

فضاهای نام، الگوها و معتبرسازی



در فصل ۶ (تعریف نوعهای ساده) و فصل ۷ (تعریف نوعهای پیچیده) آموختید که چگونه الگویی تهیه کنید که عنصرها و ویژگیهای سند XML را مشخص کند. با این روش یک رده از سندهای XML را می‌توان تهیه نمود.

هر یک از این سندهای XML را - که نمونه یا Instance نامیده می‌شوند - می‌توان با الگو مقایسه نمود تا درستی آنها تایید شود (صفحه ۷۶). یکپارچه بودن اطلاعات با این کار تضمین می‌گردد.

در صورتی که سند XML از فضاهای نام مختلفی ساخته شده باشد عمل معتبرسازی آن پیچیده‌تر می‌گردد. جزئیات این مطلب در فصل ۸ (استفاده از فضاهای نام در XML) توضیح داده شده است.

در این فصل به تشریح عملیات معتبرسازی می‌پردازیم (برای توضیحات بیشتر در مورد نحوه استفاده از معتبرساز الگو به مبحث معتبرسازی XML با استفاده از الگو در صفحه ۲۴۵ رجوع کنید).

الگوها و فضاهای نام

همان‌گونه که گفته شد، فضای نام مجموعه‌ای از عنصرها و ویژگی‌های مرتبط با یکدیگر است که به وسیله یک نام مشخص می‌گردد. این نام شبیه به URL می‌باشد. معمولاً فضاهای نام برای مشخص کردن عنصرهایی به کار می‌روند که به صورت سراسری تعریف شده‌اند و علاوه بر آن نامهایی شبیه به یکدیگر نیز دارند (به طور کلی اجزایی که به صورت محلی تعریف می‌شوند به وسیله محتوای خود از یکدیگر متمایز می‌شوند). نام فضای نام منحصر به فرد است، نام فضای نام و نام عنصر مورد نظر نیز باید منحصر به فرد باشند.

هنگامی که سندی را که عنصرهایش با یک فضای نام مشخص شده‌اند معتبرسازی می‌کنید، باید بدانید که یک عنصر از آن منحصر به فرد است و یا خیر. همچنین باید از نحوه تعریف آن، این که در چه زمانی و چند بار باید تکرار شود و دارای چه عنصرهای دیگری باید باشد و . . . اطلاع داشته باشید. تمامی این اطلاعات در یک الگو موجود می‌باشند، عنصرهای فضای نام نیز در این الگو تعریف شده‌اند.

الگو می‌تواند تعریف کند که یک سند XML چگونه باید باشد. همچنین می‌تواند به طور هم‌زمان با عنصرهای خود (که به وسیله این الگو تعریف می‌شوند) یک فضای نام ایجاد کند و یا آن را توسعه دهد. بعد از اینکه عنصرهای یک الگو به یک فضای نام خاص مرتبط شدند، عنصرهایی با پیشوند آن فضای نام در سند قابل استفاده‌اند. این سندها به وسیله الگوی مربوط قابل معتبرسازی هستند.

توسعه یک فضای نام

می‌توان اجزایی از الگو را که به صورت سراسری تعریف شده‌اند (یا به عبارت دیگر در بالاترین مرتبه می‌باشند) با یک فضای نام مرتبط کرد. بدین وسیله می‌توان از این اجزاء در سندهای الگوهای دیگر استفاده نمود. جزئی که به صورت سراسری تعریف شده باشد فرزند مستقیمی از عنصر `xsd:schema` است. این جزء می‌تواند تعریفهای عنصرها، تعریفهای ویژگیها، تعریفهای نوعهای ساده و یا پیچیده، تعریفهای گروههای نام‌گذاری شده و یا تعریفهای گروههای ویژگی باشد.

با این که احتیاجی به انجام کار خاصی در قسمت تعریفها نمی‌باشد اما باید فضای نام هدف را که اجزا با آن مرتبط می‌شوند مشخص نمود.

