table>...</table>
<asp:textbox>	<input type="text">

Basic Web Controls: TextBox

- ◆ Creates single-line or multiline text-box
- ◆ Lets the user to enter text
- ◆ Properties
 - ◆ Text
 - ◆ TextMode – SingleLine, MultiLine, Password
 - ◆ MaxLength
 - ◆ ReadOnly
 - ◆ AutoPostBack
- ◆ Events
 - ◆ TextChanged – combined with AutoPostBack



Basic Web Controls: Label

- ◆ Display static text in a <label> input
- ◆ Allows programmatically to manipulate it
- ◆ Properties
 - ◆ Text
- ◆ AssociatedControlID – on click focus goes to the specified control
- ◆ Events
 - ◆ TextChanged – combined with AutoPostBack

CAUTION: the Text is NOT HTML encoded before it is displayed in the Label control.

This make it possible to embed script within HTML tags in the text.

```
asp:label#LabelExample
```

```
I am a label and I hold some text
```

Basic Web Controls: Literal

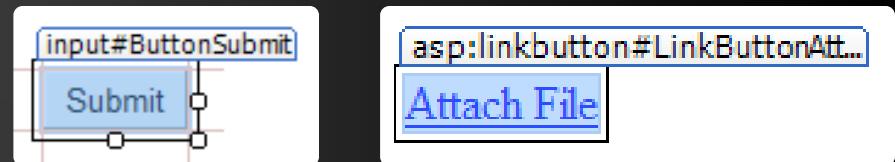
- ◆ Display static text
- ◆ Allows programmatically to manipulate it
 - Unlike the Label control, Literal does not let you apply styles to its content
- ◆ Properties
 - Text
- ◆ Renders the Text property value directly
- ◆ Use Mode="Encode" for automatic escaping

```
asp:Literal#LiteralExample  
[ Literal "LiteralExample" ]
```

CAUTION: the Text is NOT HTML encoded before it is displayed in the Literal control. This make it possible to embed script within HTML tags in the text.

Basic Web Controls – Buttons

- ◆ Implement **IButtonControl**
 - ◆ **Button, LinkButton, RadioButton, ...**
- ◆ **Properties**
 - ◆ **Value** – button's title
 - ◆ **CommandName** – pass a command
 - ◆ **CommandArgument** – pass command arguments
 - ◆ **PostBackUrl** – posts back to specified page
 - ◆ **CausesValidation** – perform validation or not
 - ◆ **ValidationGroup** – which validation group to be validated



◆ Button Events

- ◆ Click
- ◆ Command
 - ◆ **CommandName** and **CommandArgument** are passed to the C# code at the server-side



- ◆ Different button types

- ◆ Standard button

Submit

- ◆ **System.Web.UI.WebControls.Button**
 - ◆ Submits the form by default
 - ◆ Has a command name associated (CommandName property)
 - ◆ Programmatically determine which button is clicked in the Command event handlers

```
<asp:Button ID="ButtonOK" runat="server"
    Text="Click here ..." />
```

- ◆ Different button types

- ◆ Link button

[Link Button](#)

- ◆ Same functionality as Button
 - ◆ Renders as hyperlink
 - ◆ Use Hyperlink if you want to link to another page
 - ◆ Renders JavaScript on the client browser

```
<asp:LinkButton ID="ButtonHomePage"  
runat="server" Text="Home Page" />
```

Basic Web Controls – Buttons (6)

- ◆ Different button types

- ◆ ImageButton

- ◆ Display an image that responds on mouse click
 - ◆ ImageURL – URL to displayed image
 - ◆ Both Click and Command events are raised



```
<asp:ImageButton ID="ButtonSubscribe" runat="server"  
    ImageUrl="~/img/btn_subscribe.png" />
```



Buttons – Example

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Buttons.aspx.cs" Inherits="Buttons" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="formMain" runat="server">
        <asp:Button ID="ButtonEx"
            CommandName="ButtonEx"
            runat="server"
            OnClick="OnBtnClick"
            OnCommand="OnCommand"
            Text="Normal Button" />
        <br />
    </form>
</body>
</html>
```

Buttons – Example (2)

```
<asp:LinkButton ID="LinkButtonEx"
    runat="server"
    OnClick="OnBtnClick"
    Text="Link Button"
    CommandName="LinkButtonEx"
    OnCommand="OnCommand" />
<br />
<asp:ImageButton ID="ImageButtonEx"
    runat="server"
    CommandName="ImageButtonEx"
    ImageUrl("~/images/DotNet_Logo_Small.gif"
    OnCommand="OnCommand"
    OnClick="OnBtnClick" />
<br />
<asp:Label ID="LabelMessage"
    runat="server" Text=""></asp:Label>
</form>
</body>
</html>
```



Buttons Borrow?

Live Demo



Server-Controls - Microsoft Visual Studio (Administrator)

File Edit View JustCode JustMock Website Build Debug Team Data Tools Architecture Test Analyze Window Help

XHTML 1.0 Transition

Buttons.aspx.cs* Buttons.aspx

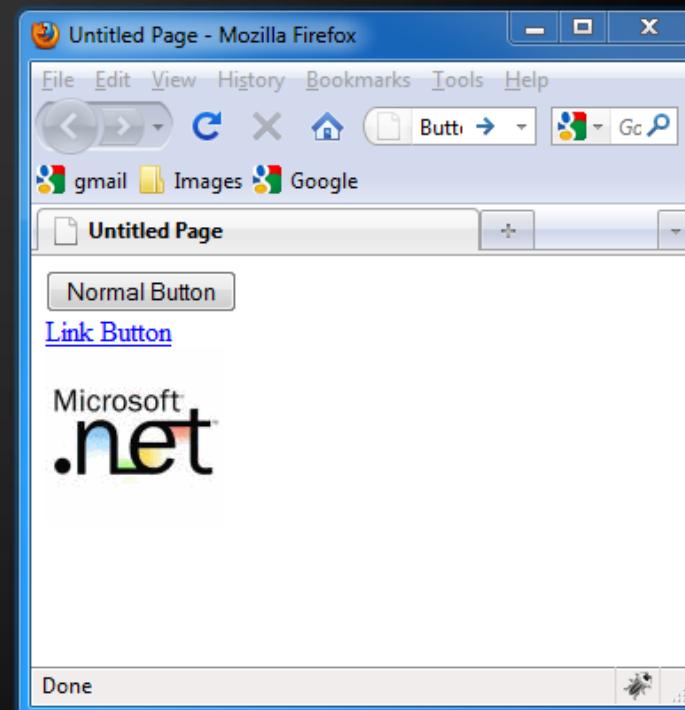
Client Objects & Events (No Events)

