

شناسنامه مقاله

نویسنده: فرشاد فکری نجات

وب سایت: www.fekrinejat.com

ایمیل: fekrinejat@yahoo.com

چاپ شده توسط: ماهنامه دانش و کامپیوتر

شماره: ۶۸ (آبان ۸۶)

صفحه: ۱۳

نگاهی عمیق به تکنیک Ajax

مقدمه :

در دهه اخیر، گسترش اینترنت و وب جهانی دستخوش تحولات بسیاری گردیده که از دو دیدگاه متفاوت (برای مصرف کنندگان آن) خوشایند می‌باشد. اما از طرفی انتظار مردم از طراحان، برنامه نویسان و مجموعه های نرم افزاری بسیار بالا رفته است. به هر حال امروزه تمامی انسانها از تمام نقاط جهان به اینترنت و وب سایت های خوش ساختی چون Yahoo ، Gmail ، Microsoft و غیره .. دسترسی داشته و امکانات آنها را با وب سایتهای فارسی زبان مقایسه کرده و یقینا خواهان استفاده از بهترین امکانات می‌باشند.

مسئله بررسی تمامی جوانب مختلف تکنیک Ajax در قالب یک مقاله میسر نمی‌باشد. ولی در این مقاله به بررسی گسترش وب ۲ در عرصه جهانی، معنا و مفهوم ساختار Ajax و دستاوردهای آن در قالب مثال هایی ساده و قابل درک، بررسی یک مثال بسیار ساده Ajax با توضیحات کامل خواهیم پرداخت.

شاید مخاطبان مستقیم این مقاله طراحان و برنامه نویسان مجرب نرم افزارهای تحت وبی باشند که از ابتدای ظهور وب تا به امروز تمامی توان و دانش خود را صرف تولید محصولات تحت وب کرده باشند. البته کسب دانش در رابطه با Ajax و موارد استفاده آن، برای مصرف کنندگانی که روزانه به گشت و گذار در اینترنت می‌پردازند نیز خالی از لطف نیست. حقیقت امر این است که از چندی پیش با استفاده از ابزارهایی چون XML و JavaScript تحولات بسیاری در وب بوجود آمده تا بتوان آن را از زمینه های ظهور وب نسخه ۲ تلقی نمود. اگرچه هنوز با وب ۲ فاصله ای قابل ملاحظه وجود دارد، ولی زمینه های بوجود آمده همانند Web Service ها، Ajax و RSS و غیره ... نشان از تحولاتی شیرین در وادی IT دارد.

Ajax چیست :

این تکنیک مخفف **Asynchronous Javascript And Xml** و به مفهوم استفاده نامتقارن JavaScript و XML بوده که به صورت "ای جکس" نیز تلفظ می‌گردد (گاهی به اشتباه بصورت آژاکس تلفظ می‌گردد). جالب است بدانید که مقدمات استفاده از این تکنیک از سال ها پیش در مرورگرهای وب قرار داده شده بود ولی با وجود استفاده مایکروسافت بصورت محدود در

Outlook ، کسی به موارد استفاده آن بصورت گسترده پی نبرده بود. در حقیقت تاریخچه واقعی Ajax به عنصری مخفی به نام IFrame در HTML بر می‌گردد که با استفاده از آن، برنامه نویسان قادر به انجام اعمالی شبیه Ajax بودند.

جهت درک بهتر مفهوم Ajax مثال های ملموس زیادی وجود دارد. به عنوان مثال فرض کنید که در Inbox ایمیل خود در Gmail قرار گرفته اید، به محض دریافت یک ایمیل جدید بدون اینکه شما بر روی دکمه و یا لینکی کلیک کرده باشید ، خبر دریافت ایمیل جدید در همان صفحه وب به شما داده می‌شود. توجه داشته باشید که تا پیش از ظهور تکنیک Ajax ، جهت مطلع شدن از تغییرات در صفحه وب، حتما می‌بایست صفحه Refresh شده و یا روی لینکی کلیک می‌نمودید.

با استفاده از این تکنیک کد نویسی در وب ، دیگر لازم به بارگزاری مجدد (Refresh) کل صفحه نیست، بلکه با روش هایی می‌توان تنها بخشی از یک صفحه وب را به روز رسانی کرده و بخش های دیگر را به قوه خود باقی گذاشت.

برای درک بهتر مزایای استفاده از این تکنیک، مثال دیگری را مطرح می‌کنیم. وب سایتی را در نظر بگیرید که از منوها ، تصاویر مختلف و بسیاری اجزای دیگر، در صفحات وب آن استفاده شده است. شما جهت ارتباط با مدیران سایت به بخش "تماس با ما" و یا Contact مراجعه نموده و یک فرم وب را جهت ارسال درخواستتان مشاهده کرده و آن را پر می‌کنید. سپس روی دکمه ارسال کلیک کرده و منتظر عملیات کامپیوتر جهت ارسال درخواستتان می‌شوید (در این مدت صفحه برای یک لحظه سفید شده و Refresh می‌شود). پس از چند ثانیه ، بسته به سرعت اینترنت شما و تعداد کاربران Online آن وب سایت، همان صفحه و یا صفحه دیگری ظاهر شده و پیغامی با عنوان "اطلاعات شما ارسال شد." نمایان می‌گردد. در این سناریو مطلقاً از Ajax استفاده نشده و تمامی تصاویر و اجزای دیگر صفحه وب مورد نظر مجدداً از اینترنت دریافت شده و در مرورگر ظاهر می‌گردد.

