

## تعریف عناصر و ویژگیها در یک

### DTD

در فصل دوم با عنوان "ایجاد یک DTD" طرز تنظیم یک DTD را فرا گرفتید. در این فصل با چگونگی ایجاد محتویات آن آشنا خواهید شد. در واقع می‌بینید که نوشتن قوانین تعیین عناصر و ویژگیها در اسناد XML یکسان است و برای DTD های داخلی و خارجی مشابه می‌باشد.

یک DTD باید برای تمام عناصر و ویژگیهایی که در سند XML وجود دارند، قوانینی تعریف کند. در غیر این صورت، سند XML معتبر نخواهد بود. اگر نیازی به اضافه کردن عناصری به سند XML پیدا کردید باید تعاریف آنها را نیز به DTD مربوط بیفزایید یا در صورت تمایل یک DTD جدید ایجاد نمایید.



## تعریف عناصر

```
code.dtd
<!ELEMENT endangered_species (animal)>
```

**شکل ۱-۳**. تمام عناصری که در یک سند XML ظاهر می‌شوند باید تعریف گردند. در این مثال عنصر endangered\_species به صورتی تعریف شده که شامل عنصر دیگری به نام animal می‌باشد.

برای اینکه اسناد XML محتویات و ساختار مشخصی داشته باشند باید محتویات و ساختار هر یک از عناصر موجود در سند XML را تعریف کنید.

برای تعریف یک عنصر:

```
code.dtd
<!ELEMENT picture EMPTY>
```

**شکل ۲-۳**. عناصری که به داده‌های دودویی اشاره دارند معمولاً به صورت EMPTY تعریف می‌گردند؛ زیرا فاقد داده‌های XML خواهند بود. گاهی با ویژگی‌هایی نیز همراه هستند (به صفحه ۴۹ نگاه کنید).

۱. عبارت `<!ELEMENT tag>` را تایپ کنید. منظور از

tag نام عنصری است که می‌خواهیم تعریف کنیم.

۲. اگر عنصر محتویاتی ندارد واژه EMPTY را تایپ نمایید.

۳. در غیر این صورت عبارت (contents) را تایپ کنید.

منظور از contents توضیحاتی درباره عنصر یا متن

محتویات آن است. پرانتزها را فراموش نکنید.

گزینه‌های مجاز برای این متغیر در صفحه‌های ۴۸

و ۴۴ توضیح داده شده‌اند. اگر می‌خواهید عنصر

شما شامل ترکیب‌های گوناگونی از عناصر و متن

باشد واژه ANY را تایپ کنید.

۴. در آخر برای تکمیل اعلان عنصر علامت > را تایپ

نمایید.

نکته‌ها

◀ ویژگی‌ها جزء محتویات محسوب نمی‌گردند. حتی عناصر

خالی می‌توانند دارای ویژگی‌هایی باشند (صفحه ۴۹).

◀ هنگام استفاده از ANY باید دقت بیشتری داشته باشید.

تمام قسمتهای DTD از قوانینی تشکیل شده که

مشخص می‌کند هر عنصر می‌تواند چه محتویاتی داشته

باشد و چه محتویاتی نداشته باشد. اگر تمام عناصر

بتوانند هر نوع محتویاتی داشته باشند دیگر نیازی به

استفاده از DTD نخواهد بود. چرا که DTDها به

سازگاری داده‌ها کمک می‌کنند و الزامی نیستند.

◀ اگر برای عنصری از واژه ANY استفاده کنید آن گاه این

عنصر نمی‌تواند از عناصری تشکیل شود که در DTD

تعریف نشده‌اند.

**شکل ۳-۳**. مقدار ANY به قدری نامشخص است که بدون استفاده می‌ماند. اگر نمی‌خواهید سند XML محدود باشد، باید از DTD صرف‌نظر کنید. در این مثال عنصر endangered\_species می‌تواند شامل هر چیزی مثل متن یا عناصر دیگر باشد. توجه داشته باشید که عناصر تشکیل دهنده endangered\_species نیز باید در DTD تعریف شده باشند.

```
code.dtd
<!ELEMENT endangered_species ANY>
```

- ◀ هر عنصر می‌تواند از عناصر دیگری تشکیل شود. با وجود این هر عنصر باید یک بار به صورت دقیق تعریف شود. هیچ عنصری تا وقتی که در DTD تعریف نشده باشد نمی‌تواند در یک سند XML معتبر ظاهر گردد.
- ◀ شما می‌توانید تعداد عناصر مشخصی را که می‌توانند در یک محل معین وجود داشته باشند کنترل کنید (به صفحه ۴۸ نگاه کنید).
- ◀ ترتیب اعلان و تعریف عناصر اهمیتی ندارد. برای مثال، می‌توانید ابتدا عنصر درون یک عنصر را اعلان کنید و سپس عنصر خارجی را تعریف نمایید بدون اینکه مشکلی به وجود آید.
- ◀ ترتیب ظاهر شدن عناصر یک سند XML را می‌توانید کنترل نمایید (صفحه ۴۶).
- ◀ در تمام اجزای XML بین حروف بزرگ و کوچک فرق گذاشته می‌شود. برای مثال عبارت ELEMENT!< با Element!< فرق دارد. می‌توانید از ترکیب حروف بزرگ و کوچک نیز استفاده کنید؛ ولی باید هر جایی که می‌خواهید به چیزی اشاره کنید همان ترکیب را به کار ببرید. بهتر است در نامگذاریها تمام حروف را به صورت کوچک بنویسید تا مجبور نباشید ترکیبهای مختلف را به خاطر بسپارید.
- ◀ DTD ها عناصر XML نیستند و برای بستن آنها به علامت / پیش از > نیاز نیست.

```
code.dtd
<!ELEMENT name (#PCDATA)>
<!ELEMENT weight (#PCDATA)>
<!ELEMENT threat (#PCDATA)>
```

