فصل سوم :
معرفی کنترل های HTML و نحوه کاربرد آنها در صفحات ASP.NET

مقدمه :

کنترل های HTML سرور در حقیقت عناصر استاندارد HTML هستند که در سرور پردازش می شوند. تمام کنترل های سرور HTML (که بعنوان کنترل های HTML هم شناخته می شوند) دقیقا معادل یک عنصر HTML تفسیر و اجرا شده و خواص اغلب آنها با عناصر HTML یکسان است.

در طی فصول آتی ما از این کنترل ها به ندرت استفاده خواهیم کرد! کنترل های وب توانایی های بسیار بیشتری را ارائه می دهند و ادامه ی کار تقریبا با آنها تکمیل می گردد و همراه خواهد بود. این فصل صرفا برای یادآوری اسکریپت نویسی کلاینت ساید با دید VS.NET مفید می باشد و تاثیری آنچنانی بر روی برنامه نویسی سمت سرور ما در فصول آتی نخواهد داشت. این فصل در حقیقت یک نوع DHTML نویسی به سبک VS.NET می باشد.

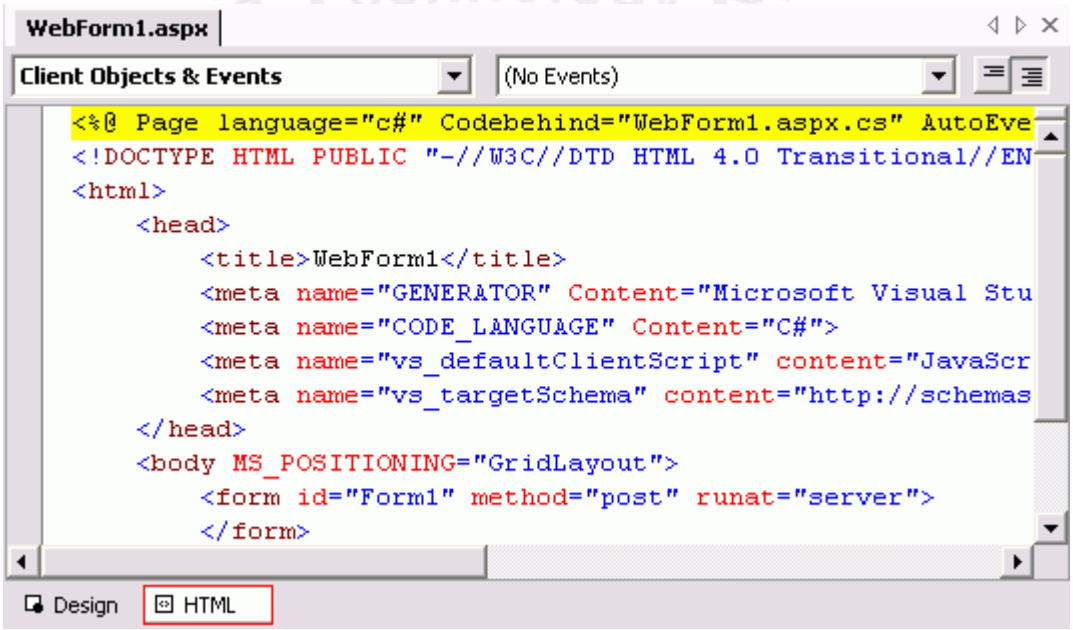
پردازش در خواست ها از طرف سرور

کنترل های HTML در فضای نام System.Web.UI.HtmlControls تعریف شده اند. شما یک کنترل HTML را در اغلب حالتها با اضافه کردن ویژگی "RUNAT="Server به تگ آن ، می توانید ایجاد کنید. با فراموش شدن این مورد تمام قابلیت های پردازشی سمت سرور را از دست خواهید داد. بهتر است به

هر کنترل HTML یک ID منحصر بفرد اختصاص داده شود تا بتوان به سادگی در برنامه به آن رجوع کرد.

کنترل های سرور HTML از کلاس HTMLInput مشتق شده اند و باید درون کنترل HtmlForm قرار گیرند.

