

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## فصل ۱۲



تهیه کننده: رویا لاله زاری

رشته: نرم افزار کامپیوتر

استاد مربوطه: فرزانه دباغیان

# ابزارهای XML



XML ساده است و با تکنیک های همراه خود قدرتش را کنترل می کند. برای کار با اسناد XML دو ابزار قوی XPath , XSLT وجود دارند. ابزار XSLT امکان می دهد که اطلاعات مورد نیاز خود را استخراج و سپس به صورت دلخواه تبدیل کرد. مهمترین کاربرد XSLT تبدیل XML به HTML است. XPath روشی برای تشخیص قسمت های مختلف یک سند است.



ممکن است سوالی مطرح شود که مرورگر تگ های سفارشی را چگونه قالب بندی می کند؟

این وظیفه طراح صحنه است که مشخص کند یک تگ چگونه نمایش داده شود. XLT-FO و CSS دو سیستم اصلی برای قالب بندی اسناد XML هستند.

XLink و XPointer نیز پیوندها و تصاویر را به اسناد XML اضافه می کنند.

# قواعد تگ نویسی در XML



برای نوشتن XML قواعد زیر باید رعایت شوند:

یک عنصر پایه نیاز است.

عناصر نباید تداخل داشته باشند.

حروف کوچک و بزرگ مهم است.

مقادیر باید داخل گیومه باشند.

ارجاع به معادلهای نامی باید اعلام شوند.

نامهای مورد قبول و معتبر عناصر و ویژگی ها می توانند با یکی از

حروف، زیر خط (-) یا دو نقطه (:), شروع می شود.





در عوض از XML برای طراحی زبان نشانه ای سفارشی استفاده می شود که اصطلاحاً کاربرد XML (XML Application) نامیده می شود. سپس اسناد را با این زبان ها می نویسند.

این زبان ها با تعیین تگ ها (عناصر) و ویژگی های (Attributes) آنها مشخص می شوند. برای مثال ویژگی های تگ P می تواند شامل اندازه فونت، نوع فونت و رنگ فونت باشد.

قوانین این کار یک طرح، مدل یا اختصاراً شما گفته می شود.

# دو روش برای نوشتن شما: DTD ها و XML schema



سند DTD بصورت زیر تعریف می شود:

```
<?xml version="1.0"?>
```

```
<!DOCTYPE endangered _species[
```

...

# سیستم تعریف نوع سند DTD



همانطوریکه بیان شد XML زبانی برای تولید زبان های دیگر است و تگ های زبانی مانند XHTML را می توان با آن ایجاد کرد.

ساختار یک تگ (عنصر) بصورت زیر است:

<!ELEMENT                      >نام محتوای تگ نام تگ

نوع محتوای تگ میتواند متنی، عددی ویا در صورتیکه تگ برای معرفی استفاده می شود کلمه کلیدی EMPTY باشد.



مثال : تگ P در زبان XHTML رابا کمک روش DTD تعریف کنید.

عنصری که محتویات آن به صورت متنی است:

```
<!ELEMENT name (#PCDATA)>
```

لذا تعریف تگ P که محتوای آن یک پاراگراف متنی است بصورت زیر است:

```
<!ELEMENT P(=PCDATA)>
```

تعریف پارامترهای اختیاری بصورت زیر است:

```
<!ELEMENT student (birth _day | age)>
```

که به صورت زیر استفاده می شود:

```
<student(1370/1/26)> یا <student (21)>
```

مثال : تگ P در زبان XHTML رابا کمک روش DTD تعریف کنید.

عنصری که محتویات آن به صورت متنی است:

`<!ELEMENT name (#PCDATA)>`

لذا تعریف تگ P که محتوای آن یک پاراگراف متنی است بصورت زیر است:

`<!ELEMENT P(=PCDATA)>`

تعریف پارامترهای اختیاری بصورت زیر است:

`<!ELEMENT student (birth _day | age)>`

که به صورت زیر استفاده می شود:

`<student(1370/1/26)>` یا `<student (21)>`

# روش XML schema



یک شما، همانند یک DTD مشخص می کند که اسناد XML چگونه می توانند ظاهر شوند، بدین معنی که شامل چه عناصری هستند (و به چه ترتیبی) و چه محتوایی دارند و نیز دارای چه ویژگی هایی ممکن است باشند.