شکل ۴-۳. بیشتر DTDها شامل عناصر متنی می‌باشند.

```
code.xml
<endangered_species >
<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>
```

شکل ۵-۳. به این مثال از یک سند معتبر XML توجه کنید. عنصر name متنی است. درباره ویژگی آن در صفحه ۵۰ صحبت خواهیم کرد. عناصر جداگانه threat نیز تنها شامل متن هستند ولی عنصر threats از عناصر threat تشکیل می‌شود و محتویات آن فقط به صورت متنی نیست.

## تعریف عنصری که محتویات آن به صورت متنی است

برخی از عناصر اسناد XML تنها شامل محتویات متنی می‌باشند. برای مثال عنصر Address شامل عناصر Street ، City ، State و Zip است. عنصر State تنها از دو حرف تشکیل می‌شود.

برای تعریف یک عنصر :

- ۱- عبارت **ELEMENT tag** را تایپ کنید. منظور از tag نام عنصری است که قصد تعریف آن را دارید.
  - ۲- سپس عبارت **(#PCDATA)** را تایپ کنید (پرانترها فراموش نشوند). این عبارت مشخص می‌کند که محتویات عنصر مورد نظر تنها به صورت متنی است.
  - ۳- برای تکمیل اعلان نوع عنصر، علامت < را تایپ نمایید.
- نکته‌ها

- ◀ عبارت PCDATA از واژه‌های Parsed Character Data گرفته شده است و به هر چیزی به جز متن نشانه‌گذاری شده اشاره می‌کند. مواردی که PCDATA به آنها اشاره می‌نماید عبارتند از : اعداد، حروف، نمادها و موجودیتها (به صفحه ۵۵ نگاه کنید).
- ◀ عنصری که شامل PCDATA است نمی‌تواند از عنصر دیگری تشکیل گردد.
- ◀ شما می‌توانید از #PCDATA به عنوان یک سری از گزینه‌ها استفاده کنید (صفحه ۴۷) و ممکن است آن را به صورت ترتیبی مورد استفاده قرار ندهید.
- ◀ محدودیت اصلی DTDها این است که نمی‌توانید داده وارد شده را به صورت عددی، داده‌ای، متنی یا هر چیزی دیگر معین کنید. به عبارت دیگر، یک سند XML که شامل: <YEAR>dragon</YEAR> است فقط برای <YEAR>2005</YEAR> معتبر می‌باشد. به این موضوع نوع داده قابل دسترس در الگوی XML می‌گویند (صفحه ۶۷).

## تعریف عنصری که یک فرزند دارد

هنگامی که اطلاعات خود را به قسمتهای کوچکتر تقسیم می‌کنید، عناصری به وجود می‌آیند که از عناصر دیگری تشکیل می‌شوند.

برای تعریف عنصری که یک عنصر فرزند دارد :

۱. عبارت `<!ELEMENT tag>` را تایپ کنید. منظور از tag عنصری است که می‌خواهیم تعریف کنیم.
۲. عبارت `(child)` را تایپ نمایید. منظور از child نام عنصری است که عنصر مورد نظر شما را تشکیل می‌دهد.
۳. برای تکمیل اعلان علامت `>` را تایپ کنید.

### نکته‌ها

- ◀ هنگامی که اعلام می‌کنید عنصری از عناصر دیگر تشکیل شده است، آن عنصر در هر سند XML که توسط DTD شما مورد استفاده قرارگیرد شامل همان عناصر خواهد بود. درغیراین‌صورت سند شما معتبر نیست.
- ◀ اگر برجسی به گونه‌ای تعریف شده که شامل عنصر دیگری است می‌تواند فقط شامل همان عنصر باشد نه چیز دیگر. یعنی نه متنی دارد و نه عنصر دیگری.
- ◀ ایجاد یک عنصر فرزند اختیاری است و می‌تواند چندین بار در یک سند ظاهر شود. جزئیات بیشتر در در صفحه ۴۸ گفته شده است.
- ◀ یک عنصر فرزند می‌تواند عناصر والد گوناگونی داشته باشد. عناصر چه والد باشند و چه فرزند باید فقط یک بار تعریف شوند.

```
code.dtd
<!ELEMENT endangered_species (animal)>
```

