XPath چیست ؟ (بخش اول)

XML ، بهمراه خود مجموعه ای از سایر تکنولوژی ها را ایجاد کرده است . XSL یکی از مهمترین تکنولوژی های مرتبط با XML بوده که عموما" به سه تکنولوژی دیگر اشاره می نماید :

- XSLT : زبانی برای تبدیل ساختار یک سند XML
- XPath : زبانی برای آدرس دهی بخش های متفاوت یک سند XML
- XSL-FO : زبانی بمنظور فرمت دهی یک سند XML

XPath یکی از تکنولوژی های مرتبط با XML بوده که ارتباط بسیار نزدیکی با XSLT دارد . بنابراین لازم است قبل از آشنائی با XPath ،نگاهی سریع به تکنولوژی XSLT داشته و از این رهگذر جایگاه XPath را تبین نمائیم .XSLT زبانی مبتنی بر قوانین بوده و با سایر زبان های برنامه نویسی تفاوت دارد. XSLT مبتنی بر مجموعه ای از تمپلیت ها است که هر یک بر اساس قوانین تعریف شده شکل واقعی خود را پیدا خواهند کرد .در حقیقت یک تمپلیت با اتکاء بر قوانین تعریف شده، نحوه پردازش سندهای XML را مشخص می نماید.به عبارت دیگر یک Stylesheet ، نحوه ایجاد خروجی مورد نظر در زمان مواجه شدن با یک الگو در سند XML را مشخص می نماید. در XSLT بر اساس مجموعه ای از قوانین، تمپلیت (تمپلیت ها) تعریف و در زمان تبدیل یک سند XML از قوانین موجود در XSLT برای یافتن یک الگو در سند XML استفاده می گردد . در صورتیکه الگوی مورد نظر در سند XML پیدا گردد ، واکنش های پیش بینی شده ، انجام و خروجی مورد نظر ایجاد می گردد. یک StyleSheet می تواند نحوه برخورد با یک المان خاص در زمان عملیات تبدیل را مشخص و تعریف نمائید. مثلا" در صورتیکه المانی با نام NAME در سند XML پیدا گردید، می توان واکنش های مورد نظر را بمنظور برخورد با واقعیت موجود مشخص کرد. گرامر تمپلیت در این حالت بصورت زیر خواهد بود :

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

مثال: استفاده از XSLT بمنظور تبدیل یک سند XML

در این مثال با استفاده از XSLT یک سند XML به Html تبدیل و در خروجی نمایش داده می شود .

سند XML نمونه (**Test.xml**)

```
<?xml version="1.0" ?>
<?xml:stylesheet type="text/xsl" href="Test.xsl"?>
<PEOPLE>
  <PERSON>
    <NAME>Webmaster</NAME>
     <EMAIL>webmaster@Srco.ir</EMAIL>
  </PERSON>
  <PERSON>
   <NAME>Webadmin</NAME>
    <EMAIL>info@Srco.ir</EMAIL>
  </PERSON>
</PEOPLE>
```

در سند فوق از یک دستورالعمل پردازشی بمنظور مراجعه به Stylesheet بصورت زیر استفاده شده است :

```
<?xml:stylesheet type="text/xsl" href="Test.xsl"?>
```

خروجی Html مورد نظر

Name	Email
Webmaster	webmaster@Srco.ir
Webadmin	info@Srco.ir

سند XSLT بمنظور تبدیل سند XML به HTML برای نمایش در مرورگر (**Test.xsl**)

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-
xsl">
<xsl:template match="/">
<HTML>
<BODY>
 <TABLE BORDER="3">
 <TR>
 <TD>Name</TD>
 <TD>Email</TD>
 </TR>
   <xsl:for-each select="PEOPLE/PERSON">
     <TR>
     <TD><xsl:value-of select="NAME"/></TD>
     <TD><xsl:value-of select="EMAIL"/></TD>
    </TR>
   </xsl:for-each>
 </TABLE>
</BODY>
</HTML>
</xsl:template>
</xsl:stylesheet>
```

در هر فایل XSL ، می بایست XSL namespace معرفی گردد . بدین ترتیب پارسر از نسخه XSLT استفاده شده ، آگاهی لازم را پیدا خواهد کرد .

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

توضیحاتی در رابطه با مثال فوق :

- قبل از انجام هر گونه پردازش در رابطه با یک سند XML ، می بایست با استفاده از یک **عبارت XPath** بخش مورد نظر در سند انتخاب گردد .عملیات فوق توسط یک عملگر match انجام می شود . در صورتیکه تمام سند انتخاب گردد، از "/" = match استفاده می گردد. یکی دیگر از رویکردهای موجود، استفاده از المانی است که نشاندهنده تمام سند است . در مثال فوق،المان موجود در سند که دارای چنین رسالتی است با استفاده از PEPOLE = match مشخص می گردد.

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

- عبارت زیر، تمام المان های PERSON در بافتار PEPOLE را با استفاده از PEPOLE/PERSON که عبارتی از **نوع XPath است**، پیدا خواهد کرد. در صورتیکه گره انتخابی، شامل تمام المان ها در ریشه باشد ، تمام المان های PEOPLE انتخاب خواهند شد . با توجه به اینکه قصد نمایش تمام المان های PERSON در سند خروجی وجود دارد ، از عبارت فوق استفاده شده است . عبارت for-each یک حلقه تکرار بوده که باعث انجام پردازش های یکسان در رابطه با المان های مورد نظر(انتخابی) خواهد شد .

