

## XSLT چیست ؟

XSLT، از کلمات Transformation: Language Stylesheet eXtensible برای تبدیل یک سند XML به نوع دیگر است. تکنولوژی فوق توسط کنسرسیوم وب با هدف اولیه تبدیل یک سند XML به نوع دیگر، طراحی شده است. XSLT، دارای قابلیت های بمراتب بیشتری بمنظور تبدیل یک سند XML به HTML و سایر فرمت های مبتنی بر متن است. بدین ترتیب می توان تعریف کامل تری در رابطه با تکنولوژی فوق را بصورت زیر ارائه داد :

تعریف XSLT
XSLT، زبانی بمنظور تبدیل ساختار یک سند XML است.

### چرا سند های XML نیازمند تبدیل می باشند ؟

XML، روشی ساده و استاندارد بمنظور مبادله داده های ساختیافته بین برنامه های متفاوت کامپیوتر است. بخشی از موفقیت XML جهت نیل به خواسته فوق، به نحوه نوشتن و خواندن این نوع فایل ها بر می گردد. با استفاده از یک ادیتور متنی می توان بسادگی و بسرعت اقدام به نوشتن یک سند XML، خواندن و اعمال تغییرات در آنان نمود. فراموش نکنیم که مهمترین هدف XML، ارتباط بین سیستم های متفاوت نرم افزاری بوده و در این راستا به دو نیاز حیاتی بخوبی جواب داده است :  
تفکیک داده از نمایش ( ارائه ). نیاز به تفکیک اطلاعات از جزئیات نمایش بر روی یک دستگاه خاص همواره مورد نظر طراحان و پیاده کنندگان بوده است. خواسته فوق، بموازات رشد دستگاه های مبتنی بر اینترنت به یک ضرورت حیاتی تبدیل شده است. سازمان ها و موسسات ارائه دهنده اطلاعات بارز، تمایل به عرضه اطلاعات خود بر روی مرورگرهای وب و سایر دستگاهها نظیر تلویزیون و یا تلفن های مبتنی بر WAP را دارند. بدین ترتیب دامنه استفاده از اطلاعات صرفاً به مرورگرهای وب محدود نبوده و دستگاه های متفاوت مبتنی بر تکنولوژی اینترنت را شامل می گردد.

ارسال داده بین برنامه ها. ارسال اطلاعات از سازمانی به سازمان دیگر بدون درگیر شدن در جزئیات، همواره از آرزوهای بزرگ طراحان و پیاده کنندگان سیستم های نرم افزاری بوده است.

دو خواسته فوق هیچگونه تعارضی با یکدیگر ندارند ، مثلاً" یک لیست قیمت کالا را می توان بر روی صفحه ، نمایش و در همان زمان از آن بعنوان ورودی در سایر برنامه ها استفاده کرد . یکی دیگر از مزایای اساسی XML ، یکپارچگی بین دنیای سندها و داده ها است . بدین ترتیب می توان از یک روش خاص بمنظور عرضه ساختار داده ها ( صرفنظر از اینکه اطلاعات توسط انسان و یا ماشین استفاده می گردد) ، بهره گرفت . صرفنظر از اینکه داده های XML سرانجام توسط انسان و یا یک برنامه نرم افزاری دیگر استفاده می گردند ، به این نکته مهم می بایست اشاره گردد که بندرت داده ها به همان صورت که دریافت می گردند، استفاده خواهند شد. در این راستا لازم است که داده ها در ابتدا به یک فرمت دیگر تبدیل گردند . تبدیل XML به Html ، متداولترین کاربرد استفاده از XSLT است . پس از تبدیل داده ها به فرمت Html ، امکان نمایش آنها در هر مرورگری فراهم خواهد شد . بنابراین جایگاه "تبدیل" در رابطه با ارائه و نمایش داده ها ، به تغییر فرمت سندهای XML به سایر فرمت های متداول نظیر Html برمی گردد . XSLT در این رابطه بخوبی پاسخگو بوده و با استفاده از قوانین موجود قادر به تبدیل نوع یک سند به نوع دیگر با توجه به نیازهای مربوطه خواهد بود .

