ASP.NET

Draft Version

2014

www.najam.in

Web Application Basic

*According to Wikipedia “A****web application****or****web app****is any application software that runs in a web browser or is created in a browser-supported programming language (such as the combination of JavaScript, HTML and CSS) and relies on a common web browser to render the application”.*

*Computer programs have evolved over the years, earlier they were very basic calculations, being executed on huge machines. As the applicability of computers increased, size of program increased. Need of making applications able to communicate across machines was felt. There were various reasons for need of this communication, specialized programs may reside on different machine, a single computer was not able to perform huge calculation, or many people need to work on same data. Concept of network emerged and Client Server application emerged. In client-server, the processing was done by both server and client. Server used to reside on one machine and client was installed on each user’s machine. Main issue with setup was that both codes were very tightly integrated, most of time any change is server code required change on client. More over if client is changed, you need to install new version on each user’s machine. In early stage this was not a problem, but at computer usage grew people were sharing information across geographical area, telling user about upgrade got troublesome. Also each different application required separate client installation.*

*Idea of web application was based on having a universal client application. This means you have a client application that you installed on your machine, now you can use lot many server applications. This is still pretty much client server, with lot of problems gone.*

*Web applications work on HTTP protocol. Let’s first understand what protocol is and then what is HTTP.*

*Protocol is a set of computer rules that govern how communication is done between computers. Communicating systems use well-defined formats for exchanging messages (this many bytes are headers, next X bytes are body….). Each message has an exact meaning intended to provoke a particular response of the receiver.*

**HTTP**: This stands for Hypertext transfer protocol, let’s start with each component. HyperText is data on web connected through hyperlink; it covers all kind of file, html (Static/dynamic), images, css, javascript etc. A newer term related to this HyperMedia, this cover additional thing like video files, audio and other things.

*HTTP is for getting resource from web (network) so on base of HTTP, is TCP IP protocol, now you might be wondering TCP IP bhi hai HTTP bhi hai 2 protocols! Ager yaad karo to TCP IP transport layer per hai but HTTP application layer protocol hai. As web is intended for large number of user, people decide to keep it stateless. State is remembering anything. To understand state, think of large shop, when you enter the shop, you have to submit any bag that you have with you to gatekeeper. Now think that gatekeeper as server and people as client connecting to server. One way is you give gatekeeper takes your bags and he remembers you, now suppose 100 more people came, is it possible for him to remember every person? What if 5000 people come, he is not going to remember. Scenario 2 we change the set up, we give gatekeeper a number of tokens and ask him to give customer a token and put a copy of same token with bags. Now no matter how many people come he will be able to serve them!! This is because he does not have to remember anything.*

*We are happy with statelessness of HTTP is most of case; this is because we want web to handle more data and we don’t want situation when client connects, we have its state and then he never comes back, resources on server will be wasted. When a client send a request to server, connection is established, server reads request, and take necessary steps by executing relevant code and then sends response (mostly html) back to client and connection is closed.*

*So in short we know that HTTP is a protocol which works over TCP IP and is stateless.*

*Now web works in following manner, client asks for some resource, (you want to GET data)*

*Then you do something on that data and you send it back to server (you POST data), this goes on and on..*

HTTP defines methods (sometimes referred to as verbs) to indicate the desired action to be performed on the resource. So for now think that HTTP has verbs which define type of action (Like we have select, insert, update, delete in sql)

Is being taken and these verbs are GET, POST, HEAD, PUT, DELETE. There are few other which you need to worry as of now.

In core if we look at HTTP request it is like

GET /index.html HTTP/1.1

Host: www.example.com

*GET is verb (Action type) index.html is resource asked, Protocol version is HTTP 1.1, and host where this request has to be send is* [www.example.com](http://www.example.com)*, I will be covering HTTP in greater depth in another notes.*

*Now when we have basic understanding of how web works and what is HTTP, let’s drill deeper.*

*Consider your normal window application, es application me UI, ko bahut deep me socho, tumne ek form ka objet liya hai to windows ne ek window banadi, uske under tumne button aur textbox liye hai to windows in who bhi paint ker diya, tum mouse kisi control per le ja ker click kerte ho, to windows ka tension hai ki kya resolution hai, ya fir window shrink kiya huwa hai to actually kis control per mouse click huwa, tumko yeh sab sochna nahi padta. Fir jis control per click kiya, uska relevant code execute huwa aur respose fir se window paint kerta hai. To actually hamare application me do part hai, ek to UI dikh raha, aur ek peeche ka logic, database interaction etc. Window application me yeh hum sepreate feel nahi kerte, kyuki UI paint bhi usi machine per ho rahi aur logic bhi wahi execute ho raha. Web me is dono part ko alag alag kerte hai. UI wale to client samjho aur logic wale ko server, but is tarah to cliet server application hota he tha, to fir yeh soha gaya ki generic client (browser banate hai) aur application specific UI bhi server he bhejega, client ko sirf itna pata ho ki, server jo bhej raha usko process kaise kerna hai. We want freedom to create any UI, but we want to run on generic same application that is browser. To decide huwa ki ek standard format me UI instruction bheje jaye, this is what we call HTML.*

*HTML is hypertext markup language, dekho isme bhi hypertext hai!. TO HTML ka format decide huwa aur browsers implement ho gaye, HTML ko ager ek angle se dekho to who UI ka description he to hota hai, like create ka button of 200px, on left 100px, with background color so on.. is description ka he syntax to HTML hai. Kyuki HTML me fixed “tags” to browser ka code generic style me likha ja saka. Over the time bahut se kami feel hui HTML me isliye HTML ke bhi naye naye version aa rahe. So if you think it is pretty simple way to solve the problem. Today it may feel a big, revolutionary thing, but web has just simply evolved and almost everything is just commonsense.*

*SO when we request any resource (page) from server, server can send us static html or dynamically execute code to produce HTML (whatever is implementation) and send us HTML, we update that HTML (by putting values in control) and send it back to server, server again reads incoming HTML and take appropriate steps and send new HTML back to us.*

*Well, if you read basic HTTP and basic HTML, you can have fun on creating your own web server, and browser ☺, although current server and browser have evolved over years, your will be just extremely basic one but will be fun!*

*So till now we have seen ki HTML me he communication ho raha, fir kuch years baad logo to feel huwa ki jo bhi page aata hai usme kuch bhi kaam kerna ho to wapas server ke pass bheja padta hai, but kuch kaam jaise kisi particular option se selection per kisi control ko hide kerna, ya color change kerna, ya fir wapas post kerne se pehle validate kerna, yeh sab kaam ke liye ab client machine he use ho jaye to server per load kam hoga, plus network ka bhi kam use hoga. To script languages use hone lagi (javascript, vbscript). Yeh full language nahi hoti, isme chote light weight work hote hai. Yeh client ki machine per execute hote hai. (shuru shuru me iske support ke liye bhi browser ka code change kerna pada hoga!)*

*Ab events yaad hai? Yaha bhi event based kaam hota hai, browser ne page request bheji, server ne HTML de, is html ko jab browser process kerta hai to specific times per event bhi raise kerta hai, jaise html document pura load ho gaya to event, these event we can handle in our javascript code. To browser jaise he page load kerta hai, dekhta hai ki javascript ka koi function is event per register to nahi hai hai, ager hai to usko call kerdo. Along with such events controls ke event bhi javascript me handle hote hai, jaise button ka click, textbox ka textchange so on..*

*So scripting language ke aane se client side per basic work hone lage.*

*Ek part aur hai CSS (Cascading Style Sheet): Used for describing the*[*look and formatting*](http://en.wikipedia.org/wiki/Presentation_semantics)*of html document.*

*So in place of putting formatting information in every element in HTML, we create some reusable sets of style and use them across application. (More detail of HTML, JS & CSS will come as we will develop applications)*

*Little later on jab web aur bade honge to server side me bahut se issues aaye, let’ see them. Pehle initially jab dynamic pages bante the, then developers used to put html tag and server code in a page*

<html>

<body>

<%

response.write("Testing testing!")

%>

</body>

</html>

*Jo part bhi* <% %> *ke under hota hai who part server execute kerke uska result iski jagah fit kerke puri html client ko bhej deta tha. Over the time is model me problem aane lagi, server code aur HTML mix hone se confusion hoe laga aur maintenance mushkil ho gaya. Fir jab .net aaya to ASP.NET me ek major nayi functionality dali gayi, “Code behind”.*

*Code behind ka simple sa matlab hai ki server code alag file, html alag file me. Compile ho ker yeh wapas ek he file bante hai, but during development they are separate file. Ek sample asp.net project banao 2010 me default.aspx ko open karo*

<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"

CodeBehind="Default.aspx.cs" Inherits="WebApplication1.\_Default" %>

*Highlighted part ko dekho, it tells ki iski code file kaun se hai, ayesa koi zaruri nahi hai ki ager page ka naam Defailt.aspx hai to code file ka naam Default.aspx.cs hoga, tum kuch bhi naam rakho codebehind file ka, bus aspx page me set ker do. Even tum ek alag se dll bana ker uski kisi class ka naam de sakte ho!!!. Class ka naam Inherits wale part me set hota hai. Class ka naam isliye dena zaruri hai kyuki ek .cs file me multiple class ho sakti hai, to Codebehind se file, aur Inherits se class ka naam pata lagta hai.*

*Ab defult.aspx ka tree open karo,to 2 file dikhegi, .aspx.cs, yeh to code file ho gayi, second is autogenerated .aspx.designer.cs. is me kabhi koi change mat kerna, aur na he koi code likhna, yeh visual studio khud generate kerta hai, isko abhi open karoge to sirf*

public partial class \_Default {

}

*Hoga, wapas aspx page me jao aur ek button add ker do* <asp:Button ID="btn" runat ="server" Text="Click"/>

*Ab fir se designer file dekho*

public partial class \_Default {

/// Auto-generated field.

/// To modify move field declaration from designer file to code-behind file.

/// </remarks>

protected global::System.Web.UI.WebControls.Button btn;

}

*Btn naam ka Button class ka object ban gaya. Ab default.aspx.cs open karo, isme dekho*

public partial class \_Default : System.Web.UI.Page

{

**Things to observe**: Both classes are partial, to partial classes who class hoti hai jo multiple file me breaked hoti hai, compile ho ker ek class ban gayi *(ayesa isliye ki multiple developer ek class me kaam ker sakte and additional benefit, WCF me bahut use hota hai yeh feature, who sab baad me). Compile ho ker ek he class banegi..*

*.aspx.cs class jo hamare modification aur coding ke liye hai, that Inherits from System.Web.UI.Page. Jaise window application me form System.Windows.Form se inherit hota hai waise he page ki apni parent class hai.*

*Ab thoda wapas .aspx file per, we wrote* <asp:Button ID="btn" runat ="server" Text="Click"/>

*, but ager HTML ke tags dekho to unme to kahi bhi <Asp: kerke koi bhi tag nahi hai!!. HTML me to <input type=”button” hota hai. Ab isko run karo aur view source karo*

<input type="submit" name="ctl00$MainContent$btn" value="Click" id="MainContent\_btn" />

*Client per to yeh input he ban gaya!. Actully browser to sirf HTML tag samjhega, but who bahut he basic controls hai, ASP.NET ne basic aur kuch extra controls banaye (grid, repeater..) in controls ki class ke code me ASP.NET ne likh rakha hai ki jab HTML generate ho to kaise hone chaeye. Aur in controls ko compiler easly pehchan le to inka naam <asp:*

*Style ka hota hai. Button input me convert ho gaya, Grid simple table tr td me resolve hota hai!!*

*Compile time per .aspx.cs and aspx.designer.cs combine ho ker ready hai, user ne .aspx ke request bheji to .aspx to compiler dekha hai, jo, jo bhi server controls hai (<asp: ) unki bhi html generate kerta hai, fir sari html client ko wapas bhej deta hai. Yeh kafi simple level per hai, ab isme ASP.NET ek aur major part deta hai who hai uska event based hona.*

*ASP.NET ko event based banaya gaya hai, chahe woh request ki life cycle ho ya page ki life cycle. ASP.NET bolta hai ki mai execute kerte kerte event raise karunga, tum un events per apna code likh lo, woh mai execute ker dunga. Isme do tarah ke event ate hai, life cycle wale plus controls ke. So jab page execute hota hai to usme yeh event kafi important hote hai. That’s why you write some code in Page\_Load, some in Render and so on.*

*ASP.NET ki details me jane se pehle last important component to bhi short me dekhte hai, that is our SERVER. Server simple ek application hai, jiska kaam hai request ko handle kerna aur response dena. HTTP, khud TCP IP per chalta hai, to tum ager program likho jo HTTP ke liye designated port (80) per listen kare, aur request aane per usko parge kerke HTML response bheje to tumne ek web server bana diya. Jis tarah bahut sare browser hai, usse bhi jayda type ke server hai, jo 2 tumne sabse jyada sune honge woh hai IIS and Apache. IIS windows per hai, Microsoft ne banaya hai aur .NET ka only server hai. Apache php, ruby, javame use hota hai. Please keep in mind ke IIS per php, ruby, java bhi chal sakte hai, but .NET application sirf IIS per chalti hai. IIS ka kaam hota hai asp.net request ko samajhna, usko proper component( web application, wcf) tak le jana response generate kerna aur result wapas client ko bhejna.*

*One thing you must always keep in mind ki window application mostly single threaded banaye honge, who execute bhi waise he honge, but ASP.NET, WCF applications to even tumne single thread banaya ho, magar jab multiple request aayegi, then request ko process alag alag thread se kiya jata hai (in build multithreading). To ager kahi static use ker rahe to dhyan rakha who sare user ke beech share ho raha, aur baki variable, user specific honge.Kitne thread banne hai yeh IIS ke settings me hota hai.*

*Now lot of questions, computer ko kaise pata ki ASP.NET ki request hai ya WCF ke? ASP.NET me he multiple tarah ke request hoti hai, page(.aspx), ya image, ya javascript file, how does computer differentiates? How does authentication works so on..*

## How does IIS handles request

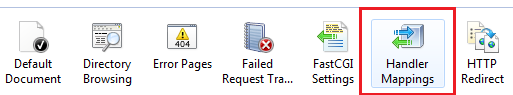
*Normally jab IIS installed ho aur tum Visual Studio install karo to who bahut sare task khud ker deta hai, jo pata nahi lagte. Magar kisi machine me Visual Studio install kerne ke baad IIS install karo to dekh sakte ho ki ASP.NET ko IIS nahi janta!*

*Microsoft ne IIS ko is way me likha hai ki uski abilities to kabhi bhi enhance ker sake. Kis type ki application ko kaise handle kerna hai yeh core IIS ka part nahi hai, balki additional dlls ka kaam hai, jinko ISAPI bolte hai (IIS ki API).*

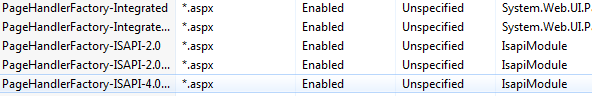
*C:\Windows\Microsoft.NET\Framework\v4.0.30319 aur ager 64 bit machine hai to C:\Windows\Microsoft.NET\Framework64\v4.0.30319 (ya fir 2.0XXXX me, not in 3.0 or 3.5) me dekho ek dll milegi* ***“aspnet\_isapi.dll****”. Yeh dll IIS ko ASP.NET request process kerne capable banati hai. To ager VS ke baad IIS install karo to ek exe “****aspnet\_regiis.exe****” execute kerni padti hai yeh uper wali ISAPI ko IIS me install ker deta hai. Itni kahani se ek baat clear hui, ki IIS request process kerne ke liye ISAPI per depend kerta hai.*

When a Web server receives a request, it examines the file-name extension of the requested file, determines which ISAPI extension should handle the request, and then passes the request to the appropriate ISAPI extension. ASP.NET handles file name extensions that have been mapped to it, such as .aspx, .ascx, .ashx, and .asmx.

*ISAPI se ASP.NET ke pass request aa gayi, but ab ASP.NET me bhi to .aspx(page), .ascx (control), .asmx (Web service) different different tarah se execute hote hai, inko handle kerne ke liye “Handler” hote hai, .ASPX ka alag, .ASMX ka alag, check kerne ke liye see below in your machine.*

*At IIS level*

*You can see/ edit/ add*

**

*So* ***handler*** *ka kaam hai request execute kerna, ager .aspx ki request hai to page kaise execute hoga yeh aspx handler ko pata hona chaeye. To ager tumko kabhi mood ko mai \*.fun se page bana ker dekhu, to tumko .fun ko handle kerne wala ek handler likh ker IIS me dalna padega.*

**If a file name extension has not been mapped to ASP.NET, ASP.NET will not receive the request. IIS 5.0** *me .htm file ASP.NET se mapped nahi thi to unper aa koi authorization check nahi ker sakte the, kyuki authorization check kerna ka power asp.net me tha.Ab request isliye execute hoti thi kyuki IIS me ek aur ISAPI, html content ke liye hai, to bhale he tum .htm ki request asp.net se karo, IIS usko asp.net request nahi manta tha aur dusre ISAPI se process kara ker response deta tha.*

*Yeh baat to ho gayi handler ke,ek aur related concept hai Modules (HTTPModule). Is dono ko samjhne ke liye let’s take example of hospital. Hospital is our IIS. Hospital me reception, per ilaj(request process) nahi ho raha. Patient(request) dekh ker bola ki is patient to le ker emergency me jao, ya OPD me, ya Cardio me. Alag alag ISAPI( departments in hospital) per bhej rahe. Ab koi deprtment na ho us hospital me (ISAPI) missing to reception he bolenge beta nikal lo.*

*Ab patient ek department(ISAPI) per aa gaya, udher alag alag specialist hain (handlers), but isse pehle ki actual Dr dekhta hai, kuch chintu mintu compunders, junior dr aate hai, who patient ko enrich kerte hai, like uski report theek se lagana, koi test missing hai to who kerwana, ie they assist by putting more info. There junior dr’s are HTTPModules. They are not any lesser than Handler, but they just put more infor in request and response (not handle it). Fir patient(request) to Dr dekhta hai (handler process kerta hai), final outcome (response) aaya, ab jab response wapas ja raha tab fir se junior dr se mil sakta hai and can ask ki dawa kab leni hai , kaise leni hai, ie HTTP module request to enrich ke he thi, who response per bhi to add, modify ker sakta hai. Request Validate kerna, Session manage kerna are few of Module work*

*So in IIS any request is processed by one handler at a time, but multiple modules can modify request response.*

*Now how does Handler execute a request that’s a separate detail, which you have to wait for few weeks to be able to understand. For now think ki first ASP.NET ka environment host hota usme application load hoti hai aur page ka code (init, load….) execute hota hai.(Page wala part next notes me, but environment wala kuch days baad bataunga)*

*Before we go in page life cycle and global event, let’s read about Session.*

***Session***: Session in ASP.NET has two related meaning; Session is started as soon as you put your first request to server to explicitly closing session or session time out. Secondly Session term is also used for Session State management.

