Server control

Server controls are tags that are understood by the server.

Most web server controls are derived directly or indirectly from the base class System.Web.Control.WebControl

ASP.NET server controls are the primary controls used in ASP.NET. These controls can be grouped into the following categories:

Types of server controls

- **Validation controls** - For input validation. These are used to validate user input and they work by running client-side script.
- **Data source controls** - These controls provides data binding to different data sources.
- **Data view controls** - These are various lists and tables, which can bind to data from data sources for displaying.
- **HTML Server Controls** - Traditional HTML tags
- **Web Controls**
- **Login** - These controls provide user authentication.
- **Navigation controls** - These controls help in navigation. For example, menus, tree view etc.
- **Rich controls** - These controls implement special features. For example, AdRotator, FileUpload, and Calendar control.

The syntax for using server controls is:

```
<asp:controlType  ID ="ControlID" runat="server" Property1=value1  [Property2=value2] />
```

List of server control

Web Server Control	**Description**
AdRotator	Displays a sequence of images
Button	Displays a push button
Calendar	Displays a calendar
CheckBox	Displays a check box
CheckBoxList	Creates a multi-selection check box group
DropDownList	Creates a drop-down list
HyperLink	Creates a hyperlink
Image	Displays an image
ImageButton	Displays a clickable image
Label	Displays static content which is programmable (lets you apply styles to its content)
LinkButton	Creates a hyperlink button
ListBox	Creates a single- or multi-selection drop-down list
ListItem	Creates an item in a list
Literal	Displays static content which is programmable(does not let you apply styles to its content)
Panel	Provides a container for other controls
PlaceHolder	Reserves space for controls added by code
RadioButton	Creates a radio button

RadioButtonList	Creates a group of radio buttons
BulletedList	Creates a list in bullet format
Repeater	Displays a repeated list of items bound to the control
Table	Creates a table
TextBox	Creates a text box
Xml	Displays an XML file or the results of an XSL transform

Common Property of control

Property	**Description**
AccessKey	Pressing this key with the Alt key moves focus to the control.
Forecolor	Set / get Foreground color.
BackColor	Set / get Background color.
BorderColor	Set / get Border color.
BorderStyle	Set / get Border style.
BorderWidth	Set / get Border width.
CausesValidation	Set / get Indicates if it causes validation.
CssClass	CSS class
Enabled	Indicates whether the control is grayed out.
Visible	It indicates whether a server control is visible.
EnableTheming	Indicates whether theming applies to the control.
EnableViewState	Indicates whether the view state of the control is maintained.
Font	Set / get Font name, size and, style.
ID	Identifier for the control.
SkinID	Gets or sets the skin to apply to the control.
Style	Gets a collection of text attributes that will be rendered as a style attribute on the outer tag of the Web server control.
TabIndex	Gets or sets the tab index of the Web server control.
ToolTip	Gets or sets the text displayed when the mouse pointer hovers over the web server control.
Width	Gets or sets the width of the Web server control.

Height	Set / get Height of control in pixels or %.
dataId	

Common Methods of the Server Controls

The following table provides the methods of the server controls:

Method	Description
DataBind	Binds a data source to the server control and all its child controls.
Dispose	Enables a server control to perform final clean up before it is released from memory.
FindControl()	Searches for child control with specific name in the current control and all contained control.
Focus	Sets input focus to a control.
HasControls	Determines if the server control contains any child controls.
Render	Renders the control to the specified HTML writer.
ToString	Returns a string that represents the current object.

= = = == = = = = = =

TEXTBOX CONTROL

- TextBox control is used to enter data or take input from user.
- By default it display single-line textbox.
- Asp tag is <asp:TextBox> and HTML tags are <input type=”text”/>

properties:

Name	**Description**
AutoPostBack	Gets or sets a value that indicates whether an automatic postback to the server occurs when the TextBox control loses focus.
Columns	Gets or sets the display width of the text box in characters.
MaxLength	Gets or sets the maximum number of characters allowed in the text box.
ReadOnly	Gets or sets a value indicating whether the contents of the TextBox control can be changed.
Rows	Gets or sets the number of rows displayed in a multiline text box.
Text	Gets or sets the text content of the TextBox control.
CauseValidation:	indicating whether validation is performed when the TextBox control is set to validate when a postback occurs.

<table>
<tr><td>TextMode</td><td>Gets or sets the behavior mode. Possible value of this properties:
<table>
<tr><td>Single-line</td><td>Display text in single line</td></tr>
<tr><td>multiline</td><td>Display text in multiple line</td></tr>
<tr><td>password</td><td>Display password character instead of text</td></tr>
</table>
</td></tr>
</table>

Properties: Accesskey, BacColor, BorderColor, BorederStyle, BorderWidth, CausesValidation, CssClass, Enabled, EnableTheming, Font, ForeColor, Height, Id, Style, SkinID, TabIndex, Visible ValidationGroup, Width

Methods

IsEnabled	Gets a value indicating whether the control is enabled.

Events
TextChanged event

Autocomplete:

AutoComplete monitors a text box and creates a list of values entered by the user. When the user returns to the text box at a later time, the list is displayed. Instead of retyping a previously entered value, the user can simply select the value from this list. Use the AutoCompleteType property to control the behavior of the AutoComplete feature for a TextBox control.

= = = == = =

BUTTON CONTROL

- Button control is used to create push button on the web page.
- You can create either a submit button or a command button. By default, a button control is a submit button.
- A submit button simply posts webpages back to server.
- Button control is generally used to post the form or fire an event either client side or server side.
- Its asp.net tag is <asp:Button>

Syntaxt:

```
<asp:Button
            ID="Button1"
            runat="server"
            Text="text to display" />
```

Properties:

Property	**Description**
Text	The text displayed on the button. This is for button and link button controls only.

CausesValidation	Determines whether page validation occurs when a user clicks the button. The default is true.
CommandName	A string value that is passed to the command event when a user clicks the button.
CommandArgument	A string value that is passed to the command event when a user clicks the button.
PostBackUrl	The URL of the page that is requested when the user clicks the button. Indicates the URL on which the Form will be posted.
ValidationGroup	Gets or Sets the name of the validation group that the button belongs to. This is used to validate only a set of Form controls with a Button.
OnClick	Attach a server side method that will fire when button will be clicked.
OnClientClick	Attach a client side (javascript) event that will fire when button will be clicked.