شکل ۶-۳. عنصر endangered\_species از یک عنصر به نام animal تشکیل شده است.

```
code.xml
<endangered_species>
  <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>
    ...
  </animal>
</endangered_species>
```

شکل ۷-۳. عنصر endangered\_species تنها شامل عنصر animal است. عنصر animal نیز بر اساس تعریف خود می‌تواند محتویات دیگری داشته باشد. محتویات عنصر animal به تعریف عنصر endangered\_species بستگی ندارد.

```
code.dtd
<!ELEMENT animal (name, threats, weight, length,
source, picture, subspecies)>
```

**شکل ۸-۳** . عنصر مرتب animal به ترتیب شامل عناصر دیگری است که نام برده شده‌اند. شاید این عنصر فقط شامل همین عناصر باشد.

```
code.html
<endangered_species >
<animal>
<name language="English">Tiger</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>
<source sectionid="101" newspaperid="21"/>
<picture filename="tiger.jpg" x="200" y="197"/>
<subspecies>
<name language="English">Amur or
Siberian</name>
<name language="Latin">P.t. altaica</name>
<region>Far East Russia</region>
<population year="1999">445</population>
</subspecies>
</animal>
</endangered_species>
```

**شکل ۹-۳** . به این مثال معتبر از یک سند XML توجه کنید. هر عنصر فقط یک بار تعریف شده است. عنصر name نمی‌تواند دو بار ظاهر شود و نمی‌توانیم دو عنصر subspecies داشته باشیم. در این باره در صفحه ۴۸ بیشتر توضیح خواهیم داد.

## تعریف عنصری با اجزای مرتب

برخی از عناصر شامل عناصر دیگری هستند که رعایت ترتیب در آنها مهم است. ترتیب عناصر فرزند تشکیل دهنده یک عنصر والد قابل تعریف است.

**برای تعریف عنصری که اجزای آن ترتیب دارند :**

۱. عبارت `<!ELEMENT tag` را تایپ کنید. tag نام عنصری است که قصد تعریف آن را دارید.
۲. عبارت `child1` را تایپ کنید. منظور از `child1` نخستین عنصر فرزند یک عنصر والد است.
۳. عبارت `child2` , را تایپ نمایید. `child2` دومین عنصر فرزند همان عنصر والد می‌باشد. بقیه عناصر فرزند را به ترتیب تایپ و با یک علامت کاما و یک فضای خالی از یکدیگر جدا کنید.
۴. مرحله ۳ را برای بقیه عناصر فرزند تکرار نمایید.
۵. برای تکمیل کار، علامت ( را تایپ کنید.

### نکته‌ها

- ◀ مهم‌ترین نکته در نوشتن عناصر مرتب، تایپ علامت کاما است. کاما کارا کتری است که عناصر مرتب (یا گروهی از عناصر) را از یکدیگر جدا می‌سازد.
- ◀ می‌توان در تمام قسمتهای یک عنصر مرتب از عبارت `#PCDATA` استفاده نکرد.
- ◀ عناصر مرتب از عناصر دیگر تشکیل می‌گردند. در شکل ۹-۳ عنصر `threats` از عناصر جداگانه `threat` تشکیل شده است.
- ◀ امکان ایجاد واحدهای مرتب نیز وجود دارد. هر واحد می‌تواند یک عنصر، گزینه‌ای از عناصر (داخل پراتنز) یا ترتیبی از عناصر (داخل پراتنز) باشد.
- ◀ هر واحد تعریف شده در مجموعه واحدهای مرتب می‌تواند چندین بار ظاهر گردد (صفحه ۴۸).

## تعریف گزینه‌ها [Choice]

عناصر می‌توانند از یک یا چند چیز دیگر تشکیل شوند.

برای تعریف گزینه‌های تشکیل دهنده یک عنصر :

۱- عبارت **ELEMENT tag!** را تایپ کنید. منظور از tag عنصری است که قصد تعریف آن را دارید.

۲- عبارت **child1** را تایپ کنید. **child1** نخستین عنصر فرزند است.

۳- علامت | را تایپ نمایید. بدین ترتیب اعلام می‌کنید که شاید عنصر اول ظاهر شود ولی عنصر دوم ظاهر نگردد و برعکس شاید عنصر اول ظاهر نشود و فقط عنصر دوم ظاهر شود.

۴- عبارت **child2** را تایپ کنید. **child2** دومین عنصر فرزند است. در صورتی که عنصر دیگر دیده نشود این عنصر ظاهر می‌گردد.

۵- اگر گزینه‌های دیگری دارید مراحل ۳ و ۴ را تکرار کنید.

۶- برای تکمیل فهرست گزینه‌ها، ( را تایپ کنید.

۷- به منظور تکمیل اعلان عنصر، > را تایپ نمایید.