حال پس از استفاده از تکنیک Ajax جهت بهبود سناریوی فوق، وضعیت بدین گونه تغییر می‌کند که شما ابتدا وارد بخش "تماس با ما" شده و سپس فرم وب را پر کرده و دکمه ارسال را کلیک می‌نمایید. پس از چند ثانیه پیغام "اطلاعات شما ارسال شد." در بالای همان فرم ظاهر می‌شود. توجه نمایید که در این وضعیت نه صفحه Refresh شده و نه لازم به بارگزاری مجدد بخش هایی از صفحه وب می‌باشد. بلکه در پشت صحنه ، مرورگر با سرور تبادل اطلاعات کرده و بدون نیاز به بارگزاری مجدد تصاویر و بخشهای یک صفحه وب ، فقط بخشی از صفحه که حاوی پیغام "اطلاعات شما ارسال شد." می‌باشد، را بروزرسانی می‌کند.

برای درک بهتر موضوع به مثالی دیگر می‌پردازیم. ممکن است بارها از دیکشنری های Online چند زبانه در اینترنت استفاده کرده و با آنها آشنایی کافی داشته باشید. لغت مورد نظر خود را در یک TextBox وارد کرده و دکمه جستجو را می‌فشارید. سپس معنای لغت مورد نظر با Refresh (بارگزاری مجدد) صفحه نمایان می‌گردد (به عنوان نمونه یکی از همین فرهنگ های لغت را می‌توانید در آدرس www.farsidic.com ملاحظه نمایید). متأسفانه در این مثال نیز از Ajax استفاده نشده اما اگر با بهینه سازی وب سایت، از تکنیک Ajax استفاده گردد، به راحتی و با تایپ کردن همزمان شما در TextBox مورد نظر، لیستی پویا از لغات یافت شده (با حذف یا اضافه یک کلمه در TextBox جستجو، محتوای لیست نیز تغییر می‌آید) ، بدون بارگزاری مجدد صفحه نمایان می‌گردد.

مثال هایی همچون سناریوی های فوق در اینترنت بسیار یافت می‌شود. تنها کافیست مفهوم آن را به خوبی درک نموده و گشتی در اینترنت بزنید. شاید بتوان از جامعه مجازی ایرانی Cloob (آدرس <http://www.cloob.com>) به عنوان یک مثال بارز وب سایت فارسی با بهره گیری تکنیک Ajax نام برد.

ذکر این نکته خالی از لطف نیست که بدانیم شرکت Google یکی از پیشگامان اصلی استفاده از تکنیک Ajax در تولیداتی همچون Google Earth ، Google Maps ، Google Suggestion ، Gmail و غیره می باشد.

همانطور که بسیاری از متخصصین IT و کاربران مطلع هستند، نارضایتی های بسیاری از سرعت پائین نرم افزارهای تحت وب که روی Browser ها اجرا می شوند مطرح بوده و از این رو همیشه توان نرم افزارهای Desktop (مانند نرم افزارهای ویندوزی) مانند چماقی بر سر Web Application ها بوده. ولی با بهره گیری تکنیک Ajax ، از این پس امید بسیاری به محبوبیت هر چه بیشتر نرم افزارهای تحت وب خواهد بود.

نکته مهم:

شناخت اینکه يك صفحه وب با تکنیک Ajax و یا بدون آن تولید شده، برای يك برنامه نویس آشنا به JavaScript و مباحث کدهای Client/Server Side شاید چندان مشکل نباشد ولی قطعا يك برنامه نویس نوپا و یا کاربر عادی، در بعضی مواقع قادر به شناسایی آن نخواهد بود.

به عنوان مثال يك صفحه وب پویا (DHTML) را در مقابل چشمان خود تجسم کنید. درون این صفحه، چند منو بصورت لینک قرار گرفته که با قرار گرفتن ماوس بر روی هر يك از لینک ها ، يك متن مجزا در سمت چپ صفحه ظاهر شده و با کنار رفتن ماوس از روی لینک، متن مورد نظر نیز محو می شود (این عمل بدون Refresh صفحه انجام می گیرد) ولی باید توجه داشت که به هیچ عنوان کد Ajax در این صفحه مورد استفاده قرار نگرفته، بلکه متن های مورد نظر که همیشه یکسان و ثابت هستند، با استفاده از JavaScript (به روش Client Side) ، بواسطه لایه ها و یا هر عنصر دیگری، آشکار و پنهان می گردند.

