

## نوشتن XML



XML سیستم گرامری ایجاد زبانهای علامت‌گذاری دلخواه است. یعنی می‌توان آن را برای انواع داده‌ها از قبیل داده‌های ریاضی، شیمی، تجاری و غیره به کار برد.

برای اینکه بتوان به کمک XML زبان دلخواهی ایجاد کرد باید اصول گرامر XML را دانست. به همین دلیل، ما نیز بحث خود را از این نقطه شروع می‌کنیم. ابتدا اصول قوانین نوشتن اسناد XML و سپس ایجاد زبان در XML را فراخواهید گرفت. در اصطلاح به هر زبانی که با XML ساخته می‌شود *XML application* می‌گویند که منظور از آن کاربرد XML است نه برنامه‌های کاربردی. چون این اصطلاح ابهام برانگیز است و برنامه‌های کاربردی مثل فتوشاپ را تداعی می‌نماید، در این کتاب از آن استفاده نمی‌شود.

### ابزارهای نوشتن XML

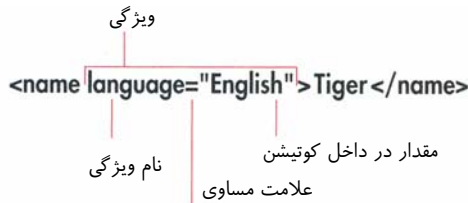
برای نوشتن XML نیز مانند HTML می‌توان از یک ویرایشگر یا واژه‌پرداز استفاده کرد. این گونه برنامه‌ها عبارتند از: TeachText یا SimpleText در مکینتاش و Notepad یا WordPad در ویندوز. ویرایشگرهای متن ویژه‌ای وجود دارند که می‌توانند هم‌زمان با نوشتن XML درستی آن را بررسی نمایند. همچنین برنامه‌هایی هستند که می‌توانند فایل‌های برنامه‌های دیگر مانند برنامه‌های صفحه‌آرایی، صفحه‌گسترده، بانکهای اطلاعاتی و ... را به فرمت اسناد XML تبدیل کنند.

اگر روش ایجاد، بازکردن، ویرایش و ذخیره فایلها یا اسناد را می‌دانید، برای شروع ذکر یک نکته باقی می‌ماند، آن هم ذخیره اسناد XML با پسوند xml است.

## عناصر، ویژگیها و مقادیر



**شکل ۱-۱.** عنصری با برچسب‌های شروع و پایان که محتویات آن متنی است، نام آن `name` و محتویات آن `Tiger` می‌باشد.



**شکل ۱-۲.** عنصر `name` دارای یک ویژگی با نام `Language` است که مقدار آن `English` می‌باشد. توجه داشته باشید که واژه `English` بخشی از محتویات عنصر `name` نیست. ویژگی درباره محتویات توضیح می‌دهد و `English Tiger` یا `English Tiger` نام عنصر محسوب نمی‌گردد.

```
<animal>
  <name language="English">Tiger</name>
  <name language="Latin">Panthera
    tigris</name>
  <weight>500 pounds</weight>
</animal>
```

برچسب پایان  
محتویات

**شکل ۱-۳.** عنصر `animal` از سه عنصر دیگر تشکیل شده است که عبارتند از دو عنصر به نام `name` و یک عنصر به نام `weight`. عنصر `animal` متنی ندارد ولی محتویات عناصر `name` و `weight` شامل متن هستند. از طرفی عناصر `name` و `weight` از عنصر دیگری تشکیل نشده‌اند. همچنین برای خواناتر شدن کد از فضای خالی استفاده شده است.