```
<form id="formMain" runat="server">
    <asp:Button ID="ButtonEx" CommandName="ButtonEx" runat="server" 
        OnClick="OnBtnClick" OnCommand="OnCommand" Text="Normal Button" />
    <br />
    <asp:LinkButton ID="LinkButtonEx" runat="server" 
        OnClick="OnBtnClick" Text="Link Button" CommandName="LinkButtonEx" 
        OnCommand="OnCommand" />
    <br />
    <asp:ImageButton ID="ImageButtonEx" runat="server" 
        CommandName="ImageButtonEx" ImageUrl="/Images/DotNet_Logo_Small.gif" 
        OnCommand="OnCommand" OnClick="OnBtnClick" />

```

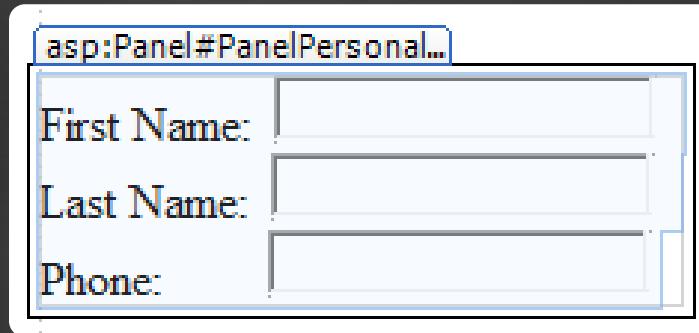
Design Split Source <body> <form#formMain> <asp:Button#ButtonEx>

No errors in solution Ready Ln 13 Col 36 Ch 36 INS



Basic Web Controls – Panel

◆ The Panel control



- ◆ Container for other controls
- ◆ Rendered as <div>
- ◆ Useful for:
 - ◆ Grouping controls
 - ◆ Easier for layout positioning and hiding/showing
 - ◆ Generating and inserting controls at runtime

◆ Properties

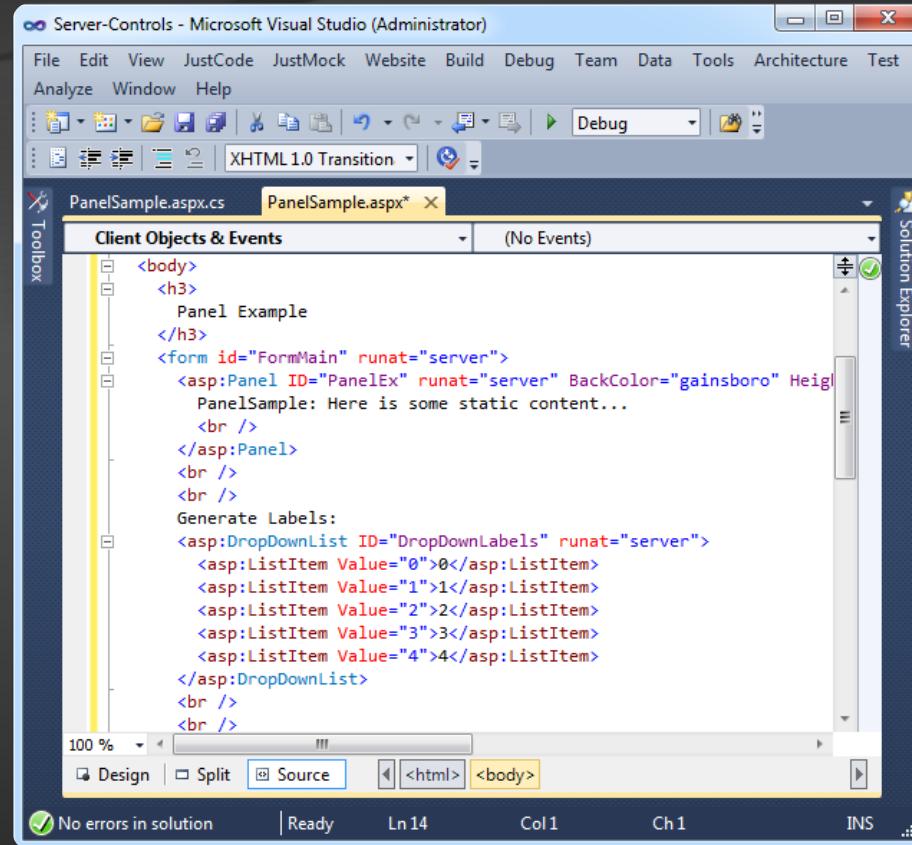
- **ScrollBars** – modify visibility and position of scroll bars
- **Wrap** – value indicating whether the content wraps within the panel
- **GroupingText** – caption for the group of controls that is contained in panel control
- **DefaultButton** – button to be pressed by default (Enter)

Dynamically Generated Controls

- ◆ You may generate ASP.NET controls in the controls tree dynamically