برای مشخص کردن فضای نام هدف :

در عنصر ریشه سند الگوی خود عبارت `targetNamespace="URL"` را تایپ کنید. URL آدرس فضای نامی است که می‌خواهید اجزای تعریف شده در این الگو را با آن مرتبط کنید. به این کار توسعه یک فضای نام می‌گویند.

نکته

◀ تنها اجزایی که به صورت سراسری تعریف شده باشند (به عبارت دیگر در بالاترین مرتبه باشند) با فضای نام مرتبط می‌شوند. البته این امر به آن معنی نیست که شما نمی‌توانید از قسمتهایی که به صورت محلی تعریف شده‌اند (و از این رو با فضای نام مرتبط نشده‌اند) استفاده کنید و یا آنها را معتبرسازی کنید (مانند عنصر `animal` در شکل ۹-۱). هنگامی که ما یک نمونه سند XML را با الگویی که در شکل نشان داده شده معتبرسازی می‌کنیم، پردازشگر به طور خودکار می‌داند در کجا به دنبال تعریف عنصر `endangered_species` بگردد. هم‌چنین می‌داند که این عنصر دارای عنصر `animal` تایید نشده است (عنصری که با یک فضای نام مرتبط نشده است).

```
code.hsd
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
targetNamespace="http://www.cookwood.com/ns/end_species1">
<xsd:element name="endangered_species">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="animal"
type="animalType"
maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="animalType">
<xsd:sequence>
<xsd:element name="name" type="nameType"
minOccurs="2"/>
<xsd:element name="threats"
type="threatsType"/>
<xsd:element name="weight"
type="xsd:string" minOccurs="0"
maxOccurs="1"/>
...
</xsd:sequence>
</xsd:complexType>
...
```

شکل ۹-۱: در این مثال، عنصر

`endangered_species` و نوع پیچیده `animalType`

با فضای نام

`http://www.cookwood.com/ns/end_species1`

مشخص شده‌اند. عنصرهای `animal`، `threats` و

`weight` با فضای نام مرتبط نشده‌اند زیرا در بالاترین

مرتبه نیستند.

```

code.hsd
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/
2000/10/XMLSchema"
targetNamespace="http://www.cookwood.com
/ns/end_species1"
elementFormDefault="qualified">
<xsd:element name="endangered_species">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="animal"
type="animalType"
maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="animalType">
<xsd:sequence>
<xsd:element name="name" type="nameType"
minOccurs="2"/><xsd:element
name="threats" type="threatsType"/>
<xsd:element name="weight"
type="xsd:string" minOccurs="0"
maxOccurs="1"/>
...
</xsd:sequence>
</xsd:complexType>
...

```

شکل ۲-۹ : اکنون عنصرهای `animal`، `threats` و `weight`، که به صورت محلی تعریف شده‌اند، با فضای نام مرتبط شده‌اند. این عنصرها در سند XML قابل استفاده‌اند.

افزافه کردن تمامی عنصرهایی که به صورت محلی تعریف شده‌اند

به صورت پیش‌فرض، تنها عنصرهایی که به صورت سراسری تعریف شده باشند (و در بالاترین مرتبه باشند) با فضای نام هدف مرتبط می‌شوند. در صورتی که بخواهید تعریفهای عنصرهای محلی را (آنهايي که در یک یا چند مرتبه پایینتر قرار دارند) اضافه کنید می‌توانید تمام عنصرها را یکبار اضافه کنید، یا تمام تعریفهای ویژگیها را یکبار اضافه کنید (همان‌گونه که در پایین نشان داده شده) و یا عنصرها و ویژگیها را (همان‌گونه که در صفحه ۱۲۵ نشان داده شده است) به صورت جدا از یکدیگر اضافه کنید.