بمنظور ارسال اطلاعات بین برنامه های متفاوت، می بایست قادر به تبدیل داده ها از مدل استفاده شده توسط یک برنامه به مدل استفاده شده توسط برنامه دیگر باشیم . بمنظور تامین اطلاعات مورد نیاز یک برنامه ، از فرمت های متفاوت استفاده می گردد . یک فایل حاوی داده های جدانشده توسط ویرگول ، یک اسکریپت SQL ، یک پیام HTTP و یا مجموعه ای از توابع مربوط به یک اینترفیس خاص برنامه نویسی ، نمونه هایی در این زمینه می باشند . بموازات رشد تجارت الکترونیکی مبتنی بر XML ، نقش XSLT در تبدیل داده ها بین برنامه های متفاوت ، بسیار حائز اهمیت شده است . در این راستا استانداردهای متعددی وجود دارد . مثلاً" در صنعت روزنامه نگاری از یک فرمت خاص برای مبادله اطلاعات، نسبت به صنعت تلویزیون استفاده می گردد . در برخی حالات دیگر ممکن است ، تمایل به استخراج یک آدرس خاص از یک لیست ثبت سفارش و الحاق آن به لیست صورتحساب را داشته باشیم . بهرحال ، استخراج و ترکیب داده ها از یک مجموعه اسناد XML و ایجاد اطلاعات مورد نیاز برای سایر اسناد XML ، یک ضرورت بوده و XSLT در این راستا ابزاری مناسب است .

## **XSLT چگونه یک سند XML را تبدیل می نماید ؟**

برای تبدیل یک سند XML و ایجاد خروجی مورد نظر، دو مرحله متفاوت را می بایست دنبال نمود :

- اولین مرحله ، شامل یک تبدیل ساختاری است . در این حالت داده ها از ساختار استفاده شده در یک سند XML ، به ساختاری دیگر که تامین کننده خروجی مورد نیاز است ، تبدیل خواهند شد .
- دومین مرحله ، ایجاد فرمت مورد نیاز است . ساختار جدید به فرمت مورد نظر نظیر Html و یا PDF تبدیل و در خروجی نمایش داده خواهد شد .

در رابطه با دومین مرحله ، در بخش قبل موارد لازم بیان گردید : ساختمان داده ئی که از اولین مرحله بدست می آید، می تواند بعنوان یک HTML ، یک فایل متن و یا یک سند XML دیگر در خروجی ارائه گردد . خروجی Html ، امکان نمایش مستقیم اطلاعات در یک مرورگر و یا تغذیه به یک برنامه پیشرفته واژه پرداز را دارا است . خروجی مبتنی بر متن ، امکان فرمت دهی داده ها را با یک روش قابل قبول بمنظور استفاده توسط سایر برنامه ها فراهم می نماید . با تبدیل یک سند XML به نوع دیگر که خود نیز از نوع XML خواهد بود ، امکان استفاده مستقیم آن برای سایر برنامه هائی که قادر به دریافت داده های خود بصورت XML می باشند ، فراهم می گردد . مثلاً" یک تبدیل کننده XSLT ، ممکن است میزان فروش ماهیانه را بعنوان یک سند XML اخذ و یک منحنی هیستوگرام را بصورت XML در خروجی ایجاد نماید .

در مرحله اول ( تبدیل ) ، عملیات لازم در خصوص ایجاد خروجی با فرمت مناسب انجام خواهد شد . در مرحله فوق عملیاتی نظیر : انتخاب داده ، گردآوری و گروه بندی داده ها ، مرتب سازی داده ها و انجام عملیات محاسباتی ( نظیر تبدیل اینچ به سانتیمتر ) صورت می پذیرد. نحوه انجام عملیات فوق به چه صورت است ؟ قبل از عرضه XSLT ، امکان پردازش سندهای XML ، صرفاً" از طریق نوشتن یک برنامه خاص میسر بود . برنامه ها با استفاده از یک رابط برنامه نویسی (API) ، پارسر را فراخوانده تا امکان اخذ اطلاعات در رابطه با سند و یا انجام عملیات مورد نظر فراهم گردد . در این راستا از دو رابط برنامه نویسی استفاده می گردد :

( XML Simple API for(SAX ) و (object Model Document(DOM) .