```
protected void Page_Load(object sender, EventArgs e)
{
    TextBox textBox = new TextBox();
    textBox.Text = "Dynamic TextBox";
    this.Controls.Add(textBox);
}
```

- ◆ Add controls in the page / control tree in the Init / Load events
 - ◆ If added later, the events from the dynamic controls will not be able to be handled

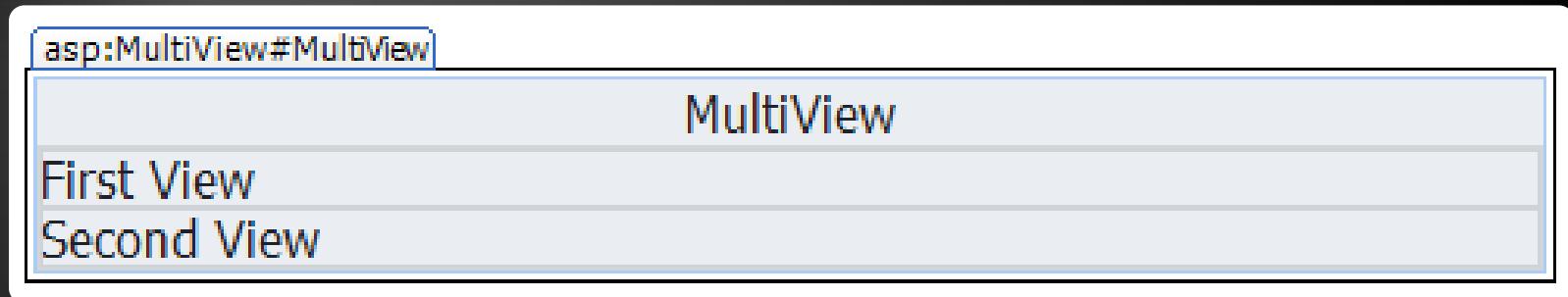


Panels and Dynamic Controls

Live Demo

Basic Web Controls – MultiView

- ◆ MultiView displays one of few panels (views) that reside inside it



- ◆ ActiveViewIndex – indicated which of the views to be displayed
- ◆ SetActiveView – changes the active view
- ◆ Views – holds the views

Basic Web Controls – CheckBox

- ◆ Select between checked / unchecked

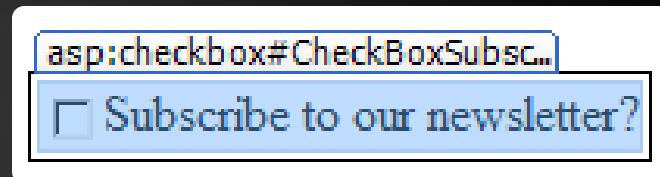
- ◆ Properties

- ◆ Checked

- ◆ Text – control caption

- ◆ AutoPostBack

- ◆ Automatically posts back the page when control state is changed

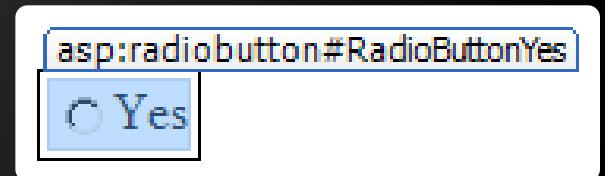
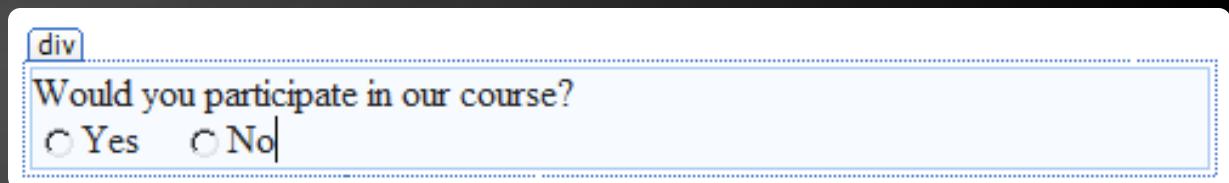