XML نیز مانند HTML از سه جزء اصلی عناصر، ویژگیها و مقادیر تشکیل می‌شود. عنصر، اصلی‌ترین قسمت سند XML محسوب می‌گردد و هر چیزی می‌تواند باشد؛ از جمله، عناصر دیگر و یا متن. هر عنصر یک برچسب شروع و یک برچسب پایان دارد. برچسب شروع نامی دارد که بین علامتهای `<` و `>` قرار می‌گیرد و گاهی این نام با صفتیایی همراه است مانند **شکل ۱-۱.** نامی که در نظر می‌گیرید باید هدف عنصر و در مواردی محتویات آن را در صورت وجود معرفی نماید. محتویات عنصر بلافاصله پس از برچسب شروع آورده می‌شوند. برچسب پایان نیز نامی دارد، همان نام برچسب شروع، که در ابتدای آن علامت `/` قرار گرفته و در بین علامتهای `<` و `>` محصور می‌گردد.

ویژگیها که در برچسب شروع آورده می‌شوند بین علامت کوتیشن قرار می‌گیرند و مقادیری را تعیین می‌کنند که درباره هدف و محتویات (در صورت وجود) عنصر خاصی توضیح بیشتری ارائه می‌دهند (**شکل ۱-۲**). یک ویژگی، اطلاعاتی درباره داده‌های موجود در سند XML می‌دهد که این اطلاعات با خود داده فرق دارد و در اصطلاح به آنها *فرآوده* می‌گویند. یک عنصر می‌تواند در صورت نیاز ویژگیهای زیادی داشته باشد؛ ولی هر کدام از ویژگیها باید نام منحصر به فردی داشته باشند. در این درس با طرز نوشتن عناصر، ویژگیها و مقادیر آشنا می‌شویم.

### فضای خالی

برای خواناتر شدن و ویرایش آسان‌تر عناصر می‌توان در اطراف آنها فضای خالی ایجاد کرد (**شکل ۱-۳**). تجزیه‌گر، فضای خالی را در نظر نمی‌گیرد و برنامه‌های مرورگر وب مانند IE5 و Mozilla (نگارش بتای Netscape6) از آن صرف‌نظر می‌کنند (همان‌گونه که با HTML رفتار می‌کنند).

## قوانین نگارش در XML

ساختار زبان XML اصول و قوانین قابل فهمی دارد که کارآیی آن را افزایش می‌دهد و استفاده از آن را بسیار آسان می‌سازد. در این قسمت به شرح مهم‌ترین اجزای قوانین این زبان می‌پردازیم. اگر در فایل‌های خود از این قوانین پیروی کنید به برنامه شما یک برنامه خوش‌فرم می‌گویید. یک برنامه خوش‌فرم می‌تواند توسط مرورگرها به نمایش درآید.

### عنصر ریشه (Root element)

هر سند یا فایل XML باید یک عنصر ریشه داشته باشد که تمام عناصر سند را شامل گردد. تنها توضیحات و دستورات پردازشی می‌توانند خارج از عنصر ریشه باشند (شکل ۴-۱).

### برچسب پایان

هر عنصری باید یک برچسب پایان داشته باشد. برای برچسب شروع و پایان می‌توان از یک برچسب استفاده کرد به گونه‌ای که پیش از > یک / قرار بگیرد (شکل ۵-۱). به این قبیل برچسبها، برچسبهای خالی می‌گویند. همچنین می‌توان از یک برچسب پایان جداگانه استفاده نمود.

### عناصر تو در تو

اگر می‌خواهید عنصر A در ابتدا و عنصر B پس از آن قرار گیرد ابتدا عنصر B و سپس عنصر A را ببندید (شکل ۵-۱).

### حروف کوچک و بزرگ

XML بین حروف کوچک و بزرگ فرق می‌گذارد. یعنی عناصر animal, ANIMAL, و Animal با یکدیگر متفاوتند و می‌توانند ارتباطی با یکدیگر نداشته باشند (شکل ۶-۱).

### مقادیر باید داخل کوتیشن قرار گیرند

مقادیر ویژگیها را باید همیشه بین کوتیشن قرار داد. در این مورد ` و " هر دو صحیح می‌باشند (شکل ۷-۱).