*Now if you have no idea what did these line say, no issues, let’s take on them one by one. To understand, remember you read that HTTP is stateless. This means that a Web server treats each HTTP request for a page as an independent request. The server retains no knowledge of variable values that were used during previous requests. It means ki you asked for a page, Server created object of that page class, did processing ,sent you response and object is cleaned off. Next time if you post back, page object will be recreated and request will be processed. As object is recreated, and no additional information is stored (Stateless), we have no way to know what user did in just 5 sec ago. Moreover, because no information is stored, if your application requires authentication, then in every trip, you will have to provide credentials!. Because application does not recognize you.*

ASP.NET session state identifies requests from the same browser during a limited time window as a session. So this cover time duration part, that until your session expire you are identified. Now how do it identifies you, when a request comes to server, it checks for session key, if it does not have any, server generates one and send it back with response, client is supposed to send session key with every subsequent request, otherwise he will be treated as new client every time*. This is the way people are identified in clubs; when you enter first they take fee and stamp on your hand, that stamp is symbol of you being identified by club, for specific duration. With that stamp, you would be allowed to reenter club for a specific duration. If you wash off that stamp, you will again ask for fee.*

*Along with time duration, Session state management says that, every user will get memory where any data relevant can be stored for cross trip. This is like your gym locker, you go to gym and can store you items in locker, next day when you visit, and identified by your session key IIS by help of modules will map you with data in your session. So inspite of HTTP being Stateless, we go to server we store some data specific to user and clean up everything, next when we go to server, by help of our session id, server can see what all data we have stored in session and that data will be accessible for you next processing.*

*Think of customer care, you know that you will get different person over the phone every time, they would not remember you, so this set up is stateless. But if you have to repeat you problem every time you call their call center, it will be frustrating, so they create log where they put all the discussion that happens. This is session state management. You close your call, Customer care executive need not to remember you, STATELESS, but because of that log, next time you call another executive can pick from log what you have already told and what has already happened. Please do not confuse this log of activity on IIS, this is just a example to tell how HTTP is stateless, but we might need to remember some data.*

*So in short session is 1. Duration 2. Storage, It is duration as it gets timeout and it is storage as we can store info. Even if you do’t store anything in session, still you have session created and a session id delivered to client and a timeout associated with session.*

*In next note we will see page cycle and start with actual asp.net implementation.*

HTML

As you must know that HTML stands for Hypertext markup language, is you remember Web Basic notes we talked about hyper text**. Hypertext** is text displayed on a [computer display](http://en.wikipedia.org/wiki/Computer_display) or other electronic devices with references ([hyperlinks](http://en.wikipedia.org/wiki/Hyperlinks)) to other text which the reader can immediately access, the hypertext pages are interconnected by hyperlinks, typically activated by a mouse click, key press sequence or by touching the screen. Apart from text, hypertext is sometimes used to describe tables, images and other presentational [content forms](http://en.wikipedia.org/wiki/Content_form) with hyperlinks. Hypertext is the underlying concept defining the structure of the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web).

**Markup** languages are designed for the processing, definition and presentation of [text](http://www.webopedia.com/TERM/T/text.html). The [language](http://www.webopedia.com/TERM/L/language.html) specifies [code](http://www.webopedia.com/TERM/C/code.html) for formatting, both the layout and style, within a [text file](http://www.webopedia.com/TERM/T/text_file.html). The code used to specify the formatting is called [tags](http://www.webopedia.com/TERM/T/tag.html).

HTML is way of representing web pages, as we read in Markup language definition above HTML is made of tags which is used with text and other content (image, sound..) to display it to user and allow to interact. When server returns HTML as string, browser reads it and paints the UI.

*To speed up things we will study about few things and have one example for all of those, then again few other thing then one example for those.*

HTML page is made of 3 major sections

HTML, HEAD and BODY

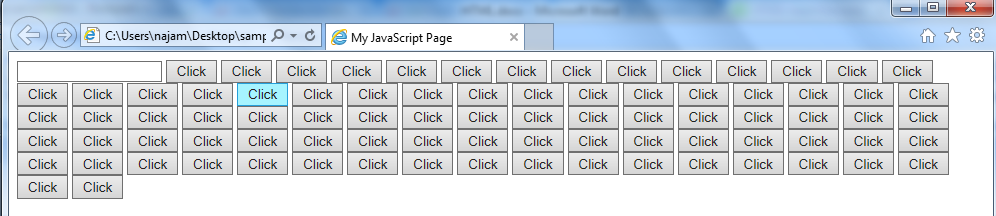
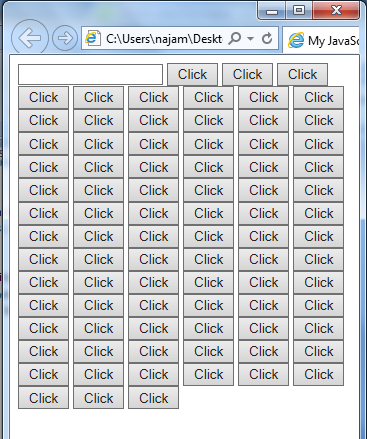
* **HTML Tag:** the HTML tags go at the start and at the end of document. This tells your browser that what falls between them should be defined as HTML. (Note there are few things that are written outside HTML tags like DocType, but don’t think about them right now)
* **Head Tag:** This contains things that are not displayed but effect rendering of page and page’s behavior. Two most important things CSS and Script are defined in Head tag.
* **Body**: Anything that you want to display should be written between body tags.

In html every tag should (read **must** ☺ )be closed. Tags is opened like <tagName> and closed with </tagName>, some tag can also be open and closed as <tagName property1=”value” property2=”value2” />. Tags can be nested, i.e. a tag inside another tag, only thing you should take care is order of closing tag must be reverse of opening tag. In simpler way think it as stack, so more recent tag opened will be closed first.

Let’s start with each of 3 major tags and start filling them with other tags.

**Title**: You page should have a title, that is displayed in tab of your browser, title goes in head part of page, please focus that title does not display anything on page, but only on browser window. You must give title to your pages. (It will throw any error, if you don’t give, it’s just stupid not give title)

By default HTML layout is kind of “flow layout”, by this I mean, browser will keep displaying elements in same row (please try not to relate with tr of table, by default there is no table) until width is full and if it finds that space in the row is smaller than required for next element it will put next element in next row. Moreover if you drag and decrease the size of the window, this repaints things. This is useful but need you to give a thought before you create page. We prefer our page to look good on all screen size.



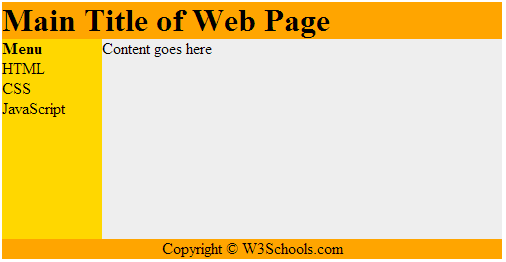
So you see that just putting elements in page would not give up the page layout we want to, we need to have control where an element is displayed. For that let’s check out our layout options.

First thing not all element let browser put them in same row as previous element, block level elements normally start (and end) with a new line when displayed in a browser. A block-level element starts on a new line and stretches out to the left and right as far as it can Examples: <h1>, <p>, <ul>, <table>. Inline elements are normally displayed without starting a new line.

When creating a page you start with layout, by layout I mean the way in which things will be arranged in page. There are 2 ways to do it. Using tables or using div. Tables are very clear by their name, with the div tag, you can group large sections of HTML elements together. One thing you must try to understand is that both table and div are to define layout of other controls. Here is now a catch, you will find lot of code using table (because it is simpler) for layout, but actually div is supposed to do that. It will be best if you learn making layout with div, if you find it difficult you can always create layouts by table. Defining layout by table is straight forward, you define a page level table then as per design of page you keep creating controls and new tables is needed.

Divs require you to have good understanding **Position** from CSS notes. If don’t remember Position, I would recommend to stop here and read about it.

Let’s say we want to create a layout as image below, a very common layout.. . *As you most probably will be reading this in your bed, without your laptop with you, so please focus on each and every element of html, css in this note. Also please note that lot of time I will write inline css, do not follow it as your habbit, always prefer to write css in external file or in style section.*



Here is how to do it by table

**Layout HTML**

<html>

<body>

<table style="width: 500px;" cellpadding="0" cellspacing="0">

<tr>

<td colspan="2" style="background-color: #FFA500;">

<h1 style="margin: 0; padding: 0;">Main Title of Web Page</h1>

</td>

</tr>

<tr>

<td style="background-color: #FFD700; width: 100px; vertical-align: top;">

<b>Menu</b><br>

HTML<br>

CSS<br>

JavaScript

</td>

<td style="background-color: #eeeeee; height: 200px; width: 400px; vertical-align: top;">

Content goes here

</td>

</tr>

<tr>

<td colspan="2" style="background-color: #FFA500; text-align: center;">

Copyright © W3Schools.com

</td>

</tr>

</table>

</body>

</html>

Let’s discuss above html, first start with table tag, table is basically for showing data in tabular layout, however as told above, people (count me in it☹), misuse it to define layout of page. Tables are defined with the **<table>** tag. It is has rows **<tr>** and rows have columns **<td>.** Each area inside a td is also called cell and it contains other element(s). The <td> elements can contain all sorts of HTML elements like text, images, lists, other tables, etc. You are not bound to have equal number of <td> in all rows, however if you don’t do this you will two options either have “empty” spaces, or tell row(s) with lesser tds to fill up space (Colspan)

See the css here is used to give back color to td, we could have written with each td, but again we wrote it once and used it in all tds. *You should keep css note open along with this.*

<html>

<head>

<style>

td

{

background-color: red;

}

</style>

</head>

<body>

<table style="width: 500px;" border="4">

<tr>

Only one td in a row

<td>row 1 Col1</td>

</tr>

<tr>

Two tds in a row

<td>row 2 Col1</td>

<td>row 2 Col2</td>

</tr>

<tr>

<td colspan="2">row 3 Col1</td>

</tr>

See this part doesn’t have red background; this is because row 1 has just one column, and this space is inside table but not belongs to any col.

</table>

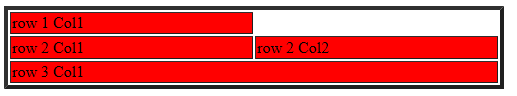
Colspan, tell that this td should expand coving space of two tds.

</body>

</html>

Output of this will be

See colspan 2 said to expand and take place of 2 tds. Hence it filled the row. If in case row 2 would have 3 col and we would have still said colspan 2 then, this col would have expanded to 2 col and 3rd will be empty as in first row



This example should have given you fair idea of what a table does and how tr and td play along. Now please go back to above **layout html** and try to understand each element. Once we have divided our page in the major part, we can have as many tables inside them to define inner layout. In place of

<td style="background-color: #FFD700; width: 100px; vertical-align: top;">

Br breaks line, anything after it will be in new line

<b>Menu</b><br>

HTML<br>

CSS<br>

JavaScript

</td>

We could have created a new table in this td and added 4 rows (tr) with one cell (td) in each row. ***Try it***

**Quiz for you**: See width part, how tds will arranged??

See width is 100px

<table style="width: 500px;" border="4">

<tr>

<td width="100px">row 1 Col1</td>

See width is 20px

</tr>

<tr>

<td width="20px">row 2 Col1</td>

<td width="480px">row 2 Col2</td>

</tr>

<tr>

<td colspan="2">row 3 Col1</td>

</tr>

</table>

So hopefully you would have clear idea of tables. Now let us our focus to second option of layouts which is by div.

<html>

<body>

<div id="container" style="width: 500px">

<div id="header" style="background-color: #FFA500;">

<h1 style="margin-bottom: 0;">Main Title of Web Page</h1>

</div>

<div id="menu" style="background-color: #FFD700; height: 200px; width: 100px; float: left;">

<b>Menu</b><br>

HTML<br>

CSS<br>

JavaScript</div>

<div id="content" style="background-color: #EEEEEE; height: 200px; width: 400px;

float: left;">Content goes here</div>

<div id="footer" style="background-color: #FFA500; clear: both; text-align: center;">

Copyright © W3Schools.com</div>

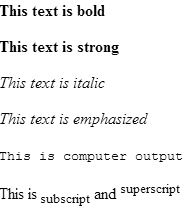
</div>

</body>

</html>

Focus of float property, also see the difference here, unlike table where table contain rows and row contain columns, here every div is independent to other, they are just placed as required.

*Actually that’s it for HTML (if you know CSS). As we need to finish off these quickly to work on your asp.net website, I won’t cover obvious things like, button, textbox, anchor (hyperlink), fieldset (group box) kind of stuff. We will see them as needed, but to be safe side I will put one line for each here. You need not worry of them; they don’t have any concept with them as in case of table and divs. Other than that, it is how much you practice, for that I am pasting few layouts in end of this note, please try to create then, by div (or atleast table). Solution of these will also be pasted in this note. As always if you need more detail on any ……….*

**Text fomartting in HTML:**

<p><b>This text is bold</b></p>

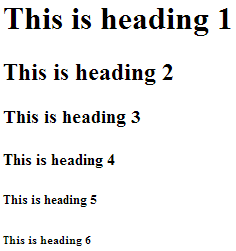
<p><strong>This text is strong</strong></p>

<p><i>This text is italic</i></p>

<p><em>This text is emphasized</em></p>

<p><code>This is computer output</code></p>

<p>This is<sub> subscript</sub> and <sup>superscript</sup></p>

**Headings**: Headings are defined with the <h1> to <h6> tags. Size decreases from H1 to H6

<html>

<body>

<h1>This is heading 1</h1>

<h2>This is heading 2</h2>

<h3>This is heading 3</h3>

<h4>This is heading 4</h4>

<h5>This is heading 5</h5>

<h6>This is heading 6</h6>

</body>

</html>

You can also write simple text and put your css to do this, but when its already in html tags we should use it.

**Paragraphs:**  Shows a text as a paragraph with row break after it.

<p>Don't forget to close your HTML tags!</p>

**Textbox:** For text box, value property contains text value <input type="text" name="FirstName" value="Mickey"/>

**Button:** <input type="button" value="click"/> or <input type="submit" value="Submit"> (to postback)

**Anchor:** <a href="http://www.google.com">Visit google!</a>, href part is actual URL and text is displayed. URL can be relative, i.e. from inside the current application or to putside the application as in this example.

**Image:** <img src="smiley.gif" alt="Smiley face" height="42" width="42"> src again can be relative

**Password:** <input type="password" name="pwd">, textbox which show bullets in place of letters.

**Span:** The difference between [span](http://www.htmldog.com/reference/htmltags/span/) and [div](http://www.htmldog.com/reference/htmltags/div/) is that a [span](http://www.htmldog.com/reference/htmltags/span/) element is in-line and usually used for a small chunk of HTML inside a line (such as inside a paragraph) whereas a [div](http://www.htmldog.com/reference/htmltags/div/) (division) element is block-line (which is basically equivalent to having a line-break before and after it) and used to group larger chunks of code.

<p>My mother has <span style="color:blue">blue</span> eyes.</p>

The <span> tag is used to group inline-elements in a document. The <span> tag provides no visual change by itself.

**Checkbox:** <input type="checkbox" name="vehicle" value="Bike">I have a bike<br>

<input type="checkbox" name="vehicle" value="Car">I have a car

**Radio button:** <input type="radio" name="sex" value="male">Male<br>

<input type="radio" name="sex" value="female">Female

**Dropdown:**

<select>

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

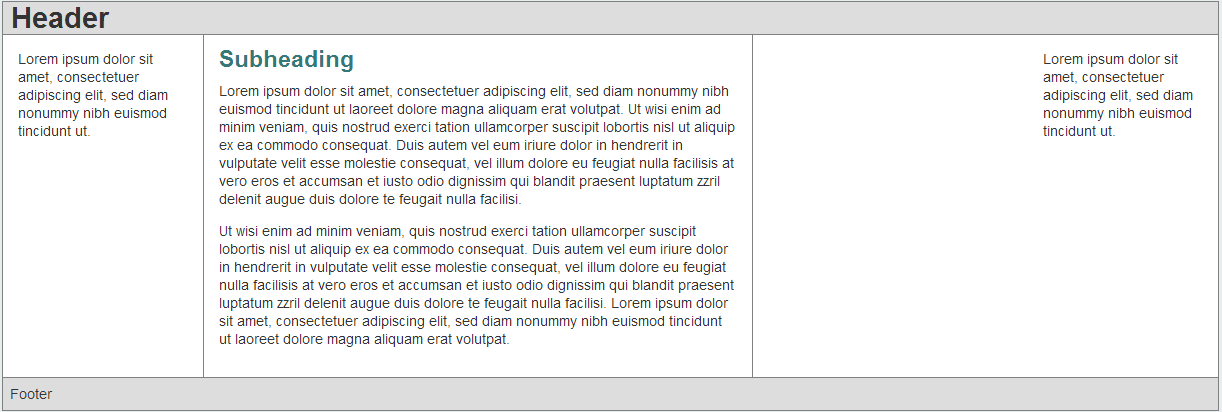
<option value="opel">Opel</option>

<option value="audi">Audi</option>

</select>

Exercise:

1. Please create following layout, see text in last cell, your text should be displayed as such.



1. Please create following layout



Solution:

#container

{

width: 90%;

margin: 10px auto;

background-color: #fff;

color: #333;

border: 1px solid gray;

line-height: 130%;

}

#top

{

padding: .5em;

background-color: #ddd;

border-bottom: 1px solid gray;

}

#top h1

{

padding: 0;

margin: 0;

}

#leftnav

{

float: left;

width: 160px;

margin: 0;

padding: 1em;

}

#rightnav

{

float: right;

width: 160px;

margin: 0;

padding: 1em;

}

#content

{

margin-left: 200px;

border-left: 1px solid gray;

margin-right: 200px;

border-right: 1px solid gray;

padding: 1em;

max-width: 36em;

}

#footer

{

clear: both;

margin: 0;

padding: .5em;

color: #333;

background-color: #ddd;

border-top: 1px solid gray;

}

#leftnav p, #rightnav p { margin: 0 0 1em 0; }

#content h2 { margin: 0 0 .5em 0; }

<div id="container">

<div id="top">

<h1>

Header</h1>

</div>

<div id="leftnav">

<p>

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh

euismod tincidunt ut.

</p>

</div>

<div id="rightnav">

<p>

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh

euismod tincidunt ut.

</p>

</div>

<div id="content">

<h2>

Subheading</h2>

<p>

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh

euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad

minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip

ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate

velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero

eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit

augue duis dolore te feugait nulla facilisi.

</p>

<p>

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis

nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit

in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis

at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril

delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet,

consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet

dolore magna aliquam erat volutpat.

</p>

</div>

<div id="footer">

Footer

</div>

</div>

1. Solution of problem 2

<html>

<head>

<style>

#header,

#main,

#sidebar,

#footer {

display:inline;

position:relative;

float:left;

background-color:#eee;

}

#header,

#footer {

width:100%;

height:60px;

}

#header {

margin-bottom:2%;

}

#footer {

margin-top:2%;

}

#main {

width:68%;

height:50%;

margin-right:2%;

}

#sidebar {

width:30%;

height:50%;

}

</style>

</head>

<body>

<div id="header">Header</div>

<div id="main">Main</div>

<div id="sidebar">Sidebar</div>

<div id="footer">Footer</div>

</body>

</html>

CSS

You have fair amount of understanding how HTML works, server send standard syntax as string (html) and browser understand it and paints a UI for you. As any real world object UI components have 2 part, core structure and “look and feel” part of it. For example you have button and you paint it blue, red…

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the [layout](http://en.wikipedia.org/wiki/Page_layout), [colors](http://en.wikipedia.org/wiki/Color), and [fonts](http://en.wikipedia.org/wiki/Typeface). Whereas the HTML is the *meaning* or *content*, the style sheet (CSS) is the *presentation* of that document. You have to understand that HTML alone can survive, but CSS need HTML. You might also think that why we need CSS at first place, we can write html and include tags for “presentation” for example

<p style="color:Blue; font-size:large"> This is text</p>

People who created HTML and CSS could have created HTML syntax as

<p><color>Blue<font-size>large</font-size>This is text</color> </p> or something similar, but this does not look clear, right?

