



تعریف نوعهای ساده

یک عنصر از نوع ساده فقط شامل متن است. این عنصر نمی‌تواند از عناصر دیگر و ویژگیها تشکیل گردد. به جای اینکه محتویات یک عنصر را به متن محدود کنید می‌توانید متن خاصی را برای آن در نظر بگیرید. چنین محدودیتهایی را می‌توانید با استفاده از تعریف نوع ساده درونی از پیش تعریف شده یا نوعهایی که خود شما تعریف کرده‌اید ایجاد نمایید.

XML Schema برای بیشتر متنهای رایج مجموعه‌ای از نوعهای ساده درونی و از پیش تعریف شده دارد که شامل مقادیر رشته‌ای، بولین، URLها، فرمتهای گوناگون تاریخ و انواع سیستمهای عددنویسی است. می‌توان با اعمال محدودیتها و شرطهایی برای یک نوع ساده درونی، نوع ساده دلخواهی ایجاد کرد. برای مثال شاید بخواهید یک عنصر به صورت رشته‌ای و فرمت خاص (مثل شماره تلفن یا کد کالا) یا عنصری که فقط شامل مجموعه‌ای از فرمتهای معین تاریخ است داشته باشید.

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

شکل ۱-۶. هنگام اعلان یک عنصر، می‌توانید نام و نوع محتویات آن را انتخاب نمایید.

```
code.xml
<weight>500 pounds</weight>
<population>28</population>
```

شکل ۲-۶. منظور از string یک سری حرف، عدد یا نماد است. چون integer شامل رشته‌های عددی می‌گردد عناصر weight و population هنگام مقایسه با اعلان شکل ۱-۶ معتبر می‌باشند.

```
code.xml
<weight>500</weight>
<population>not very many</population>
```

شکل ۳-۶. در این مثال عنصر weight که در نگاه اول نادرست به نظر می‌رسد، معتبر است. زیرا به صورت رشته‌ای از اعداد ظاهر گشته است. ولی عنصر population معتبر نیست زیرا عبارت "not very many" عدد صحیح یا integer نیست.

اعلان یک عنصر با یک نوع ساده

نوعهای ساده درونی و از پیش تعریف شده گوناگونی وجود دارند. شما نیز می‌توانید یک نوع ساده بر اساس نوعهای درونی بسازید.

برای اعلان یک نوع ساده :

- ۱- عبارت `xsd:element` را برای آغاز اعلان تایپ کنید.
- ۲- `name="label"` را تایپ نمایید. منظور از label نام عنصری است که قصد اعلان آن را دارید.
- ۳- عبارت `type="xsd:string"` را تایپ کنید. سپس اگر عنصر مورد نظر شامل رشته‌ای از کاراکترها است عبارت `xsd:string` را تایپ کنید. چنانچه عنصر از اعداد اعشاری تشکیل می‌گردد عبارت `xsd:decimal` را تایپ نمایید. برای بقیه انواع عددی به صفحه ۷۸ مراجعه کنید. اگر عنصر شامل مقادیر درست یا نادرست است (۰ یا ۱) عبارت `xsd:Boolean` را تایپ کنید. در صورتی که عنصر یک تاریخ است عبارت `xsd:date` را تایپ کرده و برای فرمت‌های دیگر تاریخ به صفحه ۷۸ مراجعه کنید. برای عنصری که زمانی از روز را نشان می‌دهد عبارت `xsd:time` را تایپ نموده و فرمت‌های گوناگون زمان را در صفحه ۷۸ بررسی نمایید. اگر عنصر شامل آدرس URL است، `xsd:uri-reference` را تایپ کنید (uri صحیح است نه url). در مواردی که عناصر از دو حرف اختصاری زبانهای موجود در فهرست ISO639 تشکیل می‌شود `xsd:language` را تایپ کنید. چنانچه قصد ایجاد نوع ساده دلخواهی دارید واژه `custom` را تایپ کنید.
- ۴- برای تکمیل نوع، علامت " را تایپ نمایید.
- ۵- به منظور تکمیل پرچسب، علامتهای `>/` را تایپ کنید.

نکته‌ها

- ◀ نوعهای ساده درونی گوناگونی وجود دارند که از طریق آدرس <http://www.w3.org/TR/xmlschema-2/#built-in-datatypes> در دسترس قرار می‌گیرند.
- ◀ نوعهای ساده دلخواه را می‌توان بر اساس نوعهای ساده درونی ایجاد نمود. به صفحه ۸۱ نگاه کنید.
- ◀ نوعهای ساده درونی اغلب با xsd: شروع می‌شوند (مورد استثنا را در صفحه ۱۲۸ ببابید)؛ در نتیجه از نوعهای ساده‌ای که توسط کاربر ایجاد شده قابل تشخیص می‌باشند (صفحه ۸۱).
- ◀ ویژگیها اغلب به عنوان نوعهای ساده محسوب می‌گردند (زیرا نمی‌توانند شامل محتویات و ویژگیها باشند) و شبیه عناصر نوع ساده اعلان می‌گردند. اما باید فقط در یک عنصر اعلان شوند (صفحه ۱۰۸).
- ◀ اگر عنصری شامل ویژگیها و عناصر دیگر باشد یک نوع پیچیده محسوب می‌گردد (فصل ۷).
- ◀ نام یک عنصر معتبر باید با یک حرف یا علامت _ آغاز گردد و با اعداد، حروف، علامت نقطه، - و _ ادامه یابد (صفحه ۲۶).
- ◀ عناصر نوع ساده می‌توانند به صورت سراسری نیز اعلان گردند (یعنی در ابتدای سند و پایین عنصر xsd:schema) در حالی که می‌توانند توسط یک نوع پیچیده فراخوانی یا نسبت‌دهی شوند. همچنین می‌توانند در تعریف یک نوع پیچیده به شکل محلی اعلان شوند که به صورت خودکار نسبت‌دهی می‌شوند (فصل ۷).

```
code.xsd
<xsd:element name="last_modified"
type="xsd:date"/>
```

شکل ۴-۶. در این مثال عنصر last_modified شامل تاریخ است.

```
code.xml
<last_modified>1999-05-16</last_modified>
```

شکل ۵-۶. برای این‌که این عنصر معتبر باشد باید فرمت تاریخ آن به صورت CCYY-MM-DD باشد. یعنی چهار رقم برای سال، خط تیره، دو رقم برای ماه، خط تیره، دو رقم برای روز. فرمت‌های دیگری نیز برای تاریخ وجود دارند که در صفحه ۷۸ به آنها اشاره شده است.

```
code.xml
<last_modified>May 16, 1999</last_modified>
```

شکل ۶-۶. عنصر last_modified معتبر نیست؛ زیرا فرمت تاریخ مناسبی ندارد.