### موجودیتها باید تعریف شوند

برخلاف HTML در XML به غیر از ۵ موجودیت داخلی (صفحه ۳۱) از هر موجودیتی می‌توان استفاده کرد. پیش از به‌کارگیری موجودیتها باید آنها را در یک DTD تعریف کرد.

```
code.xml
<?xml version="1.0" ?>
<endangered_species>
<name>Tiger</name>
</endangered_species>
```

**شکل ۴-۱.** در یک سند خوش‌فرم باید یک عنصر (مانند endangered\_species) وجود داشته باشد که تمام عناصر را شامل شود. خط اول یک دستور پردازشی است و می‌تواند خارج از ریشه قرار گیرد.

```
code.xml
<?xml version="1.0" ?>
<endangered_species>
<name>Tiger</name>
<picture filename="tiger.jpg" />
</endangered_species>
```

**شکل ۵-۱.** هر عنصر باید قسمت پایان داشته باشد. در برچسب خالی از یک برچسب برای مشخص نمودن ابتدا و انتها استفاده می‌کنند به گونه‌ای که پیش از علامت > یک علامت / قرار می‌گیرد. همان طور که مشاهده می‌کنید از عناصر تو در تو نیز استفاده شده است.

```
code.xml
<name>Tiger</name>
<Name>Tiger</Name>
```

```
code.xml
<name>Tiger</Name>
```

**شکل ۶-۱.** مثال اول درست است و از قوانین پیروی می‌کند. این دو عنصر هیچ نوع وابستگی به یکدیگر ندارند. مثال دوم غلط است زیرا برچسب شروع و پایان آن یکسان نمی‌باشند.

```
code.xml
<picture filename="tiger.jpg"/>
```

**شکل ۷-۱.** وجود علامت کوتیشن الزامی است. بدین منظور می‌توان از " یا ` استفاده نمود به شرطی که کوتیشن اول و دوم یکسان باشند.

## اعلان نگارش XML

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

به طور کلی، هر سند XML با اعلان نگارش XML مورد استفاده شروع می‌گردد. به این خط از برنامه اعلان XML می‌گویند.

### برای اعلان نگارش XML:

- ۱- در ابتدای سند پیش از هر چیزی عبارت `<?xml` را تایپ کنید.
- ۲- سپس عبارت `version="1.0"` را که در حال حاضر تنها نگارش XML است تایپ نمایید.
- ۳- برای تکمیل اعلان نگارش، علامتهای `>` را تایپ کنید.

### نکته‌ها

- ◀ به برجسبهایی که با علامتهای `<?` شروع و با `>` پایان می‌پذیرند دستورات پردازشی می‌گویند. علاوه بر اعلان نگارش XML، برای تعیین صفحه‌سبک نیز از دستورات پردازشی استفاده می‌شود. درباره صفحه‌های سبک در صفحه ۱۷۵ توضیح خواهیم داد.
- ◀ عدد مربوط به نگارش برنامه، باید بین علامتهای کوتیشن (" یا ') قرار گیرد.
- ◀ نوشتن اعلان XML اختیاری است. اگر بخواهید آن را در سند خود بیاورید باید در اولین خط نوشته شود.
- ◀ اگر سند شما به سند دیگری وابسته است باید این موضوع در برنامه قید شود (صفحه‌های ۳۹-۴۰).
- ◀ شاید برای تعیین کدگذاری کاراکتری، به دستور پردازشی اولیه XML مثل UTF-8 یا UTF-16 نیز نیاز داشته باشید.

**شکل ۸-۱.** به دلیل اینکه خط اعلان XML یک دستور پردازشی محسوب می‌گردد و یک عنصر نمی‌باشد، نیازی به برجسب پایان ندارد.

## ایجاد عنصر ریشه

هر سند XML از عنصری تشکیل شده که شامل تمام عناصر سند می‌باشد. از آنجا که این عنصر در برگزیده تمام عناصر سند می‌باشد به آن عنصر ریشه می‌گویند.