This is one reason CSS was created. It cleans up our HTML file. All style related things are moved to separate file or section. This also enables us to reuse the style over multiple HTML elements without retyping whole style.

Now before we actually start learning CSS, let’s look at few qualities of CSS. First you must keep in mind that CSS can be put at places in HTML.

**In-line styles** are put into the HTML tags using the style attribute.

<p style="color: red">text</p>

**Internal Styles**

Embedded, or internal, styles are used for the whole page. Inside the [head](http://www.htmldog.com/reference/htmltags/head/) element, the [style](http://www.htmldog.com/reference/htmltags/style/) tags surround all of the styles for the page.

<html>

<head>

<title>CSS Example</title>

<style>

.paraStyle {

color: red;

}

</style>

</head>

<body>

<p class="parastyle">Sample text</p>

</body>

</html>

**External Styles**

External styles are used for the whole, multiple-page website. There is a *separate CSS file*, which will simply look like

P { color: red;

}

a { color: blue;

}

And will be included in html page like

<html>

<head>

<title>CSS Example</title>

<link rel="stylesheet" href="style.css">

Now you have seen 3 ways/ places of putting style in HTML. You might be wondering what will happen if you put different styles in two or more places for same element, like you might put inline style as well as External Styles for a same element. CSS has priority associated with each declaration. Following table shows you that priority, you need not remember this.

|  |  |  |
| --- | --- | --- |
| High Priority | CSS Source Type | Description |
| 1 | User defined | Most browsers have the accessibility feature: a user defined CSS |
| 2 | Inline | A style applied to an HTML element via HTML ‘style’ property |
| 3 | Media Type | A property definition applies to all media types, unless a media specific CSS defined |
| 4 | Importance | The ‘!important’ value overwrites the previous priority types |
| 5 | Selector specificity | A specific contextual selector (#heading p) overwrites generic definition |
| 6 | Rule order | Last rule declaration has a higher priority |
| 7 | Parent inheritance | If a property is not specified, it is inherited from a parent element |
| 8 | CSS property definition in HTML document | CSS rule or CSS inline style overwrites a default browser value |
| 9 | Browser default | The lowest priority: browser default value is determined by W3C initial value specifications |

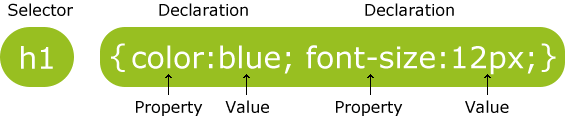
CSS can also allow the same page to be presented in different styles for different rendering methods, such as on-screen, in print.  It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. So if you want to allow users to set their own color choice (template) CSS is your friend.

Let’s start learning CSS with first taking on basic structure of a CSS style. As we know CSS stands for Cascading Style Sheets, forget cascading for a moment focus in Style sheet. Like in HTML one tag like <input> or <p> specifies one element of complete page, CSS is made of style rules. A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block.

Selectors are used to declare which part of the markup a style applies to, a kind of match expression. Selectors may apply to:

* [elements](http://en.wikipedia.org/wiki/HTML_element) of a specific type, e.g. the second level headers [h2](http://en.wikipedia.org/wiki/HTML_element#Basic_text)
* elements specified by [attribute](http://en.wikipedia.org/wiki/HTML_attribute), in particular, like
  + id: an identifier unique to the document
  + class
* elements depending on how they are placed relative to, or nested within

Classes and IDs are case-sensitive, start with letters, and can include alphanumeric characters and underscores. Okay, did it bounced over the head? ☺.

In html you write tags, right? CSS style is to decorate (effect look and format of these elements). If style has to be applied in html tags, then you need a mechanism to relate a particular html element/tag to specific style. To do this, i.e. specify on which element what particular style has to be applied, styles use selectors. As their name suggest selector is way to target html element. Once you have selectors for each selector there are “properties” inside curly brackets, which simply take the form of words such as [color](http://www.htmldog.com/reference/cssproperties/color/), [font-weight](http://www.htmldog.com/reference/cssproperties/font-weight/) or [background-color](http://www.htmldog.com/reference/cssproperties/background-color/). A value is given to the property following a colon (NOT an “equals” sign) and semi-colons separate the properties. So selector, properties and value make up one style. Collection of style is CSS sheet.

So you know that CSS selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more.

There are lots of ways to write selectors, but most common 3 are mentioned above, by element/tag type like you can say

P { color: red;

}

Here this style says set color red for all paragraph elements.

Second way is by attribute, you can write it in so many ways that I don’t want to list them here, but just for you to see them following is list, but be advised that you need not go through all this; we will concentrate only on 2 most used attributes, Id and Class.

|  |  |  |
| --- | --- | --- |
| Pattern | Meaning |  |
| E | An element of type E |  |
| E:link | an E element being the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited |  |
| E:active | an E element during certain user actions |  |
| E::first-line | the first formatted line of an E element |  |
| E::first-letter | the first formatted letter of an E element |  |
| E.warning | an E element whose class is "warning" (the document language specifies how class is determined). |  |
| E#myid | an E element with ID equal to "myid". |  |
| E F | an F element descendant of an E element |  |
| \* | Any element |  |
| E[foo] | an E element with a "foo" attribute |  |
| E[foo="bar"] | an E element whose "foo" attribute value is exactly equal to "bar" |  |
| E[foo~="bar"] | an E element whose "foo" attribute value is a list of whitespace-separated values, one of which is exactly equal to "bar" |  |
| E[foo|="en"] | an E element whose "foo" attribute has a hyphen-separated list of values beginning (from the left) with "en" |  |
| E:first-child | an E element, first child of its parent |  |
| E:lang(fr) | an element of type E in language "fr" (the document language specifies how language is determined) |  |
| E::before | generated content before an E element |  |
| E::after | generated content after an E element |  |
| E > F | an F element child of an E element |  |
| E + F | an F element immediately preceded by an E element |  |
| E[foo^="bar"] | an E element whose "foo" attribute value begins exactly with the string "bar" |  |
| E[foo$="bar"] | an E element whose "foo" attribute value ends exactly with the string "bar" |  |
| E[foo\*="bar"] | an E element whose "foo" attribute value contains the substring "bar" |  |
| E:root | an E element, root of the document |  |
| E:nth-child(n) | an E element, the n-th child of its parent |  |
| E:nth-last-child(n) | an E element, the n-th child of its parent, counting from the last one |  |
| E:nth-of-type(n) | an E element, the n-th sibling of its type |  |
| E:nth-last-of-type(n) | an E element, the n-th sibling of its type, counting from the last one |  |
| E:last-child | an E element, last child of its parent |  |
| E:first-of-type | an E element, first sibling of its type |  |
| E:last-of-type | an E element, last sibling of its type |  |
| E:only-child | an E element, only child of its parent |  |
| E:only-of-type | an E element, only sibling of its type |  |
| E:empty | an E element that has no children (including text nodes) |  |
| E:target | an E element being the target of the referring URI |  |
| E:enabled | a user interface element E which is enabled |  |
| E:disabled | a user interface element E which is disabled |  |
| E:checked | a user interface element E which is checked (for instance a radio-button or checkbox) |  |
| E:not(s) | an E element that does not match simple selector s |  |
| E ~ F | an F element preceded by an E element |  |

So for selector based on Id we write style as #<id> like

<style type="text/css">

#xyz

{

color: red;

}

</style>

Where html is <p id=xyz style="color: blue"> test </p>. Please focus that any element with Id xyz (only one element can have an id, duplicate ids not allowed in html) will get this style

Other common way based on attribute (and most used one) is based on class, most of html elements have attribute called class, you create a style and give it a name, that name is set in class attribute of html tag(s). In style class name is started with dot (.)

.center

{

text-align: center;

color: red;

}

Apply this style on any element that you want show text center aligned and color red. Now these ways of specifying selectors can be mixed, like if you write

p.center

{

text-align: center;

color: red;

}

Now P says to apply on Paragraph elements only, .center further filter says apply this sheet only on paragraph element having class center.

Other than mixing of selectors, we can group them, i.e. if we want to specify same set of properties with same value for different selectors we write them once

h1, h2, p

{

text-align: center;

color: red;

}

Now in html if you create h1 or h2 or p all will get same style. Same selector can be specified multiple times in sheet, net resultant style applied will be collection of all properties. In case of same property repeated multiple times or in case of conflict, they will be resolved on basis of priority (See first table).

Like after writing above property you want to set background color of h1 then you can write

h1

{

background-color:Gray ;

}

Net style applied on H1 tag will be combination of two styles. Before we go further, let’s take on 2 other very important facts about CSS. These two are browser style and inheritance. Every browser has some inbuilt CSS, which it applies on html element, this is the reason why you see thing differently on Chrome and IE. If you want to check this create html page without specifying any CSS at all and check page in both browsers. So if you want constant look and feel you must specify some CSS for all elements displayed in page. Another thing is inheritance, this is not inheritance what you learned in C#, but this is visual inheritance, i.e. child element will pickup CSS style of parent if it is not defined on it. This holds true for each property. So if you have background color set at parent and only color set at child, you will find both background-color and color properties at child level.  Properties that can be inherited are color, font, letter-spacing, line-height, list-style, text-align text-indent, text-transform, visibility, white-space and word-spacing. Properties that cannot be inherited are background, border, display, float and clear, height and width, margin, min- and max-height and -width, outline, overflow, padding, position, text-decoration, vertical-align and z-index. Last thing before we start exploring CSS again is F12 key on your keyboard, this is the best way to test your CSS.

Now start reading from <http://www.w3schools.com/css/css_background.asp> . Do practice them!

**CSS- Properties**

**Back ground**

CSS background properties are used to define the background effects of an element.

body

{

background-color: #b0c4de;

}

The background-color property specifies the background color of an element.

body

{

background-image: url("background.jpg");

}

By default, the background-image property repeats an image both horizontally and vertically. Like image in right.

body

{

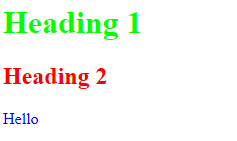
background-image: url("gradient2.png");

background-repeat: repeat-x;

}

If the image is repeated only horizontally (repeat-x), or you want only vertically (repeat-y). Have you seen borders with images or background image, normally they are small image, repeated as required. Showing the image only once is specified by the background-repeat property:

background-repeat: no-repeat

**The color property** is used to set the color of the text.

<html>

<head>

<style>

body { color: blue; }

h1 { color: #00ff00; }

h2 { color: rgb(255,0,0); }

</style>

</script>

</head>

<body>

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<div>Hello</div>

</body>

</html>

See H1 takes hex code (greenish), you can also provide color by rgb or color name. One more thing div(hello) does not have any style defined yet it is blue!, how?? It inherited from body

**Text Alignment:** The text-align property is used to set the horizontal alignment of a text. Text can be centered, or aligned to the left or right, or justified.When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight. To set the vertical alignment of an element (vertical-align)

<html>

<head>

 <style>

h1 { text-align: center; }

p.rightText { text-align: right; }

</style>

</head>

<body>

<h1> CSS text-align Example</h1>

<p class="rightText"> Right Text</p>

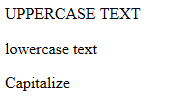
<p>Left text</p>

</body>

</html>

**Text Transformation**

<html>

<head>

<style>

p.uppercase {text-transform:uppercase;}

p.lowercase {text-transform:lowercase;}

p.capitalize {text-transform:capitalize;}

</style>

</head>

<body>

<p class="uppercase">Uppercase text</p>

<p class="lowercase">Lowercase Text</p>

<p class="capitalize">Capitalize</p>

</body>

</html>

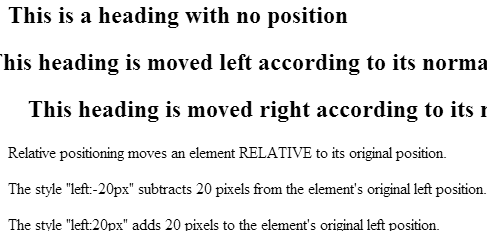
**Position (Very Very Important)** The CSS positioning properties allow you to position an element. Elements can be positioned using the top, bottom, left, and right properties. However, these properties will not work unless the position property is set first. They also work differently depending on the positioning method.

* Static is the default value and renders a box in the normal order of things, as they appear in the HTML.
* Relative is much like static but the box can be offset from its original position with the properties top, right, bottom and left.
* Absolute pulls a box out of the normal flow of the HTML and delivers it to a world all of its own. In this crazy little world, the absolute box can be placed anywhere on the page using top, right, bottom and left.
* Fixed behaves like absolute, but it will absolutely position a box in reference to the browser window as opposed to the web page, so fixed boxes should stay exactly where they are on the screen even when the page is scrolled.

Static HTML elements are positioned static by default. A static positioned element is always positioned according to the normal flow of the page. Static positioned elements are not affected by the top, bottom, left, and right properties.

Relative A relative positioned element is positioned relative to its normal position.

<html>

<head>

<style>

h2.pos\_left

{

position: relative;

left: -20px;

}

h2.pos\_right

{

position: relative;

left: 20px;

}

</style>

</head>

<body>

<h2>This is a heading with no position</h2>

<h2 class="pos\_left">This heading is moved left according to its normal position</h2>

<h2 class="pos\_right">This heading is moved right according to its normal position</h2>

<p>Relative positioning moves an element RELATIVE to its original position.</p>

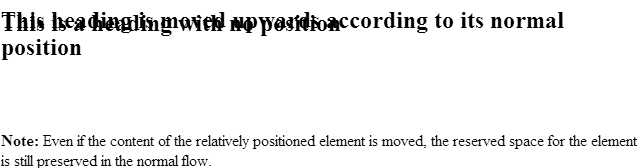
<p>The style "left:-20px" subtracts 20 pixels from the element's original left position.</p>

<p>The style "left:20px" adds 20 pixels to the element's original left position.</p>

</body>

</html>

The content of relatively positioned elements can be moved and overlap other elements, but the reserved space for the element is still preserved in the normal flow.

<html>

<head>

<style>

h2.pos\_top

{

position: relative;

top: -50px;

}

</style>

</head>

<body>

<h2>

This is a heading with no position</h2>

<h2 class="pos\_top">

This heading is moved upwards according to its normal position</h2>

<p>

<b>Note:</b> Even if the content of the relatively positioned element is moved,

the reserved space for the element is still preserved in the normal flow.</p>

</body>

</html>

Absolute An absolute position element is positioned relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>:

<html>

<head>

<style>

h2

{

position: absolute;

left: 10px;

top: 15px;

 background-color: green;

}

</style>

</head>

<body>

<h2>absolute position</h2>

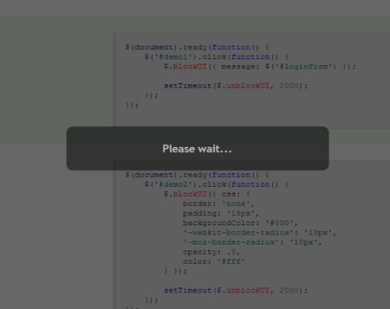
<p> With absolute positioning, an element can be placed anywhere on a page. The heading

below is placed 100px from the left of the page and 150px from the top of the page.</p>

</body>

</html>

Looking at example you might be wondering what is different between relative position and absolute position; both have made a text to be displayed over other. Well big difference is in relative position it is displayed as per top & left value but it also blocks space where it would have been displayed in normal flow. But in absolute it does not block space.

See this one, it places thing irrespective of their position in normal flow. You must have seen grey translucent background to disable page like in picture, where you page is covered with grey div and its all disabled. This is very common and used a lot in web development. So how to do it, Simple, use absolute position.

#disablingDiv

{

/\* Do not display it on entry \*/

display: none; /\* Display it on the layer with index 1001.

Make sure this is the highest z-index value

used by layers on that page \*/

z-index: 1001; /\* make it cover the whole screen \*/

position: absolute;

top: 0%;

left: 0%;

width: 100%;

height: 100%; /\* make it white but fully transparent \*/

background-color: white;

opacity: .00;

filter: alpha(opacity=00);

}

We place a div with id “disablingDiv” in html (above style is based on id, remember #id, this is not mandatory to name it like this, you can use any name or any way of writing css, just concentrate on what is written inside style) . Initially this div will be not be displayed. Above you have read two ways of hiding an element in css display and visible (refer them if you forgot them), we will use display as we don’t want it to take any space when it is not displayed. Now we have a div that is hidden in page. Let’s check out each line in style. First one says display: none. I.e. hide it. Second line is z-index :1001. You see page in two dimension length and width, but page actually has 3 dimension, you can put element over other element, that is denoted by z-index. This line says z-index 1001, mean put my this div on 1001th layer, this is just for safe side, even z-index-2 or 3 would have worked, but just be explicit and on safer side it is this large number. Postion:absolute is the main line which takes it out of normal flow and let it cover whole page, you should keep in mind that normal flow in html is that every element take required space and next element is placed next to it. Next four lines top:0% to height:100% are to cover whole page. Rest all is simple color and opacity. Once we have all set we can show hide this div whenever we want from javascript

document.getElementById('disablingDiv').style.display = 'block'; //show

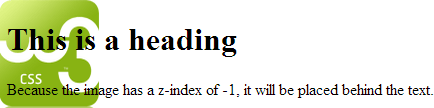
document.getElementById('disablingDiv').style.display = 'none'; // hide

position absolute is also used a lot when you have to show some div as popup or floating div kind of things.

<html>

<head>

<style>

 img

{

position: absolute;

left: 0px;

top: 0px;

z-index: -1;

}

</style>

</head>

<body>

<h1> This is a heading</h1>

<img src="w3css.gif" width="100" height="140" />

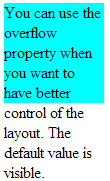
<p> Because the image has a z-index of -1, it will be placed behind the text.</p>

</body>

</html>

**Overflow:** This tells browser what to do when element’s content is bigger than size of element in html. Let say you have a div of 100px height and width and you wrote a big text in it. Possible values are **visible|hidden|scroll|auto.**

<html>

<head>

<style>

div.overFlowDemo

{

background-color: #00FFFF;

width: 100px;

height: 100px;

overflow: visible;

}

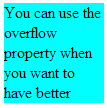
</style>

</head>

<body>

<div class="overFlowDemo">

You can use the overflow property when you want to have better control of the layout.

 The default value is visible.</div>

</body>

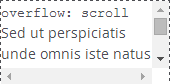
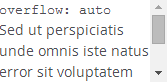
</html>

See div area is only blue area, still its text grows out of its boundary as we have set over flow as visible

If we set overflow as hidden then, see that text is cut to fit in div area.

If overflow:auto and overflow:scroll , Auto will only show a scrollbar when any content is clipped. Scroll will however

always show the scrollbar even if all content fits and you can’t scroll it.



JavaScript

Web is all about processing at server and showing result on user’s browsers. Although major business processing is done at server, it is desired to have some “logic” or validations to be done at client machine to save bandwidth and not keeping server’s resources busy unnecessarily. JavaScript was build for same reason. It is not fully fledged programming language, it has limitation on what it can do, like it not allowed to access file system. This is for security reason; you would not like to get your file and folders deleted when you open a webpage, as on web any kind of script be placed by mischievous peoples.