```
<xsl:for-each select="PEOPLE/PERSON">
```

- زمانیکه یک المان PEPOLE توسط عبارت xsl:for-each انتخاب گردید، با استفاده از عبارت xsl:value-of مقدار مربوط به المان استخراج و در فایل خروجی قرار می گیرد. در مثال فوق، مقدار ذخیره شده در المان NAME در خروجی قرار خواهد گرفت .

```
<xsl:value-of select="NAME"/>
```

CSS

هسته اولیه XSL از CSS شکل گرفته است . CSS بمنظور تعریف و افزودن فرمت به یک فایل Html استفاده می گردد . گرامر استفاده شده در یک Stylesheet مربوط به XSLT مشابه گرامر استفاده شده در CSS است . Stylesheet های

استفاده شده در XSLT دارای عملکردی بسیار متفاوت نسبت به نمونه های خود در CSS می باشند . CSS امکان تعریف زنگ ها ، زمینه ها ، نوع و اندازه فونت ها را برای یک صفحه وب Html فراهم می نماید . XSLT امکان تبدیل یک فایل XML را به فرمتی دیگر فراهم می نماید . در صورتیکه هدف صرفا" تعریف فرمت و قالب برای یک صفحه وب باشد ، می توان همچنان از CSS استفاده کرد . استفاده از CSS با توجه به عمومیت آن، برای اکثر استفاده کنندگان گزینه ای مناسب خواهد بود .

Language XML Query

XML ، امکان تعریف تگ های اختصاصی را توسط مولفین سندها ، فراهم می نماید، بنابراین تبدیل یک سند XML به نوع دیگر، همواره بعنوان یک نیاز خواهد بود . از طرف دیگر، مرورگر قادر به نمایش مستقیم اسناد XML نمی باشند . بنابراین ضروری است که یک سند XML به HTML تبدیل تا امکان نمایش آن توسط مرورگرهای وب فراهم گردد . بمنظور پاسخ به نیازهای فوق، شرکت های ماکروسافت، Texcel و WebMethods در سال ۱۹۹۸ پیشنهاد ایجاد یک زبان پرسو جو برای XML را به کنسرسیوم وب، ارائه دادند (**XQL**: Query Language XML) . بخشی از پیشنهاد فوق به نحوه استفاده از زبان XSL در اسناد XML اشاره داشت . در سال ۱۹۹۹ کنسرسیوم وب ، تصمیم به یکپارچه نمودن تمامی تحقیقات انجام شده در رابطه با " ایجاد یک مدل اساسی برای پرس و جو " ، گردید . و بر همین اساس XSLT معرفی و عرضه گردید .

XPath

در زمان پیاده سازی XSLT ، گروه دیگری در کنسرسیوم وب بر روی یکی از تکنولوژی های مرتبط با XML و با نام XPointer فعالیت خود را آغاز نمودند. XPointer از ایده تگ های anchor در یک سطح جدید استفاده می کرد . هم XPointer و هم XSLT نیازمند روشی بمنظور اشاره به بخش های متفاوت یک سند XML می باشند. XSLT نیازمند انتخاب بخش های از یک سند XML بمنظور عملیات تبدیل بوده و XPointer بمنظور الحاق دو سند به امکلن فوق نیاز خواهد داشت . بدین منظور می بایست از یک گرامر متداول در این زمینه استفاده تا امکان بکارگیری آن در XSLT و XPointer فراهم گردد . این تکنولوژی جدید، XPath نامیده شد . با اینکه XPath زیرمجموعه ای از XSLTاست ولی می تواند بصورت مستقل نیز استفاده گردد .

XPath زبانی برای یافتن اطلاعات در یک سند XML است . با استفاده از XPath می توان محل و موقعیت ساختار سند و یا داده های موجود در یک سند XML را مشخص نمود. پس از مشخص نمودن موقعیت و مکان المان مورد نظر در یک سند XML ، می توان با استفاده از XSLT پردازش های لازم را در رابطه با اطلاعات مربوطه، انجام داد .کنسرسیوم وب ، تعریف زیر را در ارتباط با XPath ارائه نموده است :