استفاده از نوعهای تاریخ و زمان

XML Schema برای تاریخ، چند نوع داده درونی دارد. یکی از فرمت‌های مناسب مرحله ۳ صفحه ۷۶ یا فرمت‌هایی که در این قسمت گفته شده رابه‌کاربرید (صفحه ۸۱).

برای استفاده از نوعهای تاریخ و زمان :

۱- اگر می‌خواهید مقدار مشخصی از زمان، مانند ۲ ساعت و ۱۲ دقیقه و ۴۵/۳ نشان را دهید عبارت `xsd:timeDuration` را تایپ کنید.

فرمت `time duration` (طول زمان) باید به صورت `PnYnMnDnTnHnMnS` باشد. P یعنی Period و اجباری است. T یعنی Time section و اختیاری است. n عدد صحیح مثبت و نشانه تعداد سال (year)، ماه (month)، روز (day)، ساعت (hour)، دقیقه (minute) و ثانیه (second) است. با اضافه کردن یک خط تیره اختیاری می‌توان طول زمان را در گذشته نشان داد.

۲- اگر می‌خواهید زمان معینی از روز را که هر روز اتفاق می‌افتد نشان دهید عبارت `xsd:time` را تایپ نمایید. برای مثال ۴ و ۱۵ دقیقه بعدازظهر هر روز. فرمت کار به شکل `hh:mm:ss.sss` است. برای نشان دادن اختلاف وقت محلی با وقت مبدأ جهانی یا UTC از `Z` با فرمت `+hh:mm` یا `-hh:mm` به صورت اختیاری استفاده می‌شود.

۳- اگر می‌خواهید زمان خاصی را در تاریخ معینی مشخص نمایید (مانند ۴ و ۱۵ دقیقه بعدازظهر روز ششم ماه مه سال ۱۹۳۵) عبارت `xsd:timeInstant` را انتخاب کنید. فرمت کار با مشخصه اختیاری وقت محلی به صورت `CCYY-MM-DDThh:mm:ss.sss` است.

۴- برای تعیین روز مشخص مثل ششم ماه مه سال ۱۹۳۵ عبارت `xsd:date` با فرمت `CCYY-MM-DD` به کار می‌رود.

`xsd:month` برای تعیین ماه مشخص (مانند ماه مه سال ۱۹۳۵) با فرمت `CCYY-MM` به کار می‌رود.

۶- برای تعیین سال معین (مثل سال ۱۹۳۵) از `xsd:year` با فرمت `CCYY` استفاده می‌شود.

```
code.xsd
<xsd:element name="gestation"
type="xsd:timeDuration"/>
```

```
code.xml
<gestation>P3M15D</gestation>
```

شکل ۷-۶ . دوران بارداری ببر حدود سه ماه ونیم است. توجه کنید که نیازی به تعیین تمام واحدها نیست. فقط همیشه به مقدار اولیه P نیاز داریم. اگر محتویات period شامل داده زمانی باشد باید T را هم اضافه کنید.

```
code.xsd
<xsd:element name="bedtime" type="xsd:time"/>
```

```
code.xml
<bedtime>20:15-05-05:00</bedtime>
```

شکل ۸-۶ . به فرمت وقت جهانی توجه کنید. محتویات عنصر `bedtime` ساعت ۸ و ۱۵ دقیقه بعدازظهر را به صورت EST نشان می‌دهد.

```
code.xsd
<xsd:element name="birth"
type="xsd:timeInstant"/>
```

```
code.xml
<birth>1999-03-14T18:27:46.2398Z</birth>
```

شکل ۹-۶ . محتویات این عنصر تولد یک ببر را در ساعت ۶ و ۲۷ دقیقه و ۴۶/۲۳۹۸ ثانیه بعدازظهر روز ۱۴ ماه مارس سال ۱۹۹۹ به وقت جهانی نشان می‌دهد.

```
code.xsd
<xsd:element name="birthdate"
type="xsd:date"/>
```

```
code.xml
<birthdate>1999-03-14</birthdate>
```

شکل ۱۰-۶ . عنصر نوع `xsd:date` شبیه `xsd:timeInstant` بدون داده زمانی است.

۷- اگر می‌خواهید قرن خاصی را مشخص نمایید (مانند سده ۱۹۰۰) باید `xsd:century` را با فرمت `CC` به کار ببرید (قرن بیستم با عدد ۱۹ مشخص می‌گردد).

۸- به منظور نمایش روز خاصی از ماه معینی بدون بیان سال (مانند اول ماه ژانویه) باید از `xsd:recurringDate` با فرمت `MM-DD` استفاده کنید. دو خط تیره اول به جای عدد سال قرار می‌گیرند و خط تیره بعدی برای جدا کردن ماه و روز به کار می‌رود.

۹- برای تعیین روز معینی از ماه (مانند روز ششم هر ماه) با فرمت `xsd:recurringDay` با فرمت `DD` به کار می‌رود. دو خط تیره اول به جای حذف مقادیر سال و ماه و خط تیره سوم برای جدا کردن گذاشته شده است.