### نکته‌ها

◀ پس از مرحله ۶ می‌توانستید با اضافه کردن یک علامت \* به عنصر این امکان را بدهید که شامل هر تعداد از هر گزینه‌ای باشد. این یکی از روشهای تعریف لیست نامرتب است که عنصر والد آن از چندین عنصر تشکیل شده باشد (صفحه ۴۸).

◀ گزینه نخست می‌تواند **#PCDATA** باشد تا بتوان عنصری با محتویات ترکیبی ایجاد کرد؛ اما باید یک علامت \* نیز اضافه نمایید.

◀ می‌توان گزینه‌ها را بین واحدها نیز تعریف کرد. منظور از واحدهای عناصر، گزینه‌های بین عناصر (داخل پرانتز) و عناصر مرتب (داخل پرانتز) می‌باشد.

```
code.dtd
<ELEMENT characteristics ((weight, length) | picture)>
```

شکل ۱۰-۳. در این مثال عنصر characteristics شامل عناصر مرتب weight و length می‌باشد. از طرفی شامل عنصر picture نیز هست در صورت نمایش یکی از عناصر دیگر می‌تواند نمایش داده نشود.

```
code.html
<characteristics>
<weight>500 pounds</weight>
<length>3 yards from nose to tail</length>
</characteristics>
```

```
code.html
<characteristics>
<picture filename="tiger.jpg" x="200" y="197"/>
</characteristics>
```

شکل ۱۱-۳. این دو مثال صحیح و معتبر هستند.

```
code.html
<characteristics>
<weight>500 pounds</weight>
</characteristics>
```

```
code.html
<characteristics>
<weight>500 pounds</weight>
<length>3 yards from nose to tail</length>
<picture filename="tiger.jpg" x="200" y="197"/>
</characteristics>
```

شکل ۱۲-۳. هیچ یک از این مثالها معتبر نیستند. مثال اول نادرست است زیرا گزینه اول عنصر مرتبی است که ابتدا weight و سپس length را شامل می‌شود. در حالی که در اینجا فقط شامل عنصر weight شده است. مثال دوم نیز غیر معتبر است زیرا ممکن است فقط یکی از گزینه‌ها مورد استفاده قرار گیرد نه هر دو گزینه.

## تعریف چند واحد

```
code.dtd
<!ELEMENT animal (name+, threats, weight?,
length?, source, subspecies*)
```

**شکل ۱۳-۳.** نمادهای ویژه، انعطاف پذیری بیشتری به تعاریف می‌دهند. حالا، عنصر animal می‌تواند از یک یا چند عنصر name تشکیل گردد. عناصر weight و length می‌توانند حذف یا فقط یک بار ظاهر شوند و عناصر subspecies یا هیچ‌وقت یا چندین بار تکرار گردند. عناصر threats, source و picture باید حتماً یک بار ظاهر شوند که این حالت پیش فرض می‌باشد.

```
code.dtd
<!ELEMENT threats (threat, threat, threat+)>
```

**شکل ۱۴-۳.** عنصر threats حداقل از سه عنصر threat تشکیل می‌شود. یعنی می‌تواند بیشتر از سه عنصر نیز داشته باشد.

در DTDها سه نماد ویژه وجود دارد که برای تعیین تعداد واحدهای یک عنصر به کار می‌روند. یک واحد می‌تواند یک عنصر، گزینه‌ای بین دو یا چند عنصر (داخل پرانتز) یا یک عنصر مرتب (داخل پرانتز) باشد.

## برای تعریف چند واحد :

۱. در بخش محتویات تعریف عنصر، عبارت **unit** را تایپ کنید. منظور از unit یکی از سه حالت: عنصرمنفرد، گزینه‌ای بین دو یا چند عنصر و یا ترتیبی از عناصر است. دو حالت اخیر باید داخل پرانتز نوشته شوند.

۲. برای اینکه واحد، حداکثر یک بار در عنصر تعریف شده ظاهر گردد علامت ؟ را تایپ کنید.

اگر علامت + را تایپ نمایید، واحد مورد نظر می‌تواند حداقل یک بار در عنصر تعریف شده ظاهر شود. در این حالت تعداد دفعات ظاهر شدن واحد نامحدود و مطابق نظر شما خواهد بود.

با تایپ علامت \* واحد مورد نظر، هم می‌تواند هر چند بار که نیاز داشته باشید ظاهر شود و هم هیچ‌گاه در عنصر تعریف شده ظاهر نگردد.

## نکته‌ها

- ◀ بهتر است تعداد معینی برای واحدها مشخص نکنید. برای مثال نگویید به طور حتم ۳ واحد وجود خواهد داشت. روش پیشرفته‌تر این است که با استفاده از عبارت **(unit, unit, uint+)** حداقل سه واحد در نظر بگیرید.
- ◀ علامت ستاره برای فهرستی از گزینه‌های تشکیل دهنده یک عنصر که در پرانتز نوشته می‌شوند به کار می‌رود و مشخص می‌کند که عنصر مذکور می‌تواند تعداد نامحدودی از گزینه‌های جداگانه با هر ترتیبی داشته باشد.