برای این که تمامی عنصرهایی را که به صورت محلی تعریف شده‌اند به فضای نام اضافه کنید :

عبارت `elementFormDefault="qualified"` را در عنصر `xsd:schema` تایپ کنید.

برای این که تمامی ویژگیهایی را که به صورت محلی تعریف شده‌اند به فضای نام اضافه کنید :

عبارت `attributeFormDefault="qualified"` را در عنصر `xsd:schema` تایپ کنید.

نکته

همان‌گونه که مشاهده شد، مقدار پیش‌فرض هر یک از این ویژگیها `unqualified` است. این به آن معنی است که تنها اجزایی که به صورت سراسری (بالاترین مرتبه) تعریف شده باشند با فضای نام مرتبط می‌شوند مگر آنکه شما به صورت دیگری این انتخاب را تعیین کنید.

اضافه کردن عنصرهای خاصی که به

صورت محلی تعریف شده‌اند

ویژگی form برای مشخص کردن تعیین ارتباط یک عنصر خاص، محلی و تک با فضای نام هدف کاربرد دارد. این کار بستگی به مقدار پیش فرض ندارد.

برای این که عنصرهای خاصی که به صورت محلی تعریف شده‌اند به فضای نام اضافه شوند :

در تعریف عنصر عبارت "form="qualified" را تایپ کنید. عنصر بدون توجه به این که در کجا تعریف شده است، با فضای نام هدف مرتبط می‌شود.

اگر از ویژگی elementFormDefault="qualified" در عنصر xsd:schema استفاده کرده باشید، می‌توانید از ویژگی form برای جلوگیری از مرتبط شدن یک عنصر خاص، که به صورت محلی تعریف شده است، با فضای نام هدف استفاده کنید.

برای جلوگیری از مرتبط شدن یک عنصر خاص، که به صورت محلی تعریف شده است، با فضای نام هدف (بدون توجه به پیش فرض) :

در تعریف عنصر عبارت "form="unqualified" را تایپ کنید.

```
code.hsd
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
targetNamespace="http://www.cookwood.com/ns/end_species1"
elementFormDefault="qualified">
<xsd:element name="endangered_species">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="animal"
type="animalType"
maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="animalType">
<xsd:sequence>
<xsd:element name="name" type="nameType"
minOccurs="2"/>
<xsd:element name="threats"
type="threatsType" form="unqualified"/>
<xsd:element name="weight"
type="xsd:string" minOccurs="0"
maxOccurs="1" form="unqualified"/>
...
</xsd:sequence>
</xsd:complexType>
...
```

شکل ۳-۹: بدون توجه به تنظیم

elementFormDefault در عنصر xsd:schema هیچ‌یک از عناصر weight و threats با فضای نام هدف مرتبط نخواهند شد. این کار به دلیل آن است که ویژگی form آنها بر مقدار elementFormDefault برتری دارد.

```

code.xsd
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/
2000/10/XMLSchema"
targetNamespace="http://www.cookwood.com/
ns/end_species1"
xmlns="http://www.cookwood.com/
ns/end_species1">
<xsd:element name="endangered_species">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="animal"
type="animalType"
maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="animalType">
<xsd:sequence>
<xsd:element name="name" type="nameType"
minOccurs="2"/>
<xsd:element name="threats"
type="threatsType"/>
<xsd:element name="weight"
type="xsd:string" minOccurs="0"
maxOccurs="1"/>
...
</xsd:sequence>
</xsd:complexType>
...

```

شکل ۴-۹: برای این سند الگو، فضای نام پیش فرض با نام end_species1 را در عنصر پایه الگو تعریف کرده‌ایم. این مطلب به معنی آن است که تعریفهای مربوط به نوعهای بدون پیشوند (animalType)، (threatsType و nameType) در الگوی مرتبط با فضای نام end_species1 قرار دارند (تعریفهای مربوط به نوعهای با پیشوند در الگوی مرتبط با فضای نام XMLSchema هستند).