### برای ایجاد عنصر ریشه:

- ۱- در ابتدای سند XML عبارت `<root>` را تایپ کنید. منظور از root نام عنصری است که بقیه عناصر سند را در برمی‌گیرد.
- ۲- برای اجزای دیگر برنامه (که در طول این کتاب نوشتن آن را یاد می‌گیرید) چند خط خالی ایجاد نمایید.
- ۳- عبارت `</root>` را تایپ کنید. این نام با نامی که در مرحله نخست تایپ کرده‌اید مطابقت کامل دارد.

### نکته‌ها

- ◀ حروف کوچک و بزرگ از یکدیگر متمایز هستند. یعنی `<NAME>` با `<Name>` و `<name>` فرق دارد.
- ◀ نام مجاز برای عناصر و ویژگیها با یک حرف، علامت \_ یا : شروع شده و پس از آن می‌توان از اعداد، حروف ، \_ ، - ، . و : استفاده کرد.
- ◀ توجه داشته باشید که استفاده از علامت : فقط برای اسامی مجاز است (صفحه ۱۱۳). همچنین هر نوع ترکیبی از حروف x ، m ، l چه به صورت کوچک و چه به صورت بزرگ توسط W3C رزرو شده است.
- ◀ عنصر ریشه باید برچسب پایان داشته باشد.
- ◀ خارج از برچسبهای شروع و پایان عنصر ریشه، عناصر دیگری نمی‌توان معرفی نمود. تنها دستورات پردازشی و الگوها را می‌توان پیش از برچسب شروع ریشه قرار داد (صفحه‌های ۲۴ و ۶۷).

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

**شکل ۹-۱.** در HTML همیشه نام عنصر ریشه HTML است؛ ولی در XML هر نام مجازی می‌تواند به عنوان نام عنصر ریشه در نظر گرفته شود. مثل `endangered_species` که در مثال بالا نشان داده شده است. پیش از برچسب شروع و پس از برچسب پایان عنصر ریشه نمی‌توان عنصر دیگری را قرار داد.



علامتهای مشخص کننده برچسب

**شکل ۱-۱۰.** اجزای تشکیل دهنده یک عنصر ساده در XML عبارتند از: برچسب شروع، محتویات (که می‌تواند متن یا عناصر دیگر و یا حتی خالی باشد) و برچسب پایان. تنها تفاوت برچسب پایان با برچسب شروع وجود علامت / پیش از نام عنصر در برچسب پایان است.

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

**شکل ۱-۱۱.** در اسناد XML تمام عناصر باید بین برچسبهای شروع و پایان عنصر ریشه قرار بگیرند.

## نوشتن عناصر غیر خالی

شما می‌توانید هر عنصری را که می‌خواهید در یک سند XML ایجاد نمایید. بهتر است از اسامی و نامهایی استفاده کنید که مشخص کننده محتویات آنها باشند تا در آینده پردازش اطلاعات آسانتر و قابل فهم‌تر گردد.

### برای نوشتن عنصر غیر خالی:

- ۱- عبارت `<name>` را تایپ کنید. نام `name` معرف محتویات در برگیرنده این عنصر می‌باشد.
- ۲- محتویات عنصر را ایجاد نمایید.
- ۳- عبارت `</name>` را تایپ کنید. این نام باید همان نامی باشد که در مرحله نخست معرفی کرده‌اید.