نکته مهم این که هرگاه يك صفحه وب عادی بدون Refresh شدن، اعمالی که مرتبط با کدهای Server-Side (مانند ارتباط با بانک اطلاعاتی در سرور) است را انجام دهند، بعنوان صفحات مزین به Ajax محسوب می گردند.

مفهوم عملکرد Ajax :

از این بخش مقاله به بعد که مخاطب اصلی آن برنامه نویسان و تولید کنندگان محصولات تحت وب هستند، با مفاهیم فنی و روش برنامه نویسی عملی Ajax آشنا خواهید شد.

برای نوشتن ساده ترین کد Ajax می بایست به مباحث HTML ، JavaScript ، DOM و يك زبان سمت سرور مانند ASP ، PHP ، ASP.NET و غیره .. آشنایی داشته باشید، اگرچه جهت گسترش يك برنامه Ajax پیچیده، داشتن دانش مکفی در رابطه با XML و CSS نیز از ملزومات است.

همانطور که تقریباً همه برنامه نویسان وب مطلع هستند، هر مرورگر وب مانند IE ، FireFox ، NetScape ، Mozilla ، Opera و غیره ... دارای تعداد زیادی Object (شئی) درونی بصورت API در زمینه های مختلفی چون DOM ، CSS ، HTML ، XML و غیره می باشند. (جهت مشاهده لیست مرجع برخی از این Object ها، آدرس زیر را ملاحظه فرمایید)

<http://www.xulplanet.com/references/objref/>

این Object ها بطور ذاتی در مرورگرها وجود داشته و نیازی به تولیدشان نیست، بلکه جهت استفاده از آنها ابتدا می‌بایست عملکرد آن شئی خاص را مورد بررسی قرار داده و سپس توسط يك زبان Client-Side مانند JavaScript ، به راحتی از آن استفاده نمود.

نکته:

نکته قابل توجه در این رابطه، عدم هماهنگی و استاندارد نبودن نام این Object ها در مرورگرهای مختلف می‌باشد که البته یکی از دغدغه های Ajax نیز همین موضوع است. این مسئله نه تنها در مورد Object ها ، بلکه در مورد Tag های HTML نیز صادق است. به عنوان مثال استفاده يك Tag جدید و ناشناخته در مرورگر X دلیلی بر کارکردن همان Tag در مرورگر Y نخواهد بود.

شاید جالب باشد اگر بدانید، سناریوی اصلی Ajax ، حول محور يك Object به نام XMLHttpRequest می‌چرخد که آن را با نام مخفف XMLHttpRequest می‌شناسند. این شئی بازیگر نقش اول و تقریباً همه کاره کدهای Ajax بوده و البته طبق گفته قبلی منی بر استاندارد نبودن مرورگرهای وب ، با نامهای متفاوت ولی عملکرد تقریباً یکسانی، قابل دسترس و استفاده است. نحوه عملکرد این شئی و متدها و خصوصیات آن، در ادامه مقاله بطور مفصل و کامل مورد بررسی قرار خواهد گرفت.

نکته قابل توجه این است که این Object در يك کد Client-Side و با استفاده از JavaScript اجرا شده و سپس در زمان معین ، يك کد Server-Side را بدون متوجه شدن بازدیدکننده سایت، در سرور اجرا کرده و نتیجه اجرای کد مربوطه را از سرور گرفته و مجدداً بدون Refresh شدن صفحه وب، در اختیار مرورگر قرار می‌دهد. سپس جاوا اسکریپت، نتیجه بازگشتی را به دلخواه برنامه نویس، مورد استفاده قرار داده و یا در قسمتی از صفحه مقابل کاربر (بدون Refresh شدن صفحه) نمایان می‌سازد.

این نوع اجرای نوین کدهای سمت سرور که توسط کدهای سمت Client انجام می‌گیرد، از اهمیت فوق العاده ای برخوردار می‌باشد. عملاً با این کاربرد مفید ، زبان برنامه نویسی JavaScript که روزگاری تنها برای Form Validation ، منوهای DHTML ، Rollover Pictures و غیره .. مورد استفاده قرار می‌گرفت مجدداً به دوران بلوغ رسیده و چشم انداز بسیار روشنی برای آن تصور می‌گردد.

کد نویسی Ajax :

برای کدنویسی Ajax ، Toolkit های آماده بسیاری در محیطهای نرم افزاری IDE (مانند Visual Studio.NET) فراهم گردیده است. اما برای پیاده سازی این تکنیک، می‌توان از ساده ترین ابزار مانند يك ویرایشگر متن مثل Notepad نیز استفاده کرده و بدون بهره‌برداری از امکانات پیشرفته محیط های IDE ، کد مورد نظر را تولید نمود. البته هیچگاه فراموش نکنید که استفاده از Toolkit ها و ابزارهای نرم افزاری مدرن امروزی، شما را از واقعیت‌های پشت پرده و عملکرد لحظه ای هر خط از کد، دور نگه خواهد داشت. به همین دلیل در این بخش از مقاله، ساده ترین کد Ajax را (که شاید بتوان آن را Hello World نامید) با توضیحات کامل و خط به خط ارائه خواهیم کرد تا هر کسی با کمترین دانش کدنویسی، تنها با استفاده از يك ویرایشگر ساده، طعم شیرین Ajax را بچشد.

