

الگوی XML



یک الگو نیز مانند DTD درباره اجزای اسناد XML صحبت می‌کند. از جمله: عناصر و ترتیب آنها، محتویات و ویژگیها (در حقیقت DTD ها نیز نوعی الگو محسوب می‌گردند. در این فصل درباره الگوهایی صحبت می‌شود که در زبان XML Schema نوشته شده‌اند).

DTD ها نسبت به الگوهایی که توسط XML Schema نوشته می‌شوند معایبی دارند. نخست آنکه قوانین نگارش DTD ها محدودیتهایی دارند و توسط تجزیه‌گر XML تجزیه نمی‌شوند. دوم اینکه تمام اعلانها در یک DTD به صورت سراسری می‌باشند؛ بنابراین نمی‌توانید دو عنصر گوناگون با یک نام داشته باشید حتی اگر در قسمتهای جداگانه باشند. نکته آخر و مهمتر اینکه DTD ها نمی‌توانند نوع اطلاعات یا ویژگی یک عنصر را کنترل نمایند.

زبان XML Schema توسط W3C و با هدف برطرف کردن مشکلات موجود ایجاد شده است. XML Schema در خود XML نوشته می‌شود و امکان تعریف عناصر سراسری و محلی را فراهم می‌کند. عناصر سراسری در تمام سند و عناصر محلی در بخشی از سند XML قابل استفاده هستند. همچنین XML Schema از یک سیستم انواع داده‌ها تشکیل شده که امکان تعیین نوع محتویات عناصر را فراهم می‌کند. داده‌ها می‌توانند عدد صحیح یا رشته‌ای باشند. خلاصه اینکه XML Schema امکان کنترل محتویات سند را افزایش می‌دهد.

نکته مهم!

به طور حتم قوانین نگارش توسط W3C تغییر خواهند کرد. اگر تغییر خاصی ایجاد شود در سایت مؤلف (صفحه ۱۸) اعمال خواهد شد. همچنین آخرین تغییرات مربوط به XML Schema از طریق آدرس زیر در دسترس قرار می‌گیرند: <http://www.w3.org/xml/schema>

نوعهای ساده و پیچیده

```
code.xsd
<xsd:element name="weight" type="xsd:string"/>
<xsd:element name="population"
type="xsd:integer"/>
```

شکل ۱-۵. این دو عنصر به کمک نوعهای ساده درونی تعریف شده‌اند. Weight از نوع رشته‌ای و population از نوع عدد صحیح است.

```
code.xsd
<xsd:simpleType name="zipcodeType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\d{5}-\d{4}?"/>
  </xsd:restriction>
</xsd:simpleType>
```