نکته‌ها

- ◀ وجود برچسب پایان الزامی است (در حالی که در HTML گاهی وجود برچسب پایان اختیاری است).
- ◀ قوانین نامگذاری عناصر مانند قوانینی است که برای عنصر ریشه به کار می‌روند. همچنین علامت : بیشتر برای تعیین فضای نام به کار می‌رود و در شروع اسامی نمی‌توان از هیچ ترکیبی از حروف x, m و a (چه کوچک، چه بزرگ) استفاده نمود. زیرا این ترکیب توسط W3C رزرو شده است.
- ◀ برای نامگذاری استفاده از حروف انگلیسی یا لاتین الزامی نیست.
- ◀ اطلاعات لازم برای نوشتن ویژگیها و مقادیر آنها در صفحه ۲۸ توضیح داده شده است.
- ◀ با استفاده از یک الگو می‌توان برچسبهای مجاز سند XML را مشخص کرد. برای کسب اطلاعات بیشتر درباره الگوها به صفحه ۶۷ مراجعه نمایید.
- ◀ اگر برای عناصر خود نامهای معنی‌دار و متناسب انتخاب کنید استفاده از داده‌ها آسان‌تر می‌گردد.

## عناصر تو در تو

ترکیب درست

```
<name>Tiger <baby> cub</baby></name>
```

```
<name>Tiger <baby> cub</name></baby>
```

گاهی یک داده بزرگ به قسمتهای کوچکتر تقسیم می‌گردد؛ بنابراین باید بتوانید هر یک از بخشهای داده را تعریف و با هر یک از آنها کار کنید.

### برای ایجاد عناصر تو در تو:

ترکیب نادرست

**شکل ۱۲-۱** . برای اینکه از درستی برجسبهای تو در تو مطمئن شوید هر گروه را در یک خط جداگانه قرار دهید. هیچ یک از مجموعه برجسبها نباید با یکدیگر تلافی پیدا کنند. هر مجموعه داخلی باید کاملاً توسط مجموعه بزرگتر خارجی احاطه شود.

- ۱- همان‌طور که در مرحله ۱ صفحه ۲۶ گفته شد برجسب شروع خارجی‌ترین عنصر را ایجاد نمایید.
- ۲- عبارت `<inner>` را تایپ کنید. منظور از inner نام اولین بخش از داده اصلی است.
- ۳- اگر برای عنصر inner محتویاتی در نظر گرفته‌اید، در قسمت مربوطه ایجاد نمایید.
- ۴- عبارت `</inner>` را تایپ کنید. این نام (inner) باید مطابق نامی باشد که در مرحله ۲ تعیین نموده‌اید.
- ۵- در صورت تمایل مراحل ۲ تا ۴ را با نامی دیگر تکرار کنید.
- ۶- برجسب پایان خارجی‌ترین عنصر را مطابق مرحله ۳ صفحه ۲۶ ایجاد نمایید.

### نکته‌ها

- ◀ این موضوع که هر عنصر داخلی باید توسط عناصر اصلی خارجی احاطه گردد بسیار مهم است. به عبارت دیگر، برجسب پایان عنصر خارجی تا وقتی که عنصر داخلی بسته نشده است نباید نوشته شود. در غیر این صورت، سند ایجاد شده یک سند خوش‌فرم نخواهد بود.
- ◀ تعداد عناصر تو در تو نامحدود است و به نظر شما بستگی دارد.
- ◀ گاهی به عنصر داخلی، عنصر فرزند و به عنصر خارجی، عنصر والد می‌گویند.

```
code.html
<endangered_species>
<animal>
<name>Tiger</name>
<threat>poachers</threat>
<weight>500 pounds</weight>
</animal>
</endangered_species>
```

**شکل ۱۳-۱** . عنصر animal از سه عنصر دیگر تشکیل شده است که هر یک از این عناصر شامل اطلاعاتی هستند که می‌توانیم به آنها دسترسی داشته باشیم و از آنها استفاده کنیم.