تعریف XPath
XPath ، زبانی بمنظورآدرس دهی بخش های متفاوت یک سند XML بمنظور استفاده در XSLT و XPointer است .

هدف اولیه XPath ،امکان آدرس دهی بخش های متفاوت یک سند XML است . بمنظور تامین خواسته فوق از امکانات و پتانسیل های متعددی بمنظور انجام عملیات بر روی رشته ها ، اعداد و منطق استفاده می شود . XPath از یک گرامر فشرده و عدم مبتنی بر XML بهمراه URI و مقادیر خصلت های XML استفاده می نماید.دلیل انتخاب نام XPath برای تکنولوژی فوق بدین علت است که در حقیقت از یک آدرس بمنظور حرکت در طول یک سند XML با ساختار سلسله مراتبی استفاده می گردد. XPath یک سند XML را بعنوان درختی از گره ها شبیه سازی می نماید . در این راستا ، گره های متفاوتی نظیر: گره های Element ، گره های Attribute و گره های Text ، وجود دارد برای هر گره توسط XPath ، یک رشته در نظر گرفته می شود . برخی از انواع خاص گره ها دارای اسامی اختصاصی خود می باشند . XPath بطور کامل XML Namespace را حمایت می نماید .بنابراین نام یک گره توسط یک زوج ، شامل یک بخش محلی و یک Namespace از نوع URIمشخص می گردد (نام توسعه یافته) .

برخی از مفاهیم اولیه XPath

گرامر استفاده شده در XPath شباهت زیادی به نحوه آدرس دهی فایل ها در یک سیتستم آدرس دهی فایل ها دارد . در صورتیکه مسیر با "/" شروع گردد ، نشاندهنده یک مسیر مطلق به المان مورد نیاز است . در صورتیکه آدرس با " //" شروع گردد، تمام المان هائی که با شرایط اعلام شده مطابقت نمایند، انتخاب می گردند . مثلا" //Price" ، باعث انتخاب تمام المان های price موجود در سند می گردد .

Context (بافتار)

بافتار یک پرس وجو ، گره ای در سند XML است که پردازشی بر روی آن در حال انجام است . بنابراین در تمپلیت زیر :

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

ما در بافتار ریشه سند XML می باشیم . زمانیکه از از حلقه تکرار xsl:for-each استفاده می گردد ، بافتار، گره ای است که در حال حاضر از طریق حلقه تکرار بر روی آن قرار گرفته ایم . شناخت بافتاری که توسط یک تمپلیت XSL پردازش می گردد ، بسیار حایز اهمیت بوده و در خیلی از موارد و زمانیکه یک فایل XSL نوشته شده ، خروجی مورد نظر را تولید نمی نماید ، ممکن است اشکال از بافتار باشد . زمانیکه عملیات اشکال زدائی XSL را انجام خواهیم داد ، اولین سوالی که مطرح خواهد شد ، ماهیت بافتار پردازش شده است .

مکان یابی مسیرها

با مکان یابی مسیرها ، بافتار مربوط به گره ای که قصد یافتن آن را داریم ، مشخص می گردد . برای تعیین موقعیت یک مسیر، می توان از دو روش کوتاه و یا غیرکوتاه استفاده کرد . بمنظور انتخاب یکی از روش های فوق، می بایست به حمایت آن از طرف پارسر مطمئن گردید(از سال ۲۰۰۰ ، پارسر MSXML ماکروسافت، دو روش فوق را حمایت می نماید) .

- گرامر کوتاه (Abbreviated) . مثال زیر، نحوه مکان یابی المان های موجود در یک سند XML را به صورت کوتاه توسط XPath نشان می دهد.

```
<xsl:for-each select="PEOPLE//PERSON">
```

- گرامر غیرکوتاه (unabbreviated) .مثال زیر ، نحوه مکان یابی المان های موجود در یک سند XML را به صورت غیرکوتاه توسط XPath نشان می ده‌د.

```
<xsl:for-each select="child::PEOPLE/descendant::PERSON">
```