

ASP.NET HTML SERVER CONTROLS

HTML elements in ASP.NET files are, by default, treated as text. To make these elements programmable, add a `runat="server"` attribute to the HTML element. This attribute indicates that the element should be treated as a server control.

Note: All HTML server controls must be within a `<form>` tag with the `runat="server"` attribute!

Note: ASP.NET requires that all HTML elements must be properly closed and properly nested.

HTML Server Control	Description
HtmlAnchor	Controls an <code><a></code> HTML element
HtmlButton	Controls a <code><button></code> HTML element
HtmlForm	Controls a <code><form></code> HTML element
HtmlGeneric	Controls other HTML element not specified by a specific HTML server control, like <code><body></code> , <code><div></code> , <code></code> , etc.
HtmlImage	Controls an <code><image></code> HTML element
HtmlInputButton	Controls <code><input type="button"></code> , <code><input type="submit"></code> , and <code><input type="reset"></code> HTML elements
HtmlInputCheckBox	Controls an <code><input type="checkbox"></code> HTML element
HtmlInputFile	Controls an <code><input type="file"></code> HTML element
HtmlInputHidden	Controls an <code><input type="hidden"></code> HTML element
HtmlInputImage	Controls an <code><input type="image"></code> HTML element
HtmlInputRadioButton	Controls an <code><input type="radio"></code> HTML element
HtmlInputText	Controls <code><input type="text"></code> and <code><input type="password"></code> HTML elements
HtmlSelect	Controls a <code><select></code> HTML element
HtmlTable	Controls a <code><table></code> HTML element
HtmlTableCell	Controls <code><td></code> and <code><th></code> HTML elements
HtmlTableRow	Controls a <code><tr></code> HTML element
HtmlTextArea	Controls a <code><textarea></code> HTML element

➤ **ASP.NET HtmlAnchor Control**

- **Definition and Usage**

The HtmlAnchor control is used to control an <a> element. In HTML, the <a> element is used to create a hyperlink. The hyperlink may link to a bookmark or to another Web page.

- **Syntax:**

```
<a id="link1" runat="server">Visit W3Schools! </a>
```

- ***Properties***

Property	Description
Attributes	Returns all attribute name and value pairs of the element
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
HRef	The URL target of the link
id	A unique id for the control
InnerHtml	Sets or returns the content between the opening and closing tags of the HTML element. Special characters are not automatically converted to HTML entities
InnerText	Sets or returns all text between the opening and closing tags of the HTML element. Special characters are automatically converted to HTML entities
Name	The name of the anchor
OnServerClick	The name of the function to be executed when the link is clicked
runat	Specifies that the control is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Target	The target window to open
Title	A title to be displayed by the browser (like the alt attribute of the img element)
Visible	A Boolean value that indicates whether or not the control should be visible

Examples

In this example we declare two HtmlAnchor controls in an .aspx file (remember to embed the control inside an HtmlForm control). Then we modify the URL, target, and title of the HtmlAnchor controls in an event handler (an event handler is a subroutine that executes code for a given event). The Page_Load event is one of many events that ASP .NET understands.

- **Source Code:**

```
<script runat="server">
void Page_Load (Object Sender, EventArgs e)

{
    link1.HRef="http://www.w3schools.com"
    link1.Target="_blank"
    link1.Title="W3Schools"

    link2.HRef="http://www.microsoft.com"
    link2.Target="_blank"
    link2.Title="Microsoft"
}

</script>

<html>
<body>

<form runat="server">
<a id="link1" runat="server">Visit W3Schools!</a>
<br />
<a id="link2" runat="server">Visit Microsoft!</a>
</form>

</body>
</html>
```

- **Output:**

[Visit W3Schools!](http://www.w3schools.com)
[Visit Microsoft!](http://www.microsoft.com)

➤ ASP.NET HtmlButton Control

- **Definition and Usage**

The HtmlButton control is used to control a <button> element. In HTML, the <button> element is used to create a push button.