JavaScript (JS) is a [**dynamic**](http://en.wikipedia.org/wiki/Dynamic_programming_language) computer [programming language](http://en.wikipedia.org/wiki/Programming_language). It is most commonly used as part of [web browsers](http://en.wikipedia.org/wiki/Web_browser), whose implementations allow [client-side scripts](http://en.wikipedia.org/wiki/Client-side_scripting) to [interact with the user](http://en.wikipedia.org/wiki/User_interface), control the browser, communicate [asynchronously](http://en.wikipedia.org/wiki/Ajax_(programming)), and alter the [document content](http://en.wikipedia.org/wiki/Document_Object_Model) that is displayed*. It is also being used in server-side programming, game development and the creation of desktop and mobile applications.*

In above paragraph first thing that might not be clear is “dynamic computer programming language”. Dynamic language resolve lot of things at run time, rather than compile time, like your variable’s type is determined at run time. You can change behavior by adding new [code](http://en.wikipedia.org/wiki/Source_code), by extending [objects](http://en.wikipedia.org/wiki/Object_(computer_science)) and definitions. i.e. you can add properties to object at runtime!. You will be amazed to know that even C# has dynamic part in it. In .NET 4.0 dynamic keyword was added in C# for supporting dynamic programming.

JavaScript tutorials normally as any programming language start with variable and loop then other things, we won’t go that way. We will jump into scenarios and learn from that.

Let’s start with places where JavaScript can be written. This can be internal, external and inline. Internal JavaScript functions are written inside the script tag in HTML.

<html>

<head>

<title>My JavaScript Page</title>

<script>

(JavaScript code goes in here)

</script>

<body>

</body>

</head>

External JavaScript is a “.js” file contains javaScript function and variables and referenced in HTML. <script src="script.js"></script>

Src if file location on javascript file on server.

Inline is least used and preferred way of writing JavaScript, mostly we use it for showing alerts or opening pop, but be advised this is not limited to this.

Let’s start

**Window**: The window object represents an open window in a browser. This is almost (see almost) your browser. Lot of objects are inside this window object

* **window.document** Allows you to access all your HTML elements
* **window.history** Access information about browsing history
* **window.innerHeight:** window's height (NOT including toolbars and scrollbars)
* **window.innerWidth:** window's width (NOT including toolbars and scrollbars)
* **window.screen**: the total height of your screen, in pixels
* **window.navigator:** Information about browser
* **window.location:** URL of page, if you change this to another url, that page should be opened.

There are methods like Window.Open, Window.Close, window.print etc, but I won’t write them here, as you won’t remember all of that, until their applicability is defined.

**Document**: This object contains reference to your html document; all the elements are inside it. Functions like getElementById of this object is one of most used functions in JavaScript. getElementById takes id of an element in HTML page and returns you object (client side) for that element. Most common usage of that object is to validate value or change style.

**Variable**: In start of this notes, we talked that JavaScript is dynamic language. All the variables in JavaScript are defined by “var”. See no data type!! When you assign value for first time, data type is determined for the variable. So all of following is valid

var myumber = 42;  
var blnValid = true;  
var firstName = 'Chintoo';

See for string you can use single quotes or double quotes, unlike C# where only double quotes is allowed. In variable if you write var nymuber = 42;

Then you can write mynumber = “hello”; See data type is changed!! This is valid in javascript. For name of variable please stick to alphabets and number (first char should not be number).

You are all most set to dive into actual development, but let’s first have a look at events. Events are something that are of interest, like mouse click, web page is loaded, value of control is changes, key is pressed or form is submitted etc. You can write functions in javascript that can be called on such events.

Let’s start

1. Validate if a textbox is blank

<html>

<head>

<title>My JavaScript Page</title>

<script>

function validateTextBox() {

var textVal = document.getElementById("myTextBox").value;

if (textVal == "") {

alert("Field is blank");

}

}

</script>

<body>

<input type="text" id="myTextBox" />

</body>

</head>

We have created a textbox (in html it is input, even your asp.net text is resolved to this) and set its id to myTextBox. In script section of page we have written a function validateTextBox. In next line we are using getElementById of document object, we passed id of textbox in this.

textVal variable now has value of mytextBox. Then we check this value to blank and show alert. See double equal. Alert is a way to show popup messages in javascript.

All clear? Any confusion??.

*Dhyan kidher hai??? Khak samajh aaya, uper function ko call to kahi kiya he nahi.*

This function can be called as need, you may call this when focus is out of textbox, or submit button click.

<input type="text" id="myTextBox" onblur="validateTextBox();" />

Or create a button

<input type="button" id="mybutton" value="Click" onclick="validateTextBox();" />

Please stay focused on all part of code, there is no need to explain what does onclick means, but you must focus on syntax as JavaScript is case sensitive.

Let’s take on variant of above functions, this will help you more than looking at multiple examples.

Here textVal refer to myTestBox object.

function validateTextBox2() {

var textVal = document.getElementById("myTextBox");

if (textVal.value == "") {

alert("Field is blank");

}

}

In following variant, we are passing reference of textbox as parameter, please focus on this variable in html, this means current element, also note that as “this” is passed, this variant cannot be used on click of button.

function validateTextBox3(obj) {

if (obj.value == "") {

alert("Field is blank");

}

}

<input type="text" id="myTextBox" onblur="validateTextBox3(this);" /> .Onblur is called when focus is moved from an element, opposite is onfocus

If you want to pass parameter other than **this** you do it like following

function validateTextBox4(obj) {

var textVal = document.getElementById(obj);

if (textVal.value == "") {

alert("Field is blank");

}

}

<input type="button" id="mybutton" value="Click" onclick="validateTextBox4('myTextBox');" />

See we are passing id as string and in function we retrieve object.

This example was not complex, but you should read it again as it tells about possible ways of calling a function and accessing and element in html. This is all you will be doing 80% times in javascript.

Once you are very clear, let’s move to next examples. In above example you have validated for blank text, now if you want to change backcolor of such textbox all you have to do is add following line

document.getElementById("myTextBox").style.backgroundColor = "yellow";

Again this line be written in any(variant) way you like.

1. Open Popup window

function popitup(url) {

newwindow = window.open(url, "blank", 'height=200,width=150');

}

<input type="button" id="mybutton" value="Click" onclick="popitup('http://www.google.com');" />

That’s it! Remember window is top most object (almost) our browser window. Open is a function of that object, first parameter is url that we want to open. Second parameter tells about where to open this new page, options are

* \_blank - URL is loaded into a new window. This is default
* \_parent - URL is loaded into the parent frame
* \_self - URL replaces the current page

Third parameter can be any combination of

|  |  |
| --- | --- |
| fullscreen=yes|no|1|0 | Whether or not to display the browser in full-screen mode. Default is no. A window in full-screen mode must also be in theater mode. IE only |
| height=pixels | The height of the window. Min. value is 100 |
| left=pixels | The left position of the window. Negative values not allowed |
| location=yes|no|1|0 | Whether or not to display the address field. Opera only |
| menubar=yes|no|1|0 | Whether or not to display the menu bar |
| resizable=yes|no|1|0 | Whether or not the window is resizable. IE only |
| scrollbars=yes|no|1|0 | Whether or not to display scroll bars. IE, Firefox & Opera only |
| status=yes|no|1|0 | Whether or not to add a status bar |
| titlebar=yes|no|1|0 | Whether or not to display the title bar. Ignored unless the calling application is an HTML Application or a trusted dialog box |
| toolbar=yes|no|1|0 | Whether or not to display the browser toolbar. IE and Firefox only |
| top=pixels | The top position of the window. Negative values not allowed |
| width=pixels | The width of the window. Min. value is 100 |

You need not to learn this, just remember Window.Open is used and there are two other params and what is their functionality. Remember Google is always there to tell you exact syntax. Use your brain as directory. Please note window open work for separate page, you can show pop type UI, by displaying a div, setting its position absolute and graying rest of page!.

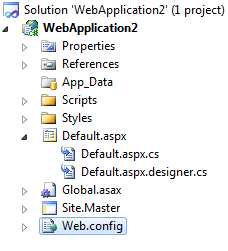
Now this is not modal dialog. Modal dialog is a way to show popup which does not allow working on base window until popup is closed. This is required a lot.

ASP.NET- Part 1

You need to quick pill to learn ASP.NET, so here it is. Very quickly we will go through ASP.NET at high level, you have read about HTML page, you can build your website from HTML page, in fact it is HTML only that goes to browser from server (yeah, along with CSS, JS and image.. stop nitpicking ☺). If we can create web site from HTML page (which you can create in notepad) why you need ASP.NET, PHP, JSP etc etc?? . HTML pages are kind of static, there content is fixed, with JS you add some dynamic behaviors that is on client side. If web site is not dynamic, it means you cannot show content in response to type of action taken by user, or even by user’s role, Preference etc. Most web applications you see today, they are dynamic, based on your role, your preference, location.. They change their content. Take one example of blog site. If you create in HTML only, you have to create a webpage (HTML page) for each article you write and in main page you will add link for that HTML page. How long you can manage such thing. What if you create a page that can display content and based on what user has asked for you, pick content from some storage (file, database, web service) and show that content in page. This is more manageable, reusable. If you focus to develop in this way you need to execute some code on server, because you want to generate HTML at server and send over to browser. PHP, ASP, ASP.NET, JSP all are framework that provide you facility to write code for server. ASP.NET is framework for web application development, coding is done in C# (or VB and lot of other .NET supported languages). So if coding is done in C# what does ASP.NET do? Well remember basics of Networks all the communication is done over TCP IP, over the sockets, you learned about lot of protocol, HTTP, HTTPS, WWW, ASP.NET takes care of all these. You don’t care about mapping request to user (there are 100’s of user at time, which user asked for what). So it’s like taking a level up from basic to make you relaxed about low level management and focus on business problem. This is exactly intent of all kind of framework; they let you focus on bigger problem rather than repetitive task.***I would highly recommend going though Web Basics notes.***

ASP (older version in 90’s to 2003 until .NET was created) code where written with HTML, code where separated from HTML by using special tags. This became messy as things grew. ASP.NET took more managed approach, kept HTML (\*.aspx) page separate from code file (\*.aspx.cs), provided config file and most importantly is based on events. Whenever ASP.NET processes a request, it goes through specific life cycle and at various stages it raises event, you write code that “hooks on” to those event. It’s like going into buffet, you are bounded by choices given by hotel, and still you have got enough. I don’t still get the idea of event based model of asp.net, I would assume you don’t like buffet☺. Never mind we will revisit this in little while.

There is lot to study in ASP.NET, but we will not go in concepts only, we will directly jump into hands on and learn concepts (in details) while learning about features. Let’s directly start with ASP.NET project structure and understand its part.

**Web.config**: This is configuration file in asp.net project. Anything that is written in config can be written in code, infact framework reads config file and set values of runtime objects. However config is used to make things configurable without need to recompile code. You might not see immediate benefit of this now, but when you develop website; you need not to get actual database name or production folder where you website will be hosted. You would like these things to be changeable without much effort. That is why such things are kept in configuration file.

**\*.Master**: You have learned about aspx page. Aspx result in html at runtime. All was good. But if you look at any web site, they have multiple pages, but lot of it kind of remains same. Biggest example is Header image, menus. One way is to copy and paste same html in all page and also put page specific HTML, this is not smart, if you have to change something, let’s say add one more menu, then in all pages need to be changed. Master page are used to managing such content of page. Technically this is called visual inheritance

**Global.asax:** The Global.asax, also known as the ASP.NET application file, is used to serve application-level and session-level events. The Global.asax file itself is configured so that any direct URL request for it is automatically rejected; external users cannot download or view the code written within it. This in simple terms it means you cannot access it from browser. There can be only one Global.asax file per application and it should be located in the application's root directory only.

In this file write code to handle Application and Session level events. For every page events like Load, init, Render are handled in the page right? but what if you want to execute some code when ever user access you site (session start) or leaves you site. Or maybe you want to execute some code when ever your web site start (before first request when ever site is hosted). This is the place for such things, Following are the events, read them you will understand them just looking at their name, and as always ask if you don’t understand.

**Application\_Start()** – This event raised when the application starts up, handles very first request after hosting.

**Session\_Start()** – This event raised for each time a new session begins, This is a good place to put code that is session-specific.

**Application\_Error()** – This event raised whenever an unhandled exception occurs in the application. This provides an opportunity to implement generic application-wide error handling.

**Session\_End()**– This event called when session of user ends.

**Application\_End()** – This event raised just before when web application ends.

**Application\_Disposed()** – This event fired after the web application is destroyed and this event is used to reclaim the memory it occupies.

Scripts and Style are very clear by name.

**Master Page**

Master pages are used for visual inheritance, ie if you have few pages that have few UI part in common, you can create master page for them. Any page specific UI, functionality will be in pages which are, in this context called as Content Pages.

**To add master page:**

Right click on project in solution explorer, Add new item -> Master Page.

**Understand default html of Master Page**

When **AutoEventWireup** is true, ASP.NET does not require that you explicitly bind event handlers to a page event such as [Load](http://msdn.microsoft.com/en-us/library/system.web.ui.control.load(v=vs.110).aspx).

Tells that page is master page

Code behind class name

Code behind file name

<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.master.cs" Inherits="WebApplication1.Site1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

<asp:ContentPlaceHolder ID="head" runat="server">

</asp:ContentPlaceHolder>

</head>

<body>

Place holder in head section for CSS and javascript, which will be filled by content page.

<form id="form1" runat="server">

<div>

<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">

</asp:ContentPlaceHolder>

</div>

Place holder in body section for html, which will be filled by content page.

</form>

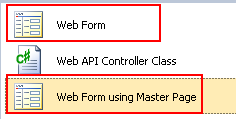
</body>

See form is part of master page

</html>

**Add Content Page**

Right click on project in solution explorer, Add new item -> Web Form using Master Page. (See you still can add normal page)

****When you add “Web Form using Master Page” Visual Studio ask you to select a master page, you can select the desired master page.

**Understand default html of Content Page (Only new tags explained)**

Master page file path (~/ means path is from main folder (root folder))

<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication1.WebForm1" %>

Include all the JavaScript reference, JavaScript code and CSS here; it will be merged will JS and CSS of master page.

<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">

</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">

</asp:Content>

Write the entire html (static or asp controls here, final html will be placed at Master page’s body content place holder.

* You can have any number of content placeholders (<, >, = 2, normally we have 2, but depends on requirement) in your Master Page.
* Depending on number of content placeholders in your Master page, same number of <asp:Content> controls will be generated in Content Pages.
* Any JS or CSS used in all pages should be placed in
* Any common UI (html ,asp controls) required in all pages are placed in master page outside any content placeholder in body section

Code behind (\*aspx.cs) files are almost same for both master and content pages, other than that Master page inherits from System.Web.UI.MasterPage and Page inherits from System.Web.UI.Page. We can access the master page in the Web Form through the Master property, The Master property returns a MasterPage object by default, ( you know Master pages inherit from MasterPage class), so effectively Master property return object in form base class (MasterPage) . If you want to get actual calls object of master page (to get properties and functions defined in that class).

Use <%@ MasterType TypeName = "Site1" %> where TypeName is class name of master page in the content pages

Site1 is name of Master Page class

public partial class Site1 : System.Web.UI.MasterPage

{

}

This gives feature to access and manipulate any control in master page from content page. Now if you want to access content page from Master page then, you have get to contentPlaceHolder (a control in Master page) and then use function FindControl to find the control you want to access.

TextBox TextBox1 = (TextBox)ContentPlaceHolder1.FindControl("TextBox1");

ContentPlaceHolder1 is id of contentPlaceholder in body section, you will have better name in your code!

FindControl returns Control type, which is base class for all the controls.

Hence we need to typecast it to actual control.

We read about structure of page and master page, let’s look at how a page is executed to generate final html output.

You now know that when create a page you get 3 files for each page; .aspx, .aspx.cs and .designer.cs. Designer class is specific to Visual studio; other 2 are used for execution. When you request for a page ASP.NET create an instance of class that represent your page. This is what you know. But if you think a bit your .aspx.cs is a partial class inheriting from Page class, what about .aspx file, it does not have any class, it only contains declaration of all controls.

Behind the scene ASP.NET generates a partial class (with same name as your code behind class) from .aspx page! . If you see step wise what happens (for .aspx file)

* Generation of a \*.cs file holding code that matches the ASPX declarations.
* Using C# compiler to compile \*.cs file into DLL.
* Running of the compiled DLL

Now after this we have classes for both .apsx.cs (already exists, written by us) and class file generated from aspx (by asp.net), both are partial class, hence are combined and executed. Now this is all good, except that if for every request ASP.NET have to generate class by parsing .aspx page then it will be slow, so it stores final class in memory to improve performance.

If want to see the these class file generated by ASP.NET at C:\WINDOWS\Microsoft.NET\Framework\<targetFramework>\Temporary ASP.NET Files\$YourWebAppName$\)



**In short you need to remember only this “We have written a partial class in code behind, ASP.NET generated another partial class with same name by parsing ASPX page, both class are merged and executed”**

Now we have a combined class, let now check out life cycle.

|  |  |
| --- | --- |
| Event | Description |
| PreInit | All the Dynamic Controls, Master pages, Profiles and themes should be set in this event. |
| Init | This should be used to set the initial value of the Control properties. |
| InitComplete | This should be used to have custom ViewState data. This is the first place where ViewState has been loaded and can be changed. |
| Preload | This can be used to set the properties of the controls. |
| Load | All the Database connections and Data Binding can be performed here. Before this event finishes up all the validations will be done. Once the event is finished, the events for all the controls will execute before calling the next event in this list. |
| LoadComplete | This event can be used for activities on controls that require them to be fully loaded.s |
| PreRender | This is the last page where the visual properties of the controls can be changed before getting them displayed on the page. |
| PreRenderComplete | This will be called when the page is ready and no changes in visual elements can be made. All data binding are done at this point. |
| SaveStateComplete | View State has been saved and from this point onwards changes in ViewState will not be preserved i.e.ViewState has been saved for the page already. |
| Unload | The Page processing is done now. This is the last event that will be called. |

The important thing to consider here is that all these events will call the respective events of their child controls. The Page will also keep the events of all the child controls in sync with its own events.

**You need not to remember all of these, but this is asked in all interviews, for development, you should know init, load, preRender, Unload. Another question asked is when view state is loaded (InitComplete)**

*If you want to go in detail, please check* [*http://www.codeproject.com/Articles/73728/ASP-NET-Application-and-Page-Life-Cycle*](http://www.codeproject.com/Articles/73728/ASP-NET-Application-and-Page-Life-Cycle)

Let’s check out few controls in ASP.NET. There are 4 types of controls in asp.net

* **HTML server controls**   HTML elements exposed to the server so you can program them. HTML server controls are very close to the HTML elements that they render.
* **Web server controls**   Controls with more built-in features than HTML server controls. Web server controls include not only controls such as buttons and text boxes, but also special-purpose controls such as a calendar, menus, and a tree view control, grid view, repeater control etc. Web server controls are more abstract than HTML server controls in that their object model does not necessarily reflect HTML syntax.
* **Validation controls**   Controls that incorporate logic to enable you to what users enter for input controls such as the [TextBox](http://msdn.microsoft.com/en-us/library/vstudio/system.web.ui.webcontrols.textbox(v=vs.100).aspx) control. Validation controls enable you to check for a required field, to test against a specific value or pattern of characters, to verify that a value lies within a range, and so on.
* **User controls**   Controls that you create as ASP.NET Web pages. You can embed ASP.NET user controls in other ASP.NET Web pages, which is an easy way to create toolbars and other reusable elements. For more information, see [ASP.NET User Controls](http://msdn.microsoft.com/en-us/library/vstudio/y6wb1a0e(v=vs.100).aspx).