شکل ۲-۵. این نوع ساده که توسط کاربر تعریف شده محتویات عنصر تعریف شده با zipcodeType را به یک رشته محدود می‌نماید. این رشته از پنج رقم به همراه یک خط تیره اختیاری و چهار رقم دیگر تشکیل شده‌است.

```
code.xsd
<xsd:complexType name="endType">
  <xsd:sequence>
    <xsd:element name="animal"
type="animalType" minOccurs="1"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

شکل ۳-۵. نوع پیچیده endType شامل یک عنصر دیگر به نام animal است که نوع پیچیده animalType را تعریف می‌کند. از نوع پیچیده endType برای تعریف عنصر دیگری که شامل عنصر animal باشد می‌توانید استفاده نمایید.

در یک الگو، محتویات یک سند می‌توانند از دو نوع تشکیل شوند؛ یکی نوع ساده و دیگری نوع پیچیده. عناصری که محتویات آنها فقط به صورت متن باشند نوع ساده و عناصری که از عناصر و ویژگیهای دیگر تشکیل شوند نوعهای پیچیده نامیده می‌شوند (ویژگیهایی که فقط محتویات متنی داشته باشند نوع ساده محسوب می‌گردند).

در یک DTD می‌توانید یک عنصر را که محتویات آن فقط به صورت متنی است با عبارت #PCDATA تعریف کنید. PCDATA می‌تواند یک نام، عدد، تاریخ یا هر چیز دیگری باشد. در XML Schema نوع دقیق متن عنصری که از نوع ساده است تعیین می‌گردد. در این خصوص چند نوع ساده درونی از پیش تعریف شده مانند date (تاریخ)، integer (عدد صحیح) و string (رشته) وجود دارد که بدون هیچ تعریفی قابل استفاده می‌باشند. نوعهای ساده را بیشتر برای کنترل ظاهر محتویات عناصر می‌توان ایجاد کرد. درباره تعریف و استفاده از نوعهای ساده در فصل ۶ صحبت خواهد شد.

عناصر نوع پیچیده به تعریف ساختار یک سند کمک می‌نمایند و با محتویات کاری ندارند. چهار نوع پیچیده اصلی داریم: عناصری که شامل عناصر دیگر هستند، عناصری که از عناصر و متن تشکیل شده‌اند، عناصری که فقط شامل متن هستند و عناصری که خالی می‌باشند. هر یک از این عناصر می‌توانند ویژگیهایی نیز داشته باشند. بر حسب نیاز می‌توان نوع پیچیده خاصی را برای سند XML تعریف کرد (فصل ۷).

اگر برای نوعهای ساده و پیچیده نامی در نظر بگیرید می‌توانید آنها را در تمام قسمتهای الگو به کار برید؛ ولی اگر برای آنها نامی در نظر نگیرید فقط می‌توانید در عنصر مربوط از آنها استفاده کنید.

اعلانهای محلی و سراسری

در DTD تمام عناصر به صورت سراسری اعلان می‌گردند؛ یعنی هر عنصر نام منحصر به فردی دارد و فقط یک‌بار تعریف می‌شود. چند عنصر می‌توانند به یک عنصر نسبت داده شوند؛ بنابراین هر عنصر می‌تواند در چند جای یک سند XML ظاهر گردد و هر جا که باشد تعریف آن همیشه یکسان خواهد بود.

نگارش XML Schema بسیار مهم است. اجزای الگو، به انضمام عناصر، ویژگیها و نام نوعهای ساده و پیچیده (تحت عناوین گروهها و گروههای ویژگی که آنها را مورد بررسی قرار خواهیم داد)، باید در ابتدای یک الگو (یعنی پایین عنصر `xsd:schema`) اعلان گردند. در نتیجه به صورت سراسری اعلان می‌شوند و برای استفاده در بقیه قسمت‌های الگو قابل دسترسی خواهند بود. توجه داشته باشید که هرگاه عنصری به صورت سراسری اعلان شود محل ظهور آن در سند XML مهم نخواهد بود؛ بلکه ظاهر عنصر تعیین می‌گردد. شما باید یک اعلان عنصر سراسری را به منظور نمایش در یک سند XML به طور صریح نسبت‌دهی کنید.

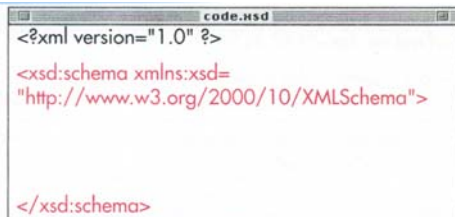
تنها استثنا در این قانون برای عنصر ریشه است که هر جا که اعلان شود به صورت خودکار نسبت‌دهی می‌گردد.

هنگامی که یک نوع پیچیده را تعریف می‌کنید می‌توانید عناصر سراسری موجود را نسبت‌دهی کنید و عناصر جدیدی را اعلان و تعریف نمایید. این عناصر محلی جدید به تعریف نوع پیچیده‌ای که اعلان شده‌اند و شاید در الگو مورد استفاده قرار نگرفته باشند محدود می‌باشند. همچنین باید نام منحصر به فردی در سراسر سندی که در آن ظاهر می‌شوند داشته باشند. چنین عناصر محلی به صورت خودکار به محلی که تعریف می‌شوند نسبت داده می‌شوند و محلی را که باید عنصر در سند ظاهر گردد تعیین می‌نمایند.

```
code.hsd
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
  <xsd:element name="endangered_species"
    type="endType"/>
  <xsd:element name="name" type="xsd:string"/>
  <xsd:complexType name="endType">
    <xsd:sequence>
      <xsd:element name="animal">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="name"
              minOccurs="2"/>
            <xsd:element name="source"
              type="sourceType"/>
          ...
        </xsd:complexType>
      </xsd:complexType>
    <xsd:complexType name="habitatType">
      <xsd:sequence>
        <xsd:element name="river">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element ref="name"
                minOccurs="1"
                maxOccurs="unbounded"/>
              <xsd:element name="source"
                type="xsd:string"/>
            ...
          </xsd:complexType>
        ...
      </xsd:complexType>
    ...
  </xsd:schema>
```

شکل ۴-۵. در این مثال چهار جزء به صورت سراسری تعریف شده‌اند؛ زیرا پس از عنصر `xsd:schema` قرار گرفته‌اند. عنصر ریشه (`endangered_species`) به صورت خودکار نسبت‌دهی شده اما همان‌گونه که می‌بینید عنصر `name` (که عنصر ریشه نیست) به شکل دستی نسبت‌دهی شده است.

به اجزای‌های لایت شده توجه کنید. دو عنصر محلی با یک نام و دو تعریف متفاوت اعلان شده‌اند. از طرفی اعلانهای عنصرهای سراسری باید نامهای منحصر به فرد داشته باشند.