- **Syntax:**

```
<button id="b1" OnServerClick="button1" style="background-color:#e6e6fa; height=25; width:100" runat="server">Blue button!</button>
```

- **Properties**

Property	Description
Attributes	Returns all attribute name and value pairs of the element
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the control
InnerHTML	Sets or returns the content between the opening and closing tags of the HTML element. Special characters are not automatically converted to HTML entities
InnerText	Sets or returns all text between the opening and closing tags of the HTML element. Special characters are automatically converted to HTML entities
OnServerClick	The name of the function to be executed when the button is clicked
runat	Specifies that the control is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Visible	A Boolean value that indicates whether or not the control should be visible

Examples

In this example we declare two HtmlButton controls in an .aspx file (remember to embed the control inside an HtmlForm control). Next, we write an event handler that specifies what to happen when a button is clicked.

- **Source Code:**

```
<script runat="server">
void button1(Object Sender, EventArgs e){
    p1.InnerHtml="You clicked the blue button!"
}
void button2(Object Sender, EventArgs e)
    {p1.InnerHtml="You clicked the pink button!";}
</script>

<html>
<body>
<form runat="server">
<button id="b1" OnServerClick="button1" style="background-color:#e6e6fa;
height=25;width:100" runat="server">Blue button!
</button>
<button id="b2" OnServerClick="button2" style="background-color:#fff0f5;
height=25;width:100" runat="server">Pink button!
</button>
<p id="p1" runat="server" />
</form>
```

➤ **ASP.NET HtmlForm Control**

- **Definition and Usage**

The HtmlForm control is used to control a <form> element. In HTML, the <form> element is used to create a form.

Note: All HTML server controls must be within the HtmlForm control!

Note: You can only have one HtmlForm control on a single page!

- **Syntax:**

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

- **Properties**

Property	Description
Action	A URL that defines where to send the data when the form is submitted. Note: This attribute is always set to the URL of the page

	itself!
Attributes	Returns all attribute name and value pairs of the element
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
EncType	The mime type used to encode the content of the form
id	A unique id for the control
InnerHTML	Sets or returns the content between the opening and closing tags of the HTML element. Special characters are not automatically converted to HTML entities
InnerText	Sets or returns all text between the opening and closing tags of the HTML element. Special characters are automatically converted to HTML entities
Method	How the form posts data to the server. Legal values are: "post" and "get". Default is "post"
Name	The name of the form
runat	Specifies that the control is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Target	The target window to load the URL
Visible	A Boolean value that indicates whether or not the control should be visible

Examples

In this example we declare one `HtmlInputText` controls, one `HtmlInputButton` control, and one `HtmlGeneric` control in an `.aspx` file (remember to embed the controls inside an `HtmlForm` control). When the submit button is triggered, the submit subroutine is executed. The submit subroutine writes a welcome message to the `p` element.

- **Source Code:**

```

<script runat="server">
void submit(Object Sender, EventArgs e)

{
if (name.value!="" )
    p1.InnerHtml="Welcome " & name.value & "!"
end if
}
</script>

<html>
<body>

```

```

<form runat="server">
Enter your name: <input id="name" type="text" size="30" runat="server" />
<br /><br />
<input type="submit" value="Submit" OnServerClick="submit" runat="server" />
<p id="p1" runat="server" />
</form>

</body>
</html>

```

- **Output:**

Enter your name:

➤ **ASP.NET HtmlGeneric Control**

- **Definition and Usage**

The HtmlGeneric control is used to control other HTML element not specified by a specific HTML server control, like <body>, <div>, , , <p>, etc.

- **Properties**

Property	Description
Attributes	Returns all attribute name and value pairs of the element
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the control
InnerHtml	Sets or returns the content between the opening and closing tags of the HTML element. Special characters are not automatically converted to HTML entities
InnerText	Sets or returns all text between the opening and closing tags of the HTML element. Special characters are automatically converted to HTML entities
runat	Specifies that the control is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name

Visible	A Boolean value that indicates whether or not the control should be visible
---------	---

Examples

In this example we declare one `HtmlInputText` controls, one `HtmlInputButton` control, and one `HtmlGeneric` control (the `<p>` tag) in an `.aspx` file (remember to embed the controls inside an `HtmlForm` control). When the submit button is triggered, the submit subroutine is executed. The submit subroutine writes a welcome message to the `p` element.