## اضافه کردن ویژگیها



**شکل ۱۴-۱.** ویژگیها در قسمت برچسب شروع معرفی می‌گردند و از دو بخش تشکیل می‌شوند. مقدار ویژگیها باید بین علائم " یا ` ` محصور گردند.

```
code.xml
<endangered_species>
<animal>
<name language="English">Tiger</name>
<name language="Latin">panthera tigris</name>
<threat>poachers</threat>
<weight>500 pounds</weight>
</animal>
</endangered_species>
```

**شکل ۱۵-۱.** ویژگیها امکان اضافه کردن اطلاعاتی درباره محتویات یک عنصر را فراهم می‌کنند.

بدون افزودن متن به عناصر می‌توان از طریق ویژگیها، اطلاعات اضافه‌ای ایجاد کرد.

### برای اضافه کردن ویژگیها:

- ۱- پیش از علامت > در برچسب شروع عبارت `attribute=` را تایپ کنید. واژه `attribute` امکان ایجاد داده‌های اضافی را فراهم می‌نماید.
- ۲- سپس عبارت `"value"` را تایپ کنید. منظور از `value` همان اطلاعات اضافی می‌باشد. وجود علامت کوتیشن الزامی است.

### نکته‌ها

- ◀ ویژگیها نیز از قوانین نامگذاری عناصر پیروی می‌کنند (به صفحه ۲۶ مراجعه نمایید).
- ◀ بر خلاف HTML در XML مقادیر ویژگیها را باید داخل کوتیشن قرار داد؛ " یا ` ` فرق نمی‌کند ولی برای یک ویژگی باید یکسان باشند.
- ◀ اگر مقدار ویژگی، خود شامل علائم جفت کوتیشن باشد آن را داخل علائم تک کوتیشن قرار دهید و بر عکس.
- ◀ برای یک عنصر نمی‌توان دو ویژگی با یک نام در نظر گرفت.
- ◀ یک ویژگی نمی‌تواند شامل یک عنصر خارجی باشد (صفحه ۵۸). همچنین برای ویژگی نمی‌توان از علامت < استفاده کرد. در صورت نیاز به این علامت به جای آن باید عبارت `&lt;` را به کار برد.
- ◀ ویژگیها اطلاعاتی درباره محتویات عناصر می‌دهند و محتویات عناصر، اطلاعاتی از داده‌ها در اختیار می‌گذارند. به اطلاعات ویژگیها *فراتر* اطلاعات می‌گویند.
- ◀ از عناصر تودرتو برای تشخیص بهتر اطلاعات استفاده می‌شود (صفحه ۲۷).



## استفاده از عناصر خالی

بعضی از عناصر فاقد محتویات می‌باشند و برای آنها نمی‌توان متنی نوشت. برای مثال عنصر `picture` را در نظر بگیرید که به یک تصویر اشاره می‌کند و دارای ویژگی است؛ ولی محتویات متنی ندارد.

**برای نوشتن یک عنصر خالی که دارای یک برچسب شروع و پایان است:**

- ۱- عبارت `<name>` را تایپ کنید. منظور از `name` واژه‌ای است که یک عنصر خالی را مشخص می‌نماید.
- ۲- ویژگی‌های مورد نیاز آن را مطابق مطالب صفحه ۲۸ ایجاد کنید.
- ۳- برای تکمیل عنصر، عبارت `>` را تایپ نمایید.

**برای نوشتن یک عنصر خالی با برچسبهای شروع و پایان جداگانه:**

- ۱- عبارت `<name>` را تایپ کنید. واژه `name` معرف یک عنصر خالی است.
- ۲- اگر می‌خواهید مانند آنچه که در صفحه ۲۸ توضیح داده شده ویژگی‌های عنصر را ایجاد نمایید.
- ۳- برای تکمیل برچسب شروع، علامت `>` را تایپ کنید.
- ۴- به منظور تکمیل عنصر، عبارت `</name>` را تایپ نمایید. عبارت `name` باید منطبق بر عبارتی باشد که در مرحله ۱ تایپ کرده‌اید.