# نقطه ضعف های DTD



ترکیب کلی و ساختار DTD همانند اسناد XML نیست (یا شباهت کمی دارند).

تمام تعریف ها در DTD به صورت عمومی هستند.

در نهایت مهمترین ضعف این است که DTD ها کنترلی بر نوع داده های داخل عناصر و ویژگی ها ندارند.



روش شمای XML سعی می کند تا نقاط ضعف فوق را برطرف نماید.

دریک سند شمای XML محتوای سند به دو نوع تقسیم میشود: ساده و پیچیده.

تگ هایی که فقط شامل متن هستند از نوع ساده و تگ هایی که شامل عناصر دیگر ویک یا چند ویژگی باشند از نوع پیچیده می باشند.



# ایجاد یک سند با محتوای ساده



یک شما، یک سند XML، با قالب text-only و با پسوند xsd است. برای ایجاد یک سند XML schema در محیط یک ویرایشگر مانند Notepad یک فایل با پسوند xsd باز کنید و مراحل زیر را انجام دهید:

1. ابتدا تگ `<?xml version="1.0">` را تایپ کنید.
2. سپس عبارت `<xsd: schema` را تایپ کنید.
3. برای موضوعات مختلف سایت هایی با دامنه های مختلف تعریف شده است.



4. آدرس این شما در متغیر بنام xsi ذخیره می شود:

```
xmlns:xsi="http://www.w3.org/2000/10/xml  
schema-instance"
```

باتایپ ساده > ، تگ ابتدایی شما را کامل کنید.

5. محتویات سند را بنویسید.

6. در انتها عبارت </xsd: schema> را تایپ کنید.

# تعریف تگ ساده




یک تگ ساده تنها می تواند شامل متن باشد و مجاز به داشتن تگ های دیگر و یا حتی ویژگی هم نیست. با این حال واژه متن یک اطلاق کلی است و می توان بجای محدود نمودن محتوای تگ ها به متن، آنها را به نوع خاصی از متن محدود نمود. بطور کلی یک تگ ساده بصورت زیر تعریف می شود:

```
<xsd:element name="lable" type = "xsd:data  
type" />
```

آدرس متغیر شمای sdt عبارت است از :

Xsd=<http://www.w3.org/2000/10/XMLSchema>

Lable نام عنصری است که قرار است تعریف شود و  `xsd:data_type` نوع داده تگ که در آدرس شمای فوق تعریف شده است. در زیر تعدادی از انواع داده آورده شده است:



<code>Xsd:string</code>	نوع داده رشته ای
<code>Xsd:decimal</code>	نوع عددی دهدهی داده
<code>Xsd:integer</code>	نوع عددی صحیح
<code>Xsd:double</code>	نوع عددی دقت مضاعف
<code>Xsd:Boolean</code>	نوع داده بولی
<code>Xsd:date</code>	نوع داده تاریخ
<code>Xsd:time</code>	نوع داده زمان
<code>Xsd:uri-refrence</code>	عنصر شامل یک URL باشد
<code>Xsd:language</code>	عنصر شامل معادل دو حرفی نشان دهنده زبان باشد



تعدادی از تگ های موجود در متغیر شمای xsd عبارت است از:

✓ **تعیین مقادیر قابل قبول:** برای تگ های ساده مانند جنسیت با

مقادیر قابل قبول مذکر یا مونث از تگ زیر استفاده می شود:

```
<xsd:enumeration value="male" /> ,  
<xsd:enumeration value="female" />
```

✓ **تعیین یک الگو:** برای تعریف یک الگو از تگ زیر استفاده می شود:


```
<xsd:pattern value="regex" />
```





✓ **تعیین طول دقیق یک تگ:** این یک تگ کنترلی است و برای مشخص کردن دقیق تعداد کاراکترهایی که یک تگ باید داشته باشد بکار می رود:

`<xsd:length value="x" />`

X  تعداد کاراکترهایی است که تگ باید داشته باشد.

☞ regex یک عبارت معمولی است که محتوای مورد نظر باید مشابه آن باشد و از اجراء زیر تشکیل شده است.