ارجاع اجزا به فضاهای نام

بعد از این که اجزای الگو (نوعهای ساده و یا پیچیده، عنصرها، ویژگیها، گروههای ویژگی و گروههای نام) با یک فضای نام مرتبط شدند می‌توان درون همان الگو و یا در الگوهای دیگر به آنها ارجاع کرد. به دلیل این که این اجزا با یک فضای نام مرتبط شده‌اند، برای ارجاع به آنان باید فضای نام نیز مشخص گردد.

در این کتاب، از پسوند xsd برای تمامی عنصرها XML Schema استفاده شده است. هم‌چنین الگوی XML Schema (که الگوی الگوها نامیده می‌شود) در عنصر ریشه الگو تعریف شده است. به همین دلیل وقتی به نوعهای موجود (نوعهایی که در الگوی الگوها موجودند) ارجاع داده می‌شود، باید از پسوند xsd استفاده شود. به این طریق، الگو متوجه می‌شود که تعاریف را در کجا بیابد.

ارجاع به نوعهای تعریف شده توسط کاربر (ساده و یا پیچیده)، عنصرها، ویژگیها و یا گروهها کمی پیچیده‌تر است. ارجاع به هر یک از موارد فوق باید حاوی اطلاعاتی درباره فضای نام مربوط باشد.

برای مشخص کردن یک فضای نام پیش فرض برای اجزای مورد ارجاع و ارجاع کردن به آنها در الگو:

- ۱- در عنصر ریشه سند الگو عبارت "xmlns="URL" را تایپ کنید. URL آدرس فضای نامی است که اجزای مورد ارجاع با آن مرتبط شده‌اند.
- ۲- در قسمت مقدار ویژگیهای type و ref مقدار reference (بدون پیشوند) را قرار دهید. reference نام جزئی است که با فضای نام پیش فرض مرتبط شده است.

برای تعریف یک فضای نام با یک پیشوند و استفاده از آن

پیشوند برای مشخص کردن اجزای مورد ارجاع در الگو :

۱- در عنصر اصلی سند XML خود عبارت `xmlns:prefix="URL"` را تایپ کنید. نامی است که برای مشخص کردن تعاریف در این الگو استفاده می‌شود. این تعاریف به فضای نامی که توسط URL مشخص شده است تعلق دارند.

۲- در قسمت مقدار ویژگیهای `type` و `ref` مقدار `prefix:reference` را قرار دهید که در آن `reference` نام جزئی است که با فضای نام مربوط به آن پیشوند مرتبط شده است (این پیشوند همان پیشوند مرحله ۱ است).

نکته‌ها

- ◀ شما می‌توانید به هر تعداد که لازم دارید فضای نام تعریف کنید و از تعریفهای درون آنها به وسیله پیشوندهای مربوط استفاده کنید.
- ◀ مقدار یک ویژگی باید با یک فضای نام مرتبط باشد (و یا با هیچ فضای نامی مرتبط نباشد).
- ◀ این تفاوت مهم را به خاطر داشته باشید : فضای نام هدف، فضای نامی است که به طور فعال و واقعی با یک جزء از فضای نام مرتبط می‌شود. یک تعریف فضای نام مانند `xmlns="URL"` یا `xmlns:end="URL"` فضای نامی را مشخص می‌کند که یک جزء با آن مرتبط شده است.

```
code.xsd
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
targetNamespace="http://www.cookwood.com/ns/end_species1"
xmlns:end="http://www.cookwood.com/ns/end_species1">
<xsd:element name="endangered_species">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="animal"
type="end:animalType"
maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="animalType">
<xsd:sequence>
<xsd:element name="name"
type="end:nameType" minOccurs="2"/>
<xsd:element name="threats"
type="end:threatsType"/>
<xsd:element name="weight"
type="xsd:string" minOccurs="0"
maxOccurs="1"/>
...
</xsd:sequence>
</xsd:complexType>
...
```