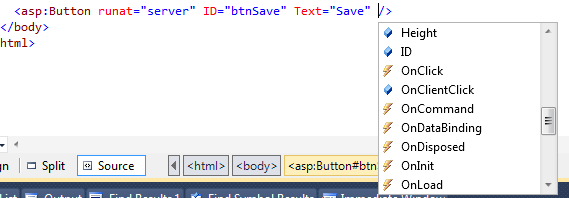
**HTML server control**: By default, HTML elements on an ASP.NET Web page are not available to the server. Instead, they are treated as opaque text and passed through to the browser. However, by converting HTML elements to HTML server controls, you expose them as elements you can program on the server.The object model for HTML server controls maps closely to that of the corresponding elements. For example, HTML attributes are exposed in HTML server controls as properties. Any HTML element on a page can be converted to an HTML server control by adding the attribute **runat="server"**. During parsing, the ASP.NET page framework creates instances of all elements containing the **runat="server"** attribute. If you want to reference the control as a member within your code, you should also assign an **id** attribute to the control.

Hopefully you would have a better understanding of HTML after reading HTML notes. So you know use of div, span etc. As discussed above If you need to control visibility of a div from server side then you put runat=”server” in div html tag and give it an Id, this dic is now accessible in your code behind class, you can write code around it. If you need to access a html control in server, you should write it as plain html.

**Web server Control**: They do not necessarily map one-to-one to HTML server controls. Instead, they are defined as abstract controls in which the actual markup rendered by the control can be quite complex. Like Grid generates table tr and td. Radiobuttonlist generate text and radio buttons. Web server controls include traditional form controls such as buttons and text boxes as well as complex controls such as tables, repeater, grid etc . Biggest benefit is automatic browser detection. The controls can detect browser capabilities and render appropriate html. For some controls, the ability to define your own layout for the control using [Templates](http://msdn.microsoft.com/en-us/library/vstudio/system.web.ui.design.templategroup.templates(v=vs.100).aspx). Controls that support Templates, there you specific any kind of complex html and that will be rendered for each data element. For some controls, the ability to specify whether a control's event causes immediate posting to the server or is instead cached and raised when the page is submitted.

**Validation Control:** An important aspect of creating ASP.NET Web pages for user input is to be able to check that the information users enter is valid. ASP.NET provides a set of validation controls that provide an easy-to-use but powerful way to check for invalid data and, if necessary, display messages to the user. ASP.NET provides validation controls for it, most of them don’t generate any html, but inject required code (JavaScript, server code) to validate fields and display errors.

We will see each in more details with examples. But before you process please note that in intellisense window things will lighting icon are events and blue brick are properties and methods.



So you write your control and set values for properties and attach functions to events, like onClick. You can also call javascript function on these events, in any case javascript function is called first and it does not return false, only then page is postbacked and server side code is executed.

Please note if you are asking page for first time, sequence of page event will be like init, load, prerender, unload (*major events listed only, it will follow all events listed in above section of life cycle*). Once page is at client side and user clicks on a button or change drop down and page is post backed, in that case init, load, control\_event, prerender, unload. Where control\_event may be button click or dropdown selectedindex change event based on what control caused postback.

**Few controls and how to work with them**

I feel that you can figure out how textbox, label work in asp.net. If not then ask me. We will jump in complex controls, controls that we will see is dropdown, listbox, grid, repeater and one special control File Upload control. If you know and understand these you can google out and understand any server control. Moreover you will rarely use any other control other than these.

**DropDown:**

See two htmls for dropdown list.



<asp:DropDownList runat="server" ID="cmbCities">

<asp:ListItem Text="Delhi" Value="1" />

<asp:ListItem Text="Mumbai" Value="2" />

<asp:ListItem Text="Kolkata" Value="3" />

</asp:DropDownList>

<asp:DropDownList runat="server" ID="cmbCountries"></asp:DropDownList>

DropDown list has a property “Items” that contain all the values to be shown in dropdown. You should also note that dropdown item has two kind of values (Text and value), one that is Text displayed to user, another is value. For example, you code around ids, like student id, but user needs name to work, so you display Name in dropdown, but you want to get Id of selected item. So when we define dropdown items, we give text and value for each item. If you don’t have separate value for text and value, you can use same value for both.

First example writes all items in aspx, as there may be case you know all the options. There may case when your dropdown values are coming from database, webservice or collection etc. To bind such things to a dropdown we use datasource property of dropdownlist.

We are create dt here for demo, we can get it from stored proc result or query or any other way.

DataTable dt = new DataTable();

dt.Columns.Add("Code");

dt.Columns.Add("Name");

DataRow dr = dt.NewRow();

dr["Code"] = "IN";

We set data source property, we can assign dataset, datatable, datarow array, collection, generic collection, array.

dr["Name"] = "India";

dt.Rows.Add(dr);

DataTextField tells about the field (property of object, if using collection, array) that we want to display

dr = dt.NewRow();

dr["Code"] = "GB";

dr["Name"] = "England";

dt.Rows.Add(dr);

dr = dt.NewRow();

dr["Code"] = "CH";

DataValueField tells about the field (property of object, if using collection, array) that we want to take as value when something is selected

dr["Name"] = "China";

dt.Rows.Add(dr);

cmbCountries.DataSource = dt;

cmbCountries.DataTextField = "Name";

DataBind is function which says, “okay I have told you everything now generate ListItems.” Without calling this function no item will be generated

cmbCountries.DataValueField = "Code";

cmbCountries.DataBind();

Looking above example you might also say, that I could have looped in datatable for each row and done something like

ListItem lt;

foreach (DataRow drow in dt.Rows)

{

lt = new ListItem();

lt.Text = drow["Name"].ToString();

lt.Value = drow["Code"].ToString();

cmbCountries.Items.Add(lt);

}

Yes it will also do! But prefer previous one. Once dropdown is filled, user can select values. Properties “SelectedIndex”, “SelectedItem” and “SelectedValue” are used to get and set selected value in dropdownlist. SeletedItem is readonly, you cannot set using this proeperty. Another way will search item in “Items” and set its selected property true.

cmbCountries.SelectedValue = "CH";

OR

cmbCountries.Items.FindByValue("CH").Selected = true; //will select china.

When we change dropdown selection an Event SelectedIndexChanged is raised, we can use this event to write any code we want to execute on dropdown change.

<asp:DropDownList runat="server" ID="cmbCountries"

onselectedindexchanged="cmbCountries\_SelectedIndexChanged" >

protected void cmbCountries\_SelectedIndexChanged(object sender, EventArgs e)

{

}

This is all you need to know to work with this control and clear interviews. Only thing before we move to next control is the html generated by this control.

<select name="cmbCities" id="cmbCities">

<option value="1">Delhi</option>

<option value="2">Mumbai</option>

<option value="3">Kolkata</option>

</select>

<select name="cmbCountries" id="cmbCountries">

<option value="IN">India</option>

<option value="GB">England</option>

<option value="CH">China</option>

</select>

See no dropdownlist or listitems.. only pure basic html tags! Keep this in mind when writing javascript function.

**Truth is most of us have to refer to google when writing javascript function, but in interviews you might not have this option, so remembering this a little bit will help you there.**

**ListBox:** This is cousin of dropdownList, only difference is it shows you multiple items and you can allow user to select multiple values. Again 2 ways of writing listbox.



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

<asp:ListItem Text="Delhi" Value="1" />

<asp:ListItem Text="Mumbai" Value="2" />

<asp:ListItem Text="Kolkata" Value="3" />

</asp:ListBox>

<asp:ListBox runat="server" ID="lstCountries"></asp:ListBox>

First one is again if you know items in advance, second one when you want to bind data source.

DataTable dt = new DataTable();

dt.Columns.Add("Code");

dt.Columns.Add("Name");

DataRow dr = dt.NewRow();

dr["Code"] = "IN";

dr["Name"] = "India";

dt.Rows.Add(dr);

dr = dt.NewRow();

dr["Code"] = "GB";

dr["Name"] = "England";

dt.Rows.Add(dr);

dr = dt.NewRow();

dr["Code"] = "CH";

dr["Name"] = "China";

dt.Rows.Add(dr);

lstCountries.DataSource = dt;

lstCountries.DataTextField = "Name";

lstCountries.DataValueField = "Code";

lstCountries.DataBind();

Exaclty same as dropdownlist. The difference that we talked about is letting user select multiple values, is done by setting property “SelectionMode” to “Multiple”

<asp:ListBox runat="server" ID="lstCountries" SelectionMode="Multiple"></asp:ListBox>

Rest all, getting selected value, or event of selection change are same as dropdown list. Interesting part is html generated

<select size="4" name="lstCities" id="lstCities">

<option value="1">Delhi</option>

<option value="2">Mumbai</option>

<option value="3">Kolkata</option>

</select>

<select size="4" name="lstCountries" multiple="multiple" id="lstCountries">

<option value="IN">India</option>

<option value="GB">England</option>

<option value="CH">China</option>

</select>

You must be wondering the difference in html of listbox and dropdown list! They are same other than size property.

**Grid:**  *This is biggie, it has so many functions, and so many properties that, we can write 20 pages only for grid view, As you don’t have time for this, we take short cut; cover all the most used features and questions in interviews.*

This control allows you not only display data, but functionality to add/edit/delete data. Simplest grid to display data is

<asp:GridView runat="server" ID="grdCountries" AutoGenerateColumns="true"></asp:GridView>

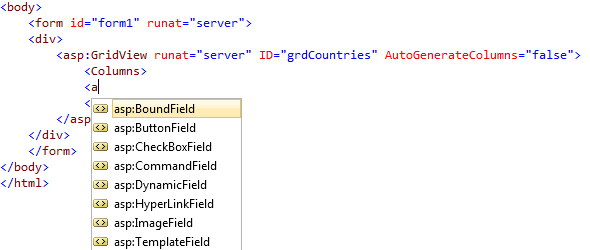
In code

grdCountries.DataSource = dt; //where dt is datatable we created in above examples.

grdCountries.DataBind();

Grid control has rows and columns, so when we bind it with a datasource then it generates equal number of rows and if AutoGenerateColumns is also true then same number of columns will be generated. AutoGenerating columns may not be usefull in all cases, let say you have an ID column that you don’t want to show to user, but you need it anyways for data manipulation. A obvious disadvatange is, that it is not possible to explicity say, which properties should be displayed as columns, what the HeaderText or width of each column should be.

The second way to create the columns allows to explicitly defining, which columns should be displayed, how they look and in which order they are displayed. GridView provides a Columns Property, in which we add columns.



See in image we have different kind of columns that we can add in GridView. ButtonField, CheckBoxField, HyperLinkField,ImageField are simple to understand just by name. You must understand what ever column type we write inside column will be executed for each row. Let’s have look on other types.

**Using BoundField:**  The BoundField class is used by GridView to display the value of a field as text. It is the most easy to use column type. You just need to set the DataField property to the name of the property; you want to show in this column. The GridView will automatically render the display control. To hide a column, set Visible = false.

<asp:GridView runat="server" ID="grdCountries" AutoGenerateColumns="false">

<Columns>

<asp:BoundField HeaderText="Country Code" DataField="Code" ItemStyle-Width="80" />

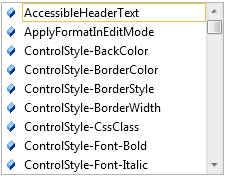
<asp:BoundField HeaderText="Country Name" DataField="Name" ItemStyle-Width="80" />

</Columns>

</asp:GridView>

Output



If you go and check on properties of asp:BoundField field you will be amazed to the number of properties , there is so much that you can customize.

**Using TemplateField:**

In a TemplateField you can specify per column, which control should be rendered for which Display-Mode (display/edit).

***ItemTemplate*:** Template that is used, if the GridView is in the Display-Mode, or if no template is specified for the current mode the GridView.

<asp:GridView runat="server" ID="grdCountries" AutoGenerateColumns="false">

<Columns>

<asp:BoundField HeaderText="Country Code" DataField="Code" ItemStyle-Width="120" />

<asp:TemplateField HeaderText="Country Name">

<ItemTemplate>

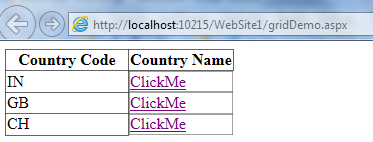
<asp:HyperLink runat="server" NavigateUrl="~/ListBoxDemo.aspx" Text="ClickMe" />

</ItemTemplate>

</asp:TemplateField>

</Columns>

</asp:GridView>

See we are using itemtemplate for generate a hyperlink for each row. Output is as shown in image. As you can there is no difference in links as in item template we have not defined any “Name” column specific code.

Let’s change html a bit and see the effect.

<asp:GridView runat="server" ID="grdCountries" AutoGenerateColumns="false">

<Columns>

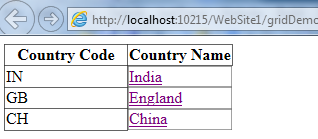
<asp:BoundField HeaderText="Country Code" DataField="Code" ItemStyle-Width="120" />

<asp:TemplateField HeaderText="Country Name">

<ItemTemplate>

<asp:HyperLink runat="server" NavigateUrl="~/ListBoxDemo.aspx" Text='<%# Eval("Name") %>' />

</ItemTemplate>

 </asp:TemplateField>

</Columns>

</asp:GridView>

This one looks better as we have country name displayed as hyperlink.

A new thing was <%# Eval("Name") %>, another cousin of this is <%# Bind("Name") %> . These are two syntaxes, used to get value of specific column for each row. Use Eval when you have to only display data, Use bind when you have to edit data.

***InsertItemTemplate*:** Template that is used, if the GridView is in the Insert-Mode, ***EditItemTemplate*:** Template that is used, if the GridView is in the Edit-Mode. To understand these better we need to look at edit and delete support of gridview.

**Data editing from GridView**

<asp:GridView ID="grdCountries" runat="server" AutoGenerateColumns="False" DataKeyNames="Id"

OnRowCancelingEdit="grdCountries\_RowCancelingEdit" OnRowDataBound="grdCountries\_RowDataBound"

OnRowEditing="grdCountries\_RowEditing" OnRowUpdating="grdCountries\_RowUpdating" ShowFooter="True"

OnRowCommand="grdCountries\_RowCommand" OnRowDeleting="grdCountries\_RowDeleting">

<Columns>

<asp:TemplateField HeaderText="Name" HeaderStyle-HorizontalAlign="Left">

<EditItemTemplate>

<asp:TextBox ID="txtName" runat="server" Text='<%# Bind("Name") %>'></asp:TextBox>

</EditItemTemplate>

<FooterTemplate>

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

</FooterTemplate>

<ItemTemplate>

<asp:Label ID="lblName" runat="server" Text='<%# Bind("Name") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Name" HeaderStyle-HorizontalAlign="Left">

<EditItemTemplate>

<asp:TextBox ID="txtCode" runat="server" Text='<%# Bind("Code") %>'></asp:TextBox>

</EditItemTemplate>

<FooterTemplate>

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

</FooterTemplate>

<ItemTemplate>

<asp:Label ID="lblCode" runat="server" Text='<%# Bind("Code") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Edit" ShowHeader="False" HeaderStyle-HorizontalAlign="Left">

<EditItemTemplate>

<asp:LinkButton ID="lbkUpdate" runat="server" CausesValidation="True" CommandName="Update" Text="Update"></asp:LinkButton>

<asp:LinkButton ID="lnkCancel" runat="server" CausesValidation="False" CommandName="Cancel" Text="Cancel"></asp:LinkButton>

</EditItemTemplate>

<FooterTemplate>

<asp:LinkButton ID="lnkAdd" runat="server" CausesValidation="False" CommandName="Insert" Text="Insert"></asp:LinkButton>

</FooterTemplate>

<ItemTemplate>

<asp:LinkButton ID="lnkEdit" runat="server" CausesValidation="False" CommandName="Edit" Text="Edit"></asp:LinkButton>

</ItemTemplate>

</asp:TemplateField>

<asp:CommandField HeaderText="Delete" ShowDeleteButton="True" ShowHeader="True" />

</Columns>

</asp:GridView>

This is a sample gridview html to support editing. Html generated is something like



See two additional columns added for Edit and Delete, also last row (Footer) has empty textboxes for inserting new row.

We need to understand each part before we go into detail. Let’s start from gridview tag.

<asp:GridView ID="grdCountries" runat="server" AutoGenerateColumns="False" DataKeyNames="Id"

OnRowCancelingEdit="grdCountries\_RowCancelingEdit" OnRowDataBound="grdCountries\_RowDataBound"

OnRowEditing="grdCountries\_RowEditing" OnRowUpdating="grdCountries\_RowUpdating" ShowFooter="True"

OnRowCommand="grdCountries\_RowCommand" OnRowDeleting="grdCountries\_RowDeleting">

DataKeyNames: Gets or sets an array that contains the names of the primary key fields for the items displayed in a [GridView](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.gridview(v=vs.110).aspx) control. In simple words, it contains primary key for each row of data shown in grid.

*Always have a primary key on that you are going to use in any grid kid of control. Kyuki mostly hum kafi sare inbuild functions use kerte plus adapater use kerke multiple rows database me update kerte hai, remember sqladapter wale note me, we have seen ki pure datatable to kaise sql server ki table me merge kerte hai, merge se matlab hai ki tumne datatable ko database se fill ker liya, fir usr ko kisi control me dikha diya (grid) fir user ne kuch row add ke, kuch delete ke aur kuch update ki, to yeh sare changes database me wapas se store kerne hai, uske liye us note me multiple ways dikhaye gaye hai, unko ager dhyan se dekho to they all need primary key. So in general make a habbit ki grid type ke control me jab bhi databind karo, aur data editable mode me dikha rahe to primary key zaror lao, warna problem aati hai.*

OnRowEditing: Occurs when a row's Edit button is clicked.

OnRowCancelingEdit: Occurs when the Cancel button of a row in edit mode is clicked

OnRowDataBound (very important event) Occurs when a data row is bound to data in a GridView control. By bound we mean grid generated control row by row, every time a row is processed this event is raised. You will be using this event a lot, to manipulate how html is generated for row, If you need to attach a javascript function based on some value in row, this is the event you will use

OnRowUpdating: Occurs when a row's Update button is clicked, but before the GridViewcontrol updates the row

OnRowUpdated: Occurs when a row's Update button is clicked, but after the GridView control updates the row.

OnRowCommand: Occurs when a button is clicked in a GridView control.

OnRowDeleting: Occurs when a row's Delete button is clicked, but before the GridView control deletes the row.

And a lot of such events, these event are pretty much stand for their names.