```
<asp:CheckBox ID="CheckBoxAgree" runat="server" />
```

- ◆ CausesValidation – whether validation is performed
- ◆ ValidationGroup – which validation group to be validated
- ◆ Events
 - ◆ CheckChanged

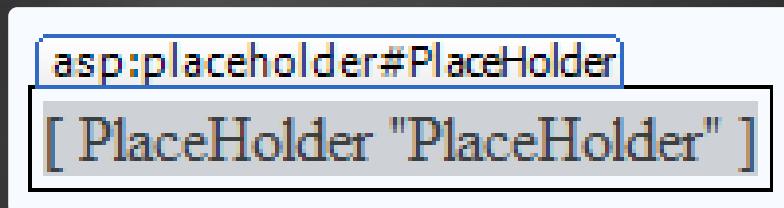
Basic Web Controls – RadioButton

- ◆ Creates a radio button on the Web Forms page
- ◆ Properties
 - ◆ Text
 - ◆ GroupName – allow a mutually exclusive selection from the group
 - ◆ AutoPostBack
 - ◆ Automatically posts back the page when control state is changed
 - ◆ See also RadioButtonList



Basic Web Controls – PlaceHolder

- ◆ The PlaceHolder control



- ◆ Reserves a space in the page control hierarchy
 - ◆ Used to add controls to the page at runtime
 - ◆ Does not produce any visible output
- ◆ The Controls properties
 - ◆ Use it to add, insert or remove controls from PlaceHolder Control

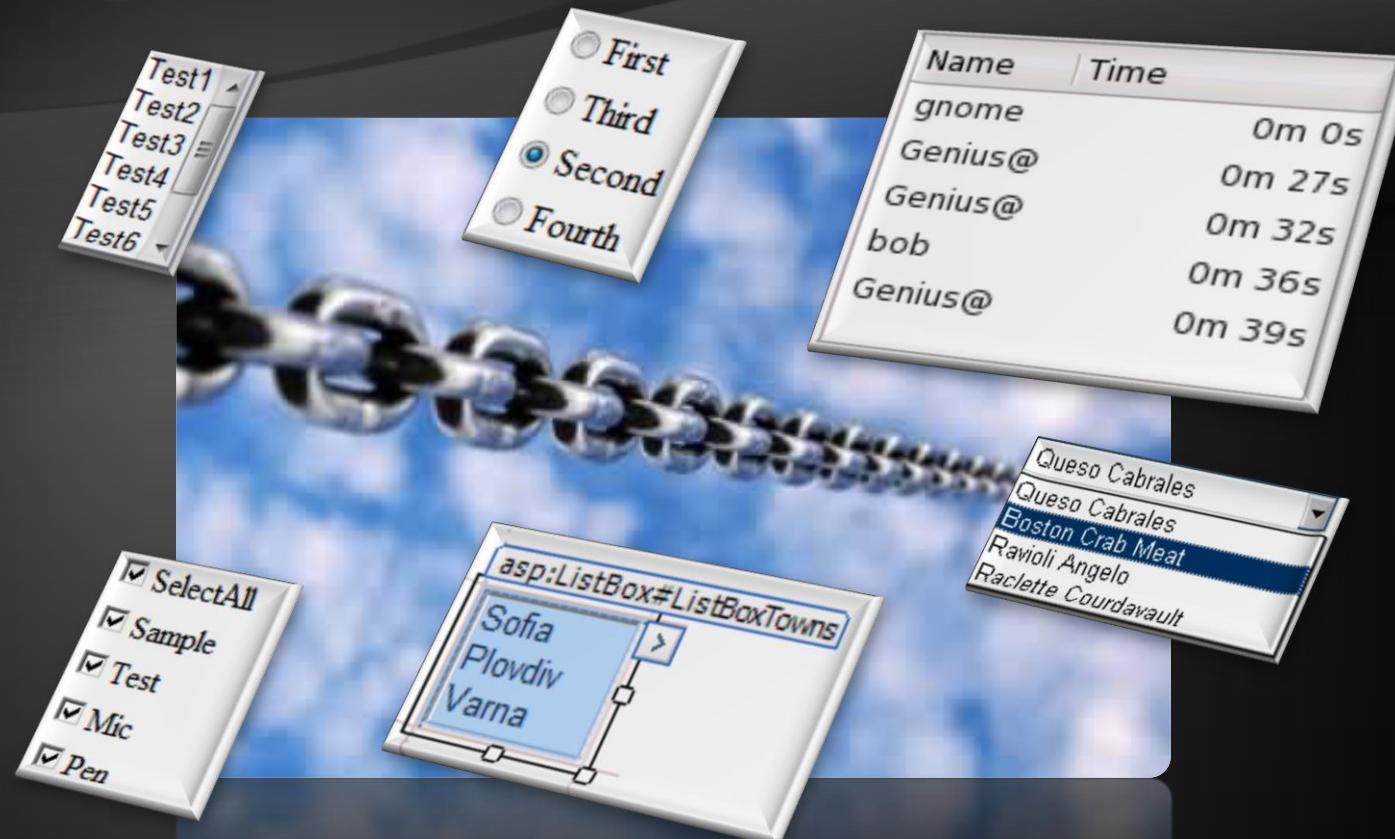


Validation Controls

Performing Control Validation

Validation Controls

- ◆ The ASP.NET Web forms validation controls
 - ◆ Validate the values that are entered into other controls of the page (e.g. in a TextBox)
- ◆ Most important validation controls:
 - ◆ RequiredFieldValidator
 - ◆ RangeValidator
 - ◆ CompareValidator
 - ◆ RegularExpressionValidator
 - ◆ ValidationSummary



List Controls

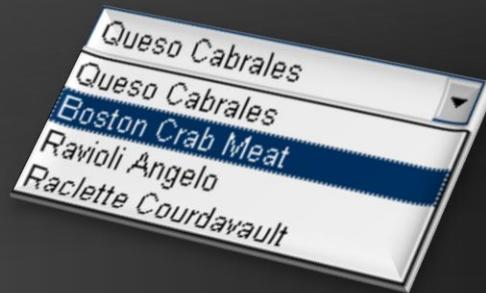
Displaying Lists of Items

- ◆ **List Web controls**

- ◆ **Display list of items, e.g. table of rows**
- ◆ **Support binding to a collection**
- ◆ **Display rows of data in templated format**
- ◆ **Expose data binding properties:**
DataSourceID, DataSource, DataMember
 - ◆ **Bind to collection that support `IEnumerable`, `ICollection` or `IListSource`**
 - ◆ **Can bind to databases through Entity Framework**

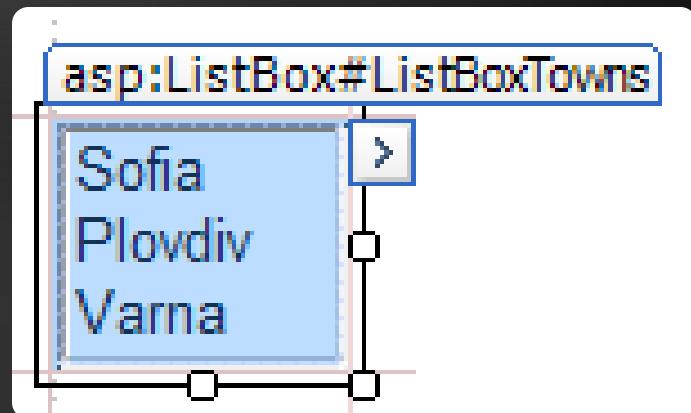
List Controls (2)

- ◆ **ListBox**
- ◆ **CheckBoxList**
- ◆ **RadioButtonList**
- ◆ **BulletedList**
- ◆ **Repeater**
- ◆ **DataList**
- ◆ **GridView**
- ◆ **DropDownList**
- ◆ **ListView**



◆ The ListBox control

- Holds a list of items
- Each item has text and value
- SelectionMode: Single or Multiple
- Items can be data-bound or provided statically