Event
Click

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click


   End Sub
```

*   **

CALENDAR CONTROL

- It is rich control
- The Calendar control is used to display a calendar in the browser.
- It creates rich functionality and good-looking calender
- This control displays a one-month calendar that allows the user to select dates and move to the next and previous months.
- AS.Net tag is <asp:Calender> and HTML tag is <Table>

Properties

Property	**Description**
Caption	The caption of the calendar
CaptionAlign	The alignment of the caption text Possible values : • Not set • Top • Bottom • Left • Right
CellPadding	The space, in pixels, between the cell walls and contents
CellSpacing	The space, in pixels, between cells
DayHeaderStyle	The style for displaying the names of the days

DayNameFormat	The format for displaying the names of the days • Full • Short • FirstTwoLetters • Shortest
DayStyle	The style for displaying days
FirstDayOfWeek	What should be the first day of week
NextMonthText	The text displayed for the next month link
NextPrevFormat	The format of the next and previous month links
NextPrevStyle	The style for displaying next and previous month links
OtherMonthDayStyle	The style for displaying days that are not in the current month
PrevMonthText	The text displayed for the previous month link
runat	Specifies that the control is a server control. Must be set to "server"
SelectedDate	The selected date
SelectedDates	The selected dates
SelectedDayStyle	The style for selected days
SelectionMode	How a user is allowed to select dates
SelectMonthText	The text displayed for the month selection link
SelectorStyle	The style for the month and weeks selection links
SelectWeekText	The text displayed for the week selection link
ShowDayHeader	A Boolean value that specifies whether the days of the week header should be shown
ShowGridLines	A Boolean value that specifies whether the grid lines between days should be shown
ShowNextPrevMonth	A Boolean value that specifies whether the next and previous month links should be shown
ShowTitle	A Boolean value that specifies whether the title of the calendar should be shown
TitleFormat	The format for the title of the calendar
TitleStyle	The style of the title of the calendar
TodayDayStyle	The style for today's date
TodaysDate	Today's date
UseAccessibleHeader	Specifying whether to use the <th> element for the day headers instead of the <td> element
VisibleDate	The date that specifies the month that is currently visible in the calendar
WeekendDayStyle	The style for weekends
OnDayRender	The name of the function to be executed when when each day cell is created
OnSelectionChanged	The name of the function to be executed when the user selects a day, week, or month
OnVisibleMonthChanged	The name of the function to be executed when the user navigates to a different month

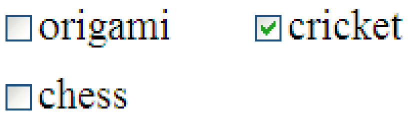
EVENT

SelectionChanged

* * **********************   ****************

CHECKBOX CONTROL

- CheckBox control allows user for mulitple selection by tick marking checkbox.
- It allows user to select one or mor choice from various values.
- When it is selected check mark ☑ will apear.
- Asp:net tag is <asp:CheckBox> and When it is rendered on the page, it is implemented through tag <input type="CheckBox"/>



□ origami ☑ cricket

□ chess

Other property: AccessKey, Attributes, BackColor, BorderColor, BorderStyle, BorderWidth, CssClass, Enabled, Font, EnableTheming, ForeColor, Height, IsEnabled, SkinID, Style, TabIndex, ToolTip, Width

Properties:

AutoPostBack	Form is automatically posted back when CheckBox is checked or Unchecked.
Checked	true/false. Specifies whether the check box is checked or not. If true, Check box is checked by default.
ValidationGroup	Used to put a checkbox under a particular validation group. It is used when you have many set of form controls and by clicking a paricular button you want to validate a particular set of controls only.
TextAlign	It is used to set alignment of text with respect to item. Possible value: Left — Text appear left of radiobutton Rifght — Text appear right of radiobutton
Text	The text next to the check box

Event

CheckedChanged	Fires when CheckBox is checked or Unchecked. This works only if AutoPostBack property is set to true.

Example:

CheckBox.aspx

```
<asp:CheckBox ID="CheckBox1" runat="server" Text="CheckBox" AutoPostBack="True"
oncheckedchanged="CheckBox1_CheckedChanged" />
<br />
<asp:Label ID="Label2" runat="server" Text="Label2"> </asp:Label>
```

CheckBox.aspx.vb

```
'On CheckBox1_CheckedChanged Event
        If CheckBox1.Checked = True Then
                Label2.Text = "Checkbox Clicked "
        Else
                Label2.Text = ""
        End If
```

= = = = == = = = = = ==

RADIOBUTTON CONTROL

- RadioButton control is used to give single select option to the user from multiple items.
- It creates mutually-exclusive option. When we click one of RadioButton in group, any other raddiobuton previously selected will automatically turn off.
- Asp.net tag is <asp:RadioButton> and html tag is <input type=”radio”>

Properties:

<table>
<tr><td>AutoPostBack</td><td>Form is automatically posted back when Radio button selection is changed.</td></tr>
<tr><td>Checked</td><td>true/false. If true, Radio button is selected by default.</td></tr>
<tr><td>CheckedChanged</td><td>Fires when Radio button selection changes. This works only if AutoPostBack property is set to true.</td></tr>
<tr><td>ValidationGroup</td><td>Used to put a radio button under a particular validation group. It is used when you have many set of form controls and by clicking a paricular button you want to validate a particular set of controls only.</td></tr>
<tr><td>GroupName</td><td>It is used to create a group of RadioButton, so only one of them can be selected at a time.</td></tr>
<tr><td>TextAlign</td><td>It is used to set alignment of text with respect to item. Possible value:
<table>
<tr><td>Left</td><td>Text appear left of check box</td></tr>
<tr><td>Rifght</td><td>Text appear right of check box</td></tr>
</table>
</td></tr>
</table>

Event:

CheckedChanged	This event occure whent selection state is changed of radiobutton. This works only if AutoPostBack property is set to true.

Example,

Following change font color of textbox according selection of radio button.

Default.aspx

```
<asp:Button ID="Button1" runat="server" Text="Button" />

<asp:RadioButton ID="RadioButton1" runat="server" AutoPostBack="True"  GroupName="g1"
Text="blue" />

<asp:RadioButton ID="RadioButton2" runat="server" AutoPostBack="True"  GroupName="g1"
Text="green" />

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
```

Default.aspx.vb

```
Protected Sub RadioButton1_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles RadioButton1.CheckedChanged

        'textbox1.text = RadioButton1.Text

       TextBox1.ForeColor = Drawing.Color.Blue

End Sub


Protected Sub RadioButton2_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles RadioButton2.CheckedChanged