برای هر فاصله خالی	\s	برای هر کاراکتر	.
برای هر غیر رقم	\d	برای هر رقم	\d
برای هیچ یا چندین xy	(xy)*	برای هیچ یا چندین x	X*
برای هیچ یا یک xy	(xy)?	برای هیچ یا یک x	X?
برای یک یا چندین xy	(xy)+	برای یک یا چندین x	X+
برای داشتن یک زشته از مقادیر ۰ تا ۹	[0-9]	برای داشتن یکی از مقادیر گروه a, b یا c	[abc]
برای داشتن دقیقاً دو xyz	(xyz){2}	برای داشتن that یا this	This   that
برای داشتن حداقل پنج و حداکثر هشت x	X{5,8}	برای داشتن دقیقاً 5x	X{5}
		برای داشتن حداقل 5x	X{5,}



مثال

```
<xsd:pattern value="\d{5}-\d{4}" />
```

این الگو تعیین می کند که عنصر باید شامل پنج رقم و علامت - و سپس چهار رقم باشد.



✓ تعیین بیشترین مقدار ممکن: برای محتوای یک تگ می تواند حداکثر مقدار  $n$  را مشخص کرد. برای مثال برای نمره دانشجو می توان حداکثر مقدار ۲۰ را مشخص کرد. بدین ترتیب با تگ `<xsd:maxInclusive value="n" />` نمره ورودی دانشجو می تواند بین ۱ تا ۲۰ ( $n <= 20$ ) و با تگ `<xsd:maxExclusive value="n" />` می تواند بین ۱ تا ۱۹ ( $n < 20$ ) باشد.



✓ تعیین کمترین مقدار ممکن: برای کمترین مقدار مانند نمره قبولی ۱۰ در یک درس تعریف می شود. با تگ `<xsd:minInclusive value="n" />` نمره دانشجو باید بزرگتر یا مساوی ۱۰ و با تگ `<xsd:minInclusive value="n" />` مشخص می شود که نمره باید بزرگتر از ۱۰ باشد.





✓ تعیین حداکثر طول یک تگ: مشابه تگ فوق است اما برای مشخص کردن حداکثر تعداد کاراکترهایی که یک تگ باید داشته باشد بکار می رود:

`Xsd:maxlength value="x" />`

X ✎ حداکثر کاراکترهایی که تگ عنصر باید داشته باشد.



✓ **تعیین حداقل طول یک تگ:** مشابه تگ های فوق اما برای مشخص کردن حداقل تعداد کاراکترهایی که یک تگ باید داشته باشد بکار می رود:

```
<xsd:minlength value="x" />
```

X حداقل کاراکترهایی است که تگ باید داشته باشد. 



• **تگ با محتوای مختلط** : بدین معنی که شامل هر ترکیبی از تگ ها، ویژگی ها و یا متن می توانند باشند. برای مثال تگ `td` در `html` را می توان از این دسته در نظر گرفت.

برای ایجاد یک محتوای پیچیده ابتدا از تگ زیر استفاده می شود:

```
<xsd:complexType name="lable">
```

```
</xsd:complexType>
```

`Lable` نام محتوای پیچیده را مشخص می کند.

# ایجاد یک محتوای پیچیده فقط متن



```
<xsd:complexType name="lable">  
  <xsd:simpleContent>  
    (محل قرار گرفتن متن)  
  </xsd:simpleContent>  
</xsd:complexType>
```



✓ تعیین تعداد کل ارقام یک عدد: با تگ زیر حداکثر تعداد ارقام  
یک عدد مشخص می شود:

`<xsd:precision value="n" />`

`n` حداکثر تعداد ارقامی است که یک عدد می تواند داشته باشد. 





✓ تعیین محتوای یک تگ: برای تگی با نام `lable` که ویژگی `fixed` را دارد مشخص می کند که محتوای تگ چه باید باشد.


```
<xsd:element name="lable" type="data_type="fixed="value" />
```

Value مقدار است که به ویژگی `fixed` نسبت داده می شود.



✓ تعیین مقدار اولیه برای یک تگ: برای تگی با نام lable که ویژگی default را دارد مشخص می کند که محتوای اولیه تگ چه باید باشد.

```
<xsd:element name="lable " type =" data  
_type" default="value" />
```

Value مقدار پیش فرض محتوای تگ است. 