- **Source Code:**

Same as Form Control

- **Output:**

Same as above Form Control

➤ **ASP.NET HtmlImage Control**

- **Definition and Usage**

The `HtmlImage` control is used to control an `` element. In HTML, the `` element is used to display an image.

- **Syntax:**

```

```

- **Properties**

Property	Description
Align	How to align the image according to surrounding elements. Legal values are: <ul style="list-style-type: none"> • top • middle • bottom • left

	<ul style="list-style-type: none"> • right
Alt	A short description of the image
Attributes	Returns all attribute name and value pairs of the element
Border	The width of the borders around the image
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
Height	The height of the image
id	A unique id for the control
runat	Specifies that the control is a server control. Must be set to "server"
Src	The URL of the image to display
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Visible	A Boolean value that indicates whether or not the control should be visible
Width	The width of the image

Examples

HTMLImage

In this example we declare an HtmlImage control in an .aspx file (remember to embed the control inside an HtmlForm control). Then we modify the src, alt and border properties of the HtmlImage control in an event handler (an event handler is a subroutine that executes code for a given event). The Page_Load event is one of many events that ASP .NET understands:

• **Source Code:**

```
<script runat="server">
void Page_Load (Object Sender, EventArgs e)

{
    image1.Src="smiley.gif"
    image1.Alt="Smiley"
    image1.Border="3"
}
</script>

<html>
<body>

<form runat="server">
<img id="image1" runat="server" />
</form>
```

```
</body>
</html>
```

- **Output:**



HTMLImage 2

In this example we declare an HtmlImage and an HTMLSelect control in an .aspx file (remember to embed the controls inside an HtmlForm control). Then we modify the src property of the HtmlImage control based on user choices. The value selected in the HtmlSelect control determines which image to display.

- **Source Code:**

```
<script runat="server">
void choose_image(Object Sender, EventArgs e)
```

```
{
    image1.Src = select1.Value
}
</script>
```

```
<html>
<body>
```

```
<form runat="server">
<select id="select1" runat="server">
    <option value="smiley.gif">Smiley</option>
    <option value="angry.gif">Angry</option>
    <option value="stickman.gif">Stickman</option>
</select>
<input type="submit" runat="server" value="Display image"
OnServerClick="choose_image">
<br /><br />

</form>
```

```
</body>
</html>
```

- **Output:**

Smiley

▼

Display image



➤ ASP.NET HtmlInputButton Control

• Definition and Usage

The HtmlInputButton control is used to control `<input type="button">`, `<input type="submit">`, and `<input type="reset">` elements. In HTML, these elements are used to create a command button, a submit button, and a reset button.

• Syntax:

```
<input type="submit" value="Submit" OnServerClick="submit" runat="server" />
```

• Properties

Property	Description
Attributes	Returns all attribute name and value pairs of the element
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the element
Name	The name of the element
OnServerClick	The name of the function to be executed when the button is clicked
runat	Specifies that the element is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Type	The type of the element
Value	The value of the element
Visible	A Boolean value that indicates whether or not the control should be visible

Examples

HTMLInputbutton

In this example we declare one HtmlInputText controls, one HtmlInputButton control, and one HtmlGeneric control in an .aspx file (remember to embed the controls inside an

HtmlForm control). When the submit button is triggered, the submit subroutine is executed. The submit subroutine writes a welcome message to the p element.

- **Source Code:**

```
<script runat="server">
void submit(Object Sender, EventArgs e)

{
if (name.value!="")
    p1.InnerHtml="Welcome " & name.value & "!";
}
</script>

<html>
<body>

<form runat="server">
Enter your name: <input id="name" type="text" size="30" runat="server" />
<br /><br />
<input type="submit" value="Submit" OnServerClick="submit" runat="server" />
<p id="p1" runat="server" />
</form>

</body>
</html>
```

- **Output:**

Enter your name:

Welcome Tom!

➤ **ASP.NET HtmlInputCheckBox Control**

- **Definition and Usage**

The HtmlInputCheckBox control is used to control an <input type="checkbox"> element. In HTML, this element is used to create a checkbox.