```
code.xsd
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd=
"http://www.w3.org/2000/10/XMLSchema">
</xsd:schema>
```

شکل ۵-۵ . در عنصر ریشه الگو، برای الگوی Schema فضای نام اعلان شده است.

آغاز یک الگوی ساده

یک الگو یک سند XML با فرمت متنی و پسوند .xsd است. این الگو با یک اعلان استاندارد XML به همراه اعلان فضای نام XML Schema آغاز می‌گردد.

برای آغاز یک الگو:

- ۱- در صورت تمایل در ابتدای سند الگو عبارت `<?xml version="1.0" ?>` را تایپ کنید (صفحه ۲۴).
- ۲- عبارت `<xsd:schema` را تایپ نمایید.
- ۳- برای اعلان فضای نام الگوی schema عبارت `xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"` را تایپ کنید. از این پس هر عنصر یا نوعی که پیشوند: xsd داشته باشد به این فضای نام نسبت داده می‌شود.
- ۴- برای تکمیل عنصر الگو علامت `>` را تایپ نمایید.
- ۵- چند خط خالی برای ایجاد قوانین الگو ایجاد کنید. در فصلهای ۶ و ۷ با این قوانین آشنا خواهید شد.
- ۶- عبارت `</xsd:schema>` را به منظور کامل شدن سند الگو تایپ نمایید.
- ۷- الگو را با فرمت text only (فقط متنی) با پسوند .xsd ذخیره سازید.

نکته

◀ برای حفظ سادگی درس و یادگیری روش ایجاد الگو در این قسمت وارد جزئیات بیشتر نشده‌ایم. در صفحه‌های ۱۲۸-۱۲۶ با اعلان فضاهای نام اضافی و اعلان XML Schema به عنوان فضای نام پیش‌فرض آشنا خواهید شد.

تعیین محل یک الگوی ساده

در برخی از معتبرسازهایی که برای مقایسه سند XML با الگو به کار می‌برید باید محل الگوی متناظر را در سند XML مشخص نمایید.

برای اعلان و تعیین محل یک الگو:

۱- در قسمت عنصر ریشه سند XML عبارت

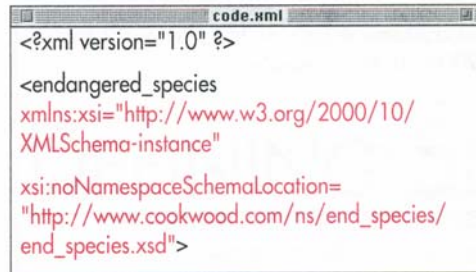
`xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"` را تایپ کنید. در نتیجه عناصر، در محل تعیین شده الگو در دسترس خواهند بود. به این کار، اعلان فضای نام می‌گویند که در صفحه ۱۱۶ به شرح کامل آن خواهیم پرداخت.

۲- عبارت `xsi:noNamespaceSchemaLocation=` را تایپ کنید.

۳- در نهایت عبارت "file.xsd" را تایپ نمایید. file.xsd آدرس URL فایل الگویی است که در مرحله ۷ صفحه ۷۲ ایجاد کرده‌اید و می‌خواهید برای معتبرسازی این فایل XML مورد استفاده قرار دهید.

نکته‌ها

- ◀ برای تعیین اعتبار اسناد با یک الگو، برنامه‌های گوناگونی وجود دارند. از جمله برنامه XML SPY که یک ویرایشگر XML است و در آدرس <http://www.xmlspy.com> قابل دسترس می‌باشد. به صفحه ۲۴۵ نگاه کنید.
- ◀ هرگاه در الگو هیچ فضای نام مقصدی اعلان نشود ویژگی `xsi:noNamespaceSchemaLocation` در نظر گرفته می‌شود. درباره این نوع فضاهای نام در فصل ۹ صحبت خواهیم کرد.



```
code.xml
<?xml version="1.0" ?>
<endangered_species
xmlns:xsi="http://www.w3.org/2000/10/
XMLSchema-instance"
xsi:noNamespaceSchemaLocation=
"http://www.cookwood.com/ns/end_species/
end_species.xsd">
```

شکل ۵-۶ می‌توانید به فایل الگویی که سند شما را با استفاده از ویژگی `xsi:schemaLocation` در عنصر ریشه سند XML تعریف می‌نماید اشاره کنید.

تفسیر الگوها

```
code.hsd
<xsd:schema xmlns:xsd="http://www.w3.org/
2000/10/XMLSchema">
  <xsd:annotation>
  <xsd:documentation>
This schema will be used to validate the set of XML
documents for the Endangered Species
project.</xsd:documentation>
</xsd:annotation>
```

شکل ۷-۵. یک یادداشت تفسیری به درک بهتر الگو کمک می‌کند. در نتیجه به روزرسانی آن در آینده نیز آسان‌تر خواهد شد.

می‌توانید درباره الگوها و عناصر آنها اطلاعات بیشتری اضافه کنید تا استفاده از الگوها در آینده برای شما و دیگران آسان‌تر گردد.

برای تفسیر الگوها:

- ۱- عبارت `<xsd:annotation>` را تایپ کنید.
- ۲- عبارت `<xsd:documentation>` را برای آغاز ایجاد یک یادداشت متنی که توسط انسانها (نه ماشین) خوانده خواهد شد تایپ کنید.
- ۳- یادداشت مورد نظر خود را تایپ کنید.
- ۴- با تایپ `</xsd:documentation>` یادداشت خود را تکمیل نمایید.
- ۵- با تایپ `</xsd:annotation>` اعلان تفسیر را به پایان برسانید.

نکته

یادداشتها را می‌توان بلافاصله پس از عنصر سراسری `xsd:schema` یا اعلانهای جداگانه عناصر یا هردوی آنها ایجاد نمود.