## ویژگیها

هنگامی که یک عنصر را به قطعه‌های کوچک‌تر اطلاعاتی تقسیم می‌کنید، بهتر است داده مکمل را به خود عنصر اضافه کنید نه به محتویات آن. این همان کاری است که یک ویژگی انجام می‌دهد.

ویژگیها درباره بخشی از محتویات اطلاعات نمی‌دهند. بلکه درباره محتویات صفحه XML اطلاعاتی در اختیار می‌گذارند. برای مثال، بانک اطلاعاتی Endangered Species را در نظر بگیرید. عنصر name شامل ویژگی language است. این ویژگی درباره زبان محتویات عنصر name توضیح می‌دهد.

شما می‌توانید همان اطلاعات را برای عناصر جداگانه به کار برید. یعنی عنصر name شامل یک عنصر language و یک عنصر local\_name باشد. هر دو روش خوبند. عناصر، اطلاعاتی را که می‌خواهید نمایش دهید در اختیار می‌گذارند و ویژگیها درباره آن اطلاعات، اطلاعاتی را ارائه می‌دهند.

ویژگیها عناصر خالی هستند که به محتویات عناصر اشاره می‌کنند.

```
code.xml
<population year="1999">445</population>
```

```
code.xml
<population>
  <year>1999</year>
  <quantity>445</quantity>
</population>
```

**شکل ۳-۱۵.** در این دو مثال کوچک XML اطلاعات مشابهی دیده می‌شوند. یعنی هر دو مثال می‌گویند که در سال ۱۹۹۹، ۴۴۵ ببر سیبری در حیات وحش وجود داشته‌اند. تفاوت این دو مثال در چگونگی سازماندهی اطلاعات است. در مثال بالا، ۱۹۹۹ یک مقدار ویژگی است. در مثال پایین ۱۹۹۹ و ۴۴۵ محتویات عناصر جداگانه هستند. هر دو روش خوبند. انتخاب با شماست.

## تعریف ویژگیهای ساده

در اسناد XML هنگامی ویژگیها ظاهر می‌شوند که در یک DTD تعریف شده باشند.

برای تعریف یک ویژگی :

- ۱- عبارت **tag** **!ATTLIST** را تایپ کنید. منظور از tag نام عنصری است که به صورت ویژگی ظاهر خواهد شد.
- ۲- واژه **attribute** را تایپ نمایید. attribute نامی است که اطلاعات بیشتری درباره tag در اختیار می‌گذارد.
- ۳- عبارت **CDATA** را تایپ کنید. توجه داشته باشید در این حالت اثری از پرانتز و علامتهای (**#P!**) نیست. این در صورتی است که بخواهید مقدار ویژگی را با ترکیب کاراکترها و بدون برچسب بسازید.

اگر می‌خواهید چند گزینه برای ویژگی بسازید که احتمال می‌رود فقط یکی از آنها در سند XML به کار روند، عبارت (**choice\_1 | choice\_2**) را تایپ کنید. هر گزینه با علامت | از گزینه قبلی جدا می‌شود و تمام گزینه‌ها باید در پرانتز قرار گیرند.

- ۴- سپس عبارت **"default"** را تایپ نمایید. اگر مقدار ویژگی در مجموعه مشخص نیست مقدار **default** برای آن در نظر گرفته می‌شود.

چنانچه مقدار **default** مقدار پیش فرض است و می‌خواهید آن را برای ویژگی مذکور در نظر بگیرید عبارت **"default" #FLXED** را تایپ کنید. اگر می‌خواهید ویژگی شامل مقادیر گوناگون باشد و فقط از مقدار پیش فرض تشکیل نشود عبارت **#REQUIRED** و اگر می‌خواهید ویژگی، مقدار پیش فرضی نداشته باشد و هر زمان که خواستید آن را حذف نمایید عبارت **#IMPLIED** را تایپ کنید.

- ۵- مراحل ۲ تا ۴ را برای تمام ویژگیهای عناصر تشکیل دهنده تکرار نمایید.

- ۶- با تایپ علامت **>** تعریف ویژگی را کامل کنید.

```
code.dtd
<!ELEMENT population (#PCDATA)>
<!ATTLIST population year CDATA #IMPLIED>
```

**شکل ۱۶-۳** . عنصر **population** شامل ویژگی اختیاری **year** است (به دلیل عبارت **#IMPLIED**). به دلیل **CDATA**. محتویات این ویژگی می‌تواند ترکیبی از کاراکترهای گوناگون باشد.

```
code.xml
<population>445</population>
```

```
code.xml
<population year="1999">445</population>
```

```
code.xml
<population year="of the
Rabbit">445</population>
```

**شکل ۱۷-۳** . براساس DTD شکل ۱۶-۳ هر سه مثال بالا معتبر هستند. ویژگی اختیاری **year** (به دلیل عبارت **#IMPLIED**) و محتویات آن که می‌توانند ترکیبی از کاراکترهای گوناگون باشند. توجه داشته باشید راهی برای اطمینان از درستی مقدار ویژگی **year** (سال) وجود ندارد. بنابراین باید از XML Schema استفاده نمایید (صفحه ۶۹).

```
code.dtd
<!ELEMENT population (#PCDATA)>
<!ATTLIST population year (1999 | 2000)
#REQUIRED>
```