- **Syntax:**

```
<input id="red" type="checkbox" runat="server" /> Red
```

- **Properties & Events :**

Property	Description
Attributes	Returns all attribute name and value pairs of the element
Checked	A Boolean value that specifies whether or not the element is to be checked
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the control
Name	The name of the element
runat	Specifies that the element is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Type	The type of the element
Value	The value of the element
Visible	A Boolean value that indicates whether or not the control should be visible

Event	Description
ServerChange	Occurs when the state of the control has changed

Examples

HTMLInputCheckbox

In this example we declare two `HtmlInputCheckBox` controls, one `HtmlInputButton` control, and one `HtmlGeneric` in an `.aspx` file (remember to embed the controls inside an `HtmlForm` control). When the submit button is triggered, the submit subroutine is executed. The submit subroutine may respond in two ways: if the "red" checkbox is checked, the server sends the message "You prefer red!" to the `p` element. If the "blue" checkbox is checked, the server sends the message "You prefer blue!" to the `p` element.

- **Source Code:**

```
<script runat="server">
void submit(Object Sender, EventArgs e)

{
if (red.Checked==True )
```

```

    p1.InnerHtml="You prefer red!";
else
    p1.InnerHtml="You prefer blue!";
    red.checked=false
blue.checked=false
}
</script>

<html>
<body>

<form runat="server">
What color do you prefer?
<br />
<input id="red" type="checkbox" runat="server" /> Red
<br />
<input id="blue" type="checkbox" runat="server" /> Blue
<br />
<input type="button" value="Submit" OnServerClick="submit" runat="server"/>
<p id="p1" runat="server" />
</form>

</body>
</html>

```

- **Output:**

What color do you prefer?

- ☐ Red
- ☐ Blue

➤ **ASP.NET HtmlInputFile Control**

- **Definition and Usage**

The HtmlInputFile control is used to control an <input type="file"> element. In HTML, this element is used to upload a file to the server.

- **Syntax:**

```
<input id="MyFile" type="file" size="40" runat="server">
```

- **Properties**

Property	Description
Accept	List of acceptable MIME types

Attributes	Returns all attribute name and value pairs of the element
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the element
MaxLength	The maximum number of characters allowed in this element
Name	The name of the element
PostedFile	Gets access to the posted file
runat	Specifies that the element is a server control. Must be set to "server"
Size	The width of the element
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Type	The type of the element
Value	The value of the element
Visible	A Boolean value that indicates whether or not the control should be visible

Example 1

In the following example we declare one `HtmlInputFile` control, one `HtmlInputButton` control, and three `HtmlGeneric` controls in an `.aspx` file (remember to embed the controls inside an `HtmlForm` control). When the submit button is triggered, the submit subroutine is executed. The file name and file type are displayed on the page, while the file itself is uploaded to the `c` directory on the server:

```
<script runat="server">
void submit(Object Sender, EventArgs e)
{
    fname.InnerHtml=MyFile.PostedFile.FileName
    clength.InnerHtml=MyFile.PostedFile.ContentLength
    MyFile.PostedFile.SaveAs("c:\uploadfile.txt")
}
</script>

<html>
<body>

<form method="post"
enctype="multipart/form-data" runat="server">
<p>
Select file to upload to server:
<input id="MyFile" type="file" size="40" runat="server">
</p>
<p>
```

```

<input type="submit" value="Upload!" OnServerclick="submit"
runat="server">
</p>
<p>
<div runat="server">
  FileName: <span id="fname" runat="server"/><br />
  ContentLength: <span id="clength" runat="server"/> bytes
</div>
</p>
</form>

</body>
</html>

```

➤ ASP.NET HtmlInputHidden Control

• Definition and Usage

The HtmlInputHidden control is used to control an <input type="hidden"> element. In HTML, this element is used to create a hidden input field.

