

Ajax : تحولی بزرگ در عرصه وب (بخش اول)

اینترنت و مهمترین سرویس آن وب علاوه بر این که حیات بشری را در هزاره سوم دستخوش تحولات فراوانی کرده است ، توانسته است منشاء تحولات گسترده ای در سایر حوزه های فناوری گردد . در واقع ، تحول در زندگی بشریت مدیون پیشرفت در سایر حوزه های فناوری است . یکی از این حوزه ها ، صنعت نرم افزار است . برای آشنائی با این تحول بد نیست مروری سریع داشته باشیم به گذشته وب و نرم افزار تا از این رهگذر بتوانیم ضرورت ایجاد فناوریهای جدیدی نظیر Ajax (برگرفته شده از Asynchronous JavaScript And XML) را بهتر درک کنیم .

مروری بر وب ، نرم افزار و تاثیر متقابل آنها بر یکدیگر

موجودات جدیدی که ما امروزه آنها را با نام برنامه های وب می شناسیم ، ماحصل تاثیر وب بر نرم افزار و رشد رو به فزاینده استفاده از اینترنت در متن زندگی ، ماحصل تاثیر نرم افزار بر وب است . اجازه دهید ببینیم موضوع از کجا و با چه مختصاتی شروع گردید و هم اینک در چه وضعیتی به سر می بریم .

وب ایستا : زمانی که تیم . برنرز لی در سال 1990 اولین پیشنهاد خود در رابطه با وب را ارائه کرد ، ایده کار بسیار ساده بود : ایجاد يك شبکه از اطلاعات متصل شده به یکدیگر با استفاده از ابرمتن ها و URIs (برگرفته شده از Uniform Resource Identifiers) . وب در آغاز به عنوان يك رسانه اشتراك اطلاعات مطرح گردید تا از طریق آن دانشمندان و محققین بتوانند دانش و اطلاعات را بین خود به اشتراك گذاشته و نتایج تحقیقات را با یکدیگر مبادله نمایند . ارائه دهندگان با استفاده از يك استاندارد مستند سازی ، اطلاعات خود را تهیه و آنها را به مخاطبان خود ارائه می کردند . در آغاز برای استاندارد مستند سازی از ابزاری که در آن زمان خود هنوز استاندارد نشده بود استفاده می گردید . HTML ، ابزاری است که با بکارگیری مجموعه ای از تگ ها ، استاندارد لازم جهت نمایش يك مستند در نرم افزارهای نمایش دهنده (مرورگرها) را فراهم می نماید . نرم افزارهای نمایش دهنده ، مجهز به مفسرهائی جهت تفسیر تگ ها و نمایش اطلاعات موجود در مستند بر اساس تعاریف از قبل تعریف شده شدند . در آن زمان که به سال های ابتدائی دهه 90 میلادی برمی گردد ، چیزی با نام برنامه های وب وجود نداشت و وب به عنوان رسانه ای جهت اشتراك و مبادله اطلاعات مطرح بود . تمامی تعاملات در عرصه وب محدود به درخواست يك مستند توسط سرویس گیرنده و ارسال آن توسط سرویس دهنده بود .

وب پویا با تمرکز بر روی پردازش های سمت سرویس گیرنده : در ادامه ، نیاز به انجام برخی پردازش های اولیه مطرح گردید . با توجه به این که اینترنت از يك بستر سرویس گیرنده و سرویس دهنده (client server based) به منظور دریافت درخواست کاربر و پاسخ به آن استفاده می کرد ، طراحان و پیاده کنندگان برای تحقق پردازش ها بر روی دو گزینه متمرکز شدند : انجام پردازش ها در سمت سرویس گیرنده و یا سمت سرویس دهنده . در آغاز ، برای انجام پردازش ها بر روی سرویس گیرنده متمرکز شدند . با این هدف که بتوان پردازش های مورد نیاز را (با حوزه ای که در آن زمان تعریف شده بود) ، در سمت سرویس گیرنده انجام داد . برای تحقق خواسته فوق لازم بود که بستر و ابزارهای لازم در بطن مرورگرها ایجاد گردد . ظهور فناوریهای نظیر HTML-DOM ، جاوااسکریپت ، CSS (برگرفته شده از sheets cascading style) و DHTML (برگرفته شده از HTML Dynamic) از جمله مهمترین تلاش های انجام گرفته در آن زمان برای تحقق پردازش ها در سمت سرویس گیرنده بود . برای آشنائی بیشتر با

تلاش های انجام گرفته در آن زمان بد نیست مروری سریع داشته باشیم به آنچه طی سالیان گذشته در عرصه وب اتفاق افتاده است .

- **عرضه جاوا اسکریپت :** Netscape Navigator اولین مرورگر وبی بود که پس از طراحی و پیاده سازی موفقیت آمیز توانست به عنوان کاتالیزوری در رشد وب و عرضه فناوری های مرتبط با آن نقش بسیار مثبتی را ایفاء کند . اغلب سازمان ها و شرکت ها از مرورگر فوق به عنوان معیار استاندارد خود برای پیاده سازی فناوری های جدید استفاده می کردند . یکی از این فناوری ها ، جاوا اسکریپت بود . جاوا اسکریپت توسط Brendan Eich از نت اسکپ ایجاد و نسخه شماره دو آن در سال 1995 عرضه گردید . برای اولین مرتبه ، پیاده کنندگان قادر به ایجاد صفحات وبی شدند که می توانست با کاربر تعامل برخورد نماید . استفاده از جاوا اسکریپت برای معتبر سازی داده در سمت سرور و گیرنده و حذف برخی مبادلات غیر ضروری بین سرور و گیرنده و سرور و دهنده از جمله مهمترین دستاورد بکارگیری جاوا اسکریپت در آن زمان بود . بدین ترتیب نطفه پردازش های ساده در سمت سرور و گیرنده به کمک جاوا اسکریپت بسته شد . قابلیت فوق در زمانی که اکثر کاربران اینترنت از طریق یک مودم با سرعت 8 . 28 (Kbps) به شبکه وصل می شدند ، باعث می شد که تعداد دفعاتی که کاربران مجبور به انتظار برای دریافت یک پاسخ از سرور دهنده باشند ، کاهش یابد . تحول فوق اولین مرحله جدی به سمت رویکرد Ajax در آن زمان بود .
- **ایجاد فریم :** در نسخه اولیه HTML با هر سند به عنوان یک موجودیت رفتار می شد تا این که در نسخه شماره چهار فریم ها معرفی گردیدند . با استفاده از فریم ها نمایش یک صفحه وب می توانست مشتمل بر بکارگیری مجموعه ای از سایر صفحات وب باشد . ایده فوق یک اصلاح ساختاری اساسی در خصوص نحوه نمایش یک صفحه وب بود . Netscape Navigator 2.0 اولین مرورگری بود که از فریم ها و جاوا اسکریپت حمایت کرد . این یک مرحله جدی به سمت رشد و توسعه Ajax در آن زمان بود . با توجه به این که یک فریم ، یک درخواست مجزا را برای سرور دهنده ایجاد می کرد ، قابلیت کنترل یک فریم و محتویات آن با استفاده از جاوا اسکریپت دنیای جدیدی را پیش روی طراحان و پیاده کنندگان صفحات وب قرار داد .
- **روش فریم مخفی :** به موازات این که پیاده کنندگان درگیر نحوه استفاده از فریم ها شدند ، یک روش جدید با هدف تسهیل در امر ارتباط بین سرور و سرور و سرور دهنده مطرح گردید . روش فریم مخفی شامل تنظیم یک frameset با طول و یا عرض صفر پیکسل است که هدف اساسی آن انجام فرآیند مقداردهی اولیه با سرور دهنده بود . فریم مخفی شامل یک فرم HTML با فیلدهای خاصی است که می توانست بطور اتوماتیک توسط جاوا اسکریپت تکمیل و برای سرور دهنده ارسال گردد . زمانی که فریم از سرور دهنده مراجعت می کرد ، یک تابع دیگر جاوا اسکریپت فراخوانده می شد تا اعلام نماید که داده از سمت سرور دهنده برگردانده شده است . روش فریم مخفی ، اولین مدل درخواست و پاسخ غیرهمزمان را برای برنامه های وب به ارمغان آورد (اولین رویکرد به مدل ارتباطی Ajax) .
- **DOM و Dynamic HTML :** تا سال 1996 ، وب یک دنیای ایستا را تجربه می کرد و برای تعامل محدود با کاربر از جاوا اسکریپت و روش فریم های مخفی استفاده می گردید . همچنان امکان تغییر در محتویات یک صفحه بدون نیاز به load مجدد آن از سرور دهنده وجود نداشت . در ادامه IE 4.0 (برگرفته شده از Internet Explorer) توسط شرکت مایکروسافت ارائه گردید . در این نسخه علاوه بر بکارگیری اکثر فناوری

های استفاده شده در Netscape Navigator ، يك ویژگی مهم دیگر با نام DHTML (برگرفته شده از Dynamic HTML) نیز ارائه گردید . DHTML به پیاده کنندگان اجازه می داد تا بتوانند هر بخشی از يك صفحه مستقر شده در حافظه سرویس گیرنده را با استفاده از جاوا اسکریپت تغییر دهند . DHTML به همراه عرضه غیره منتظره CSS (برگرفته شده از cascading style sheets) به وب و نحوه پیاده سازی صفحات وب يك نیروی تازه داد. پیاده کنندگان توانستند با ترکیب DHTML و روش فریم های مخفی هر بخشی از يك صفحه را با استفاده از اطلاعات دریافت شده از سرویس دهنده refresh نمایند . ویژگی فوق در زمان خود توانست يك تحول بزرگ در عرصه وب را ایجاد نماید .

در ادامه ، DOM (برگرفته شده از Model Object Document) ارائه گردید . برخلاف DHTML که صرفاً امکان تغییر بخش هایی خاص از يك صفحه وب را فراهم می کرد ، DOM دارای يك هدف بلندپروازانه دیگر بود . هدف ارائه يك ساختار برای تمامی صفحه بود . انجام عملیات و یا پردازش بر اساس ساختار فوق ، امکان تغییر يك صفحه را فراهم می آورد . این مرحله بعدی به سمت Ajax بود .

- **Iframes :** با این که روش فریم مخفی بطور غیرقابل پیش بینی متداول گردید ، ولی دارای چالش های مختص به خود بود . افرادی می بایست زمان زیادی را صرف نوشتن يك frameset می کردند تا شرایط استفاده از فریم های مخفی فراهم گردد . زمانی که عنصر `< iframe />` در سال 1997 و به عنوان بخشی از نسخه رسمی HTML 4.0 ارائه گردید ، يك قدم اساسی دیگر به سمت انقلاب بر روی وب حرکت شد .

در مقابل تعریف frameset ، پیاده کنندگان قادر به استفاده از فریم ها در هر مکانی از صفحه شدند . بدین ترتیب پیاده کنندگان توانستند از iframe نامرئی (با بکارگیری قابلیت های CSS) در يك صفحه استفاده نمایند و شرایط عملیاتی لازم جهت مبادله داده بین سرویس گیرنده و سرویس دهنده را فراهم نمایند . زمانی که نسخه نهایی DOM در IE 5.0 و Netscape 6.0 پیاده سازی گردید ، قابلیت ایجاد فریم ها به صورت پویا نیز فراهم گردید . در چنین حالتی از يك تابع جاوا اسکریپت به منظور ایجاد يك iframe ، ایجاد يك درخواست و دریافت پاسخ (جملگی بدون نیاز به اضافه کردن تگ های HTML اضافه در يك صفحه) استفاده می گردید . این روش معروف به روش iframe مخفی گردید .

- **XMLHttp :** پیاده کنندگان مرورگر شرکت مایکروسافت پس از مشاهده عمومیت استفاده از روش هایی نظیر فریم مخفی و iframe مخفی ، تصمیم گرفتند يك ابزار بهتر جهت تعامل بین سرویس دهنده و سرویس گیرنده را ارائه نمایند . این ابزار به شکل يك اکتیوایکس و با نام XMLHttp در سال 2001 ارائه گردید . زمانی که مایکروسافت حمایت خود از XML را از طریق يك کتابخانه با نام MSXML آغاز کرد ، شی XMLHttp نیز به همراه آن ارائه گردید .

با این که شی فوق به همراه خود نام XML را یدک می کشید ولی چیزی بمراتب بیشتر از روشی برای انجام عملیات بر روی داده XML بود . شی فوق همانند يك درخواست HTTP بود که می توانست توسط جاوا اسکریپت کنترل گردد . پیاده کنندگان دارای قدرت دستیابی به کدهای وضعیت HTTP و هدر همانند داده برگردانده شده از طریق سرویس دهنده بودند . داده می توانست به صورت XML ساختاریافته ، اشیاء سریال شده جاوا اسکریپت و یا سایر فرمت های مورد نظر پیاده کننده باشد . در مقابل استفاده از فریم های مخفی و یا iframe ، این امر میسر گردید که به سرویس دهنده از طریق کد و توسط جاوا اسکریپت دستیابی داشت (مستقل از چرخه load و reload صفحه) . شی XMLHttp پس از مدتی به عنوان يك موفقیت بزرگ برای پیاده کنندگان مرورگر IE عرض اندام کرد .

به موازات متداول شدن شی فوق ، پیاده کنندگان در Mozilla نسخه اختصاصی از XMLHttpRequest را پیاده سازی کردند . در مقابل اجازه دستیابی از طریق اکتیو ایکس ، پیاده کنندگان موزیلا متدها و خصلت ها را در يك شی ذاتی مرورگر و با نام XMLHttpRequest ارائه کردند . پس از این که دو مرورگر اصلی در بازار هر يك توانستند به نوعی از XMLHttpRequest حمایت نمایند ، مرورگرهائی دیگر نظیر Opera و Safari نیز هر يك به نوعی حمایت خود از شی فوق را اعلام کردند .

بدین ترتیب پیاده کنندگان صفحات وب توانستند ، صفحاتی را خلق نمایند که با تعبیه کدهای مورد نیاز در آنها ، امکان پردازش در سمت سرویس گیرنده را فراهم نمایند . انجام پردازش های اولیه نظیر تغییر رنگ و شکل ظاهری صفحات ، مدیریت عناصر موجود بر روی يك صفحه و تغییر آنها در زمان اجراء متناسب با عملیات کاربر ، بررسی صحت داده ورودی توسط کاربر در يك فرم ساده اطلاعاتی ، دریافت داده در پس زمینه و عدم نیاز به ارسال تمامی صفحه و بهنگام سازی بخش هائی خاص از آن (با بکارگیری فریم ها) ، نمونه هائی از تاثیر فناوری هائی بود که توان پردازش در سمت سرویس گیرنده را در اختیار پیاده کنندگان قرار می داد .

وب پویا با تمرکز بر روی پردازش های سمت سرویس دهنده

همزمان با گسترش اینترنت و وب ، ایجاد صفحات وب با محتویات پویا با اتکاء به منابع داده موجود در سمت سرویس دهنده و سایر شرایط تاثیر گذار در فرآیند پاسخ به درخواست کاربران مطرح گردید . همین موضوع باعث شد که توجه به سرویس دهنده برای انجام برخی پردازش ها نیز در دستور کار قرار بگیرد . ظهور فناوری هائی نظیر ASP (برگرفته شده از Page Server Active) و PHP (برگرفته شده از Preprocessor Hypertext) از اوایل سال 1996 پاسخی به این نیاز بود . از آن زمان به بعد بود که کم کم واژه برنامه های وب قدم در عرصه نرم افزار گذاشت . اگر به تعاریف آن زمان برگردیم مشاهده می کنیم که در ابتدا از این اصطلاح در خصوص صفحات وب پویا استفاده می گردید .

نگرش جزیره ای به فناوری های مرتبط با برنامه های وب تا اوایل سال 1999 ادامه یافت . تا این که شرکت های متعددی به فکر ارائه يك پلت فرم با در نظر گرفتن واقعیتی به نام اینترنت افتادند . دات نت ، پلت فرم ارائه شده توسط شرکت مایکروسافت يك نمونه در این زمینه است که از صدر تا ذیل آن با در نظر گرفتن واقعیتی با نام اینترنت طراحی و پیاده سازی شده است .

در این دوران بود که به برنامه های وب با يك نگاه جامع تر و سیستماتیک تر نگاه می شد . در طی پنج سال گذشته ، این نوع برنامه ها توانستند با بکارگیری مجموعه ای از فناوری ها منشاء تحولات عمده ای در خصوص بردن اینترنت در متن زندگی (و نه حاشیه !) مردم جهان باشند . ایده " نرم افزار به عنوان سرویس " و یا برنامه نویسی سرویس گرا از نگرش های جدید به مقوله نرم افزار و دنیای برنامه نویسی در عرصه وب است . موج استفاده از اینترنت برای ارائه خدمات online (ارائه شده به کاربران از طریق برنامه های وب) باعث گردید تا معماری و رفتار اینگونه برنامه ها با تامل بیشتر مطالعه و بررسی گردد . هدف ، ایجاد برنامه های وب با کارآیی مطلوب است تا ضمن ارائه خدمات خود دارای ویژگی هائی نظیر : افزایش تعامل کاربر با برنامه (همانند برنامه های desktop) ، استفاده بهینه از محیط انتقال و به حداقل رساندن ترافیک آن ، مدیریت بهینه منابع سمت سرویس گیرنده در پاسخ به رویدادهای محقق شده سمت سرویس گیرنده ، عدم ارسال اطلاعات غیره ضروری توسط سرویس گیرنده و ... باشند . ماحصل این تفکرات ظهور فناوری جدیدی با نام Ajax است که شرکت هائی نظیر

گوگل و مایکروسافت توانستند از سال 2005 با بکارگیری این فناوری تحول بزرگی را در عرصه برخی از سرویس ها و خدمات خود ایجاد نمایند .

جمع بندی !

همزمان با بحث معماری سرویس گیرنده و سرویس دهنده دو واژه fat-client و thin-client قدم در ادبیات نرم افزار گذاشتند . در معماری فوق ، هدف استفاده از توان پردازشی سرویس گیرنده و سرویس دهنده برای اجرای يك برنامه است . استفاده از توان پردازشی هر يك از عناصر اصلی درگیر در معماری فوق و تعریف سهمی خاص برای هر يك (سرویس گیرنده و سرویس دهنده) ، از اهداف اولیه و در عین حال بسیار مهم معماری فوق است . پیاده کنندگان برنامه های کامپیوتری خصوصا " برنامه های وب ، همواره با این سوال مواجه بوده و هستند که در تقسیم سهم پردازش ، چه میزان سهم را می بایست به سرویس گیرنده داد ؟ چه پردازش هایی را می توان در سمت سرویس گیرنده انجام داد ؟ در پاسخ به این سوال رویکردهای مختلفی مطرح شده است . thin و client fat دو نمونه متداول در این زمینه می باشند .

در fat client ، هدف بکارگیری پتانسیل های مرورگر برای انجام اکثر پردازش ها در يك برنامه وب است . رویکرد فوق در اواسط دهه 90 میلادی در کانون توجه طراحان و پیاده کنندگان وب قرار گرفت . علت این امر به تلاش شرکت های بزرگ نرم افزاری در جهت تولید و ارائه فناوریهای بود که توانمندی مرورگر در انجام پردازش را افزایش می داد . ظهور فناوریهای نظیر جاوا اسکریپت ، ایجاد يك اینترفیس برنامه نویسی در مرورگر ، ایجاد يك مدل شی گراء (ولو ناقص) برای يك سند HTML با هدف امکان دستیابی به هر يك از عناصر موجود در آن ، عرضه DHTML و CSS ، جملگی نشاندهنده تلاش هایی است که در آن زمان با هدف افزایش توان پردازش در سمت سرویس گیرنده مطرح شد .

پس از گذشت مدت زمان کوتاهی (از ابتدای سال 1996) ، تمایل به سمت برنامه های وب با ویژگی thin client مطرح گردید . در مدل فوق ، هدف کاهش سهم پردازش در سمت سرویس گیرنده و افزایش پردازش های سمت سرویس دهنده بود . ظهور فناوریهای متعددی نظیر ASP و PHP تنها نمونه ای اندک از تلاش های انجام یافته در آن دوران برای افزایش توان پردازش در سمت سرویس دهنده است .

با این که در سالیان گذشته همچنان از پتانسیل های سمت سرویس گیرنده برای انجام پردازش در برنامه های وب استفاده می گردید ، ولی رفتار اکثر برنامه های وب مبتنی بر این واقعیت بود که : " داده را وارد نمائید ، صفحه را برای سرویس دهنده ارسال نمائید و منتظر دریافت پاسخ باشید " .

هم اینک ما شاهد يك تحول بزرگ دیگر در عرصه برنامه نویسی هستیم . تحولی که مهمترین دستاورد آن ، تولد نسل جدیدی از برنامه های وب است که در آنها کاربر با برنامه دارای تعامل بمراتب بیشتری نسبت به نسل گذشته برنامه های وب است . این تحول عظیم نرم افزاری به کمک مجموعه ای از فن آوری ها که در مجموع به آنها Ajax گفته می شود ، محقق می گردد .

آیا Ajax قصد دارد با تغییر در الگوی رفتاری برنامه های وب فاصله آنها را با برنامه های desktop کاهش دهد ؟ آیا زمان آن رسیده است که برای پردازش های سمت سرویس گیرنده از يك پلت فرم جامع ، همانند پلت فرم های ارائه شده سمت سرویس دهنده ، استفاده گردد ؟ آیا تاکنون از این فناوری به منظور ایجاد برنامه های وب استفاده شده است ؟ با استفاده از چه مکانیزم هایی می توان مبادله اطلاعات بین سرویس گیرنده و سرویس دهنده را کاهش و از پهنای باند موجود استفاده بهینه کرد ؟

اجازه دهید به این سوال و سوالات بسیار دیگر در رابطه با Ajax ، فناوریهای مرتبط با آن و از همه مهمتر کاربرد آن در

بخش های بعدی پاسخ دهیم .