رابط برنامه نویسی SAX ، یک اینترفیس مبتنی بر رویداد بوده که پارسر در رابطه با هر یک از بخش های موجود در سند ، اطلاعات لازم را در اختیار برنامه متقاضی قرار خواهد داد. در صورتیکه از رابط برنامه نویسی DOM استفاده گردد، پارسر از طریق ایجاد یک ساختار شبه درختی در حافظه ، امکان ارتباط با آن را پیدا می نماید . در ادامه ، با نوشتن یک برنامه خاص ( مثلاً" با استفاده از زبانهای برنامه نویسی نظیر VB ، جاوا ) امکان ارتباط با درخت بوجود می آمد . در مدل فوق ، برای پردازش هر نوع سند XML ، لازم است که یک برنامه خاص و اختصاصی نوشته شده تا با دنبال نمودن مراحل مورد نظر، عملیات لازم در رابطه با یک سند XML صورت پذیرد.

چگونه می توان از XSLT بمنظور تبدیل یک سند XML در مقابل نوشتن برنامه های اختصاصی استفاده کرد ؟ طراحی XSLT ، مبتنی بر این واقعیت بوده است که این نوع برنامه ها بسیار مشابه هم بوده و می توان بمنظور تشریح عملکرد مربوطه ، از یک زبان تعریفی سطح بالا ( در مقابل نوشتن هر یک از برنامه ها توسط زبانهای نظیر VB و یا جاوا) استفاده کرد . تبدیلات انجام شده ، از مجموعه ای قوانین نشات خواهد گرفت . قوانین فوق ، نحوه ایجاد خروجی را در صورت وجود یک الگوی خاص در سند ، مشخص می نمایند . زبان فوق تعریفی بوده و تبدیل مورد نیاز تشریح می گردد ( در مقابل ارائه دنباله ای از دستورالعمل های رویه ای برای عملیات تبدیل) . XSLT ، تبدیل مورد نیاز را تشریح و در ادامه با تاکید بر نقش پردازنده XSL ، مناسبترین و کارآمدترین روش برای عملیات انتخاب می گردد .

XSLT ، همچنان بر نقش پارسرها تاکید ( پارسر سازگار با DOM و یا پارسر سازگار با SAX) ، و از آن برای تبدیل یک سند XML به یک ساختار درختی استفاده می نماید . در حقیقت ، پردازش بر روی یک سند XML توسط XSLT ، از طریق ساختار درختی ارائه شده از سند ، محقق می گردد . در DOM ، هر آیتم در یک سند XML ( المان ها ، خصلت ها ، دستورالعمل های پردازش ) ، بعنوان یک گره در نظر گرفته می شود . با بکارگیری XSLT ، مجهز به یک زبان برنامه نویسی سطح بالا شده که قادر به حرکت در طول درخت و گره های موجود ، انتخاب گره ها و انجام عملیات پیچیده پردازش بر روی گره ها ، خواهیم بود . مدل درخت XSLT ، در برخی مفاهیم مشابه DOM بوده ولی دقیقا " مشابه آن نیست .

## مقایسه XSLT و SQL

در یک بانک اطلاعاتی رابطه ای ، داده ها شامل مجموعه ای از جداول می باشند . داده ها در فایل هائی تخت با فرمت های مورد نظر ذخیره می گردند . قدرت یک بانک اطلاعاتی رابطه ای از ساختمان داده آن نبوده و به زبانی که قادر به پردازش داده ها است (SQL) ، وابسته است . در روشی مشابه ، XML ، صرفا " یک ساختمان داده مناسب از اطلاعات را تعریف و با استفاده از یک زبان سطح بالا می توان عملیات دلخواه در رابطه با ساختمان داده را انجام داد (XSLT) .

SQL و XSLT دو زبان کاملا " متفاوت نسبت بیکدیگر می باشند . در حالتی خاص ، می توان این ادعا را داشت که آنان دارای شباهت هائی در برخی موارد عملیاتی هستند . مثلا " بمنظور پردازش داده های موجود در یک بانک اطلاعاتی رابطه ای و یا یک سند XML ، زبان پردازش می بایست مبتنی بر یک گرامر مشخص باشد . در SQL از عبارت SELECT و در XSLT از عبارات XPath استفاده می گردد .

زبان عبارات XPath، یکی از بخش های اساسی XSLT بوده که توسط کنسرسیوم وب استاندارد شده است. امکان استفاده از XPath مستقل از XSLT نیز وجود دارد. گرامر XPath، امکان بازیابی گره های خاصی را از یک سند XML فراهم می نماید. بدین منظور ممکن است از طریق مسیری در سند XML و یا از طریق فهرستی که گره ها در آن قرار خواهند گرفت، این امر تحقق یابد. با استفاده از XPath، امکان دستیابی به گره هائی خاص فراهم و در ادامه با استفاده از XSLT امکان اخذ نتایج مورد نظر بوجود می آید. (اجراء پرس وجوی لازم)

XSLT و SQL دارای شباهت های دیگری نیز می باشند. هر دو زبان، دارای یک ویژگی مهم با نام closure می باشند. ویژگی فوق، بدین مفهوم است که خروجی، دارای ساختمان داده مشابه ورودی است. مثلاً "SQL خروجی خود را بصورت یک جدول و XSLT بصورت یک درخت ارائه خواهد داد. بدین ترتیب می توان خروجی یک عملیات را بعنوان ورودی در اختیار عملیات بعدی قرار داد. در SQL این عملیات با استفاده از تعریف view و یا Subquery و در XSLT با ارسال داده از طریق مجموعه ای stylesheet انجام می گیرد.

در دنیای واقعی وجود XSLT و SQL ضرورت داشته و ارتباطات متعددی بین آنها وجود خواهد داشت. داده ها عموماً در بانک های اطلاعاتی رابطه ای ذخیره و با فرمت XML بین سیستم های متنوع ارسال خواهند شد. مدل های داده در هر یک از سیستم های فوق با یکدیگر متفاوت بوده و XSLT قادر به ایفای نقشی حساس در رابطه با تبدیل مدل های متفاوت داده است. تولیدکنندگان متفاوت بانک های اطلاعاتی در تلاش برای ارائه محصول خود بگونه ای هستند که امکان ارتباط XML و SQL را فراهم نماید. SQL Server 2000 امکان استفاده از پرس و جو های مبتنی بر XPath را حمایت می نماید.

## پردازنده XSLT

مهمترین رسالت پردازنده XSLT، بکارگیری یک stylesheet XML در رابطه با یک سند XML و ایجاد (تولید) خروجی مناسب است. لازم به یادآوری است که هر یک از موارد اشاره شده، بمنزله یک برنامه XML بوده و بدیهی است که ساختار هر یک از آنها یک درخت باشد. عملکرد پردازنده XSLT وابسته به وجود درخت های فوق است. پردازنده های متعددی در رابطه با XSLT نظیر Saxon, xt, MSXML3 وجود دارد. پردازنده های فوق را می توان بصورت رایگان تهیه نمود.