• Syntax:

```
<input id="hidden1" type="hidden" runat="server" />
```

• Properties

Property	Description
Attributes	Returns all attribute name and value pairs of the element
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the element
Name	The name of the element
runat	Specifies that the element is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Type	The type of the element
Value	The value of the hidden field
Visible	A Boolean value that indicates whether or not the control should be visible

Event	Description
ServerChange	When the content of the element has changed

Examples

HTMLInputHidden

In this example we declare one HtmlInputHidden control, one HtmlInputText control, one HtmlInputButton control, and one HtmlGeneric control in an .aspx file (remember to embed the controls inside an HtmlForm control). When the submit button is triggered, the submit subroutine is executed. The submit subroutine sets the value of the hidden field equal to value of the input field and then displays the value of the hidden field in the p element.

- **Source Code:**

```
<script runat="server">
void submit(Object Sender, EventArgs e)

{
    hidden1.Value=string1.Value
    p1.InnerHtml="Hidden value= " + hidden1.Value
}
</script>

<html>
<body>

<form runat="server">
Enter some text: <input id="string1" type="text" size="25" runat="server" />
<input type="submit" value="Submit" OnServerClick="submit" runat="server" />
<input id="hidden1" type="hidden" runat="server" />
<p id="p1" runat="server" />
</form>

</body>
</html>
```

- **Output:**

Enter some text:

➤ **ASP.NET HtmlInputImage Control**

- **Definition and Usage**

The HtmlInputImage control is used to control an <input type="image"> element. In HTML, this element is used to create an input button using an image, instead of a regular push-style button.

- **Syntax:**

```
<input type="image" src="angry.gif" OnServerClick="button2" runat="server"
width="32" height="32" />
```

- **Properties and Events:**

Property	Description
Align	The alignment of the image
Alt	An alternate text to display for the image
Attributes	Returns all attribute name and value pairs of the element
Border	The width of the borders around the element
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the control
Name	The name of the element
OnServerClick	The name of the function to be executed when the image is clicked
runat	Specifies that the element is a server control. Must be set to "server"
Src	The source of the image
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Type	The type of the element
Value	The value of the element
Visible	A Boolean value that indicates whether or not the control should be visible

Examples

HTMLInputImage

In this example we declare two HtmlInputImage controls and one HtmlGeneric control in an .aspx file (remember to embed the controls inside an HtmlForm control). If the user clicks on the first image, the button1 subroutine is executed. This subroutine sends the message "You clicked the smiley button!" to the p element. If the user clicks on the second image, the button2 subroutine is executed. This subroutine sends the message "You clicked the angry button!" to the p element.

- **Source Code:**

```
<script runat="server">
void button1(Object Sender, ImageClickEventArgs e)
{
```

```

        p1.InnerHtml="You clicked the smiley button!";
    }
void button2(Object sender, ImageClickEventArgs e)

{
    p1.InnerHtml="You clicked the angry button!";
}
</script>

<html>
<body>

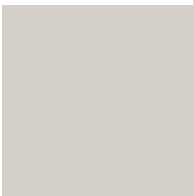
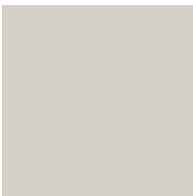
<form runat="server">
<p>Click on one of the images:</p>
<p>
<input type="image" src="smiley.gif"
OnServerClick="button1" runat="server" width="32" height="32" />
</p>
<p>
<input type="image" src="angry.gif"
OnServerClick="button2" runat="server" width="32" height="32" />
</p>
<p id="p1" runat="server" />
</form>

</body>
</html>

```

- **Output:**

Click on one of the images:



➤ ASP.NET HtmlInputRadioButton Control

• Definition and Usage

The HtmlInputRadioButton control is used to control an <input type="radio"> element. In HTML, this element is used to create a radiobutton.

• Syntax:

```
<input id="r1" name="color" type="radio" runat="server">Red</input>
```

• Properties

Property	Description
Attributes	Returns all attribute name and value pairs of the element
Checked	A Boolean value that specifies whether or not the element is to be selected
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the element
Name	The name of the radio button group
runat	Specifies that the element is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Type	The type of the element
Value	The value of the element
Visible	A Boolean value that indicates whether or not the control should be visible

Examples

HtmlInputRadiobutton

In this example we declare three HtmlInputRadioButton controls, one HtmlInputButton control, and one HtmlGeneric control in an .aspx file (remember to embed the controls inside an HtmlForm control). When the submit button is triggered, the submit subroutine is executed. The submit subroutine may respond in three ways: if the radiobutton with id="r1" is selected, the server sends the message "Your favorite color is red" to the p element. If the radiobutton with id="r2" is selected, the server sends the message "Your favorite color is green" to the p element. If the radiobutton with id="r3" is selected, the server sends the message "Your favorite color is blue" to the p element.