**شکل ۱۸-۳** . در این مثال فقط امکان وجود دو مقدار برای ویژگی **population** در عنصر **year** فراهم شده که عبارتند از ۱۹۹۹ یا ۲۰۰۰. فهرست گزینه‌ها در بین پرانتزها قرار گرفته و با علامت | از یکدیگر جدا شده‌اند. به دلیل وجود **#REQUIRED** ویژگی مذکور باید محتویاتی داشته باشد.

```
code.xml
<population year="1999">445</population>
```

```
code.xml
<population>445</population>
```

```
code.xml
<population year="1998">445</population>
```

**شکل ۱۹-۳** . مثال اول براساس DTD شکل ۱۸-۳ معتبر و مثال دوم به دلیل اینکه ویژگی **year** عبارت **#REQUIRED** را در نظر نگرفته غیرمعتبر می‌باشد. مثال آخر نیز به دلیل اینکه ۱۹۹۸ از گزینه‌های مجاز محتویات ویژگی نیست غیرمعتبر است.

نکته‌ها

- ◀ هر گزینه در فهرست گزینه‌ها باید از قوانین اسامی معتبر در XML پیروی کند (صفحه ۲۶).
- ◀ می‌توان تمام ویژگیها را در قسمت اعلان یک ویژگی تعریف نمود (مرحله ۵). همچنین می‌توان هر ویژگی را به صورت جداگانه تعریف کرد.
- ◀ چند نوع ویژگی مخصوص وجود دارند که در صفحات ۵۲-۵۳ توضیح داده می‌شوند و عبارتند از: IDREF, ID و IDREFS. درباره ویژگیهای مخصوص NMTOKEN و NMTOKENS در صفحه ۵۴ و جزئیات ورودی خروجی ویژگی ENTITY در فصل ۴ توضیح خواهیم داد.
- ◀ چنانچه یک ویژگی را با یک مقدار پیش فرض تعریف کنید، تجزیه‌گر XML به صورت خودکار مقدار پیش فرض را اضافه خواهد کرد. البته این موضوع در صورتی عملی می‌گردد که ویژگی مذکور به طور صریح در سند XML ذکر نشده باشد (شکل ۲۱-۳).
- ◀ اگر ویژگی را به‌همراه عبارت "#FIXED \"default\"" تعریف کرده باشید، مقدار ویژگی در سند XML باید با مقدار پیش فرض یکسان باشند. اگر چنین نباشد، تجزیه‌گر به صورت خودکار مقدار ویژگی را با مقدار پیش فرض تنظیم می‌نماید (شکل ۲۳-۳).
- ◀ اگر ویژگی DTD با عبارت "#REQUIRED" تعریف شود ولی در سند XML برای ویژگی مقداری در نظر گرفته نشود، تجزیه‌گر اعلام خطا خواهد کرد.
- ◀ تجزیه‌گر، اطلاعات بازگشتی ویژگیهایی را که با عبارت "#IMPLIED" تعریف شده ولی با سند XML تنظیم نشده باشند پیش بینی می‌کند.
- ◀ توجه داشته‌باشید تمام بخشهای تعریف ویژگی نسبت به حروف کوچک و بزرگ حساس می‌باشند. اگر عبارتی مثل "#Required" تایپ کنید با آنچه که در DTD گفته شده فرق خواهد داشت.
- ◀ برای مقادیر پیش فرض نمی‌توان به‌صورت هم‌زمان از عبارتهای "#REQUIRED" و "#IMPLIED" استفاده کرد.

```
code.dtd
<!ELEMENT population (#PCDATA)>
<!ATTLIST population year CDATA "1999">
```

شکل ۲۰-۳. در اینجا مقدار پیش فرض ۱۹۹۹ را به ویژگی year اضافه کرده‌ایم.

```
code.xml
<population year="1999">445</population>
```

```
code.xml
<population year="1998">445</population>
```

```
code.xml
<population>445</population>
```

شکل ۲۱-۳. این سه مثال معتبر می‌باشند. ویژگی year هر مقداری می‌تواند داشته باشد و حتی می‌تواند حذف گردد. نکته جالب این است که اگر مقداری حذف شود، مانند مثال سوم، در صورت لزوم تجزیه‌گر مقدار ۱۹۹۹ را برای ویژگی year در نظر خواهد گرفت.

```
code.dtd
<!ELEMENT population (#PCDATA)>
<!ATTLIST population year CDATA #FIXED "1999">
```

شکل ۲۲-۳. یک مقدار ثابت می‌تواند برای اطمینان از اینکه یک ویژگی مقدار دهی شده یا نشده مفید واقع شود.

```
code.xml
<population year="1999">445</population>
```

```
code.xml
<population year="1998">445</population>
```

```
code.xml
<population>445</population>
```