```
<asp:ListBox ID="ListBoxTowns" runat="server">
    <asp:ListItem Value="1">Sofia</asp:ListItem>
    <asp:ListItem Value="2">Plovdiv</asp:ListItem>
    <asp:ListItem Value="3">Varna</asp:ListItem>
</asp:ListBox>
```

DropDownList

- ◆ The DropDownList control (combo-box)

- ◆ Allows choosing among a list of items, just like ListBox
- ◆ Each item has text and value
- ◆ Items can be data-bound or provided statically

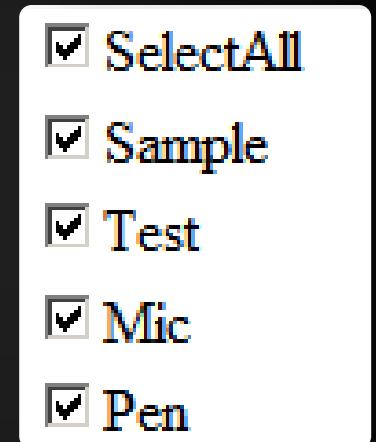


```
<asp:DropDownList ID="DropDownListTransport"  
runat="server" AutoPostBack="True">  
    <asp:ListItem Value="2">Bus</asp:ListItem>  
    <asp:ListItem Value="1">Train</asp:ListItem>  
</asp:DropDownList>
```

- ◆ **BulletedList** displays data in the form of a list of bullets
 - ◆ **DisplayMode**
 - ◆ **Text, HyperLink, LinkButton**
 - ◆ **Ordered or unordered**
 - ◆ **BulletStyle**
 - ◆ **Can be Circle, Disk, LowerRoman, ...**
 - ◆ **BulletImageUrl**
 - ◆ **FirstBulletNumber**

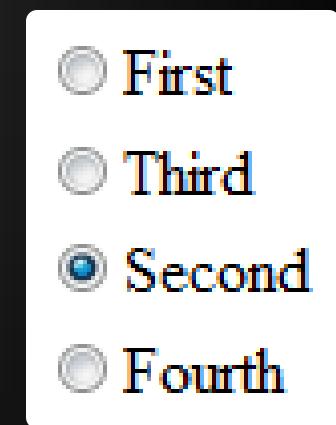
- [Home](#)
- [Books](#)
- [Electronics](#)
- [DVDs](#)
- [Computers](#)

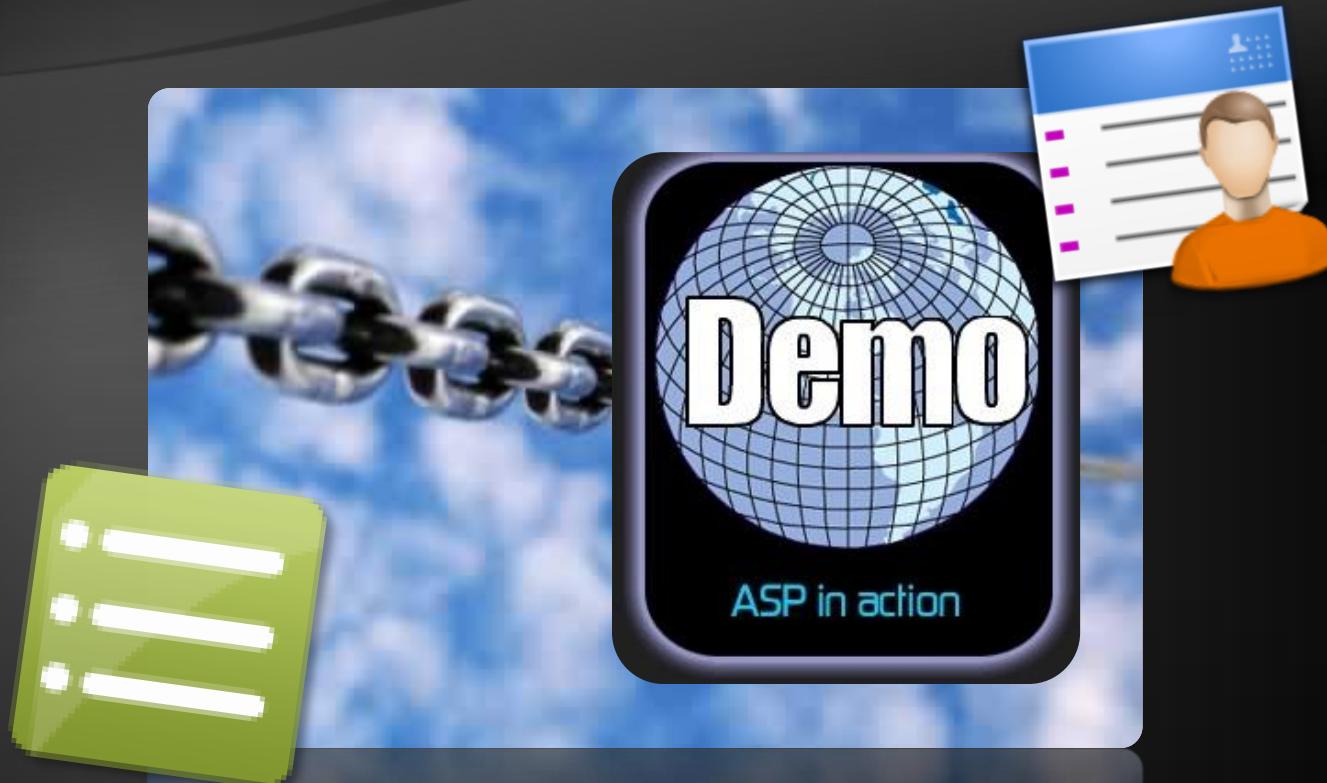
- ◆ CheckBoxList displays items as check boxes
 - ◆ RepeatColumns
 - ◆ The number of columns
 - ◆ RepeatDirection
 - ◆ Vertical, Horizontal
 - ◆ RepeatLayout: Table, Flow



