Framework & Web Contents

UNIT - 1

Overview ASP.NET Framework

- **ASP.NET** is an open source server-side Web application framework designed for Web development to produce dynamic Web pages.
- It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.
- It was first released in January 2002 with version 1.0 of the.NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology.
- ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET languages like VB.NET and C#.
- The Microsoft .NET Platform provides all of the tools and technologies that are needed to build distributed Web applications.

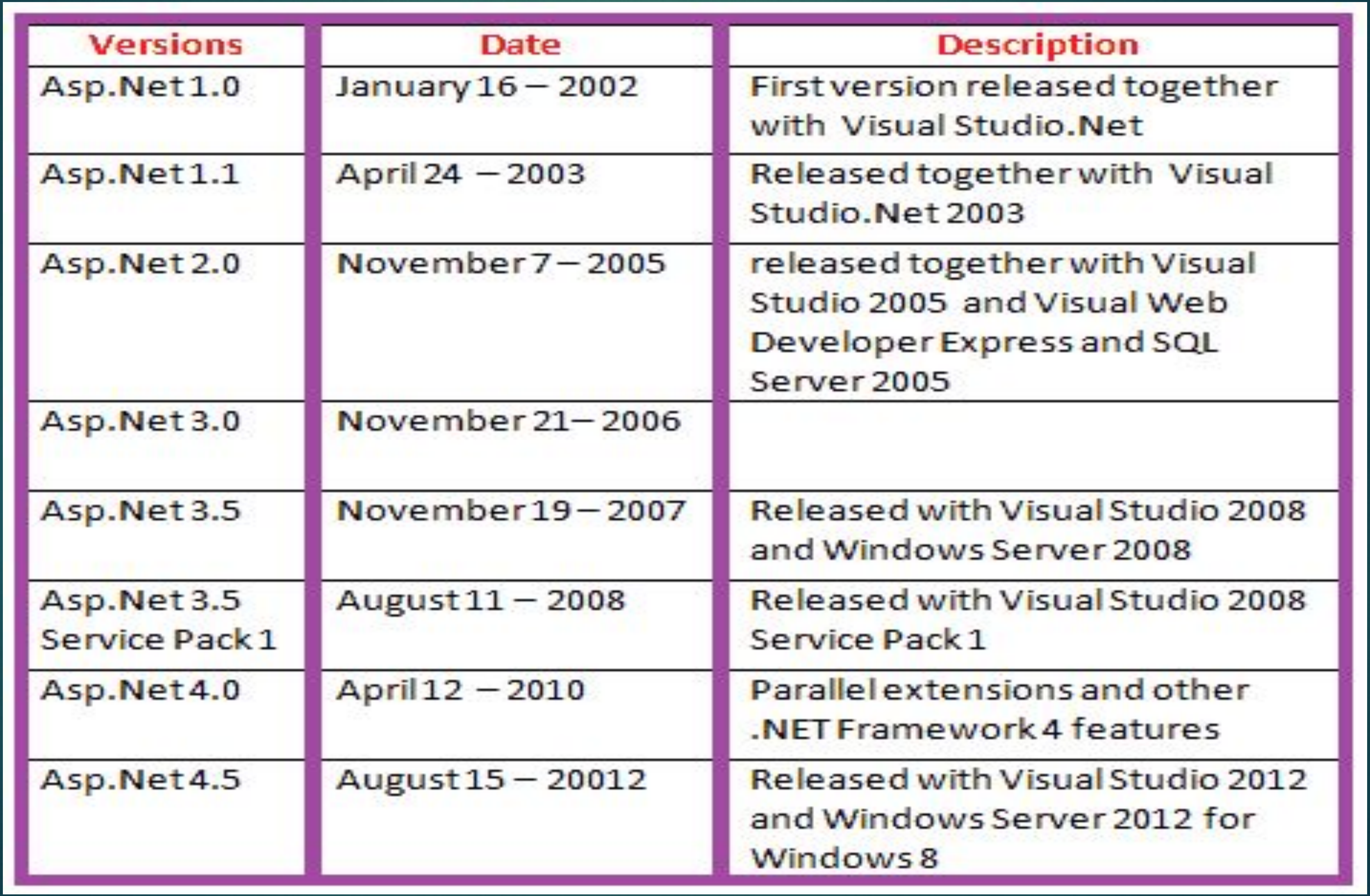
Overview ASP.NET Framework

- ASP.NET is integrated with Visual Studio .NET, which provides a GUI designer, a rich toolbox, and a fully integrated debugger.
- In ASP.NET, you can write the HTML code in the .aspx file and the code for programming logic in the code-behind file (.aspx.vb or .aspx.cs). Also ASP.NET introduces two sets of controls, the HTML controls and the Web controls, which are collectively known as "server controls."
- ASP.NET incorporates all the important standards of our time, such as XML and SOAP, plus with ADO.NET and the foundation class libraries.

Overview ASP.NET Framework

- ASP.NET is great for building standards-based websites with HTML5, CSS3, and JavaScript.
- ASP.NET supports three approaches for making web sites.
  - ASP.NET Web Forms uses controls and an event-model for component-based development.
  - ASP.NET MVC values separation of concerns and enables easier test-driven development.
  - ASP.NET Web Pages prefers a single page model that mixes code and HTML markup.