        TextBox1.ForeColor = Drawing.Color.Green

        'TextBox1.Text = RadioButton2.Text

End Sub
```

* ***

LIST BOX

- ListBox control is used to give a single or multiple select options to the user (based on the property set) from multiple listed items.

- When it is rendered on the page, it is implemented through <select/> HTML tag. It is also called as Combo box.
- **Properties**

Rows	No. of rows (items) can be set to display in the List.
SelectionMode	Single or Multiple. **Single**: Allow user to select single item from list. **Multiple:** Allows user to select multiple items from the list by holding Ctrl or Shift key.
SelectedValue	Get the value of the Selected item from the list box.
SelectedIndex	Gets or Sets the index of the selected item in the list box.
SelectedItem	Gets the selected item from the list.
Items	collection of items in the list box.
DataTextField	Name of the data source field to supply the text of the items.
DataValueField	Name of the data source field to supply the value of the items.
DataSourceID	ID of the datasource component to provide data.
DataSource	The datasource that populates the items in the listbox box.
AutoPostBack	true or false. If true, the form is automatically posted back to the server when user changes the dropdown list selection.

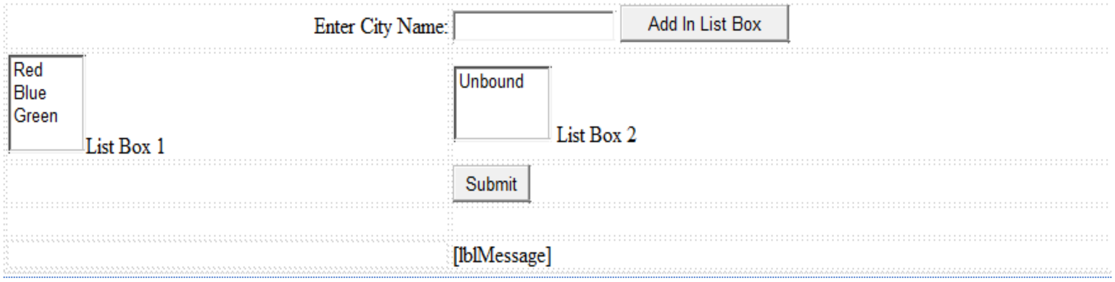
Method

Add, remove, RemoveAt, Clear

Event

SelectedIndexChanged	Method name that fires when user changes the selection of the dropdown box. (Fires only when AutoPostBack=true.)

example



Default.aspx

```
<table class="style1">

        <tr>

            <td align="right">

                Enter City Name:</td>

            <td>

                <asp:TextBox ID="txtCity" runat="server"></asp:TextBox>

                                &nbsp;<asp:Button ID="btnAdd" runat="server" Text="Add In List Box"
                                style="height: 26px" />

            </td>

        </tr>

        <tr>

            <td>

                <asp:ListBox ID="lstColor" runat="server">

                    <asp:ListItem>Red</asp:ListItem>

                    <asp:ListItem>Blue</asp:ListItem>

                    <asp:ListItem>Green</asp:ListItem>

                </asp:ListBox> List Box 1

        </td>

            <td>

                                <asp:ListBox ID="lstCity" runat="server" SelectionMode="Multiple"
                                Rows="3"></asp:ListBox>&nbsp;List Box 2
            </td>
        </tr>
        <tr>
            <td></td>
            <td>
                <asp:Button ID="btnsubmit" runat="server" Text="Submit" />
            </td>
        </tr>
        <tr>
            <td class="style2">
                </td>
            <td class="style2">
                <asp:Label ID="lblMessage" runat="server"></asp:Label>
            </td>
        </tr>
    </table>
```

Default.aspx.vb

‘On btnAdd_Click

```
        'Add Items thru Code
        lstCity.Items.Add(txtCity.Text)
        txtCity.Text = ""
```

‘On btnsubmit_Click

```
        lblMessage.Text = "You selected from the ListBox:<br>"
        For Each li As ListItem In lstCity.Items
        If li.Selected = True Then

            lblMessage.Text = lblMessage.Text + li.Text & "<br>"

        End If

    Next
```

= = =

DROPDOWNLIST CONTROL

- DropDownList control allows user to select option form multiple listed items.
- ASP.Net tag is <asp:DropDownList>
- When it is rendered on the page, it is implemented through <select/> HTML tag.
- It is also called as Combo box.
- It is combination of listbox and textbox.

Properties

SelectedValue	Get the value of the Selected item from the dropdown box.
SelectedIndex	Gets the index of the selected item in the dropdown box.
SelectedItem	Gets the selected item from the list.
Items	collection of items from in dropdown box.
DataTextField	Name of the data source field to supply the text of the items.
DataValueField	Name of the data source field to supply the value of the items.
DataSourceID	ID of the datasource component to provide data.
DataSource	The datasource that populates the items in the dropdown box.
AutoPostBack	true or false. If true, the form is automatically posted back to the server when user changes the dropdown list selection.

- Common proprties: BackColor, ForeColor, BorderColor, BorderStyle, BorderWidth, Height, width, visible etc

methods

add , Remove, RemoveAt(), Clear, Count

event

SelectedIndexChanged	This event occurs when user changes the selection of the dropdown box. (Fires only when AutoPostBack=true.)

To add data into drop down list using coding. (default.aspx.vb)

```
‘ On Page_Load Event

        ddlCountry.Items.Add("India")
        ddlCountry.Items.Add("USA")
        ddlCountry.Items.Add("UAE")
```

************   **

FILEUPLOAD CONTROL

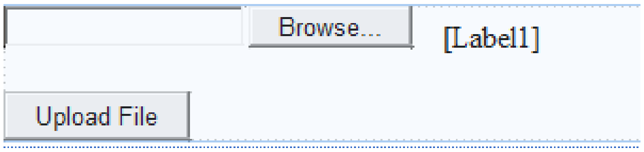
- FileUpload control allows users to upload file to the server.
- Asp tag is <asp:FileUpload /> and related html tag is <input type="file" >
- This control displays the textbox control and a browse button that enables user to select a file.

Properties:

FileBytes	Gets an array of the bytes in a file that is specified by using a FileUploadcontrol.
FileContent	Gets a Stream object that points to a file to upload using the FileUploadcontrol.
FileName	Gets the name of a file on a client to upload using the FileUpload control.
HasFile	Gets a value indicating whether the FileUpload control contains a file.
PostedFile	Gets the underlying HttpPostedFile object for a file that is uploaded by using the FileUpload control.

other proprties: BackColor, ForeColor, BorderColor, BorderStyle, BorderWidth, Height, width, visible etc

Example,



Default.aspx

```
<asp:FileUpload ID="FileUpload1" runat="server" />
    <br />
<asp:Button ID="Button1" runat="server" Text="upload" />
<asp:Label ID="Label1" runat="server"></asp:Label>
```

Default.aspx.vb

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim dir As String = "\jmp\"
        Dim app_path As String = Request.PhysicalApplicationPath
       If FileUpload1.HasFile = True Then
            Dim save_path As String = app_path + dir + Server.HtmlEncode(FileUpload1.FileName)
            FileUpload1.SaveAs(save_path)
            Label1.Text = "upload successfully"
        Else
            Label1.Text = "not successfully"
        End If

    End Sub
```

=   =   =======   =   =   =

❖ Hyperlink

- Hyperlink control is used to jump to another location or to execute the script code.
- The **HyperLink** control creates links on a web page that allow users to move from page to page in your application.
- The **HyperLink** control can display clickable text or an image.
- Asp.net tag is <asp:HyperLink> and HTML tag is <a/> tag.
- properties