با استفاده از کنترل HTMLForm می توان در خواست های رسیده به سرور را پردازش کرد. این کنترل همانند form معمولی در صفحات HTML است بعلاوه اینکه ویژگی "RUNAT="Server نیز به آن اضافه می شود. به صورت خودکار وقتی یک پروژه جدید را در ویژوال استودیو باز می کنید اینکار از طرف VS.NET صورت می گیرد (شکل ۱) و لازم به ذکر است که در هر فایل ASPX شما فقط یک فرم را می توانید تعریف کنید (برخلاف نگارش های قبلی آن) .



شکل ۱- نمایی از پشت صحنه یک فرم وب که در آن ایجاد فرم به صورت خودکار صورت می گیرد.

اگر شما به تگ فرم در VS.NET توجه کنید مواردی را ملاحظه می نمایید که باید مرور شوند. ویژگی Method به مرور می گوید که چگونه اطلاعات را به سرور بفرستد. اگر مساوی GET قرار گیرد ، داده ها به صورت یک رشته به URL اضافه شده و فرستاده می شوند و یا به صورت یک درخواست HTTP که در این حالت مساوی Post خواهد بود.

چند خاصیت دیگر هم وجود دارند که به صورت پیش فرض در محیط VS.NET ظاهر نمی شوند. برای مثال خاصیت Disabled که پیش فرض آن False است و اگر True شود تمام کنترل های متعلق بـه فـرم غیر فعال شده و به صورت خاکستری نمایش داده می شوند.

خاصیت دیگر Action است که در برنامه های ASP.NET نیازی به تنظیم کـردن آن وجـود نـدارد. در نگارش های قبلی ASP از این خاصیت برای تنظیم اینکه داده ها به چه صفحه ای فرستاده شوند ، استفاده می شد.

مروری سریع بر کنترل های HTML

برای اینکار دو راه وجود دارد: ۱- کد نویسی مستقیم و ۲- استفاده از محیط ویـژوال. بـدیهی اسـت کـه مورد دوم ساده تر بوده و نیازی به محفوظات پیشین ندارد. اگر به ToolBox ویژوال استودیو در سـمت چپ صفحه توجه کنید چندین Tab مختلف برای طبقه بندی کنترل ها وجود دارد. تمام کنتر ل هـای مـورد بحث ما در این فصل در Tab ایی به نام HTML قرار گرفتـه انـد. اگـر بـا Visual InterDev کـار کـرده باشید این Tab برای شما تازگی ندارد!

به راحتی می توان از این Tab یک دکمه (Button) را روی صفحه قرار داد و بقیه کنترل ها هم بـه همـین صورت هستند. بنابراین ذکر نکات مهم و کاربردی این کنترل ها برای فصل جاری لازم و مکمل می باشند.

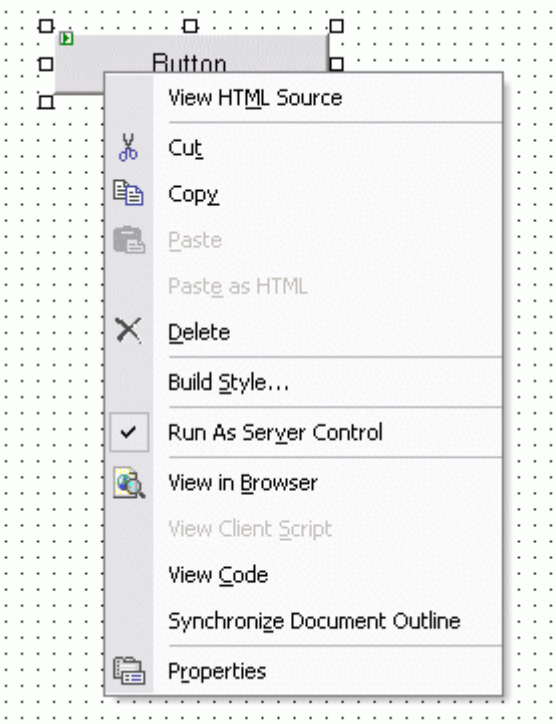
اگر به پایین صفحه سمت چپ نگاه کنید (در محیط VS.NET) می توانید بین حالت Design و مشاهده کد HTML یکی را انتخاب نمایید (شکل ۲). با انتخاب کردن حالت HTML کدی را که VS.NET بـرای قـرار دادن یک دکمه روی صفحه نوشته است را می توان ملاحظه کرد.

شکل ۲- انتخاب کردن حالت طراحی و یا مشاهده سورس صفحات در VS.NET .

