**ASP.Net**

ASP.NET (Active Server Pages) is a development framework for building web pages and web sites with HTML, CSS, JavaScript and server scripting.

ASP.NET supports three different development models:  
Web Pages, MVC (Model View Controller), and Web Forms.

# HTML VS ASP.NET

An HTML page has the extension .html. If a browser requests an HTML page from the server, the server sends the page to the browser without any modifications.

An ASP.NET page has the extension .aspx. If a browser requests an ASP.NET page, the server processes any executable code in the page, before the result is sent back to the browser.

## What is Web Forms?

Web Forms is one of the 3 programming models for creating ASP.NET web sites and web applications.

The other two programming models are Web Pages and MVC (Model, View, Controller).

Web Forms are compiled and executed on the server, which generates the HTML that displays the web pages.

Visual Studio contains:

* MVC and Web Forms
* Drag-and-drop web controls and web components
* A web server language (Razor using VB or C#)
* A web server (**IIS Express**)
* A database server (SQL Server Compact)
* A full web development framework (ASP.NET)

## ASP.NET - Server Controls

Server controls are tags that are understood by the server. This are objects placed on a page with the purpose of read and write data.

There are three kinds of server controls:

* HTML Server Controls - Traditional HTML tags
* Web Server Controls - New ASP.NET tags
* Validation Server Controls - For input validation

## ASP.NET - HTML Server Controls

HTML server controls are HTML tags understood by the server.

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. The id attribute is added to identify the server control. The id reference can be used to manipulate the server control at run time.

**Note:** All HTML server controls must be within a <form> tag with the runat="server" attribute. The runat="server" attribute indicates that the form should be processed on the server. It also indicates that the enclosed controls can be accessed by server scripts.

Example:

**<script runat="server">  
Sub Page\_Load  
link1.HRef="http://www.w3schools.com"  
End Sub  
</script>  
<html>  
<body>  
<form runat="server">  
<a id="link1" runat="server">Visit W3Schools!</a>  
</form>  
</body>  
</html>**

## ASP.NET - Web Server Controls

Web server controls are special ASP.NET tags understood by the server.

Like HTML server controls, Web server controls are also created on the server and they require a runat="server" attribute to work. However, Web server controls do not necessarily map to any existing HTML elements and they may represent more complex elements.

The syntax for creating a Web server control is:

<asp:control\_name id="some\_id" runat="server" />

**Basic Web Controls**

|  |  |
| --- | --- |
| **<asp:button>** | **<input type="submit">** |
| **<asp:checkbox>** | **<input type="checkbox">** |
| **<asp:hyperlink>** | **<a href="…"></a>** |
| **<asp:image>** | **<img src="…">** |
| **<asp:imagebutton>** | **<input type="image">** |
| **<asp:linkButton>** | **<a href="…"></a>** |
| **<asp:label>** | **<span>…</span>** |
| **<asp:listbox>** | **<select size="5"></select>** |
| **<asp:panel>** | **<div>…</div>** |
| **<asp:radiobutton>** | **<input type="radio">** |
| **<asp:table>** | **<table>…</table>** |
| **<asp:textbox>** | **<input type="text">** |

In the following example we declare a Button server control in an .aspx file. Then we create an event handler for the Click event which changes the text on the button:

**<script runat="server">  
Sub submit(Source As Object, e As EventArgs)  
button1.Text="You clicked me!"  
End Sub  
</script>  
<html>  
<body>  
<form runat="server">  
<asp:Button id="button1" Text="Click me!"  
runat="server" OnClick="submit"/>  
</form>  
</body>  
</html>**

## ASP.NET - Validation Server Controls

Validation server controls are used to validate user-input. If the user-input does not pass validation, it will display an error message to the user.

Each validation control performs a specific type of validation (like validating against a specific value or a range of values).

By default, page validation is performed when a Button, ImageButton, or LinkButton control is clicked. You can prevent validation when a button control is clicked by setting the CausesValidation property to false.

The syntax for creating a Validation server control is:

<asp:control\_name id="some\_id" runat="server" />

**Various types of validation controls are as follows:**

\_ **RequiredFieldValidator** Checks if the input control has any value.

\_ **RegularExpressionValidator** Checks the value against a regular expression (pattern).

\_ **CompareValidator** Checks if the value is acceptable compared to a given value or compared to the content of another control.

\_ **RangeValidator** Checks if the input control’s value is within a specified range.

\_ **CustomValidator** Allows you to develop custom validation.