شکل ۹-۵: این سند با شکل ۹-۴ معادل است. در اینجا، از پیشوند `end` برای فضای نام `end_species` استفاده شده است. پس باید از پیشوند استفاده شود تا مشخص گردد که `end:animalType`، `end:nameType` و `end:threatsType` به فضای نام `end_species1` تعلق دارند. با وجود اینکه عنصرها به همان فضای نام تعلق دارند (که توسط ویژگی `targetNamespace` تعیین شده است) نحوه نمایش آنها تغییر کرده است.

```

code.xsd
<?xml version="1.0" ?>
<schema xmlns="http://www.w3.org/
2000/10/XMLSchema"
targetNamespace="http://www.cookwood.com/
ns/end_species1"
xmlns:end="http://www.cookwood.com/
ns/end_species1">
<element name="endangered_species">
<complexType>
<sequence>
<element name="animal"
type="end:animalType"
maxOccurs="unbounded"/>
</sequence>
</complexType>
</element>
<complexType name="animalType">
<sequence>
<element name="name"
type="end:nameType" minOccurs="2"/>
<element name="threats"
type="end:threatsType"/>
<element name="weight" type="string"
minOccurs="0" maxOccurs="1"/>
...
</sequence>
</complexType>
...

```

شکل ۶-۹: این سند کاملاً معادل شکل ۴-۹ است. تنها تفاوت در نشانه‌گذاری است. در اینجا الگوی الگوها به عنوان فضای نام پیش‌فرض تعیین شده است و از این رو احتیاجی به استفاده از پیشوند در مورد نامهای عنصرها (schema, element, complexType) و غیره) و نوعهای پیش‌ساخته شده (رشته‌ها) توسط xsd وجود ندارد. اما باید برای هر نوعی که مربوط به فضای نام دیگری باشد از پیشوند استفاده کرد (مانند end:nameType, end:animalType, end:threatsType).

الگوی الگوها به عنوان پیش‌فرض

اجباری به استفاده از پیشوند xsd برای عنصرها و نوعهای الگوی الگوها وجود ندارد بلکه از هر پیشوند دیگری نیز می‌توان استفاده نمود. اگر الگو بیشتر از نوعهای پیش‌ساخته استفاده می‌کند، استفاده از الگوی الگوها به عنوان فضای نام پیش‌فرض مناسبتر خواهد بود؛ با این کار احتیاجی به استفاده از پیشوند برای فضای نام مربوط نمی‌باشد.

برای تعریف الگوی الگوها به عنوان فضای نام پیش‌فرض یک الگو:

۱- برای شروع الگو، بعد از تعریف XML عبارت `<schema` را تایپ کنید (دقت کنید که از `xsd:schema` استفاده نشد).

۲- الگوی الگوها به عنوان فضای نام پیش‌فرض الگوی خود را با این دستور تعریف کنید:

۳- عبارت `targetNamespace="URL"` را تایپ کنید. URL فضای نامی است که مایلید تعریفهای نوعها و عنصرهای موجود در بالاترین مرتبه را با آن مرتبط کنید (این قسمت مانند قسمتهای گذشته می‌باشد).

۴- عبارت `xmlns:prefix="URL"` را تایپ کنید. prefix نامی است که برای مشخص کردن تعاریفی در این الگو استفاده می‌شود که به فضای نامی که توسط URL مشخص شده است تعلق دارند.

۵- مرحله ۴ را برای تمامی فضای نامهایی که دارای تعریفهایی هستند که در این الگو استفاده شده است تکرار کنید.

۶- برای کامل شدن این قسمت از الگو یک `>` تایپ کنید.

نکته

◀ معمولاً نیاز به تخصیص یک پیشوند برای فضای نام هدف وجود دارد (مگر اینکه این تعریفها را از یک فضای نام دیگر استخراج کنید (صفحه ۱۳۲)).