نکات مهم مربوط به این کنترل به شرح زیر هستند:

اگر به سورس این کنترل دقت کنید و یا به صفحه ی خواص در گوشه ی سمت راست صفحه هنگامیکه کنترل <Input> یا همان دکمه ما انتخاب شده است دقت نمایید ، خاصیت Type آنرا می توان تنظیم کرد. پیش فرض آن Button است (در این حالت) . اگر به Submit تغییر نوع پیدا کند، می توان بوسیله ی آن اطلاعات را به سرور فرستاد و اگر به Reset تغییر یابد می توان برای ریست کردن تمام کنترل های روی صفحه از آن استفاده کرد. تا هنگامیکه خاصیت RUNAT="Server" به تگ آن اضافه نشود این کنترل کلاینت ساید بوده و فشردن آن باعث ایجاد رخدادی در این سمت می گردد. اضافه کردن رخداد برای این کنترل و کنترل های HTML هم دقیقا مانند اسکریپت نویسی JavaScript می باشد.

برای اینکه این کنترل بعنوان یک کنترل سرور بتواند عمل کند همانطور که ذکر شد یا می توان خاصیت RUNAT="Server" را به صورت دستی وارد کرد و یا روی آن کلیک راست کنید و گزینه ی Run as server control را انتخاب نمایید (شکل ۳) . حالا با دوبار کلیک کردن بر روی آن صفحه ی Code Behind باز شده و می توانید در رخداد ServerClick آن کدتان را بنویسید.



شکل ۳- چگونگی تبدیل یک کنترل معمولی HTML به کنترل سرور.

اگر کاربر از مرورگر نگارش ٤ به بالا استفاده می کند می توان از تگی به نام <Button> هم استفاده کرد.

اگر نوع Type را مساوی Image قرار دهید می توان از کنترل تصاویر گرافیکی بجای دکمه استفاده کرد و با کمک رخداد onMouseOver و onMouseOut تصاویر آنها را هنگام نزدیک شدن ماوس تغییر داد.

از کنترل Image هم می توان برای نمایش دادن تصاویر استفاده کرد و با برنامه نویسی در زمان های لازم تصاویر آنرا عوض کرد. اگر خاصیت ALT آن مقدار دهی شود ، یک ToolTip هنگام نگه داشتن ماوس روی آن نمایش داده می شود.

اگر نوع Type را مساوی Text قرار دهیم یک TextBox معمولی حاصل می شود و اگر نوع آنرا مساوی Password قرار دهید این TextBox هنگام نوشتن حروف کلمه ی رمز ، ستاره نشان می دهد. برای نشان دادن و دریافت یک TextBox که MultiLine باشد در اینجا از کنترل TextArea استفاده می شود.

برای دریافت ورودی از نوع Boolean از کنترل CheckBox استفاده می کنیم و اگر خاصیت Checked آن True باشد به معنای انتخاب آن از طرف کاربر است.

برای تعریف RadioButtons در اینجا که برای انتخاب یک گزینه از بین یک گروه بکار برده می شود می توان از کنترل RadioButton استفاده کرد. هر کنترلی در این حالت باید یک ID منحصربفرد داشته باشد اما Name آنها باید یکی باشد تا بتوان به راحتی با آنها کار کرد. برای دریافت اطلاعات از آنها برای مثال می توان Radio[0].Checked را چک کرد.

استفاده از **List** DropDown راه دیگری در مقابل RadioButton است هنگامیکه تعداد گزینه های ما خیلی زیاد باشند. برای Bind کردن آن یک آرایه به آن می توان از خاصیت DatSource و DataBind آن استفاده کرد. با استفاده از خاصیت SelectedIndex می توان گزینه ای را که کاربر مشخص کرده ،

دریافت کرد. اگر می خواهید کاربر چندین آیتم را با هم در این کنترل انتخاب کند خاصیت Multiple آنرا True کنید.

برای ایجاد یک لینک از Anchor Tag ایی به نام <a> استفاده می شود که در خاصیت HRef آن آدرس لینک قرار داده می شود.