نکته

منظور از UTC (Coordinated Universal Time) وقت مبدأ جهانی است که توسط انجمن بین‌المللی وقت (International Time Bureau) تنظیم می‌گردد و همان زمان گرینویچ است. EST یا Eastern Standard Time در ایالات متحده به صورت UTC-5 بیان می‌گردد. PST یا Pacific Standard Time نیز UTC-8 است. UTC+1 مربوط به نقاط غربی اروپا و UTC+3 روسیه، UTC+6 برای ژاپن و UTC+10 برای استرالیا به کار می‌رود. جدول زمانهای جهانی در سایت وب مربوط (صفحه ۱۸) قرار دارد.

```
code.xsd
<xsd:element name="campaign_start"
type="xsd:month"/>
```

```
code.xml
<campaign_start>1999-03</campaign_start>
```

شکل ۱۱-۶. اگر روز مشخصی مد نظر شما نیست می‌توانید در عنصر نوع `xsd:month` فقط سال و ماه را تعیین نمایید.

```
code.xsd
<xsd:element name="last_seen"
type="xsd:year"/>
```

```
code.xml
<last_seen>1950</last_seen>
```

شکل ۱۲-۶. آخرین بزرگسپین در سال ۱۹۹۵ دیده شده است. روز و ماه آن مشخص نیست. در این گونه مواقع فقط به نمایش سال عنصر نیاز داریم.

```
code.xsd
<xsd:element name="greatest_loss"
type="xsd:century"/>
```

```
code.xml
<greatest_loss>19</greatest_loss>
```

شکل ۱۳-۶. توجه داشته باشید که عدد ۱۹ به قرن بیستم اشاره می‌کند. یعنی از سال ۱۹۰۰ تا ۱۹۹۹. در این مدت ۹۶٪ بزرگسپین جهان از بین رفته‌اند.

```
code.xsd
<xsd:element name="birthday"
type="xsd:recurringDate"/>
```

```
code.xml
<birthday>--03-14</birthday>
```

شکل ۱۴-۶. اگر بیان سال اهمیتی نداشته باشد و فقط به ماه و روز نیاز باشد می‌توان از `xsd:recurringDate` استفاده کرد.

```
code.xsd
<xsd:element name="payday"
type="xsd:recurringDay"/>
```

```
code.xml
<payday>---30</payday>
```

شکل ۱۵-۶. برای رویدادی که در روز مشخصی از هر ماه اتفاق می‌افتد از `xsd:recurringDay` استفاده می‌شود.

نوعهای عددی

XML Schema چند نوع عددی داخلی دارد (شماره ۳ صفحه ۷۶) که برای محدود کردن محتویات عناصر و ویژگیها به کار می‌روند. از این نوعها برای ایجاد نوعهای عددی دلخواه نیز می‌توان استفاده نمود (صفحه ۸۱).

برای استفاده از نوعهای عددی:

۱- برای محتویات عددی مثبت و منفی که ارقام اعشاری محدودی دارند (مثل ۴/۲۶، -۱۰۰، ۰ یا ۰) نوع `xsd:decimal` را انتخاب کنید.

۲- از `xsd:integer` برای اعداد مثبت یا منفی بدون اعشار (مانند ۵۴۲ یا -۷) استفاده نمایید.

۳- `xsd:positiveInteger` برای {۱، ۲، ...}، `xsd:negativeInteger` برای {-۱، -۲، ...}، `xsd:nonPositiveInteger` برای {۰، -۱، -۲، ...} و `xsd:nonNegativeInteger` برای {۰، ۱، ۲، ...} به کار می‌رود.

۴- اعداد اعشاری ۳۲ بیتی مثل 43E2، صفر، مثبت و منفی، منفی و مثبت بینهایت (INF، -INF) و فقدان عدد (not a number) یا NaN از نوع `xsd:float` استفاده می‌کنند.

۵- محتویاتی که شامل اعداد اعشاری ۶۴ بیتی هستند `xsd:double` را به کار می‌برند.

نکته

◀ در آدرس <http://www.w3.org/TR/xmlschema-2>

نوعهای عددی دیگری با جزئیات بیشتر قرار دارند.

```
code.xsd
<xsd:element name="population"
type="xsd:nonNegativeInteger"/>
<xsd:element name="density"
type="xsd:decimal"/>
```

شکل ۱۶-۶. برای `population` (جمعیت) به اعداد منفی نیاز نداریم. ولی چون `xsd:positiveInteger` شامل عدد صفر است نمی‌توانیم آن را به کار ببریم. عنصر `density` که به تعداد حیوانات در کیلومتر مربع اشاره دارد با اعداد اعشاری بیان می‌گردد.

```
code.xml
<population>342</population>
<density>4.2</density>
```

شکل ۱۷-۶. چون `population` می‌تواند مقدار صفر یا بیشتر داشته باشد معتبر است. عنصر `density` (پراکندگی) نیز می‌تواند عدد صحیح یا اعشاری باشد.

```
code.xml
<population>112.5</population>
<density>7</population>
```

شکل ۱۸-۶. عنصر `population` معتبر نیست چون عدد صحیح است و نمی‌تواند مقدار اعشار داشته باشد.

نوعهای ساده دلخواه

زبان XML Schema نوعهای ساده درونی از پیش تعریف شده زیادی دارد که می‌توانند بر حسب نیاز شما تغییر کنند.

برای ایجاد نوع ساده دلخواه:

۱- برای آغاز تعریف نوع ساده عبارت `<xsd:simpleType>` را تایپ کنید.

۲- عبارت `name="label"` را تایپ نمایید. label نوع ساده دلخواه شماست و عنصر نیست؛ زیرا یک نوع می‌تواند برای یک یا چند عنصر به کار گرفته شود.

۳- عبارت `<xsd:restriction base="foundation">` را تایپ کنید. منظور از foundation نوع ساده‌ای است که نوع دلخواه بر اساس آن تعریف می‌شود.

۴- برای تعریف نوع دلخواه جدید محدودیتهای لازم را اعمال کنید (صفحه ۸۴ تا ۹۰).

۵- عبارت `<xsd:restriction>` را تایپ کنید.

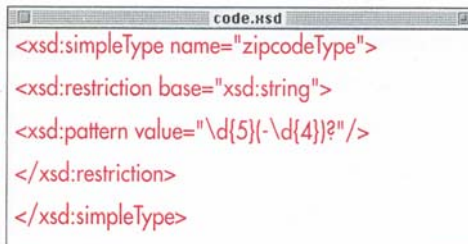
۶- برای تکمیل تعریف نوع ساده دلخواه عبارت `</xsd:simpleType>` را تایپ نمایید.