```
<asp:CheckBoxList ID="Extras" runat="server">
    <asp:ListItem Text="Audio System" Value="1" />
    <asp:ListItem Text="Parktronic" Value="2"
        Selected="True" />
</asp:CheckBoxList>
```

- ◆ RadioButtonList displays data as a list of RadioButton controls
 - ◆ RepeatColumns – the number of columns displayed
 - ◆ RepeatDirection
 - ◆ Vertical, Horizontal
 - ◆ RepeatLayout
 - ◆ Table, Flow
 - ◆ Use the Items property to access its elements





List Controls

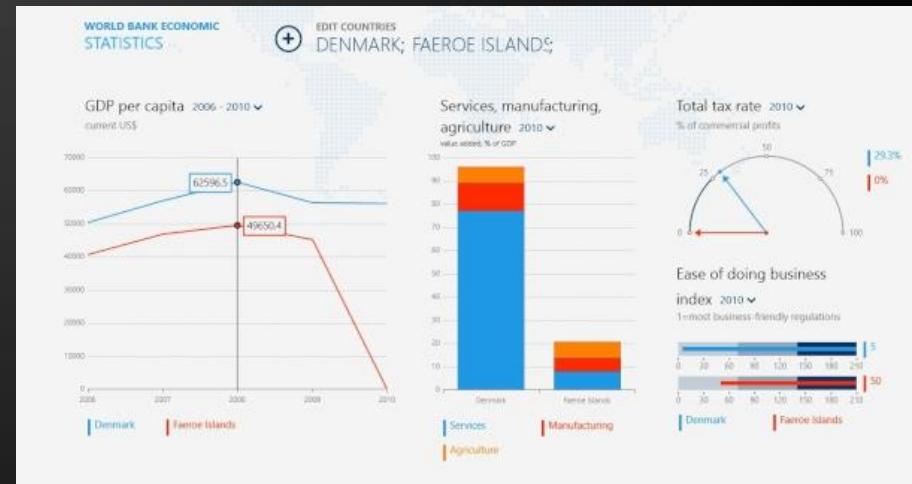
Live Demo

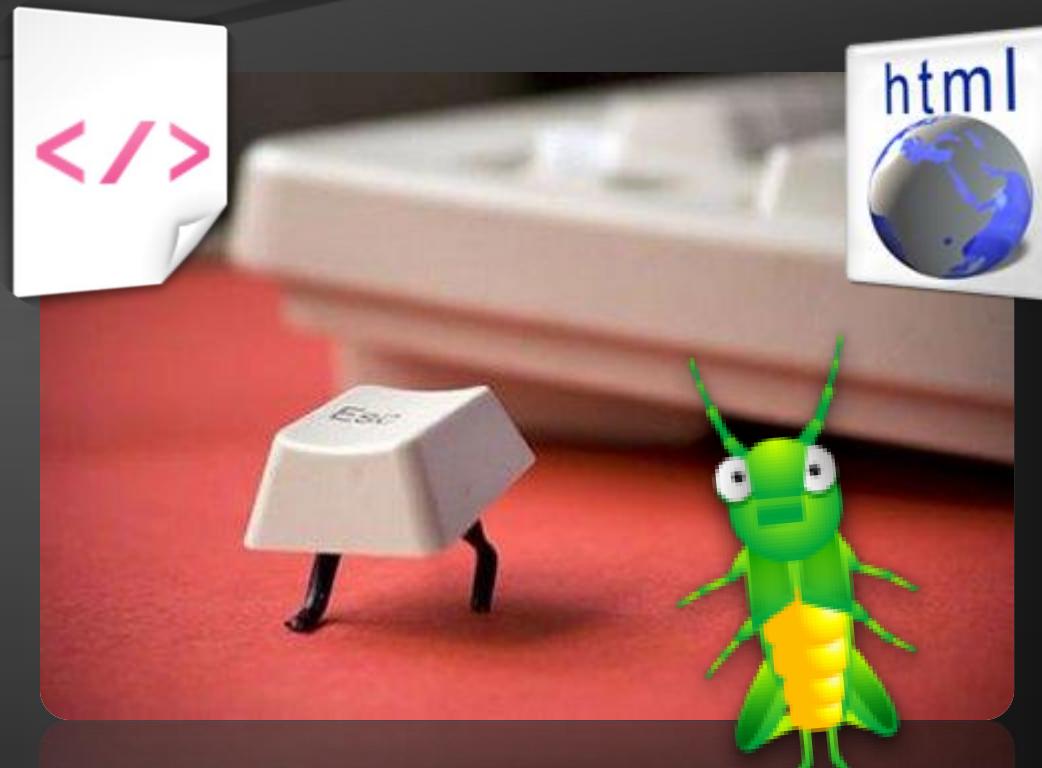
Web Server Controls

Rich Controls



- ◆ Task-specific controls
- ◆ Built with multiple HTML elements
 - ◆ Encapsulate more complex functionality
- ◆ Rich functionality
- ◆ Examples:
 - ◆ Calendar
 - ◆ Menu
 - ◆ AdRotator
 - ◆ CreateUserWizard





HTML Escaping

What is HTML Escaping?

- ◆ HTML escaping is the act of replacing special characters with their HTML entities
 - ◆ Escaped characters are interpreted as character data instead of mark up
- ◆ Typical characters to escape
 - ◆ <, > – start / end of HTML tag
 - ◆ & – start of character entity reference
 - ◆ ', " – text in single / double quotes
 - ◆ ...



HTML Character Escaping

- ◆ Each character could be presented as HTML entity escaping sequence