کنترل مهمی که در این مجموعه قرار دارد و در کنترل های وب یافت نمی شود مربوط به Upload کردن فایل ها به سرور است که باید راجع به آن قدری بیشتر توضیح داد.
با استفاده از کنترل HTMLInputFile کاربران می توانند فایل های خود را به سرور Upload کنند. برای اینکار تگ مربوط به form را باید در سورس HTML بهبود بخشید و عبارت Encrypt="MultiPart/form-data" را به آن اضافه نمود. این خاصیت سبب می شود تا مرورگر متوجه این مطلب گردد که یکی از کنترل های روی صفحه برای Upload کردن فایل به سرور بکار گرفته می شود. برای اینکار از تگ <Input> با نوع File یعنی Type="File" استفاده می گردد. بعلاوه گزینه RUNAT="Srever" هم لازم می باشد. پس از انتخاب فایل از طرف کاربر و فشردن دکمه Submit بقیه کار باید به صورت برنامه نویسی انجام شود.

```
fileSuggestionsControl.PostedFile.SaveAs(...);
```

نکته :

در زبان سی و مشتقات آن مسیر دایرکتوری ها همراه با \\ آورده می شود یعنی بجای c:\temp\ می نویسیم c:\\temp\\

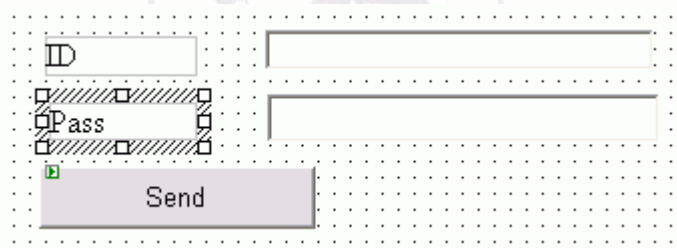
برای اینکه اطلاعات ارائه شده در این فصل جنبه ی کاربردی پیدا کنند ، کنترل های یاد شده را به صورت چند برنامه کوچک مورد استفاده قرار خواهیم داد تا نکات مربوط به آنها در عمل تجربه شوند و

خصوصا در هنگام برنامه نویسی اسکریپتی کلاینت ساید در طی فصول آتی (در صورت لزوم) آشنایی لازم با پشت صحنه فرم های وب حاصل گردد.

برنامه اول :

می خواهیم یک صفحه ی لاگین بسیار ساده درست کنیم که از کاربر شناسه ی کاربری و رمز عبور را خواسته و پس از کلیک بر روی دکمه Submit ، در سرور مشخص شود که آیا این کاربر و کلمه ی رمز او معتبر است یا خیر.

برای اینکار یک پروژه جدید در VS.NET باز کنید و روی صفحه با استفاده از کنترل های HTML ، ۲ لیبل و ۲ عدد TextFiled قرار دهید بعلاوه یک Button (شکل ٤) .



شکل ٤- طراحی فرم مربوط به صفحه ی لاگین

اگر دقت کرده باشید می توان با ۲ یا ۳ بار کلیک کردن روی Label ها آنها را به حالتی در آورد که بتوان داخل آنها عنوان دلخواهی را تایپ کرد. عنوان این دو لیبل را به ID و Pass تغییر دهید.
با استفاده از پنجره Properties خاصیت ID تکست فیلدها را به txtID و txtPass تغییر دهید و همچنین ID دکمه را به btnSubmit . در پروژه های بزرگ نامگذاری صحیح کنترل ها به شدت کنترل برنامه را ساده می کند پس هیچگاه از نام های پیش فرض استفاده نکنید.

عنوان دکمه را هم به Send تغییر دهید. برای اینکار باید از پنجره خواص خاصیت Value را به Send تنظیم کنید. نوع txtPass را به Password تغییر دهید (در سورس صفحه) .

روی تک تک کنترل ها می توان کلیک راست کرد و خاصیت Run as server control را انتخاب نمود.

حالا بر روی دکمه که خاصیت اجرا بر روی سرور را پیدا کرده دوبار کلیک نمایید و در صفحه Code Behind در داخل تابعی که رخداد کلیک را نمایش می دهد کد زیر را بنویسید.

```
private void btnSubmit_ServerClick(object sender, System.EventArgs e)
{
        if (txtID.Value =="a" && txtPass.Value =="b" )
                Response.Write("Ok!");
        else
                 Response.Write("Try again!");
}
```

در مورد اشیاء ذاتی ASP مانند Response در طی فصول آتی بیشتر بحث خواهد شد.