Property	Description

<table>
<tr><td>Text</td><td>Specifies the text displayed as a hyperlink in the user's browser. You can include HTML formatting in the property.</td></tr>
<tr><td>ImageUrl</td><td>Creates a graphical link when you set the property to the URL of a .gif, .jpg, or other web graphic file. If you set both the ImageUrl and Text properties,
the ImageUrl property takes precedence.</td></tr>
<tr><td>NavigateUrl</td><td>Specifies the URL of the page to which you want to link.(location to jump to).
When user click, page at specified url is display.</td></tr>
<tr><td>Target</td><td>Indicates the ID of a target window or frame in which to display the linked page.
You can either specify a window by name or use predefined target values:
<table>
<tr><td>_top</td><td></td></tr>
<tr><td>_parent</td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
</table>

Other property: AccessKey, Attributes, BackColor, BorderColor, BorderStyle, BorderWidth, CssClass, Enabled, Font, EnableTheming, ForeColor, Height, IsEnabled, SkinID, Style, TabIndex, ToolTip, Width

```
<asp:HyperLink ID="HyperLink1" runat="Server"
          Text="Go to page2
          NavigateUrl="default2.aspx" />

 <asp:HyperLink ID="HyperLink2" runat="Server"
         NavigateUrl=”default3.aspx"
         ImageUrl="~/images/demobutton.gif"
         ToolTip="Go to ImageButton control tutorial" />
```

Advantages of Using the HyperLink Control

(1) You can set link properties in server code.
(2) You can use data binding to specify the target URL for the link.

= = = == = = == =

Image control

- It display image.

- ImageButton control is generally used to post the form or fire an event either client side or server side.
- It support format like jpg, gif, png, bmp,

<table>
<tr><td>AlternateText</td><td>Text to be displayed if image not loaded properly or not available.</td></tr>
<tr><td>DescriptionUrl</td><td>Gets or sets the URL or location where detailed description for the image is available.</td></tr>
<tr><td>ImageAlign</td><td>Gets or sets the alignment of the Image control.
<table>
<tr><th>Alignment</th><th>Description</th></tr>
<tr><td>NotSet</td><td>The alignment is not set.</td></tr>
<tr><td>Left</td><td>The image is aligned on the left edge of the Web page with text wrapping on the right.</td></tr>
<tr><td>Right</td><td>The image is aligned on the right edge of the Web page with text wrapping on the left.</td></tr>
<tr><td>Baseline</td><td>The lower edge of the image is aligned with the lower edge of the first line of text.</td></tr>
<tr><td>Top</td><td>The upper edge of the image is aligned with the upper edge of the highest element on the same line.</td></tr>
<tr><td>Middle</td><td>The middle of the image is aligned with the lower edge of the first line of text.</td></tr>
<tr><td>Bottom</td><td>The lower edge of the image is aligned with the lower edge of the first line of text.</td></tr>
<tr><td>AbsBottom</td><td>The lower edge of the image is aligned with the lower edge of the largest element on the same line.</td></tr>
<tr><td>AbsMiddle</td><td>The middle of the image is aligned with the middle of the largest element on the same line.</td></tr>
<tr><td>TextTop</td><td>The upper edge of the image is aligned with the upper edge of the highest text on the same line.</td></tr>
</table></td></tr>
<tr><td>ImageUrl</td><td>Gets or sets the URL or path to an image</td></tr>
</table>

= ==

ImageMap

- ImageMap link different part of images to different URL.
- ImageMap control is used to create an image that contains clickable hotspot region.
- Example: It may use to diplay map of country. When a user clicks on specific state of the map, the control navigates to url that provides additional dta about selected state.

<table>
<tr><td>ImageUrl</td><td>Gets or sets the URL or path to an image</td></tr>
<tr><td>AlternetText</td><td>Text to be displayed if browser cannot display the image or not loaded properly</td></tr>
<tr><td>ImageAlign</td><td>Used to align the Text beside image.</td></tr>
<tr><td>HotSpotMode</td><td><b>Hotspotmode:</b>
<table>
<tr><td>Notset</td><td>The HotSpot object uses the behavior set by the ImageMap control's HotSpotMode property. If the ImageMap control does not define the behavior, the HotSpot objects all navigate to a URL.</td></tr>
<tr><td>Navigate</td><td>user is navigated to a different URL</td></tr>
<tr><td>PostBack</td><td>the page is posted back to the server.</td></tr>
<tr><td>Inactive</td><td>The HotSpot object does not have any behavior</td></tr>
</table></td></tr>
<tr><td>HotSpots</td><td>There are three kinds of hot spot regions defined in ImageMap control.
1. RectangleHotSpot: you define coordinates of the rectangle. (property: Top, Right, Bottom, Left)
2. CircleHotSpot : you define the x and y coordinates of the circle's center and the circle's radius.(property: Radius, x,y)
3. PolygonHotSpot: define coordinate of polygonHotspot. (property: Coordinates)</td></tr>
<tr><td>OnClick</td><td>Attach a server side event that fires after clicking on image when HostSpotMode is PostBack.</td></tr>
<tr><td>PostBackValue</td><td>You can access it in the server side click event through ImageMapEventArgs.</td></tr>
</table>

- Other properties like BackColor, ForeColor, BorderColor, BorderStyle, BorderWidth, Height etc. are implemented through style properties of <img>.

You can define as many or as few hot spots for the image as you require. You do not need to define hot spots to cover the graphic entirely.

To define hot spots for an ImageMap control

1. In Design view, right-click the ImageMap control and click Properties on the shortcut menu.
2. Click the elipsis button …beside the HotSpots property to open the HotSpot Collection Editor dialog box.
3. Click the arrow on the right side of the Add button and click the type of HotSpot you want to add: CircleHotSpot, RectangleHotSpot, or PolygonHotSpot.

4. In the Properties area, set the properties for your HotSpot.

Example:

```
<asp:Image ID="Image1" runat="server" AlternateText="error"

        DescriptionUrl="~/HTMLPage.htm" Height="62px" />

    <asp:ImageMap ID="ImageMap1" runat="server" Height="158px"

        ImageUrl="~/image/923545_600479380045113_918040335_n.jpg" Width="250px"

        HotSpotMode="Navigate">

        <asp:CircleHotSpot Radius="25" X="25" Y="25" NavigateUrl="~/HTMLPage.htm" />



    </asp:ImageMap>
```

= = = = = = = = =

Image BUTTON CONTROL

- Imagebutton is like button control. Instead of displaying text, it displays images.

Syntaxt:

```
<asp:ImageButton
            ID="ImageButton1"
            runat="server"
/>
```

Properties:

Property	Description
Text	The text displayed on the button. This is for button and link button controls only.
ImageUrl	URL or path to an image to be displayed on the button.
AlternateText	Text to be displayed if browser cannot display the image or not loaded properly
DescriptionUrl	Gets or sets the URL or location where detailed description for the image is available.
ImageAlign	Gets or sets the alignment of the Image control
CommandName	Gets or sets the command name associated with the ImageButton control.