در سناریوی این کد، دو فایل HTML و ASP در نظر گرفته شده است. در فایل HTML يك فرم با ۲ عدد TextBox قرار دارد که با وارد کردن اطلاعات در یکی از این TextBox ها، پس از کلیک در قسمتی از صفحه وب ، بدون Refresh شدن صفحه ، کد ASP اجرا شده و نتیجه آن بلافاصله

در TextBox دوم ظاهر می‌گردد. در نظر داشته باشید که هر دو فایل مورد نظر در مثال ما میبایست بر روی یک Web Server با پشتیبانی ASP 3.0 و در یک Folder قرار گرفته و سپس فایل HTML فراخوانی گردد.

مرحله اول:

ابتدا در یک ویرایشگر صفحات وب مانند Expression Web و یا یک ویرایشگر متن ساده (اگر به ساختار و چیدمان Tag های وب آشنایی کافی دارید) یک فایل جدید با نام Ajax.htm ایجاد نموده و کدهای جاوا اسکریپت و HTML زیر را در آن قرار دهید.

```
<html>
```

```
<body>
```

```
<script type="text/javascript" >
```

```
function ajaxFunction()
```

```
{  
    var xmlHttp = new ActiveXObject("Msxml2.XMLHTTP") ?  
    new ActiveXObject("Microsoft.XMLHTTP") : new XMLHttpRequest();
```

```
    xmlHttp.onreadystatechange=function() {  
        if (xmlHttp.readyState==4) {  
            document.AjaxForm.time.value= xmlHttp.responseText ;  
        }  
    }  
}
```

```
xmlHttp.open("GET", "time.asp",true);  
xmlHttp.send(null);
```

```
}
```

```
</script>
```

```
<form name="AjaxForm">
```

```
Name: <input type="text" onchange="ajaxFunction()";>
```

```
Time: <input type="text" name="time">
```

```
</body>
```

```
</html>
```

توضیح کدهای فوق:

در ابتدا برای قرار دادن یک کد جاوا اسکریپت در یک صفحه وب، میبایست از تگهای <Script> و </Script> و با نوع Text/JavaScript استفاده نموده و کدهای جاوا را در میان این دو تگ قرار دهیم تا Browser به هنگام اجرای این بخش از صفحه وب، از مفسر جاوا برای اجرای آن استفاده نماید. در ضمن فراموش نکنید که کدهای جاوا اکسریپت Case Sensitive بوده و به حروف کوچک و بزرگ حساس هستند.

سپس يك تابع به نام `ajaxFunction` تعريف مي‌گردد كه عملي را انجام داده و هيچ خروجي خاصي به فراخواننده خود ندارد. در نظر داشته باشيد كه كد فراخوان اين تابع در ادامه فايل HTML قرار دارد.

همانطور كه قبلا ذكر شد، شئي اي به نام `XMLHttpRequest` وجود دارد كه تمامي عمليات Ajax را به عهده خواهد گرفت. در ابتداي تابع `ajaxFunction` ، يك نمونه (`Instance`) از شئي مورد نظر گرفته شده و از اين پس با استفاده از نام `xmlHttp` ، از متدها و خصوصيات شئي `XMLHttpRequest` استفاده خواهد شد.

```
var xmlHttp = new ActiveXObject("Msxml2.XMLHTTP") ?  
new ActiveXObject("Microsoft.XMLHTTP") : new XMLHttpRequest();
```

در ساختار كد فوق ، از شرط بدون `IF .. Then` و با علامتهاي `?` و `:` كه همان مفاهيم را دارند (جهت مختصر شدن) استفاده شده است. در واقع يك خط كد فوق در صورت استفاده از يك بلوك `Catch .. Try` همانند كد زير خواهد بود:

```
var xmlHttp;  
try{  
    xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");;  
    catch (e){  
        try{  
            var xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");;  
            catch (e){  
                var xmlHttp=new XMLHttpRequest();;  
            }  
        }  
    }
```

دقيقا اينجاست كه عدم هماهنگي و عدم وجود استاندارد مابين Browser ها، ما را مجبور به استفاده از اين روش مي‌نمايد. زيرا گويا مايكروسافت كه خود براي اولين بار توليد كننده شئي `XMLHttpRequest` در مرورگر خود مي‌باشد، با تغيير نسخه IE ، نام آن را نيز تغيير مي‌دهد. در حقيقت شئي مورد نظر در مرورگر `Internet Explorer` ، `XMLHTTP` نام داشته و به عنوان يك `ActiveXObject` شناخته مي‌شود. به عنوان مثال براي IE 5.5 از `Microsoft.XMLHTTP` و IE 6.0 از `Msxml2.XMLHTTP` استفاده مي‌گردد.

از طرفي در مرورگرهاي ديگر، اين شئي ابتدا در سال ۲۰۰۲ با نام `XMLHttpRequest` و در `Mozilla 1.0` مورد استفاده قرار گرفت (برخلاف مايكروسافت كه از `ActiveXObject` استفاده مي‌كند، در `Mozilla` به عنوان يك API كوچك در نظر گرفته شده بود تا در صورت غيرفعال بودن اجراي `ActiveX` ها در مرورگر، همچنان كدهاي Ajax قابل اجرا باشند). سپس ديگر مرورگرها مانند `Safari` ، `FireFox` ، `Opera` و غيره .. از `Mozilla` تبعيت كرده و با يك استاندارد يکسان و بدون توجه به `Internet Explorer` ، استفاده از Ajax را پشتيباني نمودند.

خوشبختانه مايكروسافت از IE 7.0 به جمع استاندارد ديگر مرورگرها پيوسته و شئي `XMLHttpRequest` را با همين نام و همين ساختار و بصورت API (نه `ActiveX`) پشتيباني مي‌نمايد. اما همچنان بسياري کاربران اينترنت از مرورگر IE نسخه ۵,۵ و ۶,۰ استفاده نموده و برنامه نويسان همچنان مي‌بايست بالاچار از كدهاي پيچيده تري (جملات شرطي) جهت پشتيباني تمامي کاربران اينترنت استفاده نمايند.

پس از ايجاد يك `Instance` به نام `xmlHttp` ، هم اکنون به بررسي وظيفه قطعه بلوك زير مي‌پردازيم:

```
xmlHttpRequest.onreadystatechange=function() {
    if (xmlHttpRequest.readyState==4) {
        document.AjaxForm.time.value= xmlHttpRequest.responseText ;
    }
}
```

بطور كلي شئي XMLHttpRequest داراي متدها و Property هايي مي باشد كه در انتهاي اين بخش به اختصار توضيح داده خواهد شد. اما در شئي مورد نظر، يك Property به نام onreadystatechange وجود دارد كه با استفاده از ساختار فوق، به عنوان يك Event Handler مورد استفاده قرار مي گيرد. اين رويداد، پس از فراخواني يك كد Server Side (كه اين فراخواني پس استفاده از متد Send كه متعاقبا توضيح داده خواهد شد، انجام مي پذيرد)، بلافاصله بطور خودكار فراخواني شده و كدهاي داخل آن اجرا مي گردد.

اين شئي علاوه بر Property ذكر شده، يك Property ديگر نيز با نام readyState دارد كه از نوع عددي بوده و نشان دهنده وضعيت عملکرد سرور در ارسال پاسخ مي باشد. در اين Property عددي مابين ۰ الي ۴ (بصورت خودكار) قرار خواهد گرفت كه هر عدد مطابق با جدول زير معنا و مفهوم خاص خود را دارد.

۰	درخواستي كه به سرور ارسال شده، هنوز كار خود را آغاز نكرده است.
۱	درخواست در سرور در حال بررسي و اجرا است.
۲	درخواست از سرور براي Client ارسال شده است.
۳	درخواست در حال دريافت توسط Client است.
۴	عمل پاسخگويي به درخواست اتمام يافته است.

همانطور كه ملاحظه مي كنيد، با استفاده از اين Property به راحتی وضعيت درخواست ارسالي به سرور، قابل پيگيري است. در مثال اين مقاله، شرطي فراهم گرديده كه به محض قرار گرفتن عدد ۴ در خصوصيت readyState عملي را انجام دهد.

پس از تكميل پاسخ سرور، از كد زير جهت قرار دادن نتيجه كار به داخل TextBox دوم استفاده مي گردد.

```
document.AjaxForm.time.value= xmlHttpRequest.responseText ;
```

شئي XMLHttpRequest داراي يك Property ديگر به نام.responseText و از نوع String مي باشد كه نتيجه ارسالي از سرور بطور خودكار در آن قرار مي گيرد. همانطور كه ملاحظه مي فرمائيد، سمت چپ علامت مساوي كد فوق، از ساختار DOM (Document Object Model) استفاده شده كه نتيجه ارسالي از سرور را مستقيما در TextBox ي به نام time، در فرم وب به نام AjaxForm، از همين Document (صفحه وب) قرار مي دهد. (توجه داشته باشيد كه در همين صفحه وب، يك فرم وب به نام AjaxForm وجود دارد كه داراي يك عنصر از نوع Textbox و به نام time است)

هم اکنون به بررسی دو متد بسیار مهم به نامهای Open و Send که عملیات ارسال درخواست را به سرور انجام می‌دهند، می‌پردازیم.

```
xmlHttp.open("GET", "time.asp",true);  
xmlHttp.send(null);
```

متد Open وظیفه اعمال تنظیمات قبل از ارسال و متد Send وظیفه ارسال درخواست را بر عهده داشته و با وجود اینکه این دو خط کد، بعد از بلوک رویداد onreadystatechange قرار گرفته اند، ولی تا زمانیکه متد send شئی XMLHttpRequest فراخوانی نگردیده، Event Handler فراخوانی نخواهد شد.

متد Open دارای دو پارامتر ضروری (دو پارامتر اول) و یک پارامتر غیرضروری (آخرین پارامتر) می‌باشد. در اولین پارامتر این متد می‌توان از دو مقدار GET و POST (و یا مقادیری دیگر که توضیح آنها از حوصله این مقاله خارج است) استفاده نمود که البته هر یک از این دو مقادیر در مواقع خاص خود کاربرد دارد. دومین پارامتر نیز URL (نام و مسیر) یک فایل Server Side مانند ASP ، PHP ، ASPX ، CFM و غیره .. تعیین می‌نماید که در مثال فوق، نام فایلی به نام time.asp در همان پوشه فایل HTML، تنظیم شده است. سومین پارامتر (که بصورت پیش فرض مقدار true می‌باشد)، وضعیت نامتقارن بودن یا نبودن اجرای Event Handler مربوطه را تعیین می‌نماید. در صورت قرار دادن مقدار true، پس از اجرای متد send بلافاصله رویداد onreadystatechange نیز فراخوانی می‌گردد، اما در صورت قرار دادن false، ابتدا منتظر اتمام پردازش سرور و ارسال پاسخ شده و سپس رویداد مورد نظر فراخوانی می‌گردد. در نظر داشته باشید که در صورت استفاده از مقدار false (که ریسک بیشتری را می‌طلبد)، اگر به یکباره سرور دچار مشکلی گردیده و یا دریافت پاسخ از سرور به تعویق بیافتد، نرم افزار وب مدت کوتاهی مختل شده و کنترل از دست برنامه نویس خارج می‌گردد. البته متد Open دارای ۲ پارامتر غیر ضروری دیگر نیز می‌باشد که در کد Ajax این مقاله مورد استفاده قرار نگرفته است.

و بالاخره در نهایت، متدی با نام send، مقداری را بصورت پارامتر به سمت سرور ارسال می‌کند که در مثال فوق به دلیل عدم نیاز به ارسال مقادیری به سرور از واژه null استفاده گردیده است.

حال با اتمام توضیحات تقریباً کاملی در رابطه با عملکرد کدهای JavaScript و DOM فوق، هم اکنون به توضیح فرم وب در کد HTML خواهیم پرداخت.

```
<form name="AjaxForm">  
Name: <input type="text" onchange="ajaxFunction();">  
Time: <input type="text" name="time">
```

همانطور که در کد HTML فوق ملاحظه می‌فرمائید، فرمی به نام AjaxForm با ۲ عدد TextBox که نام یکی از آنها time می‌باشد، تعریف شده است. در رویداد onchange اولین TextBox (این رویداد پس از اعمال تغییراتی در محتویات TextBox به وقوع می‌پیوندد)، تابع ajaxFunction (که پیش تر بطور کامل مورد بررسی قرار گرفت) فراخوانی می‌شود.

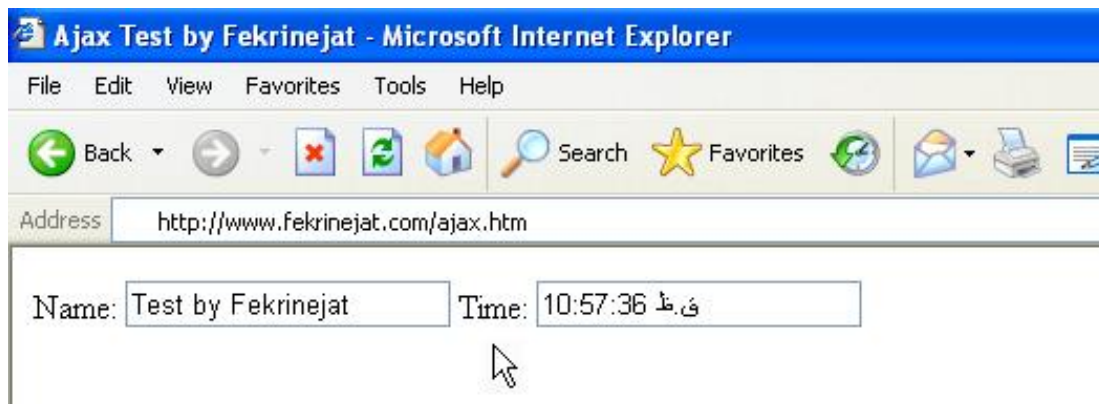
هم اکنون یک فایل جدید به نام time.asp در ویرایشگر متن و یا وب ایجاد کرده و کدهای زیر را عیناً در آن وارد نمایید.

```
<%  
response.expires=-1
```



```
response.write(time)
%>
```

كد بسيار ساده فوق، ساعت سرور را از طريق تابع داخلي Time (يكي از توابع VBScript) گرفته و بواسطه متد Wrtie در خروجي نمايش مي‌دهد. هم اكنون فايل Ajax.htm و time.asp را در پوشه اي در يك سرور وب قرار داده و فايل Ajax.htm را اجرا نماييد. سپس در TextBox سمت چپ، عبارتي را تايپ کرده و ماوس را در نزديكي TextBox سمت راست (خارج از خود TextBox)، يکبار کليک نماييد تا زمان سرور بدون Refresh شدن صفحه در TextBox سمت راست ظاهر گردد. (همانند تصوير زير)



همانطور که قبلا اشاره شد، شئي XMLHttpRequest داراي مندها و Property هاي ديگري نيز مي‌باشد که برخي از مهمترين آنها را به همراه توضيح مختصري، در جدول زير ملاحظه فرمائيد.

Method	abort()	با استفاده از اين متد، تمامي عمليات درخواست از سرور در هر مرحله اي ، بالافاصله Cancel خواهد شد.
Property ها	status	Property از نوع عددي که وضعيت درخواست HTTP به سرور را نمايش مي‌دهد. به عنوان مثال قرار گرفتن عدد ۴۰۴ در اين Property نشاندهنده عدم وجود URL ذکر شده در متد Open مي‌باشد. اما عدد ۲۰۰ دليلي بر وجود اين فايل در سرور دارد.
	statusText	Property از نوع رشته اي است که وضعيت درخواست HTTP به سرور در قالب يك متن (مانند Not Found يا OK) در آن مقداردهي مي‌گردد. (همانند status عمل مي‌کند)
	responseXML	همانند reponseText مي‌باشد، با اين تفاوت که مقدار خروجي در قالب XML به Client باز مي‌گردد که بصورت DOM ميتوان به تگ هاي آن دسترسي داشته و بخشي از محتويات XML را مورد استفاده قرار داد.

نکته:

X در Ajax مخفف XML است. ولي برخلاف آن، هيچ الزامي مبني بر استفاده XML در برخي از برنامه هاي Ajax وجود ندارد. بلکه با استفاده از خصوصيات responseText و responseBody مي توان حالتهاي ديگري از خروجي را (که با XML مرتبط نيستند) مورد استفاده قرار داده و يا بواسطه InnerHTML مستقيما يك کد HTML را به بعصري در صفحه وب انتقال داد.

مزاي و معايب :

هر عنصر جاندار و بي جان در اين کره خاكي داراي مزاي و معايب مي باشد. Ajax نيز به نوبه خود ترازوي مقايسه اي دارد که تقريباً دو کفه مزاي و معايب آن با يکديگر برابري مي کند. در اين بخش از مقاله به اين موارد بطور تقريباً کلي اشاره خواهد شد.

الف) مزاي:

محبوبيت تکنیک Ajax ميان برنامه نويسان وب ، از سال ۲۰۰۵ تاکنون، يك حقيقت قابل رويت مي باشد. اگرچه فعالان عرصه IT تقريباً از تمامي فناوري ها و دستاوردهاي نوين اين عرصه حمايت مي کنند.

۱- توليد نرم افزارهاي تحت وب با خصلت Desktop :

شايد پروژه اي به نام Windows Live مايکروسافت را شنیده باشيد. در اين پروژه، سرويس هاي نرم افزاري شرکت مايکروسافت در قالب Web Application ، از طريق اينترنت و Browser هاي وب به راحتی قابل دسترس بوده و کاربران با يك دسترسي سرعت بالاي اينترنت قادر به استفاده از تمامي امکانات آن مي باشند. جهت مشاهده مثالي در زمينه پياده سازي يك سيستم عامل در محيط وب با استفاده از Ajax به وب سايت www.Ajaxwindows.com مراجعه نماييد.

همچنين يك شرکت هندي در وب سايت خود (به آدرس www.zoho.com) ، مجموعه تحت وبی به معرض نمايش گذاشته است که کاربران به راحتی به تمامي نرم افزارهاي مجموعه Office اعم از Access ، Excel ، Word ، PowerPoint (البته با نامهاي مستعار Zoho Writer ، Zoho Sheet ، Zoho Show و غيره...) و بانک اطلاعاتي و بسياري نرم افزارهاي تحت وب ديگر، بصورت Online و رايجان دسترسي داشته باشند. با استفاده از اين امکانات، بدون نصب مجموعه آفيس بر روي کامپيوتر خود و تنها با اتصال به اينترنت، از تمامي قابليت هاي اين نرم افزارها به صورت آن لاین و در مرورگر وب خود استفاده مي نماييد.

ايده پياده سازي نرم افزارهاي Desktop در محيط وب، امروزه در حال گسترش بوده و تکنیک Ajax يکي از مناسب ترين روش هاي معقول پياده سازي اين ايده مي باشد. اگرچه فايل هاي Flash نيز امکاناتي مشابه Ajax در اختيار برنامه نويسان قرار مي دهند ولي همانند Ajax ، چندان مورد حمايت و توجه قرار نگرفته اند.

۲- استفاده کمتر از پهنای باند سرور :

پهنای باند يك سرور براي وب سايت هايي که بازدیدکنندگان بسياري دارند، از اهميت بالايي برخوردار است. به عنوان مثال سايت سنجش آموزش کشور (www.sanjesh.org) را در اولين روز ارائه نتيجه کنکور، جهت سرويس دهی به صدها هزار بازدیدکننده مجسم کنيد. هر کاربر با وارد شدن به سايت و مراجعه به بخش مربوطه، باعث ايجاد درخواست هايي براي وب سرور شده و با اشغال پهنای باند، باعث افت سرعت خواهد شد. معمولاً مسئولین سايت ها

در چنین مواقعی با ارتقاء پهنای باند سرور یا ارتقای سخت افزاری، سعی در بهینه سازی سرویس دهی دارند. اما نکته قابل توجه این است که با استفاده از تکنیک Ajax، میزان درخواست های ارسال شده به سرور و حجم پاسخهای ارسال شده از سرویس دهند، به شدت کاهش یافته و شاید دیگر نیازی به ارتقا نباشد.

۲- User Friendly بودن با سرعت کارکرد :

سألهاست که انتظار جهت Load شدن يك صفحه و قطع ارتباط با سرور، پس از كليك شدن بر روي يك لينك، کاربران اینترنت را با دغدغه هایی مواجه کرده است. با استفاده از تکنیک Ajax علاوه بر سرعت بخشیدن به عملکرد کاربر، باعث زیبایی کارکرد نرم افزار و رضایتمندی کاربران نیز خواهید بود.

الف) معایب:

۱- پیچیدگی کد نویسی :

استفاده از تکنیک Ajax ظاهراً يك تکنیک دوست داشتنی است ولي در عمل کدنویسی پیشرفته جهت انجام اعمالی در صفحه وب، باعث ایجاد کدهای پیچیده می گردد. در بسیاری مواقع که تقریباً بیش از ۸۰ درصد يك صفحه وب (پس از Refresh شدن) می بایست تغییراتی داشته باشند، از تکنیک Ajax استفاده نمی گردد. بنابراین تیم طراح و برنامه نویس با در نظر گرفتن نقاط قوت و ضعف و وضعیت موجود، تصمیم به استفاده یا عدم استفاده از Ajax می گیرند. لازم به ذکر است که زیاده روی در استفاده از Ajax بعضاً باعث افت کارایی نرم افزار وب می گردد.

۲- عدم سازگاری مرورگرها :

در خلال مقاله، بطور خلاصه به عدم وجود استاندارد، مابین مرورگرهای وب اشاره شد. متأسفانه تعدد مرورگرهای مختلف و کاربران اینترنت بر این مشکل دامن می زند. پیشنهاد می شود که حتماً کد Ajax خود را با آخرین نسخه مرورگرهای رایج دنیا امتحان نموده تا از صحت عملکرد آن اطمینان حاصل فرمائید.

۳- عدم کاربرد دکمه های Back ، Stop و Refresh :

همانطور که میدانید، مرورگرها همیشه آدرس صفحه قبلی را در حافظه نگه داشته و با فشردن دکمه Back، به آن صفحه (يك گام به عقب) مراجعه می نمایند. اما يك صفحه وب را در نظر بگیرید که با استفاده از تکنیک Ajax بدون منتقل شدن به صفحه بعد، عملی را در همان صفحه انجام داده و سپس نتیجه را نمایش خواهد داد. اساساً در این سناریو هیچ صفحه قبلی وجود ندارد که با فشردن دکمه Back به آن مراجعه نموده و بنابراین اطلاعات صفحه وب پس از تغییر محتوا، غیر قابل بازگشت خواهد بود. این موضوع برای دکمه های Stop و Refresh نیز صادق است.

۴- اتکای بیش از حد به زبان جاوا اسکریپت:

تقریباً همه مرورگرهای وب از زبان برنامه نویسی جاوا اسکریپت به عنوان يك زبان Client-Side پشتیبانی می کنند. بنابراین این زبان برنامه نویسی، نقشی بسیار مهم و کلیدی در استفاده از تکنیک Ajax بازی می کند. در این میان اگر بطور اتفاقی اجرای کدهای جاوا اسکریپت غیرفعال شده باشد، عملاً آن صفحه وب قابل استفاده کاربر مورد نظر نخواهد بود.

سخن پایانی:

در صورت علاقه شما خوانندگان عزیز به میحث Ajax، مقالات دیگری در مورد این فناوری بسیار موثر، در شماره های آینده ماهنامه مطالعه خواهید نمود. جهت پرسش سوالات تخصصی خود به وب سایت www.fekrinejat.com مراجعه نمایید.