Let’s now see the html of 1 column, yes! This big html for one column

<asp:TemplateField HeaderText="Name" HeaderStyle-HorizontalAlign="Left">

<EditItemTemplate>

<asp:TextBox ID="txtName" runat="server" Text='<%# Bind("Name") %>'></asp:TextBox>

</EditItemTemplate>

<FooterTemplate>

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

</FooterTemplate>

<ItemTemplate>

<asp:Label ID="lblName" runat="server" Text='<%# Bind("Name") %>'></asp:Label>

</ItemTemplate>

</asp:TemplateField>

Highlighted part is what you are familiar with, you read this in Itemtemplate part, it to display data. EditItemTemplate are the controls to be shown when user wants to edit the record of a row. FooterTemplate are the controls to be shown to insert data. See that when editing data we need to show current value, hence Text='<%# Bind("Name") %>'> is used, but to new row, we can show blank text box hence textbox in footerTemplate does not has its text property set.

**Second last column (Edit) HTML**

<asp:TemplateField HeaderText="Edit" ShowHeader="False" HeaderStyle-HorizontalAlign="Left">

<EditItemTemplate>

<asp:LinkButton ID="lbkUpdate" runat="server" CausesValidation="True" CommandName="Update" Text="Update"></asp:LinkButton>

<asp:LinkButton ID="lnkCancel" runat="server" CausesValidation="False" CommandName="Cancel" Text="Cancel"></asp:LinkButton>

</EditItemTemplate>

<FooterTemplate>

<asp:LinkButton ID="lnkAdd" runat="server" CausesValidation="False" CommandName="Insert" Text="Insert"></asp:LinkButton>

</FooterTemplate>

<ItemTemplate>

<asp:LinkButton ID="lnkEdit" runat="server" CausesValidation="False" CommandName="Edit" Text="Edit"></asp:LinkButton>

</ItemTemplate>

</asp:TemplateField>

See ItemTemplate says use a linkbutton (a control is asp.net that display a button as hyperlink), Set text = “Edit”, you can put any text here. FooterText again displays a linkbutton, text here is Insert. These 2 are simple, only new thing you will find is “CommandName” this has different value in ItemTemplate and different in FooterTemplate, which is correct as Edit in row with data will be edited, but last row is for insert. CommandName will be used to handle different commands on OnRowCommand event.

Last column (Delete) HTML

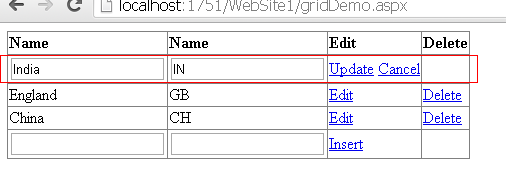
<asp:CommandField HeaderText="Delete" ShowDeleteButton="True" ShowHeader="True" />

See this is CommandField, ShowDeleteButton will show Delete linkbutton for every row. See there is no footertemplate for this column, which again make sense as last row is for inserting new row so there is nothing that you will delete.

****

**GridView is very vast control, but you need to learn this for work and interviews, if you are not clear about things till now, please re-read it.**

So our html is all set, we need to write code in events to handle all the operations. Before that if you run the page created till now you click on Edit then UI shown will be



If you scroll above in this note you will see that we had used

<EditItemTemplate>

<asp:LinkButton ID="lbkUpdate" runat="server" CausesValidation="True" CommandName="Update" Text="Update"></asp:LinkButton>

<asp:LinkButton ID="lnkCancel" runat="server" CausesValidation="False" CommandName="Cancel" Text="Cancel"></asp:LinkButton>

</EditItemTemplate>

For edit column, because we want users to have functionality to either update or cancel their update.

Now we should put code to handle Insert, update, cancel and delete, brace yourself for a little more code.

*Please note that I am using a datatable to showcase functionality of gridview, to actual code me, jaha per getdata, updatadata,insertdata jaise function unme tum database se interact ker rahe hoge.*

public partial class gridDemo : System.Web.UI.Page

{

DataTable dt;

protected void Page\_Load(object sender, EventArgs e)

{

// Page load hote he humko data chaye grid me bind kerne ke liye. Actual code me tum database se data laoge, magar page\_load to her baar call hota hai (first time page manga user ne, ya fir postback kiya). First time me tum db se data laye or grid me bind ker diya, who data html ka part ban ker user tak chala gaya, post back me controls ke saath who data wapas aa he raha hai, to tum dubara db se kyu laoge. Dusri baat, man lo user ne data me kuch change kiya, ya naya insert kiya, who data data tumhare gridview ke control me hai, unse pad ker he tum database me update kerna chaehte ho. Ab control ke event to page\_load ke baad call hote hai, to postback ke case me ager page\_load per tumne database se data la ker grid ko fir se fill ker diya to tumhare sare changes overwrite ho jayenge. In short, database se data lane wale code “mostly” ek baar call hona chaheye, hence unko hum !Page.IsPostBack ki if me likhte hai, to yeh sirf first request me data laye. Please note ki yeh depend kerta hai tumhari data need per, ager tumko hamesha naya data dikhana hai to tum her baar laoge. Is code me else wale part per jyada dhyan mat do, is example me db se data to la nahi rahe, isliye thoda jugadbazi hai.

if (!Page.IsPostBack)

{

RefreshGrid();

}

else

{

dt = getData();

}

}

// Yeh function db se data fetch kerne wali functionality ki jagah per hai, actual code me yaha database se data lane wala code hoga.

private DataTable getData()

{

dt = new DataTable();

dt.Columns.Add("Id");

dt.Columns.Add("Code");

dt.Columns.Add("Name");

DataRow dr = dt.NewRow();

dr["Id"] = "1";

dr["Code"] = "IN";

dr["Name"] = "India";

dt.Rows.Add(dr);

dr = dt.NewRow();

dr["Id"] = "2";

dr["Code"] = "GB";

dr["Name"] = "England";

dt.Rows.Add(dr);

dr = dt.NewRow();

dr["Id"] = "3";

dr["Code"] = "CH";

dr["Name"] = "China";

dt.Rows.Add(dr);

return dt;

}

// Yeh function datagrid me data fill kerne ke liye hai, First time tumko grid fill kerne hai, kuch insert to tum save kerne ke baad grid me naya databind kerna chahoge (kyuki other users bhi to db me data dal sakte hai, isi page se), is tarah update and delete me bhi save kerke naya refreshed data dikhana chahoge to ayesa ek common function bana lo, aur jab refresh kerna ho grid ko, is function ko call ker lo.

private void RefreshGrid()

{

if (dt == null)

{

dt = getData();

}

grdCountries.DataSource = dt;

grdCountries.DataBind();

}

// Yeh bada kaam ka function hai jo is example me kaam nahi aaya ☺, isko pure example ke baad alag se dekhenge.

protected void grdCountries\_RowDataBound(object sender, GridViewRowEventArgs e)

{

}

// Ager koi row delete kerne hai to tum DataKeys property se us row ki primary key uthaoge, dekho GridViewDeleteEventArgs e may e.RowIndex se tumko index number mil jayega, fir is index number se tum DataKeys se primary key ki value nikaloge, fir tumne job hi delete ka function likha hoga(database me se delete kerne ka) usko call ker doge. Yeh primary key dataKeys me rakhne ki jagah tum ek aur jugad ker sakte ho, dekho database se to primary key ka data baki data se saath aa he raha hoga, lets assume ki primary key ke colomn ka naam Id hai, Tum grid me ek column ID ka banao and uska visible false ker do, to column html me generate hoga, magar user ko dikhega nahi. Nahi samajh aaye to batana example dikha dunga.

protected void grdCountries\_RowDeleting(object sender, GridViewDeleteEventArgs e)

{

int Id;

int.TryParse(grdCountries.DataKeys[e.RowIndex].Values[0].ToString(), out Id);

var rows = dt.Select("Id = " + Id.ToString());

foreach (var row in rows)

{

row.Delete();

}

RefreshGrid();

}

// Grid ki html me ja ker dekho, humne Commands set kiye they, edit, cancel, insert, delete me. To delete, cancel, edit ka to alag se bhi event hai, jaise is example me dekh rahe ho, insert ke liye yahi use kerna pageda, and by the way, tumhare liye ek task hai, go and find out ki in sare events ke execution ka sequence kya hai.

protected void grdCountries\_RowCommand(object sender, GridViewCommandEventArgs e)

{

string Name;

string Code;

if (e.CommandName.Equals("Insert"))

{

//insert logic here, Ab yeh bahut dhyan se dekho FindControl ka use kiya jata hai control dhunde ke liye, maine yeh code chota sa likh diya, magar tum pehle findcontrol se control dhundhoge, check karoge ki who null to nahi hai, fir text property set karoge, kyuki ager control nahi mila to yeh code exception marega.

Name = ((TextBox)grdCountries.FooterRow.FindControl("txtNewName")).Text;

Code = ((TextBox)grdCountries.FooterRow.FindControl("txtNewCode")).Text;

InsertRow(Name, Code);

RefreshGrid();

}

}

//insert logic, yaha db me isnert kerne ka logic likho, biggestIdInDataTable wala part, again jugad hai kyuki hum database se data nahi la rahe, to id generate kerne ke liye maine jugad lagya, tum ager database se laoge, to id most probably tm identity column bana doge, to apne aap he generate hogi insert per, ager tumko identity column me confusion hai to ask me.

private void InsertRow(string Name, string Code)

{

DataRow dr = dt.NewRow();

Int32 biggestIdInDataTable;

Int32.TryParse(dt.Rows[(dt.Rows.Count - 1)]["Id"].ToString(), out biggestIdInDataTable);

dr["Id"] = biggestIdInDataTable + 1;

dr["Name"] = Name;

dr["Code"] = Code;

dt.Rows.Add(dr);

RefreshGrid();

}

//Grid has a property EditIndex, which is used to store the index of row of grid that is being edited., row editing event me isko set kerdo, fir row\_updating wale event me use karenge.

protected void grdCountries\_RowEditing(object sender, GridViewEditEventArgs e)

{

grdCountries.EditIndex = e.NewEditIndex;

}

//Is function me EditIndex ko read kerke, updated data read kiya jata hai. Tumne abhi tak ek baat ager gaur ke ho ki humne textboxes ka naam txtName and TxtCode diya tha, ab yeh control her row ke liye generate hoga, magar tumko yeh pata hai ki page me ek naam ka ek he control ho sakta hai to fir yeh kaise ho raha? Actully asp.net is taraha ke controls ki id change kerta hai, to application to run karo, view source karo and generated ids ko dekho. Kyuki her row in controls ki id alag hogi hai, to ager server side me tumko control chaeye to tum findcontrol uskaroge and row ka index use karoge. Row index user kerne se Find control sirf uf row ki html ke under dhundhega, to us row ki html me txtName and txtCode naam ke sirf ek he control honge.

protected void grdCountries\_RowUpdating(object sender, GridViewUpdateEventArgs e)

{

int Id;

string Name;

string Code;

Int32.TryParse(grdCountries.DataKeys[e.RowIndex].Values[0].ToString(), out Id);

Name = ((TextBox)grdCountries.Rows[e.RowIndex].FindControl("txtName")).Text;

Code = ((TextBox)grdCountries.Rows[e.RowIndex].FindControl("txtCode")).Text;

UpdateRow(Id, Name, Code);

grdCountries.EditIndex = -1;

RefreshGrid();

}

private void UpdateRow(int Id, string Name, string Code)

{

DataRow dr = dt.Select("Id = " + Id.ToString())[0];

dr["Name"] = Name;

dr["Code"] = Code;

}

//Grid control me editIndex naam ki property hoti hai, yeh tum jo row edit ker rahe ho, uska index rakhta hai, to ager tum kuch edit kerne wale the, magar ab nahi kerna chaehte to cancel click karoge. Jab tum edit kerne wale the tab tumne editindex set kiya, ab cancel kerna hai to isko -1 ker do.

protected void grdCountries\_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)

{

grdCountries.EditIndex = -1;

}

}

*Dekho ayese to yeh grid samajh aane se raha, abhi padoge fir bhul jaoge. Aur grid me bahut kuch hai, simple example jaise alternate rows ka color change kerna, ya fir value basis per color change kerna… To thoda haath pair hilao, laptop open karo, aur ek page banao grid ke saath aur database se data la ker usko edit and save kerke dekho. Tumko sirf data lane aur save karane me he first time me problems hogi.. But that’s the only way to do it…. Ek baar yeh ker liya to tumko grid, asp page, database se insert, select, update, delete sab aa jayega.*

*Uper ek function reh gaya tha RowBound wala, usko dekhte hai, jaise ke humne dekha ki her row ke liye control generate hote hai, ab maan lo tumko dropdown dikhana ho grid ke under, dropdown tumko pata hai ki usme multiple value bind hoti hai, uska tareeka uper tum dekh chuke ho. Ab jab her row ke liye alag control hoga, to matlab her dropdown ko hume data se bind kerna hai. Ayesa nahi hota ki ek jagah value rakh le aur sare drop down un values ko use kare. Tumko yaad ho to jab drop down ki html generate hoti hai to uske sare option bhi html me aate hai, to ager tumko grid me dropdown dikhana hai aur suppose 10 row hai grid me to 10 dropdwon generate honge aur 10 ke 10 ko tumko datasource se bind kerna padega.*

// Dekho rowbound event her row ke liye call hoga, to tum yaha per her row ke dropdown ko data se sakte ho, yahi event value base per color change kerne ke kaam bhi aata hai. Net per dhundhna bahut sare example mil jayenge.

protected void grdCountries\_RowDataBound(object sender, GridViewRowEventArgs e)

{

if (e.Row.RowType == DataControlRowType.DataRow)

{

DropDownList ddlDepartment = (DropDownList)e.Row.FindControl("ddlLanguage");

if (ddlDepartment != null)

{

ddlDepartment.DataSource = new GetLanguage();

ddlDepartment.DataBind();

}

}

else if (e.Row.RowType == DataControlRowType.Footer)

{

DropDownList ddlDepartment = (DropDownList)e.Row.FindControl("ddlNewLanguage");

ddlDepartment.DataSource = new GetLanguage();

ddlDepartment.DataBind();

}

}

*Grid ko yahi rokte hai, tum practice ker lo baki practice hum Blog me karenge. Ager aur koi control baad me dekhenge, once you have understood grid, repeater wagehra 2 min me samajh aa jayega. Ab kuch state management ki concept dekhte hai. Next note me Session, Caching and Ajax lenge.Thodi jagah hai to kuch special syntax dekh lete hain*

**State Management -**

***Is part ko mai chota nahi ker sakta, kyuki interview me hamesha pucha jata hai***

*Http is stateless, stateless bole to isko kuch yaad nahi rehta hai, ki pechli baar jab tum aaye the to tumne kya kya kiya tha. Ab ayse static page to chal sakte hai magar, socho ki tumne online shoping kerne hai tumne item select kiye, next page per aaye who sare gayab ho gaye, kyuki http stateless hai. Ayese to kaam nahi chalega.*

*Ab ayesa hota kyu hai,kyuki jab bhi tum request kerte ho first time or postbacks, page ka object banta hai, uski life cycle hoti hai aur who destroy bhi ho jata hai. To man lo tumne kuch value enter ke textbox, ya fir button ke click per kisi label ka text change kiya ho woh round trip me lost ho jayega. Infact tumko application identify ker paye iske liye ager who her request me credential mange to tum pareshan ho jaoge, tum chahoge ki ek bar password dal diya to ek duration tak who tumko yaad rakhe. Windows form application me ayesa nahi hota, kyuki tumhare object tum zinda rakh sakte ho, magar web server kyuki multiple users ko support kern eke liye bane hai, to woh kam se kam data store kerke rakhna chahete hai. To tumne request ko process kerne ke liye object banayee, response gaya aur fir sare object clean, ab tumko explicitly kuch save kerke rakhna hai agli baar ke liye to tum save karo.*

There are mainly two types of state management that ASP.NET provides:

1. Client side state management
2. Server side state management

*Ab baat ko bahut dhyan se socho. Ek example banate hai aur uske context me samajhne ki koshish kerte hai, yeh exactly 100% fit to nahi baithea, but tumko samajh aa jana chaeye. Suppose tumhare phone bill me kuch problem hai, you have some facts with you jisse tumhara bill sahi ho jayega, aur tumko call center call kerna hai aur resolve karana hai, is case me suppose ker lo call center me ek he banda hai. Ab tum call kerte ho representative se baat hoti hai, tum usko puri kahani samjhate ho, who tumko ek mail kerne ko bolta hai. Is case me tum client ho, call center wale server hai, tumne jo information de who tumhara information hai. Ab tumko fir se call kerne padi 1-2 din baad. Ab call center ko dekho, ek banda hai who tumhara case yaad kaise rakhe, roj 100 call aati hai. woh yaad to rakh nahi sakte ki kisse kya baat hui thi. Yani mamla stateless hai.*

*Ab yaad to who rakh sakhta nahi to fir to kya option hai, ek option to hai kit tum jab call karo to sari khanai fir se sunao (client side state management) ya fir who hero koi software khareede, ya file banaye her case ki usko sari information mil sake ki ab tak kya progress hui.(sever side state management, session ek server side statement technique hai).*

When we use client side state management, the state related information will be stored on client side. This information will travel back and forth with every request and response. *Ab tum client side statement use kerte ho dekho her call per tumko sari info batani padti hai, yani jyada bandwidth use ker rahe, jyada data transfer ker rahe.*

*Abhi data client side store kerne me acha yeh hai ki tumhare server ki memory bach gayi, but bahut se nuksan bhi hai, ek to her baar extra data tumko behjna hoga, bandwidth khaoge. Usse badi baat, client per data hai to koi bhi change ker dega. Security ka issue hai, client side per tum koi bhi secure information nahi bhej sakte.*

*Thoda is bare me batata hu, uska abhi jo pad rahe hai usse he connection hai, but normal applications me hum is per dhyan nahi dete. Man lo ek website hai jisme accounts ki information dikhte hai, update ker sakte ho. Ab accounts based on permission dekhte hai. Man lo tumko 100-500 account number tak wale account dekhne ki permission he hai. Ab developer ne ayesa banaya kit tum ek account ki information open kerte hai to who page me hidden field me account id chupa ker rah leta hai, aur jab update information save kerna hota hai to who us id ka use kerke save ker deta hai. Sahi banaya, tum bhi ayesa he banaoge, kyuki bhai common se baat hai jis account ko edit ker rahe information bhi to usme he save karoge.*

*Account number 503 jiska hai hai, usse tumhari ladayi ho gayi. Tumko usko sabak sikhana hai to tumne uske account ki watt lagani ki sochi, magar tumko permission nahi hai 503 khole ke, tum 100 se 500 tak he open ker sakte ho… Ab dekho koi bhi account khol lo data change kerke 0 balance ker do, aur jab post back karo to tum hidden field ki value change kerke 503 ker do, developer ne postback per koi check nahi lagya hai to jot um save ker rahe who 503 ke against me update ho gaya!!!!!!!*

*Infact RBPS me yeh issue aaya tha!!! Tab maine waha dual authorization implement kiya tha, matlab pehle to aap secure info page me ya client side per rakho nahi, aur rakh rahe to postback ke baad sari values reverify karo, seedhe action mat le lo.*

So there is no way we can store the sensitive information like passwords, creditcard number and payable amount on client (*tumko case bataya tha na rajasthan roadways and times group ka*) side, we need server side state management for such things.

Server side state management, in contrast to client side, keeps all the information in user memory. The downside of this is more memory usage on server and the benefit is that users' confidential and sensitive information is secure.