پردازنده saxon، قادر به تبدیل سند XML به سایر مدل های دیگر است (یک سند Html). برای استفاده از برنامه فوق، می

بایست برنامه Instan Saxon را از آدرس <http://users.iclway.co.uk/mhkay/saxon/instant.html> دریافت و بر

روی سیستم نصب کرد . برنامه فوق به زبان جاوا نوشته شده و می توان آن را مستقیماً" از طریق خط دستور اجراء نمود ( ضرورتی به داشتن مرورگرهای وب و یا سرویس دهنده وب نخواهد بود ) . مرورگرها و سرویس دهندگان وب در این حالت با سند تبدیل یافته سروکار خواهند داشت .

xt ، یکی دیگر از پردازنده های XSLT است . برنامه فوق با زبان جاوا نوشته شده و امکان اجرای آن از طریق خط دستور وجود دارد . برای دریافت پردازنده فوق می توان به آدرس <http://www.jclark.com/xml/xt.html> مراجعه نمود. پردازنده فوق همانند saxon ، قادر به فعالیت با پارسر های مبتنی بر SAX می باشد .

امکان اجرای XSLT stylesheet به همراه مرورگر IE نیز وجود دارد . بدین منظور می بایست نسخه پنج مرورگر فوق و آخرین نسخه پردازنده MSXML مربوط به شرکت مایکروسافت ، بر روی سیستم نصب گردند. برای دریافت آخرین نسخه پردازنده فوق می توان به آدرس [www.microsoft.com/xml](http://www.microsoft.com/xml) مراجعه کرد .

مثال : در ادامه به بررسی مثالی خواهیم پرداخت که نحوه استفاده از XSLT بمنظور تبدیل یک سند ساده XML را نشان خواهد داد . فرض کنید ، یک سند ساده XML مطابق زیر را داشته باشیم :

یک سند ساده XML
<pre>&lt;?xml version="1.0" encoding="iso-8859-1"?&gt; &lt;greeting&gt;Hello, world!&lt;/greeting&gt;</pre>

خروجی مورد نظر سند XML یک فایل Html بصورت زیر خواهد بود .

خروجی Html
<pre>&lt;html&gt; &lt;head&gt;   &lt;title&gt;Today's greeting&lt;/title&gt; &lt;/head&gt; &lt;body&gt;   &lt;p&gt;Hello, world!&lt;/p&gt;</pre>

```
</body>
</html>
```

XSLT زیر بمنظور تبدیل سند XML و ایجاد خروجی HTML استفاده می گردد .

### XSLT بمنظور تبدیل سند XML و ایجاد خروجی HTML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <head>
    <title>Today's greeting</title>
  </head>
  <body>
    <p><xsl:value-of select="greeting"/></p>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

### اجرای StyleSheet

برای اجرای stylesheet با استفاده از سه پردازنده اشاره شده در بخش قبل ، مراحل زیر را می بایست دنبال کرد.

استفاده از پردازنده **saxon** . بمنظور اجرای Stylesheet فوق با استفاده از پردازنده **saxon** ، عملیات زیر را می بایست انجام

داد :

- دریافت برنامه پردازنده ( آدرس مربوطه در بخش قبل اشاره گردید )
- نصب برنامه **saxon.exe** در یک فولدر مناسب

- با استفاده از برنامه notepad دو فایل اشاره شده را در فایل هائی با نام hello.xml و hello.xml ذخیره نمائید.
- خط دستور DOS را فعال نمائید. (start|programs|MSDOS prompt).
- دستور saxon hello.xml hello.xml را در خط دستور تایپ نمائید .
- خروجی html بر روی صفحه نمایش داده خواهد شد .

در صورتیکه قصد مشاهده خروجی را در مرورگر داشته باشید ، خروجی خط دستور را در یک فایل Html بصورت زیر ذخیره نمائید .

```
Saxon hello.xml hello.xml > hello.html
```

استفاده از پردازنده xt . برای اجرای پردازنده xt ، از روشی مشابه Saxon استفاده شده با این تفاوت که در عوض استفاده از برنامه saxon از برنامه xt استفاده می گردد .