# ایجاد محتوای پیچیده با تگ مختلط



```
<xsd:complextypе name="lable" mixed="true"  
...  
</xsd:complextypе>
```

# ایجاد یک سند با محتوای پیچیده



تگی که خود می تواند شامل تگ دیگر ویا یک یا چند ویژگی باشد، یک تگ پیچیده تلقی می شود. در محتوای پیچیده تگ ها به چهار دسته تقسیم می شوند:

- **پیچیده** : تگی که شامل تگ های دیگر یا ویژگی ها باشد. برای مثال تگ `table` در `html` را می توان از این دسته در نظر گرفت.
- **تگ خالی** : بدین معنی که دارای تگ دیگر یا متن نیست، البته میتواند دارای ویژگی باشد. برای مثال تگ `hr` در `html` را میتوان از این دسته در نظر گرفت.

# زبان تبدیل XML : XSLT



پیشنهاد کامل، که توسط انجمن W3C برای انتقال و قالببندی اسناد XML داده شده، دربرگیرنده مشخصه ای اصلی به نام XSL می باشد. انجمن W3C، XSL را به دو بخش تقسیم نموده است: XSLT و XSL-FO

برای انجام تغییر حالت واقعی، به یک پردازنده XSLT نیاز دارید، که در شبکه اینترنت می توان به تعدادی از آن دست یافت. اولین کاری که توسط پردازنده XSLT صورت می گیرد، تجزیه و تحلیل سند XML و تبدیل آنها به نمودار شاخه ای است که نشان دهنده ی نمودار سلسله مراتبی کل سند XML می باشد. به یاد داشته باشید که گره چیزی جز یک قسمت از سند XML نمی باشد.



# ایجاد یک محتوای پیچیده برای تگ خالی



```
<xsd:complextypename="lable">  
  <xsd:complexcontent>  
    <xsd:extentionbase="anytype"/>  
    ...  
  </xsd:complexcontent>  
</xsd:complextypename>
```

# چاپ کد HTML



اساساً برگه XSLT از دو قسمت تشکیل شده است: دستورالعمل ها و حروف. توسط دستورالعمل های XSLT چگونگی تغییر حالت سند مبدأ XML انجام می شود و اجراء حرفی خروجی `style sheet` خواهد بود.

به منظور چاپ کدهای HTML، درون تگ الگو بین

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

کدهای HTML مورد نظر خود را بنویسید.

# تعریف یک تگ ساده در تگ مختلط



```
<xsd:element type="lable" />
```

Lable همان کلمه ای است که برای مشخص کردن مدل پیچیده  
بکار برده اید.

# تعریف یک ویژگی



```
<xsd:attribute name="lable" type="simple"  
ref="lable">
```

...

```
</xsd:attribute>
```

Lable مربوط به name نامی است که برای ویژگی درسند XML به کار می برید.

Simple همان مدل ساده است که ویژگی به آن مرتبط است.

Lable مربوط به ref ویژگی است که قبلا تعریف نموده اید.



هنگامی که در سند مبدأ XML پردازنده به گره ها برخورد کند، به دنبال دستور العملی برای انجام عملیات لازم با آنها می شود. این دستور العمل ها در الگوها قرار گرفته اند. هر الگو خود شامل دو قسمت می شود که عبارتند از: ۱- یک قسمت از برچسب هایی که در قسمتی از سند XML وجود داشته و می توان آنها را در الگوها بکار برد. ۲- دستور العمل هایی که بایستی برای تغییر حالت واقعی انجام شوند.

یکی از دستور العمل های XSLT، مجموعه ای از گره ها که می توانند با مناسبترین الگو قابل دسترسی ارتباط داشته باشند، مشخص می شود.





به منظور اراده محتوای گره، عبارت زیر را تایپ کنید:

```
<XSL:value-of select="expression" />
```

Expression مجموعه گره مورد نیاز از سند مبدا XML که

خروجی نشان داده شد از آن می باشد را مشخص می کند.

# ایجاد قواعد الگو



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

...

`</xsl:template>`

# پردازش دسته ای شاخه ها



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

...

```
</xsl:for-each>
```

مجموعه شاخه هایی که پردازش می شود را **Expression** مشخص می کند.

# پردازش شرطی گره ها



```
<xsl:if test="expression">
```

...

```
</xsl:if>
```

# تفاوت استفاده از CSS در HTML , XML



طبیعت برگه های CSS قالبندی نمودن اطلاعات موجود در لایه ها می باشد، به طوری که قوانین هر لایه بر قوانین و حالات موجود در لایه پیشین برتری دارد. یکی از تفاوت های اصلی در هنگام استفاده از CSS در سند XML با سند HTML این است که در بعضی از تگ های HTML، قالب بندی اطلاعات به خودی خود وجود دارد، در صورتی که در XML اینگونه نمی باشد.