*Isme ek he issue hai ki server ki memory bahut lag rahi, but yahi jyada sensible hai secure info ke liye, Isko better kerne ke liye bahut sare options hai jo hum age dekhenge. Ek baat dhyan rakhna, koi kanun nahi hai ki client ya server side he, dono mix kerke bhi use hota hai, (infact mix he jyada use hota hai) kuch tum server per rakhoge, kuch client per. Ab tum soch rahe ki mix kyu kerna, ab apna call center wala example lo usme tumhari information call wale me track kerne shuru bhi ker de, magar 1000 customer hai, tumko pehchanne ke liye, tum kuch to information doge, jaise apna customer id kind of… Exactly yahi server side state managent (session) me hota hai, tumhari information to server per hoti hai but ek cookie (aage bataunga iske bare me) jiske under tumhari session id hoti hai, woh client side store hoti hai, he request ke saath who cookie ( effectively tumhari session id ) server per jati hai, us id se sever tumhari identiy, stored values nikalta hai. Bahut huwa gyan aao ab options dekhe.*

**Client side state management techniques**

* View State
* Control State
* Hidden fields
* Cookies
* Query Strings

**Server side state management techniques**

* Application State
* Session State

Odi baba itna shora toreeka!, yes these are different ways of statement management and Session is just one of them.

**View State:** The [ViewState](http://msdn.microsoft.com/en-us/library/vstudio/system.web.ui.control.viewstate(v=vs.100).aspx) property provides a **dictionary** object for retaining values between multiple requests for the same page. This is the default method that the page uses to preserve page and control property values between round trips. When the page is processed, the current state of the page and controls is hashed into a string and saved in the page as a hidden field. When the page is posted back to the server, the page parses the view-state string at page initialization and restores property information in the page. *Grid wale sample to run karo view source karo, deko kuch ayesa dikhega.*

<input type="hidden" name="\_\_VIEWSTATE" id="\_\_VIEWSTATE" value="avOlAJtCmBT2N5+OdC4j+Dikc3nlPdXIa3qoY8DEhg9aFVcZhLFTo4EtvEss7jriIa6zrOattaziUP7X9B5JwoYgcpOV9aIGdbYk0Y7EaLcQUQS24t9B/+JCcm9kzn3vH9ddAf8KsGQcY2IDiswukFJr5aJwSYRt3gPJfQ2/9HRNXx/VpgMozH8xZVErpsH/Y6J56GEHbvigEn/0zLeg5kF76e9p3hIkeMlkHZmrLgW1tzmkRQRT16UAc6mHT+5T1TUuUFVItKLxlMd1vfkqNn2FoVHqLgcOxWLkAm/WOP93qMYS/PoTimqt7dJCjxkUpFZ/dTiRH0JhiQz76+4d81ULKvfXHDb2gtZSW8S1IXrMR3q9ErLtb+OLazbxC/+SL5Qcf97IFz66fzWHp31Bn57Af0rdQVzHIfGp6MFxJcRyNnnAQWfH7B9KW7Kk0uf6SqrMafhVN+fFcbKBTjl8wnQZMJcfO8BUG/k1FAkRdQ0=" />

Kyuki viewstate actually ek dictionary hai to hum apni bhi koi value store ker sakte hai.

Store ker ne ke liye

ViewState["number"] = 1;

And retrieve kerne ke liye

if (ViewState["number"] != null) //pehle check kerlo hai bhi ya nahi

{

ViewState["number"] = Convert.ToInt32(ViewState["number"]);

}

*Server side per yeh ek dictionary hai, jab page I html generate hoti hai, to ASP.NET ek hidden field page me dalta hai, aur uski value me View state ki sare value hash kerke daal deta hai.*

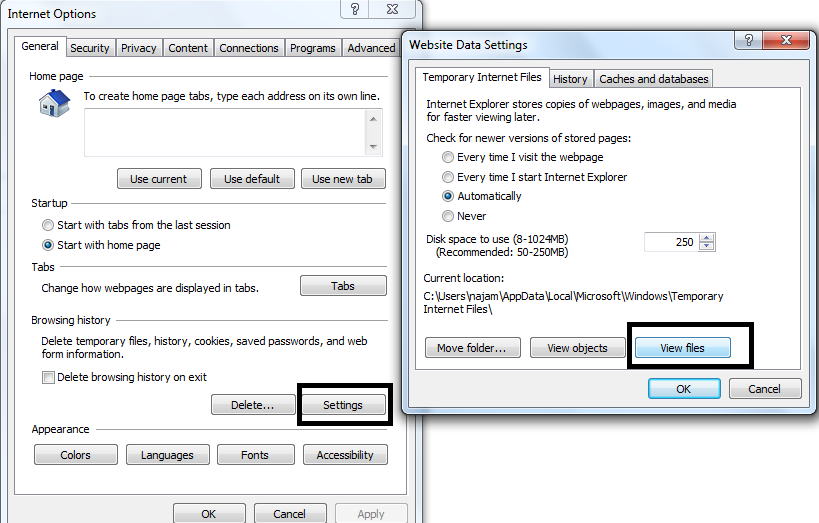
*This is all good, but because as usual mai adat se majbur hu*! *Tumhare dimag me ek question aana chaeye, ki yeh jo controls hote hai inki value to inke saath aati jati rehti he hai, ab chaeye kitne bhi round trip ho, magar textbox ki text property to html ka part he hai who to ud nahi jayegi, to jab controls apni value khud he save ker sakte hai, to yeh view state kya save ker rahi????????*

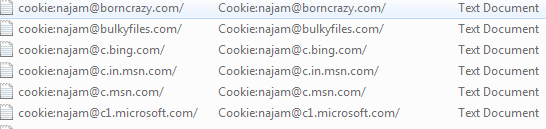
**Control State***: View state ko on off ker sakte ho page level per, to ager tumne koi user control banaya jiso view state me kuch rakhna hai, but tumhare control use kerne wale ne page level per view state off ker de to??.* we can have viewstate like behavior which cannot be disabled by control users and it is called ControlState. Control states lies inside custom controls and work the same as viewstate works.

**Hidden Fields:** ASP.NET allows you to store information in a [HiddenField](http://msdn.microsoft.com/en-us/library/vstudio/system.web.ui.webcontrols.hiddenfield(v=vs.100).aspx) control, which renders as a standard HTML hidden field. A hidden field does not render visibly in the browser, but you can set its properties just as you can with a standard control. When a page is submitted to the server, the content of a hidden field is sent in the HTTP form collection along with the values of other controls.

*Infact tum view state ko deho, html me who bhi hidden field me he hai. Bahut normal hai Hidden field use kerna, her page me developer’s ek do hidden field use ker lete hai.*

**Cookies:** So far, the techniques we have discussed store the data for the single page requests. Now we look at the techniques that store information between page requests. Cookies are small pieces of information that can be stored in a text file on users' computer. To see cookies of IE, go to internet options in tools then as shown in image





when the browser requests a page, the client sends the information in the cookie along with the request information. The server can read the cookie and extract its value

if (Request.Cookies["number"] != null) //Lets retrieve, increase and store again

{

postbacks = Convert.ToInt32(Request.Cookies["number"].Value) + 1;

}

*Dekho abhi tak chahe tumne hidden field dekha, viewstate dekha yeh sab page ki value save rakhte hai, but jaise tumhari session ki id, ya fir koi setting jaise tumne kaun se color theme select ke hai, is type ki information ek choti se textfile, cookie me likh ker client ko bhej dete hai, who fir agli her request me cookie bhejta hai server ko.*

**QueryString**: A query string is information that is appended to the end of a page URL. They can be used to store/pass information from one page to another to even the same page

<http://www.contoso.com/listwidgets.aspx?category=basic&price=100>

In the URL path above, the query string starts with a question mark (?) and includes two attribute/value pairs, one called "category" and the other called "price." Information that is passed in a query string can be tampered with by a malicious user. Do not rely on query strings to convey important or sensitive data.

if (Request.QueryString["number"] != null) //Lets retrieve, increase and store again

{

Label4.Text = Request.QueryString["number"];

}

*Query string ka kaam ek page se dusre page me value le jane ke liye hota hai, magar hamesha dhyan rakho isme koi bhi secure value, ya ayesi value nahi na pass karo jisper logic depend kerta hai, for example , tumhare application me edit and view ke page hai, jo user ke role ke hisab se show hote hai, ab tum ager querystring me view ya edit se decide karoge ki kya dikhana hai to koi bhi user querystring change kerke edit page me chala jayega.*

**Application State:** ASP.NET allows us to save values using application state. A global storage mechanism that is accessible from all pages in the Web application. Application state is stored in the Application key/value dictionary. This information will also be available to all the users of the website

Application["number"] = 0;

*Application state application level per hai matlab value tum kisi bhi user se dalo available sabe ke liye hogi, koi bhi change ker sakta hai*

**Session State:** Like Application state, this information is also in a global storage that is accessible from all pages in the Web application. Session state is stored in the Sessionkey/value dictionary. This information will be available to the current user only, i.e., current session only.

Session["number"] = 0;

ASP.NET session state identifies requests from the same browser during a limited time window as a session, and provides a way to persist variable values for the duration of that session.

*Dekho dono baat hai ek time widnow hai aur storage bhi hai, session to banega he, tumne application per hit mari, tumahra session ban gaya, tum session may data store karo ya na karo, session to banega he. Application state and Session me fark sirf itna hai ki session per user hota hai, to tum user specifc info he session me daloge.*

***Please create sample to check all these options.***

*We need to write some server code in html, like if you check the grid html you will find special syntax here are all of them and a line about when to use them.* A code nugget is a C# expression placed between the < % and % **>** tags

**Nugget Tag Description**

* **<%** denotes a standard code nugget that contains code statements that are evaluated by the ASP.NET Framework. You must use the Response.Write method in the code nugget if you want to include HTML in the response to the browser.
* **<%=** denotes a content code nugget. Similar to a standard code nugget, but the result is inserted into to the response to the browser without needing an explicit call to Response.Write.
* **<%:** Denotes an encoded code nugget. Similar to < %=, but the response is HTML encoded.
* **<%#** denotes a data-binding code nugget, used to refer to the current data object.
* **<%#:** Denotes an encoded data binding code nugget where the data-bound value is encoded.
* **<%$** A property code nugget. Used to refer to configuration value, such as those defined in Web.config.
* **<%@** Denotes a directive, which is used to configure the Web Form (or control or master page, depending on the kind of directive. We describe directives later in this chapter).

Samajh na aaye to sirf grid wala syntax yaad rakho baki kaam kerte kerte aa jayega.

ASP.NET- Part 2 [Cache and Ajax]

*You must have heard about caching, if not, just think it of a mechanism of storing some data in memory, so that it can be reused again, quickly.. Why do we need caching? Simply to reduce processing time, thus getting better performance.*

*Caching typically web applications me use hoti hai, kyuki waha per multiple users hai to performance matter kerti hai. Caching windows application me bhi hoti hai aur yeh kam log he jante hai. .NET 4 se windows application me bhi caching ka feature hai, kyuki performance windows application ke bhi achi kerna chahoge, bhale he sirf ek user ho. Is note me we will focus on only web applications.*

*Ab tumko yaad hai ki web application ka response html hota hai, to maan lo koi page hai jiska response same aata hai, usko baar baar process kerna, maltab uska object bane, init.. load.. render… yeh sab call ho, isse acha hai ki jo final html banni hai who he mil jaye to request aate he response chala jayega… Dusra case pura page na sahi kuch part hai jo ki same rehna hai, unta html (user control ki html) he mil jaye to bhi performance achi hogi.. third case html na sahi, koi data hai jo ki page me use hota hai who data he ager jaldi se mil jaye (bina db me jaye, ya file se read kiye, ya web service se) to be performance achi hogi.*

*In 3 cases ka ASP.NET me support hai, pehle wale ko* ***Page caching*** *kehte hai jisme pura page cache hota hai,****Fragment Caching*** *for caching part of page/ user control and third* ***Data caching*** *jisme data to cache kerte hai. Ab aur detail me jane se pehle samjho ki caching kya hai aur kis scenario me use ker sakte hai..*

*Dekho most of our sites are dynamic in content, matlab based on user or input, page content change hota hai.But still kuch pages static hote hai, ya fir kuch data over some duration change nahi hota, jaise tumhare site me students ki list, jab tak koi student add, edit, delete na karo who data same he rehna hai, to bar bar lene database tak kyu jao, memory me he save ker lo.*

*Caching ke peeche idea yeh hai ki jo data kuch duration tak change nahi hona hai, jisko bar bar user kerna hai aur usko lane me thoda time lagta hai usko la ker memory me rakh (heap ka ek section). Yeh to basic se baat hai ki koi data hard disk ya database se read kerne se fast hoga ager who data memory me he hai.*

*Caching application level per hoti hai, matlab, jaise session her user ka alag hota hai, caching common hoti hai, yeh obviously sahi bhi hai, jaise student ki list sarey users to jaldi mile to acha hai, koi bhi user specific data session me jayega and ayesa data to sare user use ker sake who cache me jayega..*

Let’s see how to cache complete page..

At very basic put this line in markup (html) of page. <%@ OutputCache Duration="60" VaryByParam="None"%>

OutputCache says that cache this page. Duration tells for how many seconds.

Now see VaryByParams.

You know about querystring.. [http://localhost/someApplication/somPage.aspx?**keyName=KeyValue**&**keyName2=KeyValue2**](http://localhost/someApplication/somPage.aspx?keyName=KeyValue&keyName2=KeyValue2)

You know that based on value of “keyName” & “keyName2” you normally produce different html output..

When we specify VaryByParam="None" we say no matter what is querystring just cache one copy of this page. So for first request whatever are querystrings, based on them specific output html is created by following page life cycle. Then this result is cached. Next time (for duration value) page is requested, page life cycle is not executed, cached result is send as response.

Now let’s see one example..

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

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:Label runat="server" ID="lblTime" />

</div>

</form>

</body>

</html>

In code behind

public partial class Default : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

lblTime.Text = DateTime.Now.ToString();

}

}

Run application, it will display current time.. Refresh page, it will again display latest time, i.e. response will change..

Now add cache

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

<%@ OutputCache Duration="60" VaryByParam="None"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:Label runat="server" ID="lblTime" />

</div>

</form>

</body>

</html>

Now run application, keep on refreshing page, you will get same result i.e time will be not upated, to check more, put breakpoint in page load code, you will find it is not executed for any request other than first request. After duration value, if any request comes for this page, then page is re-executed and again response is cached…

We have not used VaryByParam.. Lets use it

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

<%@ OutputCache Duration="60" VaryByParam="p"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:Label runat="server" ID="lblTime" />

</div>

</form>

</body>

</html>

protected void Page\_Load(object sender, EventArgs e)

{

string strP = Request.QueryString["p"];

switch (strP)

{

case "1":

lblTime.Text = DateTime.Now.AddDays(1).ToString();

break;

case "2":

lblTime.Text = DateTime.Now.AddDays(2).ToString();

break;

case "3":

lblTime.Text = DateTime.Now.AddDays(3).ToString();

break;

default:

lblTime.Text = DateTime.Now.AddDays(4).ToString();

break;

}

}

Now if you open this page and append ?p=1 or ?p=2 or ?p=3 or ?p=4 for example

http://localhost:3740/WebSite2/Default2.aspx?p=2

Then based what value of “p” you pass different result will be displayed (see switch statement). Now focus on

<%@ OutputCache Duration="60" VaryByParam="p"%> we are saying cache different result of this page based on value of “p”. So 4 version of this page will be saved.. For first request of every different parameter, response will be saved, for subsequent requests result from cache will be send back as response.

*There are other parameters in this syntax, most of them are very less used, and however I would suggest that once you have understood everything told above, please go through*

[*http://msdn.microsoft.com/en-us/library/vstudio/hdxfb6cy(v=vs.100).aspx*](http://msdn.microsoft.com/en-us/library/vstudio/hdxfb6cy(v=vs.100).aspx)

Moving to second part, caching a part of response (response of a user control).. Just put same OutputCache directive in ascx html (html part of user control).

*Ab 3rd dekhte hai jo sabse jyada use hota hai. Ek class hoti hai Cache, jo ki per page aur user control me accessible hoti hai, to jab koi data cache me dalna ho..*

Cache["name"] = "John";

*Add kerne ka dusra syntax bhi hai*

Cache.Insert("name", "John");

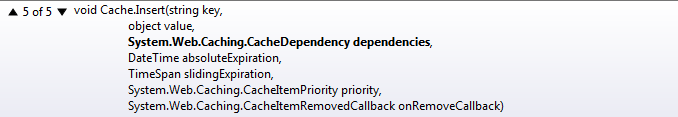
Isme ”name” key hai and “John” value hai, tum kisi bhi type ki value dal sakte ho. Jab value nikalana ho to

string strUserName = Cache["name"].ToString();

*Yeh sab se sasta tikau tareeka hai cache use kerne ka… Ab isse hamare sare case cover nahi hote.. Thoda socho to jaise tumhe student ki list cache ker le, ager sirf duration based rakho to, but beech me kisi ne ek student add/edit/delete kuch bhi ker diya.*

*Utni duration tak tumko updated data nahi milega, kyuki result cache se uth ker aa raha.. Yani tumko ek mechanism chaeye jisse tum duration ke beech me kisi event per cache to referesh ker sako..*

*Ab fir se Cache.Insert to dekhte hai*

**

*5 overload hai iske.. Hum seb se jyada option wala dekhte hai, tum apne need ke hisab se joo sa sahi fit ho who use ker lena.*

*Pehla parameter key hai, jis naam se tumko cache banana hai.*

*Second parameter value jisko cache kerna hai uske liye hai, isliye he object type ka hai..*

*Third parameter dependency ka hai, dependency me tum bolte ki yeh cache kis cheez per dependent hai, yani jisper depend hai ager who change hota hai to yeh cache bekar hai (stale word use hota hai) 3 option hai iske*

* Caching Dependency on key
* Cache Dependency on a File
* Cache Dependency on SQL

*Pehle wale me tum ek cache item ke dependency kisi aur cached item per set kerte ho.. kaise kerte hai yeh thodi der me dekhte hai..Cache Dependecy on File me tum ek file ka naam dete ho, ager who file change hui to cache stale ho jayegi. Cache dependency onSQL thoda lamba process hai, isse tum SQL ki kisi table per dependeny banate ho, jaise he table hange hogi, cache invalid ho jayegi…*

*AbosulteExpiration: The time at which the inserted object expires and is removed from the cache.jo time bataoge us time per cache invalid, ho jayegi, maan lo tumhara system 10 baje kahi se naya data lata hai to la ker cache lo next 10 baje tak.*

*SlidingExpiration: Yeh uper wale ka bhai hai, magar bahut difference hai, Yeh kehta hai ki tum kitne der tak (See data type is timespan not date time) kisi cheez to cache me rakhoge, usme bhi ek extra functionality hai, maan lo tum 20 min tak rakhte ho, iska naam dekho sliding expiration hai, iska matlab ager expire hone se pehle kisi ne read ker liya to yeh agle 20 min tak aur rahegi, ager 20 min tak kisi ne read nahi kiya tab yeh expire ho jayegi.. SlidingExpiration and Abosolute expiration interview me puche jate hai.. Ek baat hamesha dhyan rakha hai ki ager absoluteExpiration use kerna hai to SlidingExpiration me* ***System.Web.Caching.Cache.NoSlidingExpiration*** *hoga.*

*Priority naam se he bata raha kya hai, actully jab memory kam hone lagegi, to low priority items pehle hali honge..*