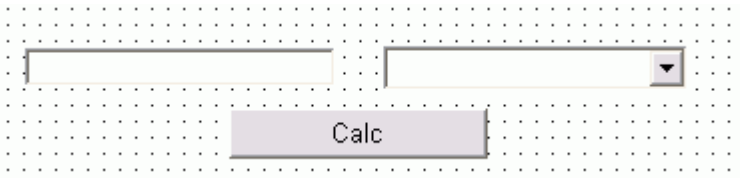
بدیهی است که چک کردن پسورد به این صورت نباید در داخل کد قرار گیرد و این مورد صرفا مثالی از کاربرد کنترل های HTML بودند.

برنامه دوم :

تمام پردازش ها در این برنامه کلاینت ساید بوده و در اینجا می خواهیم کاربر با استفاده از یک تکست باکس عددی را وارد نموده و سپس با کلیک کردن بر روی دکمه calc با یک MessageBox حاصلضرب عدد در یک عدد انتخاب شده از DropDown List نشان داده شود.

یک TextFiled و یک DropDown List و یک دکمه روی فرم قرار دهید (شکل ۵) . Id دکمه را به btnCalc ، Id تکست فیلد را به txtNo و Id کنترل DropDown را به ddlNo2 تغییر دهید.

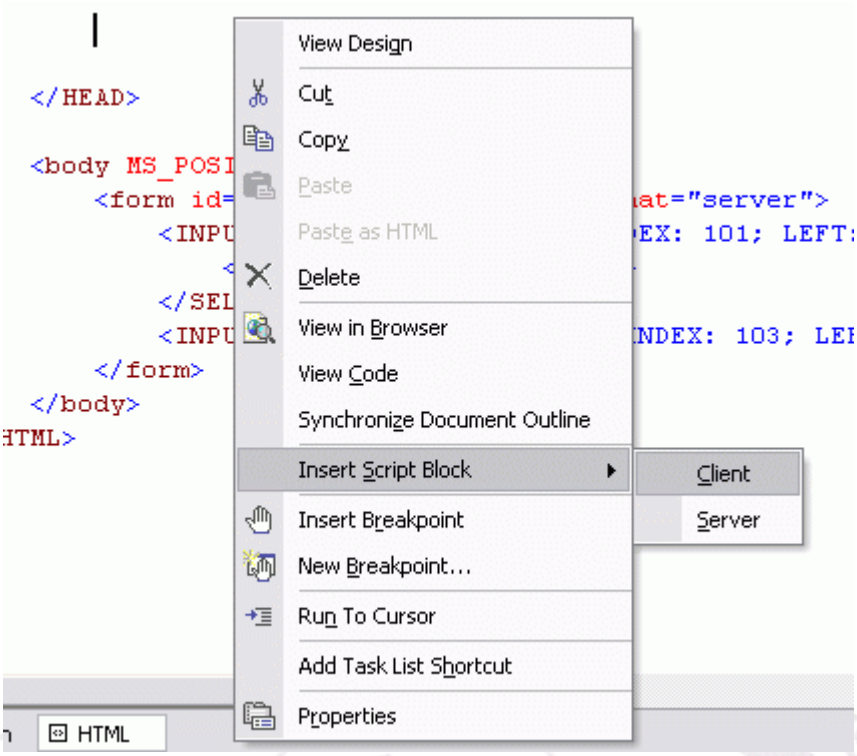
در پایین صفحه روی قسمت HTML برای مشاهده سورس صفحه کلیک کنید.



شکل ۵- ظاهر فرم برنامه دوم در حالت طراحی.

قبل از پایان تگ مربوط به Head یعنی </Head> روی صفحه کلیک راست کنید. از منوی ظاهر شده Insert script block و سپس Client را انتخاب کنید (شکل ٦).

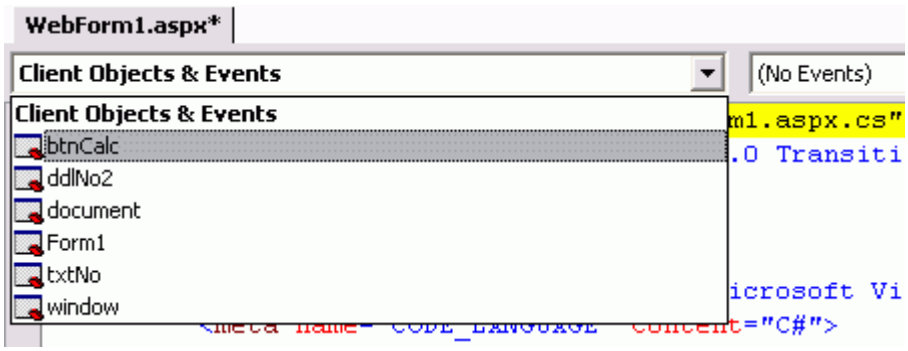
به صورت اتوماتیک ساختار اسکریپت نویسی ما تشکیل می شود. حالا مکان نما را داخل محدوده اسکریپت قرار دهید و از منوی پایین افتادنی بالای صفحه که اولین آیتم آن Client Objects & Events است گزینه ی btnCalc یعنی همان دکمه ی روی صفحه را انتخاب نمایید (شکل ۷) . حالا در منوی پایین افتادنی سمت چپ صفحه رخدادهای مربوط به دکمه نمایش داده شده اند. گزینه ی onClick را از آن انتخاب نمایید (شکل ۸) .



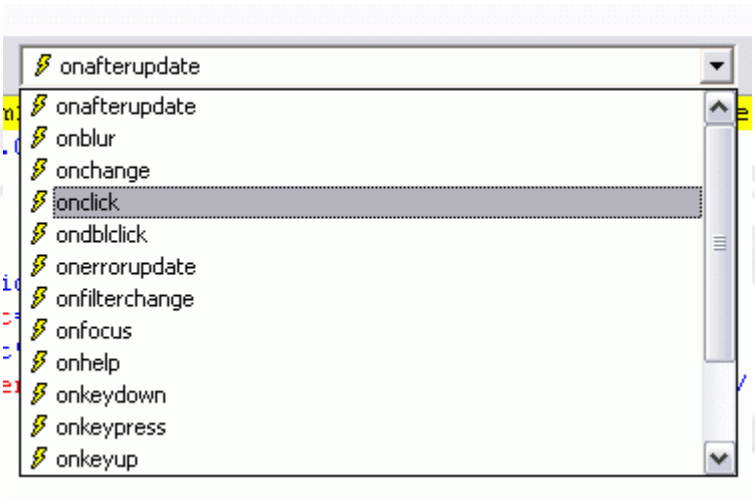
شکل ٦- نحوه ی استفاده از ابزارهای ویژوال برای سهولت بیشتر در کد نویسی.

به صورت خودکار تابعی که رخداد کلیک را بیان می کند ایجاد می شود و هر دستوراتی که داخل بدنه ی تابع btnCalc_click بنویسید هنگام کلیک شدن بر روی دکمه اجرا می شود. کسانی که قبلا از محیط های غیر ویژوال برای انجام اینکار استفاده می کردند ، می دانند چه نعمت بزرگی به VS.NET اضافه شده است! حالا اگر به تگ مربوط به دکمه هم دقت کنید به صورت خودکار عبارت زیر به آن اضافه شده است.

language="javascript" onclick="return btnCalc_onclick()"



شکل ۷- استفاده از ابزارهای ویژوال برای ایجاد روتین های مربوط به رخدادهای اشیاء.



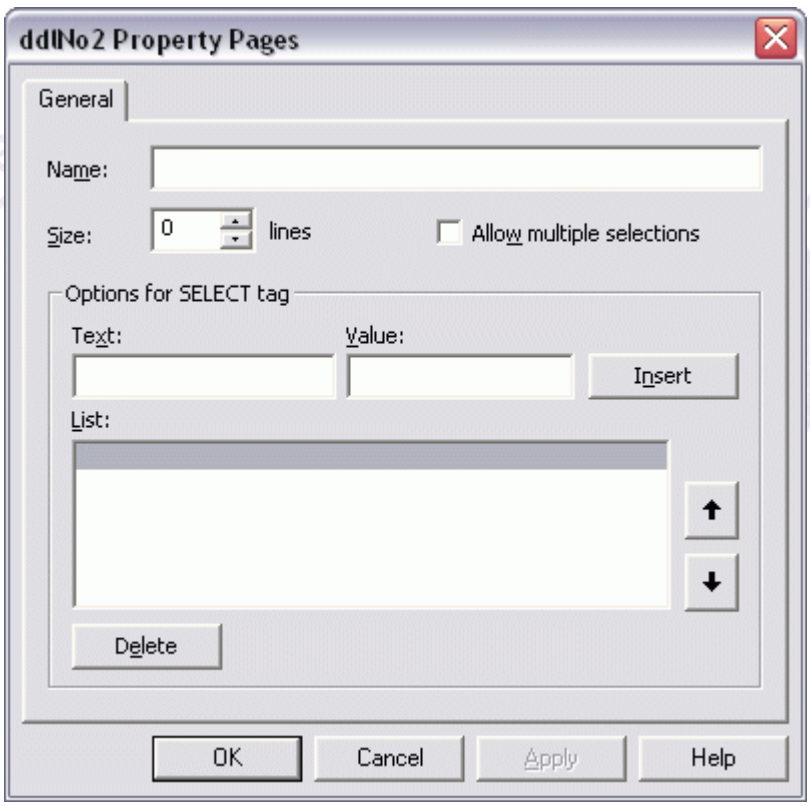
شکل ۸- منویی که لیست رخدادهای وابسته به یک دکمه را نمایش می دهد.