فضاهای نام و معتبرسازی XML

تا این مرحله شما تعدادی اجزا را با استفاده از یک فضای نام کنترل کردید و اکنون برای معتبرسازی سندی که از این اجزاء استفاده می‌کند آماده هستید. در فصلهای ۶ و ۷ در هنگام معتبرسازی احتیاجی به توجه به فضاهای نام نبود زیرا در آن سندها هیچ یک از اجزای تعریف شده را با فضای نامی مرتبط نکردید. اکنون که تمامی و یا قسمتی از اجزای شما به یک فضای نام تعلق دارند، باید در هنگام معتبرسازی سندهای خود آن فضای نام را نیز مشخص کنید.

ممکن است نیاز باشد محلی را مشخص کنید که در آن پردازشگر، الگوی حاوی تعریفهای اجزای مرتبط با فضای نام را بیابد.

نوشتن سندهای معتبر XML با استفاده از اجزای کنترل شده:

۱- ابتدا باید فضای نام اجزای مورد نظر مشخص شوند. این کار را می‌توان به یکی از طرق زیر انجام داد : با تعریف یک فضای نام پیش فرض ("xmlns="URL") و استفاده نکردن از پیشوند برای اجزا یا با تعریف یک فضای نام با پیشوند ("xmlns:prefix="URL") و استفاده از پیشوند برای اجزای مورد نظر. توضیحات بیشتر در فصل ۸ (استفاده از فضاهای نام در XML) آمده است.

۲- گاهی اوقات لازم است آدرس الگویی را تعیین کنیم که فضای نام مورد استفاده برای کنترل کردن عناصرها را تعریف و یا توسعه می‌دهد.

نکته

◀ به خاطر داشته باشید تنها عنصرهایی را که با فضای نام مرتبط شده‌اند کنترل کنید.

```
code.xml
<?xml version="1.0" ?>
<end:endangered_species
  xmlns:end="http://www.cookwood.com/ns/
  end_species1">
<animal>
<name language="English">Tiger</name>
<name language="Latin">panthera tigris</name>
<threats>
  <threat>poachers</threat>
  <threat>habitat destruction</threat>
  <threat>trade in tiger bones for traditional
  Chinese medicine (TCM)</threat>
</threats>
<weight>500 pounds</weight>
<length>3 yards from nose to tail</length>
...
</end:endangered_species>
```

شکل ۷-۹: در الگوی نشان داده شده در شکل ۴-۹ عنصر endangered_species با فضای نام http://www.cookwood.com/ns/end_species1 مرتبط شده بود (به دلیل targetNamespace موجود در الگو). در صورتی که بخواهیم سندی که حاوی این عنصر باشد معتبرسازی کنیم، باید endangered_species را کنترل کنیم. از آنجا که animal با فضای نام هدف مرتبط نیست، به این کار احتیاج ندارد.

```
code.xml
<?xml version="1.0" ?>
<end:endangered_species
  xmlns:end="http://www.cookwood.com/ns/
  end_species1"
  xmlns:xsi="http://www.w3.org/2000/10/
  XMLSchema-instance"
  xsi:schemaLocation="http://www.cookwood.com/
  ns/end_species1
  http://www.cookwood.com/xml/schemas/
  end_species1.xsd">
<animal>
<name language="English">Tiger</name>
<name language="Latin">panthera tigris</name>
<threats>
  <threat>poachers</threat>
  <threat>habitat destruction</threat>
  <threat>trade in tiger bones for traditional
  Chinese medicine (TCM)</threat>
</threats>
<weight>500 pounds</weight>
<length>3 yards from nose to tail</length>
...
```

شکل ۸-۹: اولین سطری که مشخص شده است فضای نام مربوط به اشیایی را که از پیشوند XSI استفاده می کنند تعریف می کند. سطر سوم قسمت دومی که مشخص شده است فایلی بر روی سرور را مشخص می کند که فضای نام درون آن تعریف شده است.