- **Source Code:**

```
<script runat="server">
void submit(Object Sender, EventArgs e)

{
if (r1.Checked==True )
    p1.InnerHtml="Your favorite color is red";
else
    if (r2.Checked==True )
        p1.InnerHtml="Your favorite color is green";
    else
        if (r3.Checked=True )
            p1.InnerHtml="Your favorite color is blue";
}
</script>
<html>
<body>
<form runat="server">
<p>Select your favorite color:
<br />
<input id="r1" name="col" type="radio" runat="server">Red</input>
<br />
<input id="r2" name="col" type="radio" runat="server">Green</input>
<br />
<input id="r3" name="col" type="radio" runat="server">Blue</input>
<br />
<input type="button" value="Submit" OnServerClick="submit" runat="server"/>
<p id="p1" runat="server" />
</form>
</body>
</html>
```

- **Output:**

Select your favorite color:

☐ Red

☐ Green

☐ Blue

➤ ASP.NET HtmlInputText Control

- Definition and Usage

The HtmlInputText control is used to control <input type="text"> and <input type="password"> elements. In HTML, these elements are used to create a text field and a password field.

- Syntax:

```
<input id="name" type="text" size="30" runat="server" />
```

- Properties

Property	Description
Attributes	Returns all attribute name and value pairs of the element
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the element
MaxLength	The maximum number of characters allowed in this element
Name	The name of the element
runat	Specifies that the element is a server control. Must be set to "server"
Size	The width of the element
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Type	The type of the element
Value	The value of the element
Visible	A Boolean value that indicates whether or not the control should be visible

Examples

HtmlInputText

In this example we declare one HtmlInputText controls, one HtmlInputButton control, and one HtmlGeneric control in an .aspx file (remember to embed the controls inside an HtmlForm control). When the submit button is triggered, the submit subroutine is executed. The submit subroutine writes a welcome message to the p element.

- Source Code:

```
<script runat="server">  
void submit(Object Sender, EventArgs e)
```

```

{
if( name.value!="" )
    p1.InnerHtml="Welcome " & name.value & "!";
}
</script>

<html>
<body>

<form runat="server">
Enter your name: <input id="name" type="text" size="30" runat="server" />
<br /><br />
<input type="submit" value="Submit" OnServerClick="submit" runat="server" />
<p id="p1" runat="server" />
</form>

</body>
</html>

```

- **Output:**

Enter your name:

➤ **ASP.NET HtmlSelect Control**

- **Definition and Usage**

The HtmlSelect control is used to control a <select> element. In HTML, the <select> element is used to create a drop-down list.

- **Syntax:**

```

<select id="select1" runat="server">
    <option value="smiley.gif">Smiley</option>
    <option value="angry.gif">Angry</option>
    <option value="stickman.gif">Stickman</option>
</select>

```

- **Properties**

Property	Description
Attributes	Returns all attribute name and value pairs of the element
DataMember	A name of a data table to use
DataSource	A data source to use
DataTextField	A field in the data source to be displayed in the drop-down list
DataValueField	A field in the data source that specifies the value of each selectable item in the drop-down list
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the control
InnerHTML	Sets or returns the content between the opening and closing tags of the HTML element. Special characters are not automatically converted to HTML entities
InnerText	Sets or returns all text between the opening and closing tags of the HTML element. Special characters are automatically converted to HTML entities
Items	The list of items in the drop-down list
Multiple	Whether multiple items can be selected at a time
OnServerChange	The name of the function to be executed when the selected item has changed
runat	Specifies that the control is a server control. Must be set to "server"
SelectedIndex	The index of the currently selected item
Size	The number of visible items in the drop-down list
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Value	The value of the currently selected item
Visible	A Boolean value that indicates whether or not the control should be visible

Examples

HTMLSelect

In this example we declare an HtmlImage and an HTMLSelect control in an .aspx file (remember to embed the controls inside an HtmlForm control). Then we modify the src property of the HtmlImage control based on user choices. The value selected in the HtmlSelect control determines which image to display.