# انتخاب های شرطی

<xsl:choose>

  <xsl:when test ="expression">

  ...

</xsl:when>



  <xsl:when test="expression">

  ...

</xsl:when>

.

.

<xsl:otherwise>

  ...

</xsl:otherwise>

</xsl:choose>

# ایجاد ویژگی ها



```
<xsl:attribute name="att-name">
```

...

```
</xsl:attribute>
```

# تنظیم CSS



در HTML، هر تگ دارای قالب پیش فرض درونی مخصوص به خود می باشد، که در صورت نیاز به تغییر آن قالب، می توان با استفاده از تگ های وبکارگیری `style sheet` ها قالب مورد نظر را تهیه نمود.

دو روش برای قالب تگ ها در سند XML وجود دارد: یکی توسط XSLT و دیگری CSS از آنجا که XSLT می تواند سند XML را به سند HTML تبدیل کند، می توان از آن در قالب `style sheet` استفاده نمود.



با این وجود، ذکر این نکته لازم است که XSLT خود به تنهایی دارای هیچگونه دستورالعملی قالبندی نمی باشد.

به عبارت دیگر CSS، به طور اساسی برای قالب بندی بوجود آمده، مثلاً برای تعیین اندازه و نوع فونت و یا رنگ پیش زمینه مورد استفاده قرار می گیرد.



نکته دیگر توجه به مشخصات تصاویر در هنگام نوشتن دستورهای CSS که مستقیماً در سند XML استفاده می شود، همچنین در هنگام نوشتن HTML توجه به مسئله برتری تگ ها و دستورها در قسمت قالب بندی صفحه لازم و ضروری می باشد.



# فراخوانی style sheet برای سند XML

برای ایجاد دستورالعمل اتوماتیک پردازش

```
<?xml-stylesheet type="text/css" href="style.css"?>
```

برای ایجاد دستورالعمل پردازش با XSLT

```
<xsl:processing-instruction name="xml-stylesheet">
```

```
  Type="text/css"
```

```
  Herf="style.css"
```

```
</xsl:processing-instruction>
```



با CSS می توان محل و نحوه نمایش تگ های ایجاد شده توسط XML را مشخص کرد.

نحوه نوشتن انتخاب کننده به منظور تعیین محل قرارگیری style در سند XML

نحوه نوشتن انتخاب کننده در XML روش های متعددی دارد که در اسلاید بعد چند روش نشان داده شده است.



*	برای قراردادن <code>style</code> در هر جزء
<b>name</b>	برای قرار دادن <code>style</code> در نام هر جزء موجود
<b>Name1,name2</b>	برای استفاده از <code>style</code> در هر یک از اجزاء نامگذاری شده
<b>Name#id</b>	برای استفاده از <code>style</code> در اجزایی که ویژگی <code>ID</code> آن مقدار واحدی به نام <code>id</code> دارند.
<b>Name[att-name=att-value]</b>	برای استفاده از <code>style</code> در اجزایی که ویژگی <code>att-name</code> آن برابر با <code>att-value</code> می باشد.

# xlink و Xpointer



معروفیت HTML اساسا برپایه دو مورد می باشد: اول فوق پیوندها و دوم تصاویر. زیرا به وسیله پیوندها می توان به اطلاعات موجود در صفحات دیگر در شبکه اینترنت دسترسی پیدا نموده و به وسیله تصاویر می توان، جذابیت صفحه ها را افزایش داد.

در برنامه xml، دوزبان برای بیان پیوندها و تصاویر وجود دارد: xlink و xpointer

# ایجاد یک پیوند ساده



```
<root  
xmlns:xlink="http" // www.w3.org/1999/xlink  
>
```

```
<source xlink:type="simple"
```

```
xlink:href="URL" xlink:title="description"
```

```
</root>
```

URL، محل فایلی است که پیوند به آن جا ارجاع داده می شود.

Description، برای توضیح مقصد پیوند به کاربران است.





Value، یا میتواند کلمه `replace` باشد، برای جایگزینی پیوند استاندارد HTML با کل سند موجود در مرورگر، یا می تواند کلمه `embedded` باشد، برای جایگزین نمودن تصویر استاندارد HTML با کل سند موجود در مرورگر.



پایان