### نکته‌ها

- ◀ در XML هر دو روش گفته شده معادل یکدیگر می‌باشند.
- ◀ بر خلاف HTML، در اینجا شما نمی‌توانید برچسب شروع بدون برچسب پایان داشته باشید. اگر چنین کاری انجام دهید سند شما خوش‌فرم نیست و تجزیه‌گر XML اعلام خطا خواهد کرد.

علامت کوچکتر از

```
<picture filename="tiger.jpg" />
```

علامت / و علامت بزرگتر از

**شکل ۱۶-۱.** عناصر خالی می‌توانند به صورت هم‌زمان برچسب شروع و پایان را داشته باشند. همان طور که در این مثال می‌بینید پس از برچسب شروع، ویژگی، مقدار و برچسب پایان غیروابسته آورده شده است.

```
code.xml
<endangered_species>
  <animal>
    <name language="English">Tiger</name>
    <name language="Latin">panthera tigris</name>
    <threat>poachers</threat>
    <weight>500 pounds</weight>
    <source sectionid="120"
      newspaperid="21"></source>
    <picture filename="tiger.jpg" x="200" y="197"/>
  </animal>
</endangered_species>
```

**شکل ۱۷-۱.** `picture` و `source` عناصر خالی هستند. عنصر خالی `source` ویژگی‌های خود را دارد و عنصر خالی `picture` به داده‌های دودویی خارجی که به صورت متنی نمی‌باشند اشاره دارد.

## نوشتن توضیحات

علامت کوچکتر، تعجب و دو خط تیره

محتویات

```
<!--updated May 3, 2001-->
```

دو خط تیره و علامت بزرگتر

**شکل ۱۸-۱.** طرز نوشتن توضیحات در XML و HTML

یکسان است.

```
code.html
</endangered_species>
<animal>
<name language="English">Tiger</name>
<name language="Latin">panthera tigris</name>
<threat>poachers</threat>
<weight>500 pounds</weight>
<!--the source tag references the corresponding
article on the World Wildlife Fund web site-->
<source sectionid="120"
newspaperid="21"></source>
<picture filename="tiger.jpg" x="200" y="197"/>
</animal>
</endangered_species>
```

**شکل ۱۹-۱.** به کمک توضیحات امکان اضافه کردن

اطلاعات بیشتر در برنامه به وجود می‌آید. این توضیحات به ویژه هنگامی که بخواهید با طرز کار برنامه آشنا شوید یا آن را برای شخص دیگر تشریح کنید بسیار مفید هستند.

بهرتر است همیشه در اسناد XML از توضیحات استفاده نمایید. بدین ترتیب همیشه دلیل استفاده از عناصر یا زمانی را که بخشی از برنامه نیازمند به‌روزرسانی باشد می‌دانید. توضیحات برنامه داخل آن نوشته می‌شوند ولی از دید بینندگان برنامه پنهان می‌مانند.

### برای نوشتن توضیحات:

- ۱- عبارت <!-- را تایپ کنید.
- ۲- توضیحات لازم را بنویسید.
- ۳- عبارت >-- را تایپ کنید.

### نکته‌ها

- ◀ بین خط تیره‌هایی که در ابتدا و انتهای توضیحات قرار می‌گیرند نباید فاصله‌ای قرار بگیرد. همچنین بین توضیحات و خط تیره‌ها نباید فاصله باشد. برای مثال توضیحات را باید به صورت <!--this is a comment --> نوشت.
- ◀ بین توضیحات نباید دو خط تیره پشت سر هم نوشت و در ضمن نباید از توضیحات تودرتو استفاده کرد.
- ◀ بخش مربوط به توضیحات را می‌توان در طول ایجاد و اشکال‌زدایی برنامه، پنهان ساخت. به چنین توضیحاتی در اصطلاح 'Commenting out' می‌گویند. این توضیحات از دید تجزیه‌گر پنهان می‌مانند و مورد بررسی و غلط‌گیری قرار نمی‌گیرند.
- ◀ توضیحات برای به‌روزرسانی و بهینه‌سازی اسناد XML و صفحه‌های سبