

## XML چیست و چرا دارای اهمیت فراوان است ؟ ( بخش دوم )

در بخش اول، مفاهیم اولیه تکنولوژی XML بررسی و پس از معرفی خانواده بزرگ تکنولوژی های مرتبط با XML ، سه استاندارد لازم بمنظور تعریف نوع سندهای XML تشریح گردید ( XML Namespace و XML-Schema ) و یا ( DTD ) . در بخش دوم به بررسی نحوه انتشار سندهای XML خواهیم پرداخت .

Html بعنوان اسناداری جهت نمایش اطلاعات در عرصه وب ، سالهاست که مطرح و استفاده می گردد. تگ های Html مسئولیت تبیین نحوه نمایش اطلاعات را بر عهده خواهند گرفت . تکنولوژی فوق برای انجام وظایف محوله از پیش فرض های تعریف شده و ثابت (ایستا ) استفاده می نماید . مثلاً " با مشاهده تگ <B> دقیقاً مشخص شده است که می بایست چه نوع واکنشی صورت پذیرد. ( پر رنگ نمودن متن مورد نظر ) . نباید انتظار داشته باشیم که با درج تگ <B> عملیاتی غیر از آنچه از قبل تعریف شده است صورت پذیرد. مفسر تگ های Html پس از انجام تفسیر مربوطه ، واکنش از قبل تعریف شده ای را از خود نشان خواهد داد. مرورگرها بکمک مفسر های مربوطه همواره شرایط یکسانی را برای نمایش فراهم و شرایط خود را بر همه چیز منجمله سلیقه و نقطه نظرات طراح و ... ترجیح می دهند. خودخواهی در نمایش و تحمیل شرایط مربوطه از نکات قابل تامل در تکنولوژی فوق است . Html محدوده و مرزی را برای تفکیک داده ها از نمایش قائل نگردیده و با تلفیق دو مقوله فوق همواره از یک روش ثابت برای نمایش داده ها استفاده می نماید. فرض کنید که یک فایل html داشته و بخواهیم زمانیکه برای کاربر ارسال می گردد در مرورگر مربوطه ، به دو صورت کاملاً متمایز نمایش داده شود . برای نیل به خواسته فوق چاره ای نیست مگر اینکه دو فایل مجزای html را ایجاد که هر یک دارای خروجی اختصاصی خود باشند. در مثال فوق قصد ما نمایش داده های یکسان با فرمت های متفاوت از بعد نمایش است . ما بدنبال روش و یا روش هائی هستیم که قادر به تفکیک بین داده و نمایش باشد. قطعاً Html در این راستا گزینه ای مناسب نخواهد بود. سندهای Html از تگ هائی نظیر <H1> و <P> بمنظور ایجاد ساختار و از تگ هائی نظیر <I> و <Font> بمنظور ایجاد فرمت نمایش اطلاعات استفاده می نمایند . در این راستا عملاً هیچگونه تگی که نشاندهنده نوع اطلاعات و محتویات سند باشد ، وجود ندارد . بدین ترتیب مفسر قادر به تفسیر تگ های فوق صرفاً در حد نمایش اطلاعات بوده و امکان انجام هیچگونه پردازشی وجود نخواهد داشت .

## حل مشکل با استفاده از XML

XML بین دو مقوله داده و نمایش تفکیک قائل شده است. در تکنولوژی فوق بدلیل عدم وجود تگ های از پیش تعریف شده برای نمایش اطلاعات می توان سناریوی مربوط به نحوه نمایش اطلاعات را بکمک یک تکنولوژی دیگر تعریف و تبیین نمود. اطلاعات ذخیره شده بصورت XML را می توان با حالات متفاوت و اعمال سناریو های متفاوت نمایش داد. برخلاف HTML تکنولوژی XML دارای اطلاعات از قبل تعریف شده و مشخصی برای نحوه نمایش اطلاعات نیست. تگ های تعریف شده در یک سند XML ، بصراحت ساختار و محتویات را ارائه خواهند داد . در این حالت می توان نرم افزارهایی را طراحی نمود که قادر به انجام عملیات دلخواه بر روی اطلاعات موجود در سند XML باشند . هنوز یک مسئله وجود دارد : در تگ های تعریف شده در سند XML ، آیتمی وجود ندارد که به کامپیوتر اعلام نماید به چه صورت می بایست اطلاعات مربوط به هر یک از المان ها فرمت ، تا خوانندگان قادر به خواندن آنان باشند . ما به چیزی بیشتر بمنظور تدوین استراتژی نمایش اطلاعات نیاز داریم .

## زبان های تعریف سبک XML

بمنظور حل مشکل اشاره شده می توان اطلاعات مربوط به نحوه نمایش اطلاعات را در یک فایل مجزا قرار داده و در ادامه با استفاده از یک پردازنده Stylesheet فایل فوق را با سند XML ترکیب نمود. Stylesheet توسط زبان های Style نوشته می گردند. هر شخص می تواند زبان اختصاصی خود را در این زمینه پیاده سازی و استفاده نماید ولی XML با دو زبان استاندارد ارائه شده است .

## (Style Sheets Cascading(CSS

CSS یک زبان Style بوده که بمنظور استفاده در سندهای HTML ابداع گردیده ولی بخوبی قادر به فعالیت در کنار سندهای XML نیز می باشد . CSS مکانیزمی مناسب برای نمایش یک سند XML در مرورگر ها است . اکثر ابزارهای تولید صفحات وب دارای امکانات لازم برای تولید اسکریپت های CSS می باشند . CSS دارای دو محدودیت عمده است . اولاً ، قادر به تولید خروجی مناسب و با کیفیت بالا نمی باشد. ثانیاً ، صرفاً باعث تزئین سند شده و قابلیت تغییر در دنباله اطلاعات موجود در سند را نخواهد داشت . محتویات یک سند XML در اغلب حالات دارای یک نظم متفاوت نسبت به حالتی است که قصد نمایش آنها را داریم . در این راستا لازم است قبل از استفاده از CSS بمنظور نشر یک سند ، سند مورد نظر به حالتی که تمایل به نمایش آن را در خروجی داریم ، تبدیل گردد . بدین ترتیب یک مرحله پردازش دیگر به حجم عملیات اضافه خواهد گردید.

## (Style Language Extensible(XSL

XSL یک زبان مبتنی بر XML است که دارای محدودیت های اشاره شده در رابطه با CSS نمی باشد . متأسفانه در حال حاضر ابزارهای لازم برای ایجاد اسکریپت های XSL محدود بوده و در برخی حالات ممکن است نوشتن اسکریپت ها بصورت دستی انجام گیرد . XSL دارای سه ویژگی منحصر بفرد بوده که هر کدام دارای استاندارد اختصاصی خود می باشند.

- (Transformation(XSLT) . ویژگی فوق باعث انتقال یک سند XML به فرمت دیگر می شود. فرمت فوق می تواند یک سند XML دیگر و یا یک سند DHTML باشد .

- (XPath(Pointing) . ویژگی فوق امکان مشخص نمودن هر مکان در یک سند XML را فراهم می نماید . پتانسیل فوق بعنوان هسته اساسی پردازش Sheet Style مطرح بوده و مکانیزم لازم در خصوص اعمال قوانین نمایش در خصوص اطلاعات موجود در سند XML را بدون نیاز به استفاده از تگ های مربوط به Style در بطن و همراه سند ، فراهم می نماید .

- (XSL(Formating) ، فرآیندی است که بر اساس آن امکان اعمال فرمت مورد نظر به اطلاعات، بدون نیاز به نوشتن دستورالعمل هایی که مختص یک دستگاه خروجی خاص می باشند ، را فراهم می نماید . مثلاً" می توان قوانینی مشابه : " تمام عناوین می بایست بصورت پررنگ ، فونت تایمز ، در وسط قرار گرفته ، و دور آنها یک کادر کشیده شده که دارای پهناي N باشد " را بسادگی تعریف کرد . ویژگی فوق یک تحول چشمگیر نسبت به وضعیت فعلی است . در این راستا الزامی به نوشتن قوانین متعدد برای هر یک از دستگاههایی که قصد حمایت از آنان را داریم ، وجود نخواهد داشت .

## StyleSheets چگونه کار می کند ؟

مهمترین اصل تمام زبانهای XML Style ، کمک به طراحان بمنظور ایجاد قوانین ، نسبت دهی و دستورالعمل های پردازش برای انواع اطلاعات موجود در یک سند است . قوانین در یک Stylesheet در تمپلیت ها قرار گرفته و به المان ها و الگوهای اطلاعاتی موجود ، اشاره خواهند کرد . در زمان اجراء ، یک نرم افزار کوچک که پردازنده StyleSheet نامیده می شود ، سند XML و یک StyleSheet را بعنوان ورودی اخذ خواهد کرد ( پردازنده Stylesheet ، درون هر برنامه ای که Stylesheet را به XML اعمال می نماید ، نظیر مرورگرها و یا ابزارهای طراحی صفحات وب وجود دارد) . پردازنده بر اساس دستورالعمل های موجود در Stylesheet عملیات مورد نظر را انجام و بر اساس تمپلیت تعریف شده اطلاعات ذیربط را در آن درج و باعث ایجاد

یک خروجی جدید بعنوان ماحصل کار می نماید. روش فوق برنامه نویسی تعریفی در مقابل رویه ای نامیده می شود ( در این حالت قوانین مربوط به نحوه نمایش تعریف شده اند، نه اینکه دستورالعمل های رویه ای در متن کدها استفاده شده باشد ). نمونه کدهای زیر ، یک تمپلیت XSLT را ایجاد که در آن المان <title> در زمان ایجاد تگ های Html ، درشت نوشته شده خواهند شد .

#### مثال : یک تمپلیت ساده XSLT

```
<xsl:template match="title">
<H1>
<xsl:apply-templates/>
</H1>
</xsl:template>
```

در تمپلیت فوق ، خصلت match دارای مقدار title بوده و این بدین معنی است که قانون می بایست به هر المان موجود در <title> که پردازنده Stylesheet آن را در سند XML پیدا خواهد کرد ، اعمال خواهد شد. قانون موجود در تمپلیت به این نکته اشاره دارد که محتویات المان <title> در سند XML ، می بایست در فایل Html که ایجاد خواهد شد ، درج و قبل از آن تگ <H1> و بدنبال آن تگ </H1> آورده شود . فرض کنید فایل XML مشابه زیر باشد :

#### مثال : سند XML که فایل XSLT بر روی آن اعمال می گردد

```
<title>This text will be very large</title>
```

نتیجه فایل HTML پس از اعمال تمپلیت تعریف شده :

```
<H1>This text will be very large</H1>
```

مثال فوق نمونه ساده ای در این زمینه بوده و در عمل می توان با استفاده از امکانات موجود، تمپلیت های بمراتب پیچیده تر و قدرتمندتری را ایجاد نمود.

## StyleSheet و افزایش انعطاف پذیری

با تفکیک دستورالعمل های فرمت اطلاعات از محتویات اطلاعات و در ادامه استفاده از مکانیزم Stylesheet بمنظور ترکیب آنها با یکدیگر ، قابلیت های فراوانی در اختیار علاقه مندان به این تکنولوژی قرار خواهد گرفت :

- می توان StyleSheet های متفاوتی را برای دستگاه های متفاوت ( مرورگرها ، PDA ، تلفن ) بر روی محیط های اطلاعاتی متفاوت ( online ، چاپ ، CD ) و با اهداف متفاوت ( نمایش مدیریتی ، نمایش تکنیکی ) ایجاد نمود .
- می توان یک Stylesheet ایجاد و آن را بر روی سندهای متعددی اعمال نمود.
- می توان اطلاعات را ویرایش و StyleSheet آنها را یکدیگر تفکیک خواهد کرد .
- می توان تمام اطلاعات موجود را ، صرفاً " با فشردن با یک کلید ، منتشر کرد.

### خلاصه

استانداردهای اساسی در زمان انتشار سندهای XML شامل CSS و XSL/XSLT است . مهمترین نکات در رابطه با تکنولوژی های فوق ، عبارت است از :

- جدا سازی محتویات از فرمت نمایش
- نسبت دهی Style به ساختار
- استفاده از CSS بمنظور تزئین اطلاعات
- استفاده از XSLT در زمانیکه به الگوهای قدرتمندی نیاز است .