می خواهیم در رخداد onload صفحه ، dropdown list را پر کنیم. برای این منظور به همان ترتیبی که برای اضافه کردن تابع مربوط به رخداد کلیک عمل کردن ، از منوی پایین افتادنی سمت چپ window را انتخاب می کنیم و از منوی پایین افتادنی سمت راست ، گزینه ی onload را. به صورت خودکار تابع window_onload اضافه می شود و اگر به تگ مربوط به body دقت کنید ملاحظه می کنید که عبارت زیر هم به صورت خودکار به آن اضافه شده است .

```
<body MS_POSITIONING="GridLayout" language="javascript" onload="return window_onload()">
```

اگر در برنامه ای لازم شد یک سری عناصر ثابت را به dropdown list اضافه کنیم روی آن کلیک راست کنید و گزینه خواص را انتخاب کنید. در صفحه ی باز شده (شکل ۹) ، می توان آیتم های جدید را اضافه کرد و وقتی می خواهیم آیتمی را با برنامه نویسی به این نوع dropdownlist اضافه کنیم باید با استفاده از شیء سازنده option اینکار را انجام دهیم یعنی :

```
newOption = new Option(optionText,optionValue,defaultSelected,selected);
```



شکل ۹- صفحه ای که امکان اضافه کردن عناصر استاتیک را به dropdown list فراهم می کند.

کد مربوط به بارگذاری صفحه :

```
function window_onload() {

  var i,j=0;

  for(i=0 ; i<=100 ; i+=10)
  {
    dd = new Option(i,i);
    window.Form1.ddlNo2.options[j]=dd;
    j++;
  }

}
```

کد مربوط به رخداد کلیک روی دکمه :

```
function btnCalc_onclick() {

 window.alert(  window.Form1.ddlNo2.value  * window.Form1.txtNo.value  );

}
```

در این فصل نحوه ی برنامه نویسی کلاینت ساید و یا همان اسکریپت نویسی سنتی را با هم مرور کردیم و دیدیم که در این محیط جدید چقدر این نوع برنامه نویسی ساده تر و لذت بخش تر شده است و از اینجا به بعد در طی فصول آتی اگر لازم بود اسکریپت نویسی کلاینت ساید را در موارد معدودی به برنامه اضافه کنیم ، حداقل روش مکانیزه آنرا به خوبی می دانیم.

برای مطالعه بیشتر در مورد این نوع برنامه نویسی می توان به کتاب (راهنمای آموزش جاوا اسکریپت – ترجمه فرنود عسکری – نشر ادبستان) مراجعه کرد. کتابی مختصر– مفید و کاربردی در طی ۲۲۰ صفحه می باشد.

تمرین :

۱- با استفاده از کنترل های HTML برنامه ای بنویسید که در آن متن فارسی وارد شده در یک TextArea به معادل یونیکد آن در TextArea دیگر تبدیل شود.

۲- سه رادیو باتن را روی صفحه قرار دهید و برای انتخاب ۱۰ تا ۳۰ و یک چک باکس برای منفی بودن یا نبودن و یک TextField برای ضرب در آیتم های رادیو باتن ها. خروجی روی صفحه با استفاده از متد document.write نمایش داده شود .

۳- برنامه Upload کردن فایل به سرور را تکمیل کنید بطوریکه نام فایلهای ذخیره شده روی سرور یکسان نباشند. (راهنمایی : می توانید از System.Guid برای این منظور استفاده نمایید) .