*Last wala dekhte hai, uper abhi tak humne dependecy dekhi, ki file ka content change ho to cache expire ho jaye, durations dekhi ki itne der baad ya is time per expire ho jaye, ab expire to ho gayi. But tumne cache me latest data bhi to lana hai,who ager automate ho jaye to acha rahe, isi kaam ke liye yeh last wala callback function hai hai. Jaise he cache expire hogi, yaha per likha huwa function callho jayega, ab function parameter ki tarah kaise pass hota hai? Yes delegates to yeh parameter ek delegate type hai.. Infact jaruri nahi ki tum is function se cache menaya data tum sirf notify ker lo, ya log ker lo, ya kuch bhi mat karo..*

*Yeh sari kahani to dalne ki ho gayi.. ab hatana hai to Remove naam ka function hota hai*

if(Cache["Key1"] != null)

Cache.Remove("Key1");

*Now see this one, it uses it all.*

protected void Page\_Load(object sender, EventArgs e)

{

DateTime date = DateTime.Now;

Cache.Insert("Date1", DateTime.Now);

Cache.Insert("Date2"

, date

, null

, DateTime.Now.AddSeconds(40)

, TimeSpan.Zero

, CacheItemPriority.Default

, new CacheItemRemovedCallback(CachedItemRemoveCallBack));

}

private void CachedItemRemoveCallBack(string key, object value, CacheItemRemovedReason reason)

{

if (key == "Date1")

{

Cache.Remove("Date2");

}

}

*Ab socho cache store memory me ho rahi, to jaise he memory bherne lagegi to garbage collector chalega, GC cache ko bhi clean kerta hai. Yeh baat bhi sahi hai ki cache ka kaam to performance badna hai to isse memory le logi to performance degrade hogi but application chalti to rehegi, magar ager GC ne cache ko khali nahi kiya to out of memory exception aayega, isliye memory kam hone per .NET cache to khali kerne ki koshish kerta hai. Kyuki GC kabhi bhi cache khali ker sakta hai isliye hum cache ka kuch bhi use kerne se pehle kerte hi ki who null to nahi hai..*

*Abhi caching ke liye itna kafi hai, itna samajh aa gaya to use ker sakte ho baki ke liye google hai.. Jaise maine nahi bata ki cache dependency on file hoti hai, tumko term pata hai, sare options pata hai google me search karo, you will find*

Cache.Insert("SampleFile", fileContent, new System.Web.Caching.CacheDependency(Server.MapPath("~/SampleFile.txt")));

*Now let’s move another very important part, that is Ajax.*

AJAX

*Abhi tak tumne page life cycle ke bare me padha hai, tum request karoge, page ka object banega, bahut sare event fire honge fir tumko response html milti hai… Ab bahut se page ayese hote hai jinko tumne first time khola, fir kuch kuch select kiya, ya fir kuch value dali, based on tumhare selection or value, page postback huwa and kuch additional content usi page me display kiya. For example tum form me ek drop down do jo user se uski country puche, based on jo country who select kare, state wale dropdown me tum us country ke state fill karana chahoge. Ab is case ko dhyan se samajhte hai, normal postback to tum jante he ho, usme pure page ki html server ko wapas bheji jati hai aur server pure page ka final response tumko deta hai. Ab pure page ki html jyada he hogi.. plus effectively net change dekho ager before postback and after postback to tum dekhoge ki sirf ek dropdown ke options he to change huwe….*

*Yahi ager kisi tarah hum sum server ko choti request bhej sake aura ur who uska response de aur hum us response to hamara jo page ki html brower me khuli hai usme beech me fit ker sake to hamara kaam fast ho jayega, plus user experience bhi acha hoga pura page bar bar load hone ki jagah content refresh ho jayega, ajax ke benefit tumko samjhane ki jarurat nahi hai, google map ho ya facebook her jagah experience ker chuke hoge ☺*

*\*Not so important para – Ajax aane se pehle Servers to pura page execute kerna jante they and browser bhi pura page upload kerna jante the.. Jab ajax aaya to servers ka code bhi update kiya gaya aur browsers ko bhi yeh update diya gaya ki kaise process kerna hai, to server, browser, ya tumhara code, jab bhi kuch nayi concept aayegi to update hota rahega..*

*Ab ajax use kerne ki option dekhte hai*

* *ASP.NET Update Panel*
* *Javascript / Jquery*

**ASP.NET Update Panel**

*UpdatePanel ek special control hai, jiska content tum bina full page postback ke refresh ker sakte ho, yeh bhi behind the scene ajax hit he marta, bas fayeda itna hai ki tumko code nahi likhna padta, ajax ke liye jo javascript chaeye woh yeh sab include ker leta hai.*

*Ek page me ek “ScriptManager” naam ka control dalo aur ek ya use jyada UpdatePanels dal lo (updatePanel ko page per scriptManager chaeye hota hai). Jayada gyan tapkane se acha hai ki direct example dekhte hai.*

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Home.aspx.cs" Inherits="BISDashboard.Home" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

Script manager

<body>

Update panel. Iske under jo control hai sirf unka html update hoga, ajax response per. Iske bahar wale control bina full page post back ke update nahi honge..

<form id="form1" runat="server">

<div>

<asp:ScriptManager EnablePartialRendering="true" runat="server" />

<asp:Label runat="server" ID="lblTime1" />

<asp:UpdatePanel runat="server" ID="pnlUpdate">

<ContentTemplate>

<asp:Label runat="server" ID="lblTime2" />

<asp:Button runat="server" ID="btnUpdate" onclick="btnUpdate\_Click" Text="Update"/>

</ContentTemplate>

</asp:UpdatePanel>

</div>

</form>

</body>

</html>

public partial class Home : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

lblTime1.Text = DateTime.Now.ToString();

lblTime2.Text = DateTime.Now.ToString();

}

protected void btnUpdate\_Click(object sender, EventArgs e)

{

lblTime2.Text = DateTime.Now.ToString();

}

}

*Run karo two jagah time dikhega, fir button ko dabao, ek to page postback nahi hoga, dusra sirf second wale label ki date change hogi, kyuki wohi updatepanel ke ander hai.. Code samajhna aur use kerna bahut asan hai, tumko ek script manager dalna hai aur ek ya use jyada (according to need) updatepanels dalne hai, baki code normal tareqe se ker do.*

*Bas dhyan is baat ka rakhna ki html sirf unhe controls ki update hogi jo update panel se under hai. Hamare pehle example me, hum country wala dropdown and state wala dropdown update panel ke under rakhenge, country wala jab change hoga to uske change ker state wala fill karadenge.*

*Code ko ager hum yeh ker de*

public partial class Home : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

lblTime1.Text = DateTime.Now.ToString();

lblTime2.Text = DateTime.Now.ToString();

}

protected void btnUpdate\_Click(object sender, EventArgs e)

{

lblTime1.Text = DateTime.Now.ToString();

lblTime2.Text = DateTime.Now.ToString();

}

}

*To bhi button ke click per sirf second wala he update hoga kyuki sirf second wala update panel ke ander hai.*

*Abhi tak ki kahani ager samajh aayi ho to socho ki update panel ka content update kab ho raha, jab uske under ka koi control ka koi server side event call ho raha, hamare example me button ka click tha, tum drop down ka selected index change ke use ker sakte ho. Yeh wali baat most of cases me fit baith jati hai, magar kabhi kabhi situation ayese bhi aati hai ki kis control ke event per update panel update kerna chahete ho, woh control update panel ke bahar hai.*

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

Ager multiple update panels hai to updatemode conditional ker do, warna ek ke referesh per dusra bhi apne se ho jayega.

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:ScriptManager EnablePartialRendering="true" runat="server" />

<asp:Label runat="server" ID="lblTime1" />

<asp:UpdatePanel runat="server" ID="pnlUpdate" UpdateMode="Conditional">

<ContentTemplate>

<asp:Label runat="server" ID="lblTime2" />

<asp:Button runat="server" ID="btnUpdate" OnClick="btnUpdate\_Click" Text="Update" />

</ContentTemplate>

</asp:UpdatePanel>

<asp:UpdatePanel runat="server" ID="pnl2" UpdateMode="Conditional">

<ContentTemplate>

<asp:Label runat="server" ID="lblTime3" />

<asp:Button runat="server" ID="btnUpdate2" Text="Update 2" OnClick="btnUpdate2\_Click" />

</ContentTemplate>

</asp:UpdatePanel>

</div>

</form>

</body>

</html>

public partial class Home : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

lblTime1.Text = DateTime.Now.ToString();

lblTime2.Text = DateTime.Now.ToString();

lblTime3.Text = DateTime.Now.ToString();

}

protected void btnUpdate\_Click(object sender, EventArgs e)

{

lblTime1.Text = DateTime.Now.ToString();

lblTime2.Text = DateTime.Now.ToString();

}

protected void btnUpdate2\_Click(object sender, EventArgs e)

{

lblTime3.Text = DateTime.Now.ToString();

}

}

*Ab dekho isme do update panel hai btnUpdate2 se 3rd wala label refresh ho raha. Isko run karo pehla button daba0 uske samne wala time referesh hoga, kyuki wahi pehla update panel referesh huwa. Dusra button dabao, uske samne wala time refersh hoga kyuki dusra wala updatepanel refresh hoga.*

*Ab maan lo humko dusre wale button per dono update panel refresh karane ho… html ko thoda update karo “Trigger” lagao*

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<title></title>

</head>

<body>

<form id="form1" runat="server">

<div>

<asp:ScriptManager EnablePartialRendering="true" runat="server" />

<asp:Label runat="server" ID="lblTime1" />

<asp:UpdatePanel runat="server" ID="pnlUpdate" UpdateMode="Conditional">

<ContentTemplate>

<asp:Label runat="server" ID="lblTime2" />

<asp:Button runat="server" ID="btnUpdate" OnClick="btnUpdate\_Click" Text="Update" />

</ContentTemplate>

Trigger, isme control and uske kaun se event per panel refresh kerna hai yeh batate hai, dekho btnUpdate2 is panel se bahar hai

<Triggers>

<asp:AsyncPostBackTrigger ControlID="btnUpdate2" EventName="Click" />

</Triggers>

</asp:UpdatePanel>

<asp:UpdatePanel runat="server" ID="pnl2" UpdateMode="Conditional">

<ContentTemplate>

<asp:Label runat="server" ID="lblTime3" />

<asp:Button runat="server" ID="btnUpdate2" Text="Update 2" OnClick="btnUpdate2\_Click" />

</ContentTemplate>

</asp:UpdatePanel>

</div>

</form>

</body>

</html>

By default, any postback control inside an [UpdatePanel](http://www.asp.net/AJAX/documentation/live/mref/T_System_Web_UI_UpdatePanel.aspx) control causes an asynchronous postback and refreshes the panel's content. However, you can also configure other controls on the page to refresh an [UpdatePanel](http://www.asp.net/AJAX/documentation/live/mref/T_System_Web_UI_UpdatePanel.aspx) control. You do this by defining a trigger for the [UpdatePanel](http://www.asp.net/AJAX/documentation/live/mref/T_System_Web_UI_UpdatePanel.aspx)control. A trigger is a binding that specifies which postback control and event cause a panel to update. When the specified event of the trigger control is raised (for example, a button's [Click](http://msdn2.microsoft.com/en-us/d02sce24) event), the update panel is refreshed.

*Ab fir se run karo,pehle button per uske samne wala (2nd label) referesh hoga.. dusre button se 2nd and 3rd label refresh honge..*

*Thoda aur practice kerne ho to Update panel ke ander update panel daal ker dekho..* **Aur jyada gyan ke liye**

<http://www.asp.net/AJAX/documentation/live/overview/UpdatePanelOverview.aspx>

*Update panel se tumhara ajax ka kafi part cover ho jayega, but hum jquery se bhi dekhte hai kyuki who bahut use hota hai..*

***Note: Please note ki even ajax request ho, update panel se, but page ka load hamesha call hota hai.. magar usme koi control ki kuch bhi value(html ) set who refresh nahi hogi ager who control update panel ke under nahi hai.***

*Ab Jquery dekhte hai kyuki sab se jyada wahi use karoge. Javascript wala tareeka bahut purana ho gaya, usko ab koi nahi use kerta (but keep in mind ki jquery ultimately use to wahi kerti hai)*

function ShowStudents() {

$.ajax({

type: "POST",

url: "AjaxPage.asmx/GetStudents",

data: '{Id: "' + 0 + '" }',

contentType: "application/json; charset=utf-8",

dataType: "json",

success: OnGetStudentSuccess,

failure: function (response) {

alert(response.d);

}

});

}

*Yeh ek sample jqery ka function hai jo ajax ke through data la raha, isko samajhte hai*

* Type is request type it will be GET or POST, use get if you are fetching data, use POST if you are submitting data by ajax.
* url is url of server function that will give you the result. (page\_name/ function\_name or service\_name/function\_name
* data is parameter of function to be called.
* ContentType is format of request, in most of cases it will be JSON (*I will tell about JSON after this section).*
* Datatype is format of response
* Success : write a function name in front of it, that function will be called, when you ajax response is returned from server. This function is normally used to handle response of ajax response.
* Failure: write a function name in front of it, that function will be called when you ajax request fails, or there is some error.

*Yeh function hum use karenge response handle kerne ke liye.*

function OnGetStudentSuccess(response) {

var students = response.d;

$('#tbody').empty();

$.each(students, function (index, std) {

var eachrow = "<tr>" + "<td>" + std.Name + "</td>";

if (std.Age > 50) {

eachrow = eachrow + '<td class="aqua">' + std.Age + "</td>";

}

else {

eachrow = eachrow + '<td class="silver">' + std.Age + "</td>";

}

eachrow = eachrow + "</tr>";

$('#tbody').append(eachrow);

});

}

*Ab overall picture ko samajhte hai.. Uske liye shuru kerte hai JSON se.*

*Wikipedia se “***JavaScript Object Notation**, is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is used primarily to transmit data between a server and web application, as an alternative to XML.”

*Main point itna sa hai ki XML ka size jyada hota hai, to logo in JSON bana liya iska same data ke liye size chota hota hai*

*Ager ek person ka data transfer kerna ho to JSON format me who kuch is traha ka banega*

{

"firstName": "John",

"lastName": "Smith",

"isAlive": **true**,

"age": 25,

"height\_cm": 167.64,

"address": {

"streetAddress": "21 2nd Street",

"city": "New York",

"state": "NY",

"postalCode": "10021-3100"

},

"phoneNumbers": [

{ "type": "home", "number": "212 555-1234" },

{ "type": "office", "number": "646 555-4567" }

]

}

*Obivoulsy this is much smaller than XML (which you can understand)*

*Ab samajhte hai.. Tumko ajax hit marne hai, matlab pura page post back kiye bina data lana hai..Data kaha hai? Server per.. yani server side ka koi function call kerna hai, server side me code kaha ho sakta hai? Ek simple class file me, ek webpage me, ya fir ek webservice me..*

*Ab class me likhoge to object kaise banaoge? Aur static bhi bana liya to bhi call kaise connect hogi, beech me to ASP.NET baitha hai, ASP.NET tumko ek traeega deta hai direct function call marne ka who hai web service iske alawa koi nahi hai..*

*To even ager tum aspx page ka bhi koi function call kerna chahete ho, to bhi tumko us method ko as a service method expose kerna hoga.. nahi samajh aaya to dubara b padho aur fir bhi nahi aaya to yeh yaad rakho ki ager tumko jquery se koi server function call kerna hai to tum us function ko ayesa banaoge*

[WebMethod]

[ScriptMethod(UseHttpGet = true,ResponseFormat = ResponseFormat.Json)]

public string MyFuncToBeCalledFromJquery()

{

//body

}

*WebMethod laane se yeh service ka part ban jayega, aur ScriptMethod lagane se is script se (jquery) se call kiya ja sakta hai.*

*Kisi bhi page me ayesa method bana lo, better hota hai ki ek .asmx (web service purane jamane wali) file add ker lo aur ajax wale sare method usme likh lo.. Itna likh ker tumhara ek method ready hai server per jisko jquery se call ker lo. Ab jquery wala part banate hai.*

function ShowStudents() {

$.ajax({

type: "POST",

url: "AjaxPage.asmx/GetStudents",

data: '{Id: "' + 0 + '" }',

contentType: "application/json; charset=utf-8",

dataType: "json",

success: OnGetStudentSuccess,

failure: function (response) {

alert(response.d);

}

});

}

*Yeh function fir se dekho hamare asmx ya web page per ek method hoga GetStudents naam ka (yes with Webmethod and Script Method). Who id naam ka parameter leta hoga, nahi leta to data wali line hata do. Bus itna he chaheye jqeury se call kerne ke liye.. Jab tum kisi event per Show Student function call karoge (client side) per to yeh ajax request marega server ko GetStudent call hoga aur result JSON format me wapas aa jayega…*

*Ab question hai ki JSON format ke result a use kaise kare..*

function OnGetStudentSuccess(response) {

var students = response.d;

$('#tbody').empty();

$.each(students, function (index, std) {

var eachrow = "<tr>" + "<td>" + std.Name + "</td>";

if (std.Age > 50) {

eachrow = eachrow + '<td class="aqua">' + std.Age + "</td>";

}

else {

eachrow = eachrow + '<td class="silver">' + std.Age + "</td>";

}

eachrow = eachrow + "</tr>";

$('#tbody').append(eachrow);

});

}

*Isko dekho yeh function call hota hai jab hamari ajax request ka result aayega.. yeh ek parameter le raha (kuch bhi naam rakhlo is param ka).. usme d naam se ek value milege usko pad lo, wahi actual JSON result hai. Hamara GetStudents function maan lo list of students deta tha, to students variable me who list aa gayi (in JSON). Ab hum isko table format me dikhana chaehte hai.*

*ab* $('#tbody').empty();

*iska matlab hai ki wohcontrol dhundho kiski id tbody hai, aur uske under ke html hata do..*

$.each(students, function (index, std) {

*Yeh keh raha ki jo JSON list hai students naam se, uske her ek element me loop karo (kind of for loop), yaha std variable me her loop me current student aa jayega.*

var eachrow = "<tr>" + "<td>" + std.Name + "</td>";

if (std.Age > 50) {

eachrow = eachrow + '<td class="aqua">' + std.Age + "</td>";

}

else {

eachrow = eachrow + '<td class="silver">' + std.Age + "</td>";

}

eachrow = eachrow + "</tr>";

*in line me hum her student ke liye nahi tr, td bana ker uski property display kara rahe, logic yeh bhi hai ager age 50 se jyada hai to color dusra use karo ( all this just to show power of jquery)*

*ab eachROw naam ke variable me ek row ki html ready hai*

$('#tbody').append(eachrow);

*Yeh keh raha ki tbody id wale control me yeh html append kerdo..*

*Is example se tum dekho, ko kaam browser kerta hai, who kaam hum explicitly ker rahe, server se data la ker UI paint ker rahe, jo jaise chahe waise UI bana lo!!! POWER ☺*

*Thoda kam samajh aaye to no tension, do baar use karoge, clear ho jayega. Isko samajhne ka best tareeqa hai ek example mai run kerke dikha dunga, yaha samjhane se jyada samajh nahi aayega.. And anyways jo maine BISDashboard ka code diya tha, who yahi use kerta, please check that code.. baki detail me mai sajha dunga..*

*Ab ek baat aur.. sirf JSON data nahi aata, tum HTML, XML, TEXT aur bahut sara format ka data la sakte ho, infact jquery me bahut se method hai ajax request marne ke, but this most basic and most usefull.*