استفاده از پردازنده MSXML . بمنظور اجرای stylesheet به همراه مرورگر IE ، می بایست تغییرات اندکی در سند XML بمنظور مراجعه به stylesheet ایجاد گردد :

```
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet type="text/xsl"
href="hello.xml"?>
<greeting>Hello, world!</greeting>
```

در ادامه می توان فایل hello.xml را در مرورگر IE ، فعال و مشاهده نمود . مرورگر فوق ، سند XML را خوانده و از وجود یک stylesheet آگاه می گردد . در ادامه stylesheet مربوطه فعال و بمنظور تبدیل سند XML و ایجاد خروجی Html ، آن را اجراء می نماید . در صورتیکه بر روی صفحه نمایشگر عبارت hello,world نمایش داده نشود و صرفاً " سند XML مشاهده گردد، علت عدم استفاده از آخرین نسخه MSXML به همراه مرورگر بوده و می بایست آخرین نسخه را از آدرس اشاره شده دریافت و بر روی سیستم نصب نمود .



## تشریح نحوه عملکرد XSLT در رابطه با مثال فوق

اولین خط فایل stylesheet ، شامل دستور زیر است :

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

دستور فوق، اعلام می نماید که یک XSLT stylesheet ، خود بعنوان یک سند XML است . برای encoding کاراکترها ،می توان از استانداردهای متعدد موجود منجمله utf-8 استفاده کرد .

در ادامه با خط زیر مواجه می گردیم :

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
```

خط فوق، عنوان استاندارد XSLT است . یک المان شروع که نوع سند را بعنوان یک stylesheet معرفی می نماید . خصلت xmlns:xsl تعریف یک namespace بوده که مشخص می نماید پیشوند xsl در ادامه بمنظور مراجعه به المان های تعریف شده ( استاندارد کنسرسیوم وب XSLT ) ، استفاده خواهد شد .

در ادامه با خط زیر مواجه می شویم :

```
<xsl:template match="/">
```

المان <xsl:template> ، یک قانون تمپلیت را تعریف که بر اساس آن نحوه برخورد با بخش های خاص سند مبداء در هنگام پردازش، تبیین می گردد . خصلت "/" مشخص می نماید که قانون فوق از ابتدای سند مبداء اعمال خواهد شد . در حقیقت، عبارت فوق یک عبارت مبتنی بر XPath بوده و المان ریشه سند XML را مشخص می نماید . یک سند XML دارای یک ساختار سلسله مراتبی بوده و همانند یونیکس که از "/" برای مشخص نمودن ریشه استفاده می نماید، XPath نیز از "/" برای مشخص نمودن المان ریشه در یک سند XML استفاده می نماید . در مدل DOM به عنصر فوق "Document object" و در XPath به آن ریشه گفته می شود .

در ادامه با خطوط زیر مواجه می شویم :

```
<html>
```

```

<head>
  <title>Today's greeting</title>
</head>
<body>
  <p><xsl:value-of select="greeting" /></p>
</body>
</html>

```

بدنه تمپلیت، نحوه ایجاد خروجی را مشخص می نماید. اکثر بدنه تمپلیت در مثال فوق، مجموعه ای از المان های HTML و متن مورد نظر را که می بایست در فایل خروجی منعکس گردند، شامل می گردد. در دستورات فوق یک المان خاص با نام `<xsl:value-of>` وجود داشته که یک دستورالعمل XSLT است (چون از namespace مربوطه با نام xsl استفاده می نماید). دستورالعمل فوق، مقدار یک گره موجود در سند مبداء را در سند خروجی تکثیر می نماید. خصلت SELECT مربوط به المان، گره ای را که مقدار آن می بایست انتخاب شود را مشخص می نماید. (تمام المان های `<greeting>` که فرزند گره مربوطه می باشند، انتخاب می گردند). در ادامه با دستور زیر مواجه می گردیم:

```
</xsl:template>
```

برای یک تمپلیت ساده نظیر آنچه در مثال فوق وجود داشت، می توان برخی از بخش ها را حذف نمود. با توجه به اینکه صرفاً از یک تمپلیت استفاده شده است المان `<xsl:template>` را می توان حذف نمود. در این حالت Stylesheet مربوطه بصورت زیر خواهد بود:

```

<html xsl:version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<head>
  <title>Today's greeting</title>
</head>
<body>

```

```
<p><xsl:value-of select="greeting"/></p>  
</body>  
</html>
```