- **Source Code:**

```
<script runat="server">
void choose_image(Object Sender, EventArgs e)
{

    image1.Src = select1.Value
}
</script>

<html>
<body>

<form runat="server">
<select id="select1" runat="server">
    <option value="smiley.gif">Smiley</option>
    <option value="angry.gif">Angry</option>
    <option value="stickman.gif">Stickman</option>
</select>
<input type="submit" runat="server" value="Display image"
OnServerClick="choose_image">
<br /><br />

</form>

</body>
</html>
```

- **Output:**



➤ **ASP.NET HtmlTable Control**

- **Definition and Usage**

The HtmlTable control is used to control a <table> element. In HTML, the <table> element is used to create a table.

- **Syntax:**

```
<table id="t1" border="1" runat="server" visible="false"/>
```

- **Properties**

Property	Description
Align	Specifies the alignment of the table
Attributes	Returns all attribute name and value pairs of the element
BGColor	Specifies the background color of the table
Border	Specifies the width of the borders Tip: Set border="0" to display tables with no borders!
BorderColor	Specifies the color of the borders
CellPadding	Specifies the space between the cell walls and contents
CellSpacing	Specifies the space between cells
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
Height	Specifies the height of the table
id	A unique id for the control
InnerHTML	Sets or returns the content between the opening and closing tags of the HTML element. Special characters are not automatically converted to HTML entities
InnerText	Sets or returns all text between the opening and closing tags of the HTML element. Special characters are automatically converted to HTML entities
Rows	Returns an HtmlRowCollection object which represents all rows in the table
runat	Specifies that the control is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Visible	A Boolean value that indicates whether or not the control should be visible
Width	Specifies the width of the table

Examples

HTMLTable

In this example we declare two HtmlSelect controls, one HtmlInputButton control, and one HtmlTable control in an .aspx file (remember to embed the controls inside an HtmlForm control). The user may choose number of cells and rows. When the submit

button is triggered, the submit subroutine is executed. The submit subroutine generates the table based on what the user did input.

- **Source Code:**

```
<script runat="server">
void submit(Object Sender, EventArgs e)

{
int row,numrows,numcells,j,i;
row=0;
numrows=Convert.ToInt32(rows1.Value);
numcells=Convert.ToInt32(cells1.Value);
for (j=1;j<= numrows;j++)

{
    HtmlTableRow r = New HtmlTableRow();
    row=row+1;
    for (i=1 ;i<= numcells;i++)

    {
        HtmlTableCell c = New HtmlTableCell();
        c.Controls.Add(New LiteralControl("row " & j & ", cell " & i));
        r.Cells.Add(c);
    }
    t1.Rows.Add(r);
    t1.Visible=true;
}
}
</script>

<html>
<body>

<form runat="server">
<p>Table rows:
<select id="rows1" runat="server">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
</select>
<br />Table cells:
<select id="cells1" runat="server">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
</select>
<br /><br />
<input type="submit" value="Display Table" runat="server"
OnServerClick="submit">
</p>
```

```
<table id="t1" border="1" runat="server" visible="false"/>
</form>
```

```
</body>
</html>
```

- **Output:**

Table rows:

Table cells:

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2
row 3, cell 1	row 3, cell 2

➤ **ASP.NET HtmlTableCell Control**

- **Definition and Usage**

The HtmlTableCell control is used to control <td> and <th> elements. In HTML, these elements are used to create table cells and header table cells.