شکل ۲۳-۳. این مثالها مانند مثالهای شکل ۲۱-۳ می‌باشند. در صورتی که اعتبار آنها با DTD شکل ۲۲-۳ سنجیده شود مثال دوم معتبر نخواهد بود. باید مقدار ویژگی ۱۹۹۹ باشد. نه ۱۹۹۸ و نه هیچ کاراکتر دیگری مجاز نیستند. به مثال آخر توجه کنید، تجزیه‌گر مقدار ۱۹۹۹ را به ویژگی year نسبت می‌دهد.

```
code.dtd
<!ELEMENT animal (name+, threats, weight?,
length?, source, picture, subspecies+)>
<!ATTLIST animal code ID #REQUIRED>
```

**شکل ۲۴-۳.** به منظور مشخص کردن عناصر معین در سند XML خود بهتر است ویژگی ID ایجاد کنید.

```
code.xml
<animal code="T143">
<name language="English">Tiger</name>
...
</animal>
<animal code="BR45">
<name language="English">Black Rhino</name>
...
</animal>
```

```
code.xml
<animal code="T143">
<name language="English">Tiger</name>
...
</animal>
<animal code="T143">
<name language="English">Black Rhino</name>
...
</animal>
```

**شکل ۲۵-۳.** با توجه به شکل ۲۴-۳ ویژگی code باید در طول سند XML یک مقدار منحصر به فرد داشته باشد؛ بنابراین مثال اول معتبر ولی مثال دوم غیر معتبر است.

## تعریف ویژگی‌ها با مقادیر منحصر به فرد

در XML چند نوع ویژگی وجود دارد که به برخی از آنها ویژگی مخصوص می‌گویند. برای مثال ویژگی ID نیز یک ویژگی مخصوص است و برای داشتن مقدار منحصر به فرد تعریف شده است و نمی‌تواند در یک سند XML تکرار گردد. از ویژگی ID برای کلیدها و اطلاعات شناسایی مثل کد کالا، کد شناسه مشتری و غیره استفاده می‌شود.

### برای تعریف ویژگی‌های ID:

- ۱- برای آغاز تعریف ویژگی، مراحل ۱ تا ۲ صفحه ۵۰ را انجام دهید.
- ۲- اگر می‌خواهید مقدار منحصر به فردی داشته باشید که در یک سند XML قابل تکرار شدن نباشد عبارت ID را تایپ کنید.
- ۳- همان‌گونه که در مراحل ۴ تا ۶ صفحه ۵۰ گفته شد تعریف ویژگی را کامل نمایید.

### نکته‌ها

- ◀ اگر دو عنصر در یک سند مقادیر ویژگی ID یکسانی داشته باشند، آن سند معتبر نخواهد بود.
- ◀ مقدار یک ویژگی ID تابع قوانین نامگذاری XML است. یعنی باید با یک حرف یا علامت - آغاز شود و در ادامه حروف، اعداد، ، ، و - به کار رود. در مواردی مثل شماره تلفن و شماره شناسایی ملی می‌توان در ابتدای مقادیر این نوع ویژگی‌ها یک حرف یا علامت - قرار داد.
- ◀ کاربرد ID به عنوان یک نوع ویژگی مثل کاربرد NMTOKEN است که باید منحصر به فرد باشد (به صفحه ۵۴ مراجعه کنید).

## ویژگیهایی با مقادیر منحصر به فرد

هرگاه مقدار یک ویژگی به یکی از ویژگیهای ID که در صفحه تعریف کرده‌ایم نسبت داده شود به آن ویژگی IDREF می‌گویند. یک ویژگی IDREFS شامل فهرست جداگانه‌ای از مقادیر ویژگیهای ID سند می‌باشد. برای نسبت دادن ویژگیها با مقادیر منحصر به فرد:

- ۱- به منظور آغاز تعریف ویژگی مراحل ۱ تا ۲ صفحه ۵۰ را اجرا نمایید.
- ۲- با تایپ IDREF، یک ویژگی که مقدار آن با مقدار ویژگی ID دیگری مطابقت دارد تعریف می‌شود. تایپ IDREFS برای ویژگیهایی به کار می‌رود که می‌توانند از چند مقدار ID جداگانه تشکیل شوند.
- ۳- تعریف ویژگی را مطابق مراحل ۴ تا ۶ صفحه ۵۰ کامل کنید.