نکته‌ها

◀ پس از تعریف نام نوع ساده دلخواه می‌توان به صورتی که در پاراگراف پس از مرحله ۳ صفحه ۷۶ گفته شد، از آن استفاده نمود. توجه داشته باشید که به آن به صورت `xsd:label` اشاره نمی‌کنید. بلکه مانند label در مرحله ۲ با آن برخورد می‌شود.

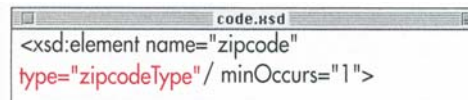
◀ در اعلان یک عنصر منفرد می‌توان نوع ساده بدون نام نیز ایجاد کرد.

◀ فهرستی از نوعهای ساده‌نیز می‌توان ایجاد نمود (صفحه ۹۰).



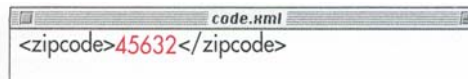
```
code.xsd
<xsd:simpleType name="zipcodeType">
<xsd:restriction base="xsd:string">
<xsd:pattern value="\d{5}(-\d{4})?" />
</xsd:restriction>
</xsd:simpleType>
```

شکل ۱۹-۶. در اینجا یک نوع جدید با نام `zipcodeType` بر اساس نوع `xsd:string` ایجاد شده و الگوی دارد که محتویات عناصر این نوع را به پنج رقم و خط تیره اختیاری و ۴ رقم دیگر محدود می‌نماید.



```
code.xsd
<xsd:element name="zipcode"
type="zipcodeType" / minOccurs="1">
```

شکل ۲۰-۶. از نوع جدید `zipcodeType` در اعلانهای عناصر نیز می‌توانید استفاده نمایید (صفحه ۷۶). ویژگی `minOccurs` در صفحه ۱۰۱ توضیح داده شده است. از این ویژگی برای تعیین تعداد دفعات ظهور یک عنصر استفاده می‌شود.



```
code.xml
<zipcode>45632</zipcode>
```



```
code.xml
<zipcode>42398-0987</zipcode>
```

شکل ۲۱-۶. هر دو عنصر `zipcode` معتبر هستند.



```
code.xml
<zipcode>4398-12349</zipcode>
```



```
code.xml
<zipcode>781001</zipcode>
```

شکل ۲۲-۶. هر دو عنصر `zipcode` غیر معتبر هستند. در مثال اول شامل ۴ رقم با یک خط تیره و پنج رقم است. در مثال دوم نیز الگوی ندارد و مانند کدپستی محل مورد نظر نیست. اغلب کد پستیهای خارج از ایالات متحده به فرمت و الگوی خاصی نیاز ندارند.

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

شکل ۲۳-۶. این اعلان عنصر را با تعریف نوع دلخواه پیچیده و اعلان عنصر شکل ۱۹-۶ و ۲۰-۶ مقایسه کنید. تعریف عنصر zipcode در هر دو مثال به کار رفته است. تفاوت مهم این است که نوع دلخواه این مثال برای عناصر دیگر قابل استفاده نیست. عنصر xsd:simpleType نه نامی دارد و نه چیزی به آن اشاره می‌کند.

استفاده از نوعهای دلخواه بی‌نام

برای تمام نوعهایی که به دلخواه ایجاد می‌کنید نیازی نیست نامی در نظر بگیرید. اگر نوعی را یک بار برای عنصر مشخصی به کار می‌برید می‌توانید تداخل ارجاع بین عنصر و نوع را حذف نمایید.

برای تعریف و استفاده از نوع دلخواه بی‌نام:

- ۱- با تایپ عبارت `<xsd:element name="label">` اعلان عنصر را آغاز نمایید. منظور از label نام عنصری است که قصد اعلان آن را دارید. این نام در سند XML شبیه عبارت `<label>` ظاهر خواهد شد.
- ۲- عبارت `<xsd:simpleType>` را تایپ کنید.
- ۳- `<xsd:restriction base="foundation">` را تایپ کنید. Foundation نوع ساده‌ای است که نوع دلخواه خود را بر اساس آن می‌سازید.
- ۴- برای نوع جدید محدودیتهای لازم را تعیین نمایید (صفحه‌های ۸۴ تا ۹۰).
- ۵- `</xsd:restriction>` را تایپ کنید.
- ۶- برای تکمیل تعریف نوع ساده جدید عبارت `</xsd:simpleType>` را تایپ کنید.
- ۷- به منظور کامل شدن اعلان عنصر نوع ساده بی‌نام عبارت `</xsd:element>` را تایپ کنید.

نکته‌ها

- ◀ کافی است نوعهای درونی را به ویژگی type نسبت دهید.
- ◀ تنها تفاوت بین نوع بی‌نام و نامگذاری شده این است که نوع نامگذاری شده می‌تواند بیش از یک بار مورد استفاده قرار گیرد (با تنظیم ویژگی type و نام). در حالی که نوع بی‌نام تنها برای عنصری که شامل آن باشد قابل استفاده است.

تعیین مجموعه‌ای از مقادیر قابل قبول

برای حفظ پیوستگی و منطق سند XML می‌توان محتویات یک عنصر یا ویژگی را به مجموعه‌ای از مقادیر، محدود نمود.

برای تعیین مجموعه مقادیر قابل قبول :

۱- در قسمت اعلان نوع دلخواه (شماره ۴ صفحه ۸۱)

عبارت `<xsd:enumeration>` را تایپ کنید.

۲- `value="choice"` را تایپ نمایید. مقدار قابل

قبولی برای محتویات عنصر یا ویژگی است.

۳- با تایپ `</xsd:enumeration>` عنصر کامل می‌گردد.

۴- برای هر مقدار اضافه‌ای که عنصر یا ویژگی می‌تواند داشته باشد مراحل ۱ تا ۳ را تکرار کنید.

نکته‌ها

◀ برای همه نوعهای ساده به غیر از بولین می‌توانید

`xsd:enumeration` را به کار برید.

◀ مقادیر تعیین شده باید منحصر به فرد باشند.