- ◆ Numeric character references:
 - 'λ' is λ, λ or λ
- ◆ Named HTML entities:
 - 'λ' is λ
 - '<' is <
 - '>' is >
 - '&' is &
 - " (double quote) is "

01	®	02	®	03	♥	04	♦	05	♣	06	♠	07	●	08	■	09	○	0A	□	0B	○	0C	♀	0D	♂	0E	♪	0F	♫
10	▶	◀	↕	!!	¶	§	▀	▀	▀	▀	▀	▀	▀	▀	▀	▀	▀	▀	→	←	↔	↑	↓	↑↓	↑↑	↓↓			
20	!	“	”	#	\$	%	“	”	“	”	“	”	“	”	“	”	“	”	“	”	“	”	“	”	“	”	“	”	
30	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	„	
40	Ⓐ	Ⓑ	Ⓒ	Ⓓ	Ⓔ	Ⓕ	Ⓖ	Ⓗ	Ⓘ	Ⓛ	Ⓜ	Ⓝ	Ⓣ	Ⓤ	Ⓛ	Ⓜ	Ⓝ	Ⓣ	Ⓤ	Ⓛ	Ⓜ	Ⓝ	Ⓣ	Ⓤ	Ⓛ	Ⓜ	Ⓝ	Ⓣ	
50	Ⓟ	Ⓡ	Ⓢ	Ⓣ	Ⓤ	Ⓥ	Ⓦ	Ⓧ	Ⓨ	Ⓩ	ⓐ	ⓑ	ⓒ	ⓓ	ⓔ	ⓕ	ⓖ	ⓗ	ⓘ	ⓙ	ⓚ	ⓛ	ⓜ	ⓝ	ⓞ	ⓟ	ⓠ	ⓡ	
60	ⓐ	ⓑ	ⓒ	ⓓ	ⓔ	ⓕ	ⓖ	ⓗ	ⓘ	ⓙ	ⓚ	ⓜ	ⓝ	ⓜ	ⓝ	ⓜ	ⓝ	ⓜ	ⓝ	ⓜ	ⓝ	ⓜ	ⓝ	ⓜ	ⓝ	ⓜ	ⓝ	ⓜ	
70	ⓟ	ⓟ	ⓡ	ⓢ	ⓣ	ⓤ	ⓥ	ⓤ	ⓥ	ⓤ	ⓥ	ⓤ	ⓥ	ⓤ	ⓥ	ⓤ	ⓥ	ⓤ	ⓥ	ⓤ	ⓥ	ⓤ	ⓥ	ⓤ	ⓥ	ⓤ	ⓥ	ⓤ	
80	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵	₵
90	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ
A0	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ
B0	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ
C0	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ
D0	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ
E0	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ
F0	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ	Ѐ

- ◆ `HttpServerUtility.HtmlEncode`
 - ◆ **HTML encodes a string and returns the encoded (html-safe) string**
 - ◆ **Page.Server is instance of HttpServerUtility**

Example:

```
<%response.write(Server.HtmlEncode("The image tag: <img>"))%>
```

Output:

```
The image tag: &lt;img&gt;
```

Web browser renders the following:

```
The image tag: <img>
```

How to Encode HTML Entities? (2)

- ◆ You could also use the following syntax to escape text with HTML tags:

```
<%: "I want to display an <img> tag here." %>
```

- ◆ The best way to safely display text in ASP.NET page is to use Literal with Mode="Encode":

```
<asp:Literal Mode="Encode" Text="You could freely  
use <img> and <p> tags here." />
```

- ◆ Never do double-escaping!
- ◆ Never store HTML escaped text in the DB!

How to Encode HTML Entities? (3)

- ◆ TextBox text does not need to be escaped

```
<asp:TextBox Mode="Encode" Text="You could freely  
use <img> and <p> tags here." />
```

- ◆ Label text needs to be escaped

```
<asp:Label Mode="Encode"  
Text="<script>alert('bug!')</script>" />
```

- ◆ Difference between %, %: and %=

```
<% “<b>This will not be escaped!</b>” %>  
<%: "I want to display an <img> tag here." %>  
<%= "<script>alert('Not good!')</script>" %
```

- ◆ Cross-site scripting (XSS) is a common security vulnerability in Web applications
 - ◆ Web application is let to display a JavaScript code that is executed at the client's browser
 - ◆ Crackers could take control over sessions, cookies, passwords, and other private data
 - ◆ How to prevent XSS?
 - ◆ Validate the user input (built-in in ASP.NET)
 - ◆ Perform HTML escaping when displaying text data in a Web control

Automatic Request Validation

- ◆ ASP.NET applies automatic request validation
- ◆ Controlled by the ValidateRequest attribute of Page directive
 - Checks all input data against a hard-coded list of potentially dangerous values
 - The default is true
 - Using it could harm the normal work on most applications
 - E.g. a user posts JavaScript code in a forum
 - Escaping is a better way to handle the problem!

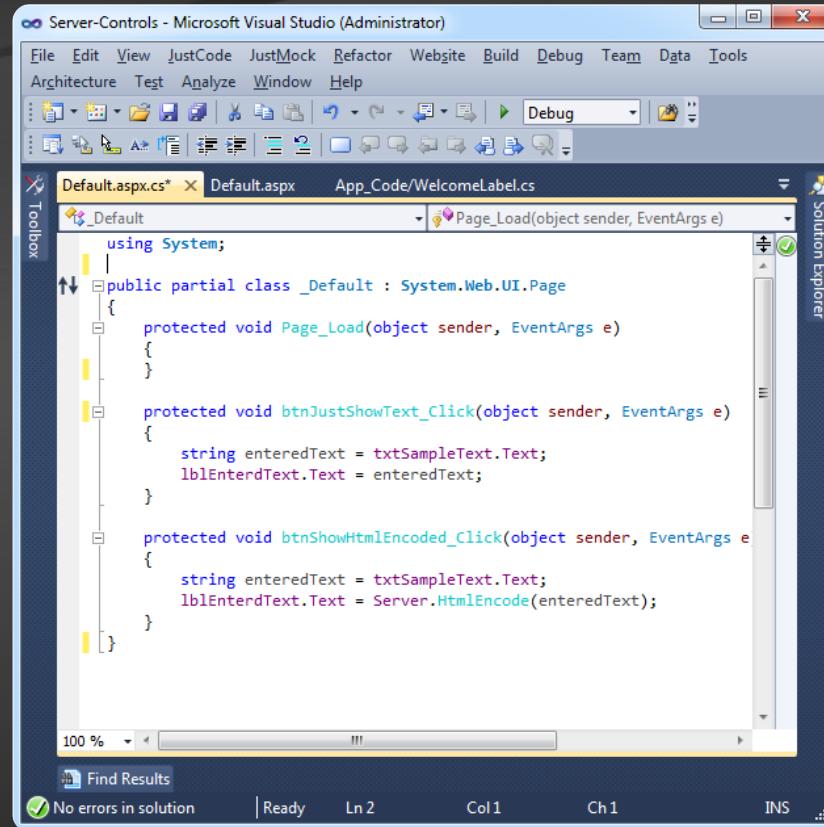
Bad Characters Protection

- ◆ The ASP.NET built-in protection against XSS
 - ◆ By default stops all HTTP requests that send un-escaped HTML code
 - ◆ An error message is shown when a form sends HTML to the server

500 Internal Server Error: A potentially dangerous Request.Form value was detected from the client (...)

- ◆ Disable the HTTP request validation for all pages in Web.config (in <system.web>):

```
<httpRuntime requestValidationMode="2.0" />
<pages validateRequest="false" />
```



HTML Escaping

Live Demo

ASP.NET Web Controls and HTML Controls

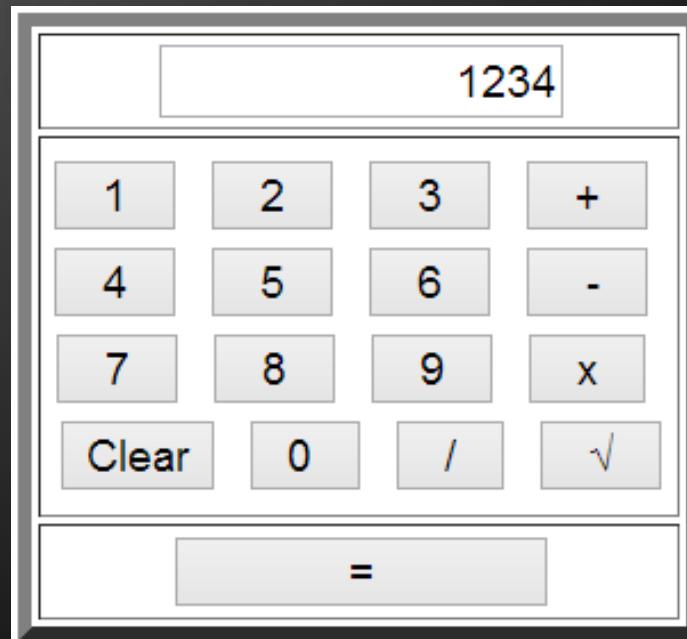
Questions?

1. Using the HTML server controls create a Web application for generating random numbers. It should have two input fields defining a range (e.g. [10..20]) and a button to generate a random number in the specified range.
2. Re-implement the same using Web server controls.
3. Define a Web form with text box and button. On button click show the entered in the first textbox text in other textbox control and label control. Enter some potentially dangerous text. Fix issues related to HTML escaping – the application should accept HTML tags and display them correctly.

4. Make a simple Web form for registration of students and courses. The form should accept first name, last name, faculty number, university (drop-down list), specialty (drop-down list) and a list of courses (multi-select list) and display them on submit. Use the appropriate Web server controls. After submission you should display summary of the entered information as formatted HTML. Use dynamically generated tags (<h1>, <p>, ...).
5. * Implement the "Tic-tac-toe" game using Web server controls. The user should play against the computer which follows some fixed strategy.

Homework (3)

6. Make a simple Web Calculator. The calculator should support the operations like addition, subtraction, multiplication, division and square root. Validation is essential!



Free Trainings @ Telerik Academy

- ◆ "Web Design with HTML 5, CSS 3 and JavaScript" course @ Telerik Academy



- ◆ html5course.telerik.com

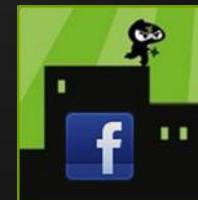
- ◆ Telerik Software Academy

- ◆ academy.telerik.com



- ◆ Telerik Academy @ Facebook

- ◆ [facebook.com/TelerikAcademy](https://www.facebook.com/TelerikAcademy)



- ◆ Telerik Software Academy Forums

- ◆ forums.academy.telerik.com