### نکته‌ها

- ◀ تعریف ویژگیهای نوع IDREF تنها زمانی کاربرد دارد که DTD نیز دارای تعریفی برای ویژگیهای نوع ID که به IDREF اشاره می‌کند باشد.
- ◀ تعریف ویژگیهای نوع ID که ویژگیهای نوع IDREF به آنها اشاره می‌کنند باید در DTD نیز وجود داشته باشد.
- ◀ شاید چند ویژگی IDREF به یک ID نسبت داده شوند (شکل ۲۷-۳). این امر مشکلی به وجود نمی‌آورد فقط باید دقت کنید ID، منحصر به یک عنصر باشد.
- ◀ هیچ راهی برای جلوگیری از تکرار اجزای ویژگی نوع IDREFS وجود ندارد. برای مثال عبارت = contents "T143T143T143" از نظر تجزیه‌گر کاملاً معتبر است و این اعتبار به نظر شما بستگی ندارد. برای کنترل بیشتر محتویات عناصر و ویژگیها باید از DTDهای XML Schema صرف‌نظر کرد (صفحه ۶۷).

```
code.dtd
<!ELEMENT specialized_website (title, url)>
<!ATTLIST specialized_website animal_focus IDREF
#REQUIRED>
```

**شکل ۲۶-۳.** فرض کنید عنصری در یک سند XML مسیر سایت‌های وب متعلق به یک حیوان معین را نگه می‌دارد؛ برای نسبت دادن سایت‌های وب به کد حیوان مورد نظر از IDREF استفاده می‌شود.

```
code.xml
<resources>
<specialized_website animal_focus="T143">
<title>Tigers in Crisis</title>
<url>http://www.tigersin crisis.com/</url>
</specialized_website>
<specialized_website animal_focus="T143">
<title>Tigers!</title>
<url>http://www.geocities.com/RainForest/6
612/</url>
</specialized_website>
<specialized_website animal_focus="O735">
<title>International Otter Survival Fund</title>
<url>http://www.otter.org/</url>
</specialized_website>
```

**شکل ۲۷-۳.** مقدار ویژگی IDREF باید شامل یک نوع ویژگی ID از سند باشد (برای مثال شکل اول ۲۵-۳ را مشاهده کنید). همان‌طور که می‌بینید بیش از یک عنصر Specialized\_website وجود دارد که می‌تواند همان مقدار ویژگی animal\_focus را داشته باشد.

```
code.dtd
<!ELEMENT general_website (title, url)>
<!ATTLIST general_website contents IDREFS
#REQUIRED>
```

**شکل ۲۸-۳.** ویژگیهای IDREFS از یک سری مقادیر ID تشکیل شده‌اند.

```
code.xml
<general_website contents="T143 O735 BR45">
<title>World Wildlife Fund</title>
<url>http://www.worldwildlife.org/</url>
</general_website>
```

**شکل ۲۹-۳.** یک ویژگی از نوع IDREFS باید شامل یک یا چند مقدار فضای خالی باشد که هر کدام در جایی از سند از ویژگیهای نوع ID موجود تشکیل شده‌اند.

## محدود کردن ویژگیها به نامهای معتبر XML

```
code.dtd
<!ELEMENT appearance (#PCDATA)>
<!ATTLIST appearance most_prominent_color
    NMTOKEN #IMPLIED>
```

**شکل ۳۰-۳.** توضیحات مربوط به حیوانات به صورت خلاصه و ساده با عبارت `most_prominent_color` مقدار دهی شده است. چون مقدار ویژگی یک واژه است (بین حروف آن فضای خالی وجود ندارد) می‌توان آن را به نوع `nmtoken` نسبت داد.

```
code.xml
<appearance most_prominent_color="orange">
The might tiger is typically orange with black stripes
and some white highlights.
</appearance>
```

```
code.xml
<appearance most_prominent_color="dark orange">
The might tiger is typically dark orange with black
stripes and some white highlights.
</appearance>
```

**کل ۳۱-۳.** تنها مثال اول معتبر است. دلیل غیر معتبر بودن مثال دوم این است که در مقدار ویژگی `most_prominent_color` فضای خالی به کار رفته است. در ویژگیهای نوع `NMTOKEN` استفاده از فضای خالی مجاز نیست.

در DTD ها می‌توان محدودیتهایی بر ویژگیها اعمال نمود تا یک نام معتبر XML باشد. یعنی با یک حرف یا علامت - شروع شده و بقیه کاراکترهای آن شامل حروف، اعداد، علامتهای - ، - و نقطه گردد.

برای اینکه مقادیر ویژگیها تابع قوانین نامگذاری در XML باشند :

۱. مراحل ۱ تا ۲ صفحه ۵۰ را برای تعریف ویژگی اجرا کنید.
۲. اگر می‌خواهید مقدار ویژگی یک نام معتبر در XML باشد (مانند آنچه که در صفحه ۲۶ گفته شد) عبارت `NMTOKEN` را تایپ نمایید.
۳. اگر قرار است مقدار ویژگی فهرستی از اسامی XML باشد عبارت `NMTOKENS` را تایپ کنید.
۴. تعریف ویژگی را مانند توضیحات مراحل ۴ تا ۶ صفحه ۵۰ به پایان برسانید.

### نکته‌ها

- ◀ در ویژگیهای نوع `NMTOKEN` استفاده از فضای خالی مجاز نیست.
- ◀ اگر می‌خواهید مقدار یک ویژگی علاوه بر اینکه تابع اسامی معتبر XML است در سراسر سند XML منحصر به فرد نیز باشد به جای عبارت `NMTOKEN` از عبارت `ID` استفاده نمایید (به صفحه ۵۲ نگاه کنید).