◀ مجموعه مقادیر می‌تواند از فضای خالی نیز تشکیل شود.

```
code.xsd
<xsd:element name="continent">
<xsd:simpleType>
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Asia"/>
    <xsd:enumeration value="Africa"/>
    <xsd:enumeration value="Australia"/>
    <xsd:enumeration value="Europe"/>
    <xsd:enumeration value="North America"/>
    <xsd:enumeration value="South America"/>
    <xsd:enumeration value="Antartica"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

شکل ۲۴-۶ . عنصر `continent` می‌تواند شامل هر یک از مقادیر باشد.

```
code.xml
<continent>Asia</continent>
```

شکل ۲۵-۶ . عنصر `continent` معتبر است؛ زیرا با یکی از مقادیر مطابقت دارد.

```
code.xml
<continent>Asia Europe</continent>
```

```
code.xml
<continent>America</continent>
```

شکل ۲۶-۶ . هیچ یک از عناصر `continent` معتبر نیستند. اولی شامل دو مقدار است. در حالی که فقط می‌تواند یک مقدار داشته باشد. دومی شامل بخشی از یک مقدار است. در فهرست مقادیر، آمریکا از دو بخش شمالی و جنوبی تشکیل شده در حالی که در این مثال فقط به آمریکا اشاره شده است.

تعیین طرحی برای یک نوع ساده

برای اینکه طرحی ایجاد شود تا با مقایسه محتویات با آن طرح، اعتبار محتویات تعیین گردد، می‌توان از زبان ویژه regex (regular expression) استفاده نمود. اساس زبان regex در XML Schema زبان عبارتهای معمولی Perl است. برای توضیح این زبان می‌توان در یک فصل جداگانه صحبت کرد؛ ولی در اینجا اشاره‌ای به آن می‌نماییم.

به منظور تعیین طرحی برای یک نوع ساده :

- ۱- در قسمت اعلان نوع دلخواه (شماره ۴ صفحه ۸۱) عبارت `<xsd:pattern>` را تایپ کنید.
 - ۲- سپس `value="regex"` را تایپ کنید. regex یک عبارت معمولی است که باید با ظاهر محتویات، مطابقت داشته باشد و به صورت زیر ایجاد می‌گردد:
- حروف، اعداد و نمادهایی را که در محتویات ظاهر خواهند شد، تعیین کنید.

(.) علامت نقطه به جای هر کاراکتری قرار می‌گیرد.

`\d` برای ارقام و `\D` برای non-digit استفاده می‌شود.

`\s` به جای فضاهای خالی (که با کلیدهای Enter ، Tab و Spacebar ایجاد می‌شوند) و `\S` برای کاراکترهای غیر خالی به کار می‌رود.

`X*` به جای هیچ یا چند `x` می‌نشیند؛ `(XY)*` نشانه هیچ یا چند `xy` است.

`X?` به جای یک یا هیچ `x` و `(XY)?` به جای یک یا هیچ `xy` به کار می‌رود.

`X+` معادل یک یا چند `x` و `(XY)+` نیز معادل یک یا چند `xy` می‌باشد.

`[abc]` شامل یک یا گروهی از مقادیر `a`، `b` یا `c` می‌گردد.

`[0-9]` مقادیر 0 تا 9 را نشان می‌دهد.

```
code.xsd
<xsd:element name="invoice_number">
<xsd:simpleType>
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="INV #99\d{3}"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

شکل ۲۷-۶ . این طرح محتویات عنصر `invoice_number` را به رشته‌ای که با `INV #99` شروع می‌شود و سه رقم در ادامه آن دارد محدود می‌نماید. هر کاراکتر که در `regex` ظاهر می‌شود باید در همان موقعیتی از محتویات که برای عنصر معتبر است ظاهر شود.

```
code.xml
<invoice_number>INV #99426</invoice_number>
```

شکل ۲۸-۶ . در اینجا مثال معتبر الگوی شکل ۲۷-۶ را می‌بینید: کاراکترهای `INV #99` به همراه سه رقم.

```
code.xsd
<xsd:element name="gestation">
<xsd:simpleType>
  <xsd:restriction base="xsd:timeDuration">
    <xsd:pattern value="P\d+D"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

شکل ۲۹-۶ . طرحها را برای کنترل محتویات عناصری که بر اساس نوعهای دیگر ایجاد شده‌اند نیز می‌توانید به کار برید. برای مثال اگر می‌خواهید عنصر `gestation` (بارداری) شامل تعداد روزهای دوره بارداری بزرگ باشد باید طرحی را که در این مثال می‌بینید پیاده سازید. توجه داشته باشید این طرح با نوع کاری ندارد و شامل حرف بزرگ `P` به همراه یک یا چند رقم می‌شود که با حرف `D` پایان می‌یابد.

```
code.xml
<gestation>P108D</gestation>
```

شکل ۳۰-۶ . مثالی برای الگوی شکل ۲۹-۶.

برای اینکه در محتویات، واژه `this` یا `that` وجود داشته باشد `that | this` به کار می‌رود. واژه‌های اختیاری دیگری را نیز به کمک خطوط عمودی دیگر می‌توانید اضافه نمایید.

`{5}` یعنی پنج `x` در یک ردیف.

`{5,}` یعنی حداقل پنج `x` در یک ردیف.

`{5,8}` معرف حداقل پنج و حداکثر هشت `x` در یک ردیف است.

`{2}(xyz)` به معنی دو `xyz` در یک ردیف می‌باشد.

۳- برای تکمیل عنصر `xsd:pattern` علامتهای `</>` را تایپ کنید.

نکته‌ها

◀ یکی از تفاوت‌های مهم بین عبارتهای معمولی XML Schema و عبارتهای معمولی Perl این است که همیشه عبارت معمولی و محتویات عنصر با یکدیگر تطابق دارند و مانند Perl برای ابتدا یا پایان یک خط از کاراکترهای `^` یا `$` استفاده نمی‌شود.

◀ برای کسب اطلاعات بیشتر درباره Perl به آدرس <http://www.perl.com/pub/doc/manual/html/pod/perl-re.html> یا کتاب Perl و CGI از همین مؤلف مراجعه کنید.

```
code.xsd
<xsd:element name="language">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="English|Latin"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

شکل ۳۱-۶. می‌توان از طرحها برای ارائه گزینه‌های گوناگون محتویات یک عنصر استفاده کرد. راه استاندارد برای ارائه گزینه‌های گوناگون در صفحه ۸۳ گفته شده است.

```
code.html
<language>English</language>
```

شکل ۳۲-۶. مثالی برای الگوی شکل ۳۱-۶.

تعیین مقادیر قابل قبول

برای محدود کردن محتویات یک عنصر یا ویژگی می‌توان بیشترین و کمترین مقدار آن را مشخص نمود.

برای تعیین بیشترین مقدار ممکن :

۱- در قسمت اعلان نوع دلخواه (شماره ۴ صفحه ۸۱) عبارت `<xsd:maxInclusive value="n">` را تایپ کنید. به حرف I که به صورت حرف بزرگ در عبارت Inclusive (جامع) قرار گرفته توجه نمایید.

۲- عبارت "n" را تایپ کنید. برای اینکه محتویات معتبر باشند باید کوچکتر یا برابر n باشند.

۳- برای تکمیل عنصر `xsd:maxInclusive` علامتهای `</>` را تایپ کنید.

روش دیگر برای تعیین بیشترین مقدار ممکن:

۱- در قسمت اعلان نوع دلخواه (شماره ۴ صفحه ۸۱) عبارت `<xsd:maxExclusive value="n">` را تایپ کنید. به حرف E در عبارت Exclusive (انحصاری) توجه نمایید.

۲- عبارت "n" را تایپ کنید. محتویات باید کوچکتر n باشد تا معتبر محسوب گردد.

۳- برای تکمیل عنصر `xsd:maxExclusive` علامتهای `</>` را تایپ نمایید.

```
code.xsd
<xsd:element name="population">
<xsd:simpleType>
  <xsd:restriction base="xsd:integer">
    <xsd:maxInclusive value="5000"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

شکل ۳۳-۶. طرح `xsd:maxExclusive` بیشترین مقدار ممکن برای یک عنصر را تعیین می‌کند.

```
code.xml
<population>5000</population>
```

```
code.xml
<population>4999</population>
```

شکل ۳۴-۶. عناصر `population` معتبر هستند؛ زیرا در مثال اول برابر و در مثال دوم کوچکتر از `xsd:maxInclusive` است.

```
code.xsd
<xsd:element name="population">
<xsd:simpleType>
  <xsd:restriction base="xsd:integer">
    <xsd:maxExclusive value="5000"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

شکل ۳۵-۶. در طرح `xsd:maxInclusive` محتویات باید کمتر از مقدار `xsd:maxExclusive` باشد.

```
code.xml
<population>5000</population>
```

```
code.xml
<population>4999</population>
```

شکل ۳۶-۶. عنصر `population` اول غیر معتبر ولی دومی معتبر است.

برای تعیین کمترین مقدار ممکن :

- ۱- در قسمت اعلان نوع دلخواه (شماره ۴ صفحه ۸۱) عبارت `<xsd:minInclusive>` را تایپ کنید. حرف I باید در Inclusive بزرگ باشد.
- ۲- عبارت `value="n"` را تایپ نمایید. محتویات باید کمتر یا برابر مقدار n باشند.
- ۳- برای تکمیل عنصر `xsd:minInclusive` علامتهای `</>` را تایپ نمایید.

روش دیگر برای تعیین کمترین مقدار ممکن:

- ۱- در قسمت اعلان نوع دلخواه (شماره ۴ صفحه ۸۱) عبارت `<xsd:minInclusive>` را تایپ کنید. حرف I باید در Inclusive بزرگ باشد.
- ۲- عبارت `value="n"` را تایپ کنید. محتویات معتبر باید کوچکتر از n باشند.
- ۳- با تایپ علامتهای `</>` عنصر `xsd:Exclusive` را تکمیل کنید.

نکته‌ها

- ◀ برای یک نوع نمی‌توانید به صورت هم‌زمان دو مقدار حداقل یا دو مقدار حداکثر در نظر بگیرید. مقادیر حداقل و حداکثر را می‌توانید ترکیب کنید یا فقط یکی از مقادیر را به کار برید.
- ◀ مفاهیم کوچکتر از و بزرگتر از برای مقادیر، مشخص است. مقدار تاریخ یا زمان بزرگتر، برای آینده و مقدار تاریخ یا زمان کوچکتر برای گذشته استفاده می‌شود.

```
code.xsd
<xsd:element name="start_date">
<xsd:simpleType>
  <xsd:restriction base="xsd:date">
    <xsd:minInclusive value="1999-07-25"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

شکل ۳۷-۶. طرح `xsd:minInclusive` کمترین مقدار ممکن را برای یک عنصر تعیین می‌کند. یعنی عنصر `start_date` باید بیست‌وپنجم جولای ۱۹۹۹ به بعد باشد.

```
code.xml
<start_date>1999-07-25</start_date>
```

```
code.xml
<start_date>1999-07-26</start_date>
```

شکل ۳۸-۶. عناصر `start_date` معتبر هستند؛ زیرا اولی برابر و دومی بزرگتر از مقدار `xsd:minInclusive` است.

```
code.xsd
<xsd:element name="start_date">
<xsd:simpleType>
  <xsd:restriction base="xsd:date">
    <xsd:minExclusive value="1999-07-25"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

شکل ۳۹-۶. در طرح `xsd:minInclusive` مقدار عنصر باید بیشتر از مقدار `xsd:minInclusive` باشد.

```
code.xml
<start_date>1999-07-25</start_date>
```

```
code.xml
<start_date>1999-07-26</start_date>
```

شکل ۴۰-۶. عنصر `start_date` در مثال اول غیر معتبر است؛ زیرا نمی‌تواند با مقدار `xsd:minInclusive` برابر باشد. عنصر مثال دوم به دلیل اینکه بزرگ‌تر از مقدار تعیین شده است معتبر می‌باشد.

محدود کردن طول یک نوع ساده

یک راه برای تعریف عنصر رشته‌ای یا نوع ساده URL، تعیین محدودیت طول آن است.

برای تعیین طول دقیق یک عنصر:

عبارت `<xsd:lengthvalue="x"/>` را در قسمت اعلان دلخواه تایپ کنید. منظور از x تعداد کاراکترهایی است که عنصر می‌تواند داشته باشد.

برای تعیین حداقل طول یک عنصر:

عبارت `<xsd:minLengthvalue="m"/>` را در قسمت اعلان نوع دلخواه تایپ نمایید. منظور از m حداقل کاراکترهای یک عنصر است.

برای تعیین حداکثر طول یک عنصر:

عبارت `<xsd:maxLengthvalue="n"/>` را در قسمت اعلان نوع دلخواه تایپ کنید. n حداکثر کاراکترهای عنصر را مشخص می‌نماید.

نکته‌ها

- ◀ اگر طول را تعیین کنید مقدار حداقل یا حداکثر را نمی‌توانید معین نمایید و برعکس اگر مقدار حداقل یا حداکثر را تعیین کردید امکان تعیین طول وجود ندارد.
- ◀ مقدارهایی که برای `xsd:length`، `xsd:minLength` و `xsd:maxLength` در نظر گرفته می‌شوند باید اعداد صحیح مثبت باشند.
- ◀ اگر عنصر بر اساس نوع دودویی تعریف شده باشد طول، اعداد داده‌های دودویی را مشخص می‌کند. اگر عنصر به صورت فهرست باشد (صفحه ۹۰) طول، تعداد اعضای فهرست را معین می‌نماید.

```
code.xsd
<xsd:element name="animal_code">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:length value="4"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

شکل ۴۱-۶. طول یک رشته (string) را می‌توان تنظیم کرد.

```
code.xml
<animal_code>TIGR</animal_code>
```

شکل ۴۲-۶. عنصر `animal_code` به دلیل اینکه شامل چهار کاراکتر است (شکل ۴۱-۶)، معتبر می‌باشد.

```
code.xsd
<xsd:element name="description">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="200"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

شکل ۴۳-۶. طول عنصر `string` را می‌توان محدود کرد.

```
code.xml
<description> The tiger (panthera tigris), largest of
all cats, is one of the biggest and most fearsome
predators in the world.</description>
```

شکل ۴۴-۶. عنصر `description` معتبر است؛ زیرا می‌تواند از ۲۰۰ کاراکتر تشکیل شود در حالی که شامل ۱۱۳ کاراکتر می‌باشد.

```
code.xml
<description> The tiger (panthera tigris), largest of
all cats, is one of the biggest and most fearsome
predators in the world. Powerfully built with fierce
retractile claws (they can be pulled into the paw, like
a house cat's), the tiger's distinctive gold coloring
with black stripes allows it to melt unseen into its
environment.</description>
```

شکل ۴۵-۶. به دلیل اینکه عنصر `description` از ۳۱۷ کاراکتر تشکیل شده است معتبر نیست.

نسخه الکت

محدود کردن رقمهای اعداد

تعداد رقمهای قسمت صحیح و اعشار اعداد را می‌توان محدود کرد.

برای تعیین تعداد رقمهای اعداد:

۱- در قسمت اعلان نوع دلخواه (شماره ۴ صفحه ۸۱) عبارت `xsd:precision` را تایپ کنید.

۲- عبارت `value="n"` را تایپ نمایید. منظور از `n` حداکثر تعداد رقمهای یک عدد است.

۳- با تایپ علامتهای `>` این مرحله را تکمیل کنید.

برای تعیین تعداد رقمهای سمت راست ممیز:

۱- در قسمت اعلان نوع دلخواه، عبارت `xsd:scale` را تایپ نمایید.

۲- عبارت `value="n"` را تایپ نمایید. `n` حداکثر تعداد رقمهای سمت راست ممیز را مشخص می‌کند.

۳- علامتهای `>` را تایپ نمایید تا این مرحله پایان پذیرد.

نکته‌ها

◀ طرح `xsd:precision` باید از اعداد صحیح مثبت یعنی `{۱،۲،۳،...}` تشکیل گردد. مقدار این طرح نمی‌تواند صفر یا کمتر از `xsd:scale` باشد.

◀ `xsd:scale` باید از مجموعه اعداد صحیح `{۰،۱،۲،۳،...}` تشکیل شود.

◀ `xsd:precision` و `xsd:scale` مقدار حداکثر را مشخص می‌نمایند یعنی تعداد ارقام کمتر نیز قابل قبول است.

◀ `xsd:precision` و `xsd:scale` برای هر نوع عددی معتبر می‌باشند؛ ولی برای نوعهای رشته‌ای، تاریخ و غیره قابل قبول نیستند.

```
code.xsd
<xsd:element name="numbers">
  <xsd:simpleType>
    <xsd:restriction base="xsd:decimal">
      <xsd:precision value="5"/>
      <xsd:scale value="2"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

شکل ۴۶-۶. مقدار `precision` تعداد کل رقمهای عدد

را تعیین می‌کند. `scale` تعداد رقمهای اعشار عدد را که در سمت راست ممیز قرار می‌گیرند مشخص می‌نماید.

```
code.html
<numbers>564.59</numbers>
```

```
code.html
<numbers>34.5</numbers>
```

شکل ۴۷-۶. عناصر هر دو مثال معتبرند؛ زیرا می‌توانند

حداکثر پنج رقم و حداکثر دو رقم اعشار داشته باشند.

```
code.xml
<numbers>1476.32</numbers>
```

```
code.xml
<numbers>4.3987</numbers>
```

شکل ۴۸-۶. این دو عنصر عددی غیر معتبر هستند؛

زیرا مثال اول شش رقم دارد و مثال دوم از چهار رقم اعشار تشکیل شده است.

ایجاد نوعهای فهرستی

تا به حال درباره عناصری صحبت کردیم که از یک بخش تشکیل می‌شدند. اگر عنصری را به صورت تاریخ تعریف کنید فقط می‌تواند شامل یک تاریخ باشد. ولی اگر به عنصری نیاز داشته باشید که فهرستی از تاریخ داشته باشد می‌توانید نوع تاریخ خود را به صورت نوع فهرستی تعریف کنید.

برای ایجاد نوع فهرستی:

- ۱- عبارت `<xsd:simpleType>` را برای آغاز تعریف نوع فهرستی تایپ کنید.
- ۲- سپس `<name="label">` را تایپ کنید. label نام عنصری است که می‌خواهید تعریف کنید و در سند XML به صورت `<label>` ظاهر می‌گردد.
- ۳- عبارت `<xsd:list base="individual">` را تایپ کنید. منظور از individual نام نوع ساده‌ای است که مقدار هر یک از اعضای فهرست را تعریف می‌کند.
- ۴- فهرستها را می‌توان با طرحهای `xsd:length`، `xsd:maxLength` و `xsd:minLength` (صفحه ۸۸) و `xsd:enumeration` (صفحه ۸۳) محدود نمود.
- ۵- برای تکمیل تعریف فهرست عبارت `</xsd:list>` را تایپ کنید.
- ۶- برای تکمیل تعریف نوع ساده، عبارت `</xsd:simpleType>` را تایپ نمایید.

نکته‌ها

- ◀ اگر به ایجاد محدودیت در فهرست نیازی ندارید می‌توانید علامتهای `</>` را تایپ و عنصر `xsd:list` را تکمیل کنید (شکل ۴۹-۶).
- ◀ نوع فهرستی را می‌توان تنها براساس نوع ساده ایجاد کرد (به فصل بعد مراجعه کنید).
- ◀ فضای خالی، تعداد اعضای فهرست را مشخص می‌کند. سپس باید هنگام تعیین طول فهرست رشته‌ای بیشتر دقت کنید.

```
code.xsd
<xsd:simpleType name="datelist">
  <xsd:list base="xsd:date"/>
</xsd:simpleType>
<xsd:element name="list_of_birthdays"
type="datelist"/>
```

شکل ۴۹-۶. یک فهرست باید بر اساس یک نوع ساده دلخواه یا درونی باشد. نوع فهرستی `datelist` بر اساس نوع `xsd:date` می‌باشد. در قسمت اعلان نوع ساده دلخواه، عنصر `list_of_birthdays` تعریف شده است.

```
code.xml
<list_of_birthdays>1893-04-20 1904-05-11
</list_of_birthdays>
```

شکل ۵۰-۶. یک فهرست می‌تواند از هیچ یا چند مقدار که بر اساس ویژگی تعیین و با فضای خالی جدا شده است تشکیل گردد.

```
code.xsd
<xsd:simpleType name="three_datelist">
  <xsd:list base="xsd:date">
    <xsd:length value="3"/>
  </xsd:list>
</xsd:simpleType>
<xsd:element name="list_of_3_birthdays"
type="three_datelist"/>
```

شکل ۵۱-۶. با طرحهای `xsd:length`، `xsd:minLength` و `xsd:maxLength` می‌توان محدودیت بیشتری برای نوع فهرستی ایجاد کرد.

```
code.xml
<list_of_3_birthdays>1893-04-20 1904-05-11
1852-06-25</list_of_3_birthdays>
```

شکل ۵۲-۶. نوع فهرستی `list_of_3_birthdays` باید از سه تاریخ تشکیل گردد تا معتبر باشد.

تعریف محتویات یک عنصر

به منظور استفاده از یک الگو برای از پیش تعریف کردن محتویات یک عنصر دو روش وجود دارد. می‌توان محتویات و مقدار عنصری را که در XML به صورت عنصر خالی ظاهر می‌شود تعریف و تعیین نمود. به حالت اول مقدار ثابت و به حالت دوم مقدار پیش فرض می‌گویند.

برای تعیین محتویات یک عنصر :

- ۱- در برچسب عنصر عبارت `fixed=` را تایپ کنید.
- ۲- سپس `"value"` را تایپ کنید. `value` محتویات عنصر را تعیین می‌نماید و نباید خالی باشد.

برای تعیین مقدار اولیه یک عنصر :

- ۱- در برچسب عنصر عبارت `default=` را تایپ نمایید.
- ۲- سپس `"value"` را تایپ کنید. `value` محتویات عنصر را مشخص می‌کند که می‌تواند خالی باشد و یا حذف شود.

نکته‌ها

- ◀ ویژگی `fixed` محتویات عنصری را تعیین می‌کند که در XML ظاهر می‌شود. اگر حذف شود محتویاتی نیز برای آن در نظر گرفته نمی‌شود.
- ◀ اگر ویژگی `fixed` تنظیم شود و عنصر خالی باشد مقدار عنصر به صورت خودکار با مقدار ثابت تنظیم می‌شود.
- ◀ چنانچه ویژگی `default` تنظیم گردد ولی عنصر از XML حذف شود مقدار عنصر به صورت خودکار با مقدار پیش فرض تنظیم می‌گردد.
- ◀ در صورتی که ویژگی `default` در نظر گرفته شود و عنصر در XML ظاهر نگردد محتویات آن به ویژگی `fixed` بستگی خواهد داشت.
- ◀ ویژگی `default` و `fixed` را نمی‌توان به صورت هم‌زمان داشت.

```
code.xsd
<xsd:element name="status" type="xsd:string"
fixed="endangered"/>
```

شکل ۶-۵۳ . عنصر `status` در سند XML ظاهر شده و شامل رشته `endangered` می‌باشد. اگر خالی باشد رشته `endangered` برای آن در نظر گرفته خواهد شد.

```
code.html
<status>vulnerable</status>
```

شکل ۶-۵۴ . عنصر `status` براساس الگوی شکل ۶-۵۳ معتبر است.

```
code.html
<status>vulnerable</status>
```

شکل ۶-۵۵ . براساس الگوی شکل ۶-۵۳ عنصر `status` معتبر نیست.

```
code.xsd
<xsd:element name="status" type="xsd:string"
default="endangered"/>
```

شکل ۶-۵۶ . اگر عنصر `status` را با یک مقدار پیش فرض تعریف کنیم در صورت ظهور یا عدم نمایش عنصر `status` در سند XML برای آن، محتویات پیش فرض در نظر گرفته خواهد شد.

```
code.html
<status>endangered</status>
```

```
code.html
<status>vulnerable</status>
```

شکل ۶-۵۷ . هر دو عنصر `status` معتبرند. ویژگی پیش فرض تنها یک مقدار اولیه را تنظیم می‌کند و مقادیر دیگر نیز قابل قبول هستند.