مشخص کردن آدرس الگو

در فصلهای گذشته با چگونگی مشخص کردن آدرس یک الگو زمانی که عنصرهای کنترل شده در سند XML وجود نداشته باشد آشنا شدید (صفحه ۷۳). مشخص کردن آدرس یک الگو برای سندی از XML که دارای عنصرهای کنترل شده باشد نیز کاملاً مشابه است.

برای مشخص کردن آدرس الگو برای عنصرهای کنترل شده:

۱- در عنصر ریشه سند XML و بعد از تعریف فضای نامی که با عنصرهای کنترل شده مرتبط شده است عبارت

**xmlns:xsi="http://www.w3-
org/2000/10/XMLSchema-instance"**

را تایپ کنید. این کار برای تعریف فضای نامی است که به وسیله آن آدرس الگو مشخص می شود.

۲- سپس عبارت **xsi:schemaLocation="URL** را تایپ کنید. URL نام فضای نامی است که شما در مرحله بعد آدرس الگوی آن را می دهید (دقت کنید که کوتیشن بسته وجود ندارد).

۳- یک فاصله و یا **return** تایپ کنید.

۴- سپس عبارت **file.xsd** را تایپ کنید. **file.xsd** آدرس URL یک فایل اصلی می باشد که حاوی الگوی تعریف کننده فضای نام مورد استفاده در این سند است (دقت کنید که کوتیشن باز وجود ندارد).

۵- برای هر تعداد الگوی به کار رفته در سند مراحل ۲ تا ۴ را تکرار کنید.

نکته

بیشتر پردازشگرهای سند های XML برای پیدا کردن فایل الگو از ویژگی **xsi:schemaLocation** استفاده می کنند. برای مطمئن شدن از این موضوع به قسمت راهنمای پردازشگر خود مراجعه کنید.

الگوها در فایل‌های مختلف

برای استفاده از اجزای یک الگو در چندین الگو یا سند و یا برای استفاده بهتر از یک الگوی بزرگ می‌توان اجزای آن را به چند فایل مختلف تقسیم کرد.

برای اضافه کردن اجزای الگو :

- ۱- اجزای الگو را در چند فایل قسمت کنید. هر فایل باید به شکل متنی و دارای پسوند .xsd باشد.
- ۲- در الگویی که می‌خواهید اجزایی را اضافه کنید، عبارت

```
<xsd:include schemaLocation="included-
includedfile.xsd" file.xsd"/>
```

آدرس URL سند الگویی است که می‌خواهید اجزایی از آن را اضافه کنید.

نکته‌ها

- ◀ ویژگی targetNamespace فایل الگویی که اضافه می‌شود باید با فایل الگویی که آن را دریافت می‌کند یکسان باشد. برای اضافه کردن اجزای الگویی که فضاهاى نام هدف متفاوتی داشته باشند به قسمت وارد کردن اجزا در صفحه ۱۳۲ مراجعه کنید.
- ◀ در صورتی که فضای نام اضافه شده دارای فضای نام هدف نباشد، فرض می‌شود که فضای نام هدف آن با فضای نام سندی که به آن اضافه می‌شود یکسان است.

```
code.xsd
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/
2000/10/XMLSchema"
targetNamespace="http://www.cookwood.com/
ns/end_species1">
<xsd:complexType name="threatsType">
<xsd:sequence>
<xsd:element name="threat"
maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>
```

شکل ۹-۹: ابتدا یک فایل جدید با تعریف نوع پیچیده threatsType ایجاد می‌کنیم (threats.xsd) که در شکل نشان داده شده است). حال می‌توانیم از این تعریف در الگوهای دیگر استفاده کنیم.

```
code.xsd
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/
2000/10/XMLSchema"
targetNamespace="http://www.cookwood.com/
ns/end_species1"
xmlns="http://www.cookwood.com/
ns/end_species1">
<xsd:include schemaLocation=
"http://www.cookwood.com/xml/schemas/
threats.xsd"/>
<xsd:element name="endangered_species">
<xsd:complexType><xsd:sequence>
<xsd:element name="animal"
type="animalType"
maxOccurs="unbounded"/>
</xsd:sequence></xsd:complexType>
</xsd:element>
<xsd:complexType name="animalType">
<xsd:sequence>
<xsd:element name="name" type="nameType"
minOccurs="2"/>
<xsd:element name="threats"
type="threatsType"/>
...</xsd:complexType>
```