CommandArgument	A string value that is passed to the command event when a user clicks the button.
PostBackUrl	The URL of the page that is requested when the user clicks the button. Indicates the URL on which the Form will be posted back.
OnClientClick	Attach a client side (javascript) event that will fire when button will be clicked.

Other property
AccessKey, Attributes, BackColor, BorderColor, BorderStyle, BorderWidth, CssClass, Enabled, Font, EnableTheming, ForeColor, Height, SkinID, Style, TabIndex, ToolTip, Width
= = = = == = =

LINK BUTTON

- It is used to create hyper-link style button.
- It has same functionality as button control and same appearance as Hyper link control.
- It implements an anchor <a/> tag that uses only ASP.NET postback mechanism to post the data on the server.
- You can't specify the target URL.

PostBackUrl	Indicates the URL on which the Form will be posted back.
ValidationGroup	Gets or Sets the name of the validation group that the button belongs to. This is used to validate only a set of Form controls with a Button.
OnClick	Attach a server side method that will fire when button will be clicked
OnClientClick	Attach a client side (javascript) method that will fire when button will be clicked.
CommandNanme	
CommandArgument	

Common property: AccessKey, Attributes, BackColor, BorderColor, BorderStyle, BorderWidth, CssClass, Enabled, Font, EnableTheming, ForeColor, Height, SkinID, Style, TabIndex, ToolTip, Width.

Example

```
// On OnClick event
```

```
                    <asp:LinkButton ID="LinkButton1" runat="Server" Text="LinkButton: Activate
Server      Side Event" OnClick="ActivateServerSideEvent" />

// On OnClientClick event and CommandName
                    <asp:LinkButton ID="LinkButton2" runat="Server" Text="LinkButton: Activate Client
Side Method" OnClientClick="GiveAlertToUser()" UseSubmitBehavior="False"
                  CommandName="ClientSideButton" />

// On OnClick & ValidationGroup
                    <asp:LinkButton ID="LinkButton1" runat="Server" OnClick="FireServerSideEvent"
                  Text="LinkButton: Fire Server Side Event" ValidationGroup="demo" />
```

= = == = = = = = =

Label:

- is used to write some text over the page.
- it makes formatting techniques when it render.
- Asp tag is <asp:label>
- It will render text within <span> tag.

Text	
AssoicatedControlID	**Get/set ID of the control associated with the label**

Literal control

- is used to write text with server side formatting options.
- Literal control to reserve a location on the Web page to display text.
- It cannot render any additional html tag.
- Default mode for literal is “Transform”.

Property:

<table>
<tr><td><b>Text</b></td><td></td></tr>
<tr><td><b>Visible</b></td><td></td></tr>
<tr><td><b>Name</b></td><td></td></tr>
<tr><td><b>Mode</b></td><td>
<table>
<tr><td><b>Mode</b></td><td><b>Description</b></td></tr>
<tr><td>PassThrough</td><td>The contents of the control are not modified.</td></tr>
<tr><td>Encode</td><td>The contents of the control are converted to an HTML-encoded string.</td></tr>
<tr><td>Transform</td><td>Unsupported markup-language elements are</td></tr>
</table>
</td></tr>
</table>

		removed from the contents of the control. If the Literal control is rendered on a browser that supports HTML or XHTML, the control's contents are not modified.	

Difference between label and literal control:

Label	**Literal**
is used to write some text over the page	is used to write some text over the page
display the text wrapped around in a span tag	Simply display text without wrapping it with anything.
	reserve a location on the Web page to display text
	If you do not need styling then its better to use literal
has much more properties than the literal control	
Label can be access via clientscript	

For example:-

Suppose you have a label and a literal control in your aspx page :-

```
<asp:Label ID="label1" runat="server"></asp:Label>

<br />

<asp:Literal ID="literal1" runat="server"></asp:Literal>
```

Now, bind your controls with some text:-

```
label1.Text = "label text";

literal1.Text = "literal text";
```

When you execute the code, you will see this:-

```
<span id="label1">label text</span>
```

<br />

literal text

In the above output, we see that the label text is wrapped around a span tag and the literal text is simply putting a text in it.

= = = = = = == =

BULLETED LIST

- The **BulletedList** control is used to create a list of items formatted with bullets. i.e. The BulletedList control creates a list in bullet format.
- Corresponding asp tag is <asp:BulletedList>
- When it is rendered on the page, it is implemented through <ol> (for numbered list) or <ul> (for unnumbered bullet) HTML tag.
- Each item in a BulletedList control is defined by a <asp: ListItem> element. Related HTML tag is <li>

DisplayMode	Determines how to display the list. • Text - Default. Standard Text • HyperLink – Hyperlink • LinkButton - Hyperlink Button
FirstBulletNumber	Sets a starting number for Bulleted list when BulletStyle is set to Numbering.
Items	Gets the colleciton of the items in the list control.
style	Specifies the style of the list item marker. Possible values: • NotSet - Not set o Circle - Empty circle • Disc - Filled circle ▪ Square - Filled square • Numbered (1,2,3,..) • LowerAlpha - (a,b,c..) • UpperAlpha - Uppercase letter (A,B,C,..) • LowerRoman - Lowercase Roman numeral (i,ii,iii,..) • UpperRoman - Uppercase Roman numeral (I,II,III,…) ➢ CustomImage - Custom image (defined by the "BulletImageUrl" property)
DataTextField	Name of the field to set as items text. Used when DisplayMode is Hyperlink or LinkButton.
DataValueField	Name of the field to set as items value. Used when DisplayMode is Hyperlink or LinkButton.
BulletImageUrl	Used to set the Bullet Image when BulletStyle is CustomImage.

<table>
<tr><td>Target</td><td>get or set where to open the target URL. This property is only used when DisplayMode property is "HyperLink"

Possible Values:
<table>
<tr><td>_blank</td><td>the target URL will open in a new window</td></tr>
<tr><td>_self</td><td>the target URL will open in the same frame as it was clicked</td></tr>
<tr><td>_search</td><td>the target URL will open in the search pane</td></tr>
<tr><td>_top</td><td>the target URL will open in the full body of the window</td></tr>
<tr><td>_parent</td><td>the target URL will open in the parent frameset</td></tr>
<tr><td>framename</td><td>specify a target frame name</td></tr>
</table>
</td></tr>
</table>

Example:

To specify the individual list items that you want to appear in a **BulletedList** control, place a **ListItem** object for each entry between the opening and closing tags of the **BulletedList** control.

```
<html>
<head></head>
<body>
   <h3><font face="Verdana">Bulleted List</font></h3>
   <form runat=server>
      <asp:BulletedList ID=BulletedList1 BulletStyle="Circle" runat=server>
       <asp:ListItem>Item #1</asp:ListItem>
       <asp:ListItem>Item #2</asp:ListItem>
       <asp:ListItem>Item #3</asp:ListItem>
       <asp:ListItem>Item #4</asp:ListItem>
      </asp:BulletedList>
   </form>
</body>
</html>
```