- **Properties**

Property	Description
Align	The horizontal alignment of cell content
Attributes	Returns all attribute name and value pairs of the element
BGColor	The background color of the cell
BorderColor	The color of the borders
ColSpan	The number of columns this cell should span
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
Height	The height of the cell
id	A unique id for the control
InnerHTML	Sets or returns the content between the opening and closing

	tags of the HTML element. Special characters are not automatically converted to HTML entities
InnerText	Sets or returns all text between the opening and closing tags of the HTML element. Special characters are automatically converted to HTML entities
Nowrap	A Boolean value that specifies whether or not the text within this control should wrap
RowSpan	The number of rows this cell should span
runat	Specifies that the control is a server control. Must be set to "server"
Style	Gets the CSS properties that are applied to the control
TagName	Returns the element tag name
VAlign	The vertical alignment of cell content
Visible	A Boolean value that indicates whether or not the control should be visible
Width	The width of the cell

Examples

Same as Table Control

➤ ASP.NET HtmlRow Control

• Definition and Usage

The HtmlTableRow control is used to control a <tr> element. In HTML, the <tr> element is used to create a table row.

• Properties

Property	Description
Align	The alignment of the row
Attributes	Returns all attribute name and value pairs of the element
BGColor	The background color of the row
BorderColor	The color of the borders
Cells	Returns the cells in this row
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
Height	The height of the row
id	A unique id for the control
InnerHtml	Sets or returns the content between the opening and closing tags of the HTML element. Special characters are not automatically converted to HTML entities

InnerText	Sets or returns all text between the opening and closing tags of the HTML element. Special characters are automatically converted to HTML entities
runat	Specifies that the control is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
VAlign	The vertical alignment of cells in the row
Visible	A Boolean value that indicates whether or not the control should be visible

Examples

Same as Table Control

➤ ASP.NET HtmlTextArea Control

• Definition and Usage

The HtmlTextArea control is used to control a <textarea> element. In HTML, the <textarea> element is used to create a text area.

• Syntax:

```
<textarea id="textarea1" cols="35" rows="6" runat="server" />
```

• Properties and Events

Property	Description
Attributes	Returns all attribute name and value pairs of the element
Cols	The number of columns of the text area
Disabled	A Boolean value that indicates whether or not the control should be disabled. Default is false
id	A unique id for the control
InnerHtml	Sets or returns the content between the opening and closing tags of the HTML element. Special characters are not automatically converted to HTML entities
InnerText	Sets or returns all text between the opening and closing tags of the HTML element. Special characters are automatically converted to HTML entities
Name	The unique name for the text area
OnServerChange	The name of the function to be executed when the contents of the textarea is changed

Rows	The number of visible rows of the text area
runat	Specifies that the control is a server control. Must be set to "server"
Style	Sets or returns the CSS properties that are applied to the control
TagName	Returns the element tag name
Value	The content of the textarea
Visible	A Boolean value that indicates whether or not the control should be visible

Examples

HTMLTextArea

In this example we declare an HtmlTextArea control, an HtmlInputButton control, and an HtmlGeneric control in an .aspx file (remember to embed the controls inside an HtmlForm control). When the submit button is triggered, the submit subroutine is executed. The submit subroutine sends the message "You wrote: " and the text the user wrote in the text area to the p element.

- **Source Code:**

```
<script runat="server">
void submit(Object Sender, EventArgs e)

{
    p1.InnerHtml = "<b>You wrote:</b> " & textarea1.Value;
}
</script>

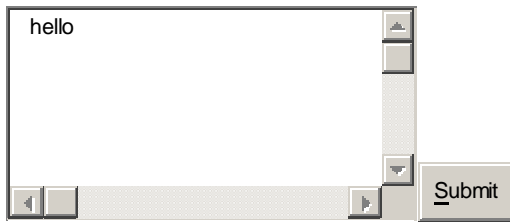
<html>
<body>

<form runat="server">
Enter some text:<br />
<textarea id="textarea1" cols="35" rows="6" runat="server" />
<input type="submit" value="Submit" OnServerClick="submit" runat="server" />
<p id="p1" runat="server" />
</form>

</body>
</html>
```

- **Output:**

Enter some text:



A web form consisting of a text input field and a submit button. The input field is a rectangular box with a thin border, containing the text 'hello'. To the right of the input field is a vertical stack of three small, square buttons with downward-pointing arrows. Below the input field is a horizontal stack of three small, square buttons with leftward-pointing arrows. To the right of these buttons is a rectangular button labeled 'Submit'.

You wrote: hello