شکل ۹-۱۰: با استفاده از عنصر xsd:include می‌توان

اجزای الگو را از الگوهای دیگر با فضای نام یکسان وارد کرد.

وارد کردن اجزا

وقتی بخواهید سندهایی از XML را که عنصرهایشان با چندین فضای نام مرتبط شده باشد معتبرسازی کنید می‌توانید اجزایی از الگو را که در بالاترین مرتبه باشند از الگوهای دیگر با فضاهای نام هدف متفاوت وارد کنید.

برای وارد کردن اجزا از الگوهایی که دارای فضای نام هدف متفاوت باشند :

- ۱- بلافاصله بعد از عنصر `xsd:schema` در سند الگویی که الگوهای دیگر را به آن وارد می‌کنید عبارت `<xsd:import` را تایپ کنید.
- ۲- سپس عبارت `namespace="URL"` را تایپ کنید که در آن URL مشخص‌کننده نام فضای نامی است که اجزای الگوی واردشده با آن مرتبط شده‌اند.
- ۳- سپس در صورت لزوم، آدرس فایل حاوی الگوی تعریف‌کننده فضای نام مرحله ۲ را با تایپ کردن عبارت `xsi:schemaLocation="URL"` وارد کنید. URL نام فضای نامی است که شما می‌خواهید سندهای الگوی آن را آدرس بدهید.
- ۴- یک فاصله و یا `return` تایپ کنید.
- ۵- اکنون عبارت `"file.xsd"` را تایپ کنید. `file.xsd` آدرس URL فایلی است که حاوی الگوی مورد نظر برای وارد کردن اجزا از آن است.
- ۶- برای کامل کردن برجسب `xsd:import` یک `>` تایپ کنید.
- ۷- همان‌گونه که در صفحه ۱۲۶ توضیح داده شد، یک پیشوند برای فضای نام واردشده تعیین کنید تا بتوانید به اجزای وارد شده در الگوی خود ارجاع کنید.

نکته

◀ در صورتی که تا این مرحله تعریف نمونه الگویی XML را انجام نداده‌اید، باید این کار را همان‌گونه که در مرحله ۱ صفحه ۱۳۰ توضیح داده شده است انجام دهید.

```
code.xsd
<?xml version="1.0" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
  targetNamespace="http://www.cookwood.com/ns/end_species1"
  xmlns="http://www.cookwood.com/ns/end_species1"
  xmlns:rivers="http://www.cookwood.com/ns/rivers1">
<xsd:import namespace=
  "http://www.cookwood.com/ns/rivers1"
  xsi:schemaLocation=
  "http://www.cookwood.com/ns/rivers1
  http://www.cookwood.com/xml/schemas/
  rivers.xsd"/>
<xsd:element name="endangered_species">
  <xsd:complexType><xsd:sequence>
  <xsd:element name="animal"
  type="animalType"
  maxOccurs="unbounded"/>
  </xsd:sequence></xsd:complexType>
  </xsd:element>
<xsd:complexType name="animalType">
  <xsd:sequence><xsd:element name="name"
  type="nameType" minOccurs="2"/>
  <xsd:element name="threats"
  type="threatsType"/>
  ...
  <xsd:element name="habitat"
```

شکل ۱۱-۹ : با استفاده از وارد کردن یک الگو در الگوی دیگر می‌توان از اجزای الگوی اول برای تعریفهای اجزای الگوی دوم استفاده کرد.