* **

PANEL CONTROL

- The Panel control is used as a container for other controls, for example, a set of radio buttons, checkboxes, etc.
- It is especially useful when you want to generate controls programmatically, hide/show a group of controls, or localize a group of controls.
- If the Panel's Enabled property is set to False then the controls which the Panel contains are also disabled.
- Panels can have scrollbars.
- Asp.net tag is <asp:Panel> . It is rendered as <>

Properties

Property	**Description**
BackImageUrl	Specifies a URL to an image file to display as a background for this control

DefaultButton	Specifies the ID of the default button in the Panel. This property is used to specify which button gets clicked when the Panel control has focus and the "ENTER" key is pressed.
Direction	Specifies the content display direction of the Panel Possible values: • NotSet - Default. The content direction is not set • LeftToRight - The content direction is left to right • RightToLeft - The content direction is right to left
GroupingText	Specifies the caption for the group of controls in the Panel. i.e. caption of panel control.
HorizontalAlign	Specifies the horizontal alignment of the content
runat	Specifies that the control is a server control. Must be set to "server"
ScrollBars	Specifies the position and visibility of scroll bars in the Panel
Wrap	Specifies whether the text should wrap or not

= = = == == = = = == = = =

PlaceHolder

- It is useful to create dynamic user interface.
- It hold other controls, or content like HTML, JavaScript, plain text etc
- It has not user interface
- The Placeholder does not render any tags for itself, so it is great for grouping content without the overhead of outer HTML tags.
- PlaceHolder control is invisible for website's visitors.

Syntax

```
<asp:PlaceHolder ID="PlaceHolder1" runat="server"></asp:PlaceHolder>
```

= = = =

diff. panel and placeholder

- Panel will render Div tags in the brower where place holder will not render any tags.
- Panel have the styling capabilites, so you can set the cssclass or style properties such as background-color, forecolor etc...But Placeholder doesn't have any style attributes associated. You can not set cssclass or forecolor etc

= = = == = = = =

RADIOBUTTONLIST CONTROL

- The RadioButtonList control is used to create a group of radio buttons
- RadioButtonList control is a single control that groups a collection of radiobuttons
- Asp.net tag is <asp:RafioButtonList> and items are implemented using <asp:ListItem>

Properties:

SelectedValue	Get the value first selected item.
SelectedIndex	Gets or Sets the index of the first selected item.
SelectedItem	Gets the first selected item
TextAlign	Gets or Sets the alignment of the radiobutton text. Must set RepeatLayout=Table
DataTextField	Name of the data source field to supply the text of the items.
DataValueField	Name of the data source field to supply the value of the items.
DataSourceID	ID of the datasource component to provide data.
DataSource	The datasource that populates the items in the radiobuttonlist. (
AutoPostBack	true/false. If true, the form is automatically posted back to the server when user click any of the radiobutton.
Items	Gets the colleciton of the items from the list.
RepeatLayout	table/flow. Gets or Set the layout of the radiobuttons when rendered to the page.
RepeatColumns	Get or Sets the no. of columns to display when the control is rendered.
RepeatDirection	Horizontal/Vertical. Gets or Sets the value to indicate whether the control will be rendered horizontally or vertically.
CellPadding	The amount of pixels between the border and the contents of the table cell. Must set RepeatLayout=Table.
CellSpacing	The amount of pixels between table cells. Must set RepeatLayout=Table

event

SelectedIndexChanged	Method name that fires when user click any of the radiobutton in the list. (Fires only when AutoPostBack=true.)

Example,

Following change font color of textbox according selection of radio button.

Default.aspx

```
<asp:Button ID="Button1" runat="server" Text="Button" />

<asp:RadioButton ID="RadioButton1" runat="server" AutoPostBack="True"  GroupName="g1"
Text="blue" />

<asp:RadioButton ID="RadioButton2" runat="server" AutoPostBack="True"  GroupName="g1"
Text="green" />

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
```

Default.aspx.vb

```
Protected Sub RadioButton1_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles RadioButton1.CheckedChanged

    'textbox1.text = RadioButton1.Text

   TextBox1.ForeColor = Drawing.Color.Blue

End Sub


Protected Sub RadioButton2_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles RadioButton2.CheckedChanged

    TextBox1.ForeColor = Drawing.Color.Green

    'TextBox1.Text = RadioButton2.Text

End Sub
```

= = = = = = =

CHECKBOXLIST CONTROL

- It Creates a multi-selection check box group.
- CheckBoxList control is a single control that groups a collection of checkable list items, all are rendered through an individual <asp:CheckBoxList>
- Its properties like BackColor, ForeColor, BorderColor, BorderStyle, BorderWidth, Height etc. >

Properties:

SelectedValue	Gets the value of first selected item.
SelectedIndex	Gets or Sets the index of the first selected item.
SelectedItem	Gets the first selected item
Items	Gets the collection of the items from the list.

RepeatLayout	table/flow. Gets or Sets the layout of the chekboxes when rendered to the page.
RepeatColumns	Gets or Sets the no. of columns to display when the control is rendered.
RepeatDirection	Horizontal/Vertical. Gets or Sets the value to indicate whether the control will be rendered horizontally or vertically.
CellPadding	The amount of pixels between the border and the contents of the table cell
CellSpacing	The amount of pixels between table cells

TextAlign	Gets or Sets the alignment of the checkbox text. Indicate On which side of the check box the text should appear
DataTextField	Name of the data source field to supply the text of the items.
DataValueField	Name of the data source field to supply the value of the items
DataSourceID	ID of the datasource component to provide data
DataSource	The datasource that populates the items in the checkboxlist box.
AutoPostBack	true/false. If true, the form is automatically posted back to the server when user click any of the checkbox.

event:

OnSelectedIndexChanged	Method name that fires when user click any of the checkbox in the list. (Fires only when AutoPostBack=true.)

Example:

CheckBoxList.aspx

```
<asp:CheckBoxList ID="cblLangKnown" runat="server" AppendDataBoundItems="True"
RepeatLayout="Flow" RepeatDirection="Horizontal"></asp:CheckBoxList>
```

CheckBoxList.aspx.vb

```
'On Page_Load Event
        cblLangKnown.Items.Add("English")
        cblLangKnown.Items.Add("Gujarati")
        cblLangKnown.Items.Add("Hindi")
        cblLangKnown.Items.Add("Marathi")

=       =       =       =
For Each item In CheckBoxList1.Items
         If (item.Selected) Then
```

```
          MsgBox(item.Value)
       End If

    Next
```

= = = = = = =

Difference between Hyperlink and linkbutton

- what is differences between link button and hyper link button?

Link button	**Hyper link button**
The LinkButton works exactly as a normal Button but it looks like an HyperLink, so it will PostBack your page to the server	It will not PostBack your page to the server. It will post a simple request to the server for the URL you set as NavigateURL
	HyperLink doesn't have the OnClick event
It is ASP.NET server control	Hyperlink control is an HTML control

= = = = = = =

❖ Access key

- Use the accesskey property to specify the keyboard shortcut for the web server control
- The access key allows a user to shift the focus of an input element, using the keyboard.
- An access key (also known as a hot key) allows users to press the ALT key plus another key to jump to a specific control on the page without using the mouse.
- Only a single character string is allowed for the accesskey property.

AccessKey: get focus using <alt> + character combination

Syntax

<asp:*webcontrol* id="*id*" AccessKey=*"accessKey"* runat="server" />

Example:

```
<asp:button id="button1" runat="server" accesskey="y" text="button" />

<asp:textbox id="textbox1" runat="server" />
```

when user press <alt> + y focus will move to button1 and click even will execute.

work with label

```
<asp:Label ID="Label1" runat="server"
  AccessKey="L"
  AssociatedControlID="TextBox1"
  Text="<u>L</u>ast name: ">
</asp:Label>&nbsp;
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
```

= = == =

AdRotator Server Control

- It is used to randomly select banner from the list specified in XML file.
- It is used to show a series of advertisements to the end users.
- If you are using an XML source for the ad information, first create an XML advertisement file.
- We can bind AdRotator server control with using SqlDataSource also.

Properties

Property	**Description**
AdvertisementFile	The path to the XML file that contains advertisement information
AlternateTextField	It gets/set a custom data field to use in instead of the Alternate text for an advertisement
ImageUrlField	A data field to use instead of the ImageURL attribute for an advertisement
KeywordFilter	A filter to limit ads after categories. Display only those which match filter.
NavigateUrlField	A data field to use instead of the NavigateUrl attribute for an advertisement
runat	Specifies that the control is a server control. Must be set to "server"

Target	Where to open the URL. Possible value: _self, _top,_blank, _parent
Height	Height of the image to be displayed
Width	Width of the image to be displayed

Elements of XML file

Element	Description
Advertisements	Specify name of the advertisement file
Ad	Delineates separate ad
ImageUrl	An absolute or relative URL to the ad image file.
NavigateUrl	The link that will be followed when the user clicks the ad. If **NavigateUrl** is not set, the image is not clickable.
AlternateText	The text that will be displayed instead of the picture if it cannot be displayed
Keyword	Keyword identifying a group of advertisements or A category for the ad. This is used for filtering.
Impressions	The number indicating how often an advertisement will appear/display. The larger the number, the more often the ad is displayed.

Example –1:

AdRotator1.aspx

```
<html>

<body>

   <h3><font face="Verdana">AdRotator Example</font></h3>

   <form runat=server>

         <asp:AdRotator id="ar1" AdvertisementFile="Ads.xml" BorderWidth="1"
         runat=server />

   </form>
```

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

ads.xml

```
<Advertisements>

  <Ad>
    <ImageUrl>images/banner1.gif</ImageUrl>
    <NavigateUrl>default2.aspx</NavigateUrl>
    <AlternateText>banner1 not display</AlternateText>
    <Keyword>Computers</Keyword>
    <Impressions>1</Impressions>
  </Ad>

  <Ad>
    <ImageUrl>images/banner2.gif</ImageUrl>
    <NavigateUrl>http://www.google.com</NavigateUrl>
    <AlternateText> banner2 not display </AlternateText>
    <Keyword>Computers</Keyword>
    <Impressions>1</Impressions>
  </Ad>

  <Ad>
    <ImageUrl>images/banner3.gif</ImageUrl>
    <NavigateUrl>http://www.yahoo.com</NavigateUrl>
    <AlternateText> banner3 not display </AlternateText>
    <Keyword>Computers</Keyword>
    <Impressions>1</Impressions>
  </Ad>


</Advertisements>

=        =        =        =        =        ==
```

HTML server control

By default, HTML elements within an ASP.NET file are treated as literal text and are programmatically inaccessible to page developers. To make these elements programmatically accessible, you can indicate that an HTML element should be parsed and treated as a server control by adding a **runat="server"** attribute.

The runat = "server" attribute indicates that:

1. the element should be treated as a server control
2. the form should be processed on the server. It also indicates that the
3. enclosed controls can be accessed by server scripts.

The unique **id** attribute allows you to programmatically reference the control. Attributes are used to declare property arguments and event bindings on server control instances.

HTML server controls must reside within a containing **<form>** tag with the **runat="server"** attribute.

HTML Server Control	Description
HtmlAnchor	corresponding an <a> HTML element
HtmlButton	corresponding a <button> HTML element
HtmlForm	corresponding a <form> HTML element
HtmlGeneric	Controls other HTML element not specified by a specific HTML server control, like <body>, <div>, <span>, etc.
HtmlImage	corresponding an <image> HTML element
HtmlInputButton (Button/Reset/Submit)	corresponding <input type="button">, <input type="submit">, and <input type="reset"> HTML elements
HtmlInputCheckBox	corresponding an <input type="checkbox"> HTML element
HtmlInputFile	corresponding an <input type="file"> HTML element
HtmlInputHidden	corresponding an <input type="hidden"> HTML element

HtmlInputImage	corresponding an <input type="image"> HTML element
HtmlInputRadioButton	corresponding an <input type="radio"> HTML element
HtmlInputText (Password/Text)	corresponding <input type="text"> and <input type="password"> HTML elements
HtmlSelect	corresponding a <select> HTML element
HtmlTable	corresponding a <table> HTML element
HtmlTableCell	corresponding <td>and <th> HTML elements
HtmlTableRow	corresponding a <tr> HTML element
HtmlTextArea	corresponding a <textarea> HTML element

Common attributes (properties) of HTML server controls:

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
Visible	A Boolean value that indicates whether or not the control should be visible. Default is true
id	A unique id for the control. In the browser, refer control with "id" property.
Name	A unique name of the control. In server side coding , refer control with "name" property.
InnerHtml	Sets / returns the content between the opening and closing tags of the HTML element. Special characters are not automatically converted to HTML entities
Inner Text	Sets / 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 link is clicked
runat	Specifies that the control is a server control. Must be set to "server"
Style	Sets / returns the CSS properties
TagName	Returns the element tag name

Target	The target window to open. possible values:
Title	A title to be displayed by the browser (like the alt attribute of the img element)

Difference between html control and server control

ASP.NET Server Controls

Advantages:

1. ASP .NET Server Controls can detect the target browser's capabilities and render themselves accordingly.

2. Newer set of controls that can be used in the same manner as any HTML control like Calender controls.

3. Processing would be done at the server side. In built functionality to check for few values(with Validation controls) so no need to choose between scripting language which would be incompatible with few browsers.

4. ASP .NET Server Controls have an object model different from the traditional HTML and even provide a set of properties and methods that can change the outlook and behavior of the controls.

5. ASP .NET Server Controls have higher level of abstraction. An output of an ASP .NET server control can be the result of many HTML tags that combine together to produce that control and its events. Example Gridview or Form control.

Disadvantages:

1. The control of the code is inbuilt with the web server controls so you have no much of direct control on these controls

- Web Server Controls (ASP.NET controls): Web Server Controls are group of controls derived directly from the System.Web.UI.WebControls base class. They are executed on the server side and output HTML sent back to the client browser. These controls are programmable and reusable that can perform function as the ordinary HTML controls.
- Web Server Controls can detect the target browser's capabilities and render themselves accordingly.

HTML Server Controls

Advantages:

1. The HTML Server Controls follow the HTML-centric object model. Model similar to HTML

2. Here the controls can be made to interact with Client side scripting. Processing would be done at client as well as server depending on your code.

3. A HTML Server Control has similar abstraction with its corresponding HTML tag and offers no abstraction.

Disadvantages:

1. You would need to code for the browser compatibility.

2. The HTML Server Controls have no mechanism of identifying the capabilities of the client browser accessing the current page.

= = = =

- Html server control map to html control. Web server controls do not necessarily map to any existing HTML elements and they may represent more complex elements.
- Server controls are easy to use and manage but HTML controls are not.
- Server control events are handled in the server side whereas HTML control events are handled in the client side browser only.
- It can maintain data across each requests using view state whereas HTML controls have no such mechanism to store data between user requests.

- -- -- - = = === =

postback

- One of the most important features of the ASP.NET environment is the ability to declare controls that run on the server, and post back to the same page. The contents of the form are *POST*ed *back* to the same URL as the form.
- server posts the same page back to the browser.
- In asp.net, the server simply posts the page back to itself and performs all the validation, display and actions. The values of the Form are posted to the same page and the very same page can process the data. This model is called post back.
- In the context of web development, a *postback* is an HTTP POST to the same page that the form is on.
- **Exampe.1** :login page

  After the user has typed his username and password, he clicks on the *'Login'* button. Upon the click, the page is sent to the server to check against the database file to check if the user with supplied details is an authenticated user.

- **Example.2**:
  PostBack is done if certain credentials of the page are to be checked against some sources (such as verification of username and password using database). This is something that a client machine is not able to accomplish and thus these details have to be 'posted back' to the server

Autopostback

- In certain situation, certain raised events require immediate action. i.e. PostBack is needed without clicking button control.
- AutoPostBack is the mechanism, by which the page will be posted back to the server automatically based on some events in the web controls
- If this property is set to TRUE the automatic post back is enabled, otherwise FALSE. Default is FALSE.
- Example: Consider the case in which you have 2 ListBoxes. Let us say the first ListBox asks you for the Country you reside in and the second one asks you for the State in that country. Based upon the Country you select, the list of States must be shown. Thus, in order to fill the values in the second ListBox, the selection in the first ListBox must be known to the server. Thus, as and when an item is selected in the first ListBox, a PostBack must be done and the appropriate list of items must be filled in the second ListBox. To handle these situations, you need to enable the *'Auto PostBack'* property for first ListBox.

IsPostBack

- This property indicates whether page is loaded because of postback or not.
- It is a Boolean property of a page when is set (=false) when a page is first loaded.
- First time that the page loads the IsPostBack is false and for subsequent PostBacks, it is true.
- Each time a PostBack occurs, the entire page including the *Page_Load* is *'posted back*' and executed.

= = == = = = = = = =

❖ Object oriented concept:

1. Object

- Object is instance of class.
- Object is representative of the class and is responsible for memory allocation of its data members and member functions. An object is a real world entity having attributes (data type) and behaviors (functions).
- An object can be considered a "thing" that can perform a set of related activities. The set of activities that the object performs defines the object behavior. For example, the hand can grip something or a Student (object) can give the name or address
- Objects are the basic run-time entities in an object-oriented system.

Example: Dim obj As stud ‘create object of class stud

2. Class

Class is a data structure that contains data members (constants files, events), member function methods, properties, constructor, destructor, indexers and nested type.Basically

- It is a user defined data type.
- It is a reference type.
- Infact class is a tag or template for object.

A class definition starts with the keyword **Class** followed by the class name; and the class body, ended by the End Class statement.

```
[ accessmodifier ]  Class name
    [ Inherits classname ]
    [ Implements interfacenames ]
    [ statements ]
End Class
```

```
Public Class stud

    Dim a as integer  ‘field name


End Class
```

3. Inheritance

○ A key feature of OOP is reusability.

- The process of deriving a new class from an existing class is called Inheritance. The old class is called the base class and the new class is called derived class. The derived class inherits some or everything of the base class.
- It's always time saving and useful if we can reuse something that already exists rather than trying to create the same thing again and again. Once a class has been written and tested, it can be used by other programs to suit the program's requirement.
- In Visual Basic we use the **Inherits** keyword to inherit one class from other.

```
Public Class One

---

---

End Class

Public Class Two

    Inherits One

---

---

End Class
```

Class one is base class and class two is derived class.

4. Encapsulation

Encapsulation is a mechanism of binding the data member & member function into a single unit known as class. Encapsulation provides a way for abstraction. The class is kind of a container or capsule or a cell, which encapsulate the set of methods, attribute and properties to provide its indented functionalities to other classes.

5. Data abstraction

Abstraction is a process of hiding the implementation details and displaying the essential features.

Data abstraction is a mechanism to provide the essential features without describing the background details. Means provide the functions to access the hidden (private) data.

It places the emphasis on what an object is or does rather than how it is represented or how it works.

Protecting the data of an object from outside functions is called Abstraction or Data Hiding

6. Data Hiding

Data hiding is a mechanism to hide the internal structure of an object from rest of the program. In a class private members are hidden from the rest of the program, hence it supports data hiding. Data hiding is also a way to implement data abstraction.

7. Polymorphism

Polymorphism means one thing in many form. Basically polymorphism is capability of one object to behave in multiple ways. There are following types of polymorphism:

1. **Overloading :**It is achieved using function overloading and operator overloading.

define the same method multiple times so that you can call them with different argument lists (a method's argument list is called its signature

```
Class stud
    Dim a As Integer
    Public Sub display()
        MsgBox("display in stud:without argument ")
    End Sub
    Public Sub display(ByVal m1 As Integer)
      MsgBox("display in stud:with one argument: oveloading: " & m1)
    End Sub

   End Class
```

2. **Dynamic polymorphism(runtime time) :** It is achieved using function overriding means using virtual function.

8. Constructors

A constructor is a special method that must be present in a class for the class to get instantiated.

- Constructors are a special kind of sub procedure. A constructor has the following properties:
- It always has the name ‘new’
- Being a sub procedure, it does not return any value

```
Public Class stud
    Sub New()
    End Sub

End Class
```