ASP.NET Versions

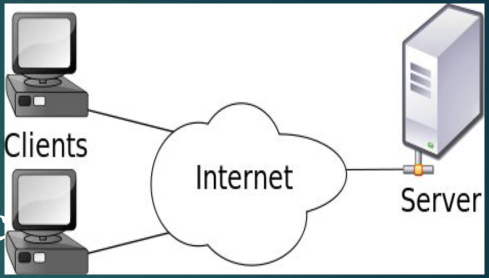


Versions	Date	Description
Asp.Net 1.0	January 16 – 2002	First version released together with Visual Studio.Net
Asp.Net 1.1	April 24 – 2003	Released together with Visual Studio.Net 2003
Asp.Net 2.0	November 7 – 2005	released together with Visual Studio 2005 and Visual Web Developer Express and SQL Server 2005
Asp.Net 3.0	November 21 – 2006	
Asp.Net 3.5	November 19 – 2007	Released with Visual Studio 2008 and Windows Server 2008
Asp.Net 3.5 Service Pack 1	August 11 – 2008	Released with Visual Studio 2008 Service Pack 1
Asp.Net 4.0	April 12 – 2010	Parallel extensions and other .NET Framework 4 features
Asp.Net 4.5	August 15 – 20012	Released with Visual Studio 2012 and Windows Server 2012 for Windows 8

ASP.NET Benifits

- Separate presentation from code
- Object-oriented approach
- Component-based development
- Event-driven architecture
- Code compilation
- Extensible architecture
- Built-in state management
- Many others (data binding, validation, master pages, etc.)

Client-Server Architecture



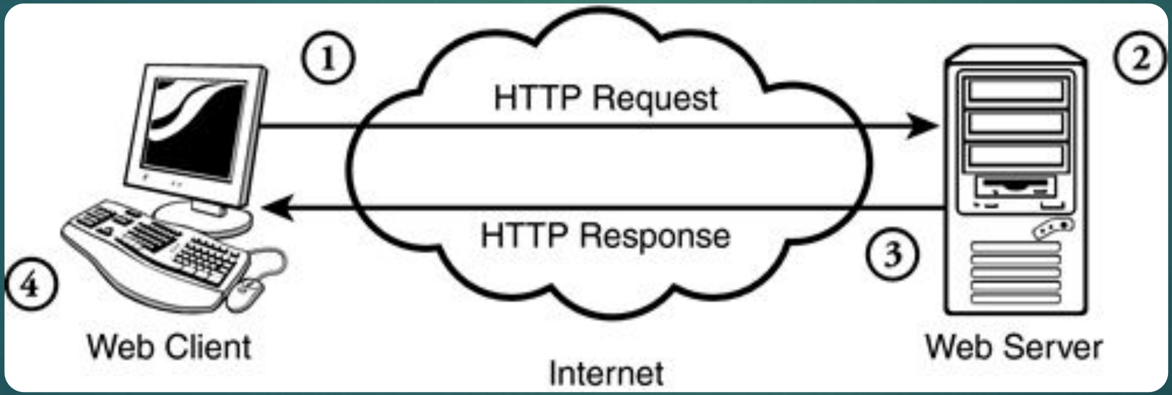
- Client-server architecture (client/server) is a network architecture in which each computer or proc the network is either a *client* or a *server*.
- Servers are powerful computers dedicated to managing disk drives (*file servers*), printers (*print servers*), or network traffic (*network servers*).
- Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.
- Examples of computer applications that use the client–server model are Email, network printing, and the World Wide Web.

Client-Server Architecture

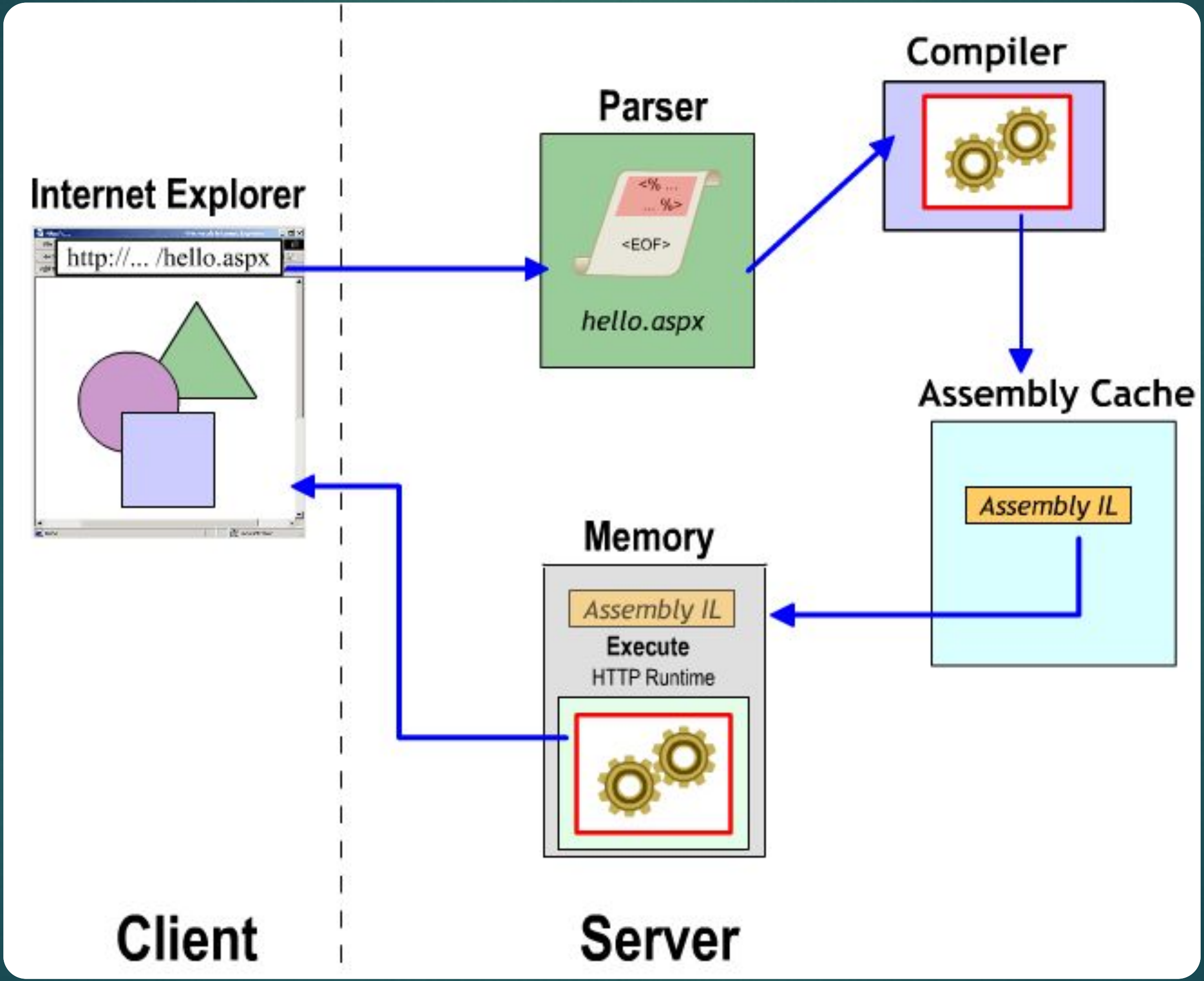
- **Advantages:**
  - All data stored at server
  - Due to thin Client application less load on client.
  - Easy to Implement Security.
- **Disadvantages:**
  - Clients are dependent on servers.
  - All load transfer on Servers.
  - Better bandwidth is required for server.

ASP.NET Execution

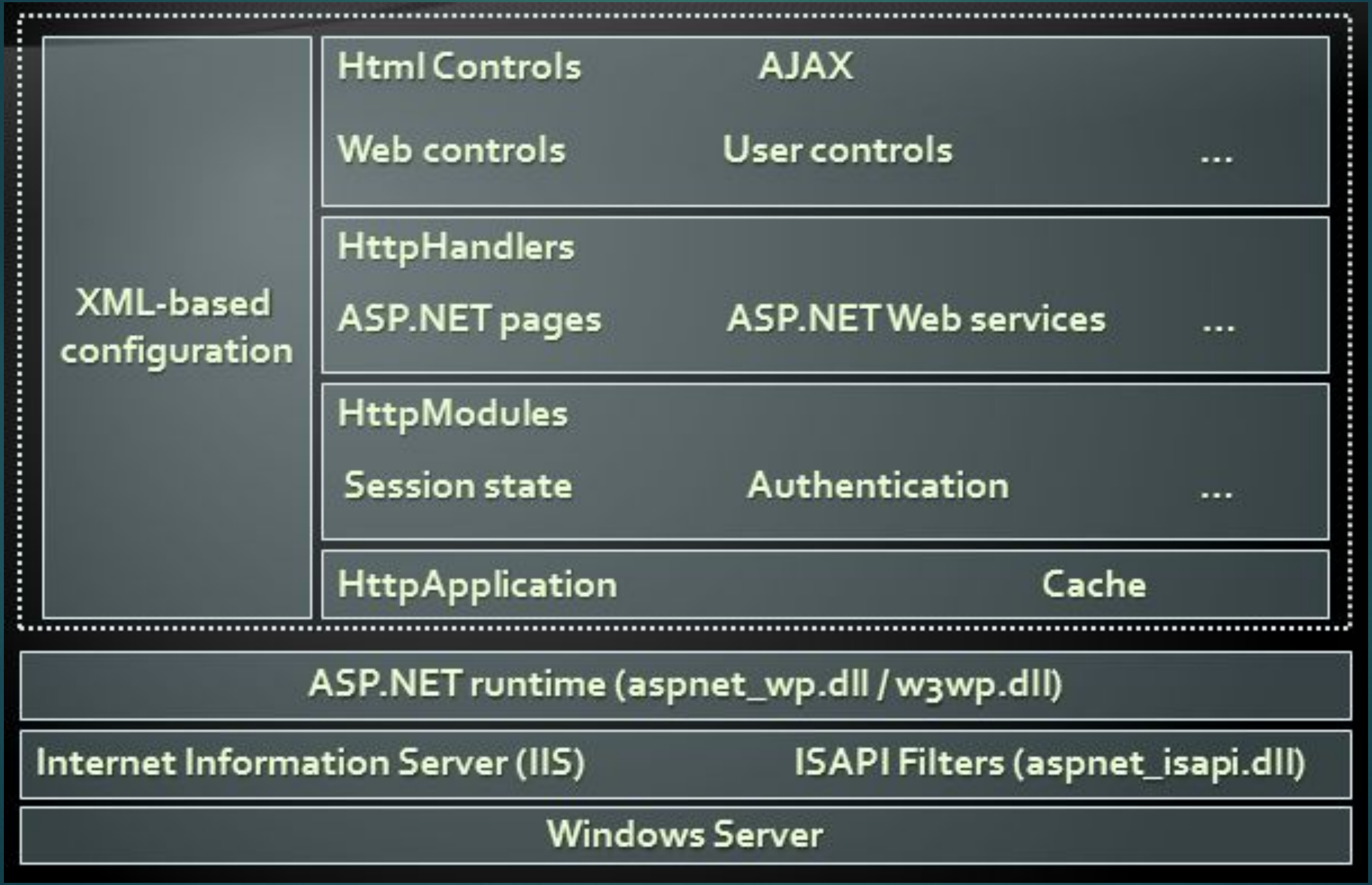
- ASP.NET applications are executed via a sequence of HTTP requests and HTTP responses
  - Client Web browser request ASPX pages
  - The Web server executes the ASPX page and produce XHTML + CSS + JavaScript



ASP.NET Execution



ASP.NET Architecture



Types of Files in ASP.NET

- ASP.NET have many types of files. They are,
  - .aspx
  - .ascx
  - .asmx
  - Web.config
  - Global.asax
  - .cs /.vb

Types of Files in ASP.NET

- **.aspx :** These are ASP.NET web pages (the .NET equivalent of the .asp file in an ASP application). They contain the user interface and optionally, the underlying application code. Users request or navigate directly to one of these pages to start your web application.

- **.ascx :** These are ASP.NET user controls. User controls are similar to web pages, except that they can't be accessed directly. Instead, they must be hosted inside an ASP.NET web page. User controls allow you to develop a small piece of user interface and reuse it in as many web forms as you want without repetitive code.

- **.asmx :** These are ASP.NET web services. Web services work differently than web pages, but they still share the same application resources, configuration settings, and memory.

Types of Files in ASP.NET

- **web.config :** This is the XML-based configuration file for your ASP.NET application. It includes settings for customizing security, state management, memory management, and much more.

- **global.asax :** This is the global application file. You can use this file to define global variables (variables that can be accessed from any web page in the web application) and react to global events (such as when a web application first starts).

- **.cs / .vb :** These are code-behind files that contain C# code or VB code. They allow you to separate the application from the user interface of a web page.

Types of Controls in ASP.NET

- The ASP.NET Framework (version 3.5) contains over 70 controls. These controls can be divided into eight groups:
  - Standard Controls
  - Validation Controls
  - Rich Controls
  - Data Controls
  - Login Controls
  - Web Part Controls
  - Html Controls
  - Navigation Cintrol

Types of Controls in ASP.NET

- **Standard Controls :** The standard controls enable you to render standard form elements such as buttons, input fields, and labels. We examine these controls in detail in the following chapter, "Using the Standard Controls.“
- **Validation Controls :** The validation controls enable you to validate form data before you submit the data to the server. For example, you can use a RequiredFieldValidator control to check whether a user entered a value for a required input field.
- **Rich Controls :** The rich controls enable you to render things such as calendars, file upload buttons, rotating banner advertisements, and multi-step wizards.

Types of Controls in ASP.NET

- **Data Controls :** The data controls enable you to work with data such as database data. For example, you can use these controls to submit new records to a database table or display a list of database records.
- **Navigation Controls :**The navigation controls enable you to display standard navigation elements such as menus, tree views, and bread crumb trails.
- **Login Controls :** The login controls enable you to display login, change password, and registration forms.
- **Web Part Controls :** The Web Part controls enable you to build personalizable portal applications.
- **HTML Controls :** The HTML controls enable you to convert any HTML tag into a server-side control.

Page Class

- Every web page is a custom class that inherits from System.Web.UI.Page. By inheriting from this class, your web page class acquires a number of properties that your code can use. These include properties for enabling caching,

Property	**Description**
Application and Session	**These collections hold state information on the server.**
Cache	**This collection allows you to store objects for reuse in other pages or for other clients.**
Controls	**Provides a collection of all the controls contained on the web page. You can also use the methods of this collection to add new controls dynamically.**
EnableViewSt ate	**When set to false, this overrides the EnableViewState property of the contained controls, thereby ensuring that no controls will maintain state information.**

Page Class

IsPostBack	This Boolean property indicates whether this is the first time the page is being run (false) or whether the page is being resubmitted in response to a control event, typically with stored view state information (true). This property is often used in the Page.Load event handler, thereby ensuring that basic setup is performed only once for controls that maintain view state
Request	Refers to an HttpRequest object that contains information about the current web request, including client certificates, cookies, and values submitted through HTML form elements. It supports the same features as the built-in ASP Request object.
Response	Refers to an HttpResponse object that allows you to set the web response or redirect the user to another web page. It supports the same features as the built-in ASP Response object, although it's used much less in .NET development.
Server	Refers to an HttpServerUtility object that allows you to perform some miscellaneous tasks, such as URL and HTML encoding. It supports the same features as the built-in ASP Server object.
User	If the user has been authenticated, this property will be initialized with user information

Control Class

- The Page.Controls collection includes all the controls on the current web form.
- You can loop through this collection and access each control.
- You can also use the Controls collection to add a dynamic control.

HttpRequest Class

- The HttpRequest class encapsulates all the information related to a client request for a web page.
- Most of this information corresponds to low-level details such as posted-back form values, server variables, the response encoding, and so on. If you’re using ASP.NET to its fullest, you’ll almost never dive down to that level.
- Other properties are generally useful for retrieving information, particularly about the capabilities of the client browser.

HttpRequest Class

Property	Description
Application Path and PhysicalPat h	These collections hold state information on the server.
Browser	This collection allows you to store objects for reuse in other pages or for other clients.
ClientCerti cate	Provides a collection of all the controls contained on the web page. You can also use the methods of this collection to add new controls dynamically.
Cookies	When set to false, this overrides the EnableViewState property of the contained controls, thereby ensuring that no controls will maintain state information.
Headers and Server Variables	This Boolean property indicates whether this is the first time the page is being run (false) or whether the page is being resubmitted in response to a control event, typically with stored view state information (true). This property is often used in the Page.Load event handler, thereby ensuring that basic setup is performed only once for controls that maintain view state

HttpRequest Class

IsAuthenticated and IsSecureConnection	Returns true if the user has been successfully authenticated and if the user is connected over SSL (also known as the Secure Sockets Layer).
QueryString	Provides the parameters that were passed along with the query string.
Url and UrlReferrer	Provides a Uri object that represents the current address for the page and the page where the user is coming from (the previous page that linked to this page).
UserAgent	A string representing the browser type. Internet Explorer provides the value MSIE for this property.
UserHostAddress and UserHostName	Gets the IP address and the DNS name of the remote client. You could also access this information through the ServerVariables collection.
UserLanguages	Provides a sorted string array that lists the client's language preferences. This can be useful if you need to create multilingual pages.

HttpResponse Class

- The HttpResponse class allows you to send information directly to the client. In traditional ASP development, the Response object was used heavily to create dynamic pages. The HttpResponse does still provide some important functionality, namely, caching support, cookie features, and the Redirect method.

Property	**Description**
BufferOutput	**When set to true (the default), the page isn't sent to the client until it's completely rendered and ready, as opposed to being sent piecemeal.**
Cache	**References an HttpCachePolicy object that allows you to configure how this page will be cached**
Cookies	**The collection of cookies sent with the response.**
Write(), BinaryWrite(), and WriteFile()	**These methods allow you to write text or binary content directly to the response stream. You can even write the contents of a file. These methods are de-emphasized in ASP.NET and shouldn't be used in conjunction with server controls.**
Redirect()	**This method transfers the user to another page in your application or a different website.**

Introduction to standard Controls

- Web Forms are the basics of building ASP.NET based web sites.
- Web site is collection of number of web Forms / pages where each Web Form contains number of Web Server controls.
- Which are described as further:

Button control

- Button control is used to submit the data to the server. Button control works like a Push Button when you click the data is submitted to the server.

Property	**Description**
AccessKey	Enables you to specify a key that navigates to the Button control.
CommandArgument	Enables you to specify a command argument that is passed to the Command event.
CommandName	Enables you to specify a command name that is passed to the Command event.
enabled	Enables you to disable the Button control.
OnClientClik	Enables you to specify a client-side script that executes when the button is clicked.
PostBackUrl	Enables you to post a form to a particular page.
TabIndex	Enables you to specify the tab order of the Button control.
Text	Enables you to label the Button control.

Button control

- The Button control also supports the following method:
  - **Focus :** Enables you to set the initial form focus to the Button control.
  - **Click :** Raised when the Button control is clicked.
  - **Command :** Raised when the Button control is clicked. The CommandName and CommandArgument are passed to this event

Example:

```
<asp:Button ID="Button2" runat="server" Text="Button"
onclick="Button2_Click" />
```

TextBox Control

- The TextBox control can be used to display three different types of input fields depending on the value of its TextMode property.
- The TextMode property accepts the following three values:
  - **SingleLine :** Displays a single-line input field.
  - **MultiLine :** Displays a multi-line input field.
  - **Password :** Displays a single-line input field in which the text is hidden.
- The TextBox control also supports the following method:
  - **Focus :** Enables you to set the initial form focus to the text box.
  - **TextChanged :** Raised on the server when the contents of the text box are changed

Property	Description
AccessKey	Enables you to specify a key that navigates to the TextBox control.
AutoCompleteType	Enables you to associate an AutoComplete class with the TextBox control.
AutoPostBack	Enables you to post the form containing the TextBox back to the server automatically when the contents of the TextBox is changed.
Enabled	Enables you to disable the text box.
MaxLength	Enables you to specify the maximum length of data that a user can enter in a text box (does not work when TextMode is set to Multiline).
ReadOnly	Enables you to prevent users from changing the text in a text box.
TabIndex	Enables you to specify the tab order of the text box.
Wrap	Enables you to specify whether text word-wraps when the TextMode is set to Multiline.

CheckBox

- CheckBox control is used to accept the choice from user. It is used to display multiple choices from which user can select none of them or many or all of them.
- For example, if you want to accept Hobbies of user, you can use CheckBox control.
- The CheckBox control also supports the following method:
  - **Focus :** Enables you to set the initial form focus to the check box.
  - **CheckedChanged :** Raised on the server when the check box is checked or unchecked.

Property	Description
AccessKey	Enables you to specify a key that navigates to the TextBox control.
AutoPostBack	Enables you to post the form containing the CheckBox back to the server automatically when the CheckBox is checked or unchecked.
Checked	Enables you to get or set whether the CheckBox is checked.
Enabled	Enables you to disable the TextBox.
TabIndex	Enables you to specify the tab order of the check box.
Text	Enables you to provide a label for the check box.
TextAlign	Enables you to align the label for the check box. Possible values are Left and Right.

Label

- Whenever you need to modify the text displayed in a page dynamically, you can use the Label control. Any string that you assign to the Label control's Text property is displayed by the Label when the control is rendered.
- You can assign simple text to the Text property or you can assign HTML content.
- As an alternative to assigning text to the Text property, you can place the text between the Label control's opening and closing tags.
- Any text that you place before the opening and closing tags gets assigned to the Text property.

Property	Description
BackColor	Enables you to change the background color of the label.
BorderColor	Enables you to set the color of a border rendered around the label.
BorderStyle	Enables you to display a border around the label. Possible values are NotSet, None, Dotted, Dashed, Solid, Double, Groove, Ridge, Inset, and Outset.
BorderWidth	Enables you to set the size of a border rendered around the label.
CssClass	Enables you to associate a Cascading Style Sheet class with the label.
Font	Enables you to set the label's font properties.
ForeColor	Enables you to set the color of the content rendered by the label.

Panel

- The Panel control enables you to work with a group of ASP.NET controls. You can use a Panel control to hide or show a group of ASP.NET controls.

Property	**Descrition**
DefaultButton	Enables you to specify the default button in a Panel. The default button is invoked when you press the Enter button.
Direction	Enables you to get or set the direction in which controls that display text are rendered. Possible values are NotSet, LeftToRight, and RightToLeft.
GroupingText	Enables you to render the Panel control as a fieldset with a particular legend.
HorizontalAlign	Enables you to specify the horizontal alignment of the contents of the Panel. Possible values are Center, Justify, Left, NotSet, and Right.
ScrollBars	Enables you to display scrollbars around the panel's contents. Possible values are Auto, Both, Horizontal, None, and Vertical.

Dropdownlist

- DropDownList control is used to give a single select option to the user 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.
- Its properties like BackColor, ForeColor etc. are implemented through style properites of <span>. It has less property to decorate in comparison with other controls. There is no property like BorderStyle, BorderWidth. in DropDownList control.
- You can add its option items by directly writing into .aspx page directly or dynamically add at run time or bind through database.
  - <asp:DropDownList ID=“dp1” runat=“server”>
  - <asp:ListItem>item 1</asp:ListItem>
  - </asp:DropDownList>

Property	Description
SelectedValue	Get the value of the Selected item from the dropdown box.
SelectedIndex	Gets or Sets the index of the selected item in the dropdown box.
SelectedItem	Gets the selected item from the list.
Items	Gets the collection of items from the 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. It will also fireOnSelectedIndexChanged method.
AppendDataBoundItems	true or false. If true, the statically added item (added from .aspx page) is maintained when adding items dynamically (from code behind file) or items are cleared.
OnSelectedIndexChanged	Method name that fires when user changes the selection of the dropdown box. (Fires only when AutoPostBack=true.)

ListBox

- All properties and its working resembles DropDownList box. However, ListBox has two extra properties called Rows and SelectionMode. ListBox control is used to give a single or multiple select option to the user (based on the property set) from multiple listed items. You can specify its height and width in pixel by setting its height and width but you will not be able give mutliple select option to the user. When it is rendered on the page, it is implemented through <select/> HTML tag. It is also called as Combo box.
- Its properties like BackColor, ForeColor etc. are implemented through style properites of <span>. It has less property to decorate in comparison with other controls. There is no property like BorderStyle, BorderWidth. in DropDownList control.
- You can add its option items by directly writing into .aspx page directly or dynamically add at run time or bind through database.

Property	Description
Rows	No. of rows (items) can be set to display in the List.
SelectionMode	Single or Multiple. If multiple, it 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 dropdown box.
SelectedIndex	Gets or Sets the index of the selected item in the dropdown box.
SelectedItem	Gets the selected item from the list.
Items	Gets the collection of items from the 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 listbox box.
AutoPostBack	true or false. If true, the form is automatically posted back to the server when user changes the dropdown list selection. It will also fireOnSelectedIndexChanged method.
OnSelectedIndexChanged	Method name that fires when user changes the selection of the dropdown box. (Fires only when AutoPostBack=true.)

FileUpload

- The FileUpload control allows the user to browse for and select the file to be uploaded, providing a Browse button and a text box for entering the filename.
- Once, the user has entered the filename in the text box, by typing the name or browsing, the SaveAs method of the FileUpload control can be called to save the file to the disk.
- The basic syntax for using the FileUpload is:
  - <asp:FileUpload ID= "Uploader" runat = "server" />
- The FileUpload class is derived from the WebControl class, and inherits all its members. Apart from those, the FileUpload class has the following read-only properties:

Property	Description
FileBytes	Returns an array of the bytes in a file to be uploaded..
FileContent	Returns the stream object pointing to the file to be uploaded.
FileName	Returns the name of the file to be uploaded.
HasFile	Specifies whether the control has a file to upload.
PostedFile	Returns a reference to the uploaded file.
ContentLength	Returns the size of the uploaded file in bytes.
ContentType	Returns the MIME type of the uploaded file
FileName	Returns the full filename.
InputStream	Returns a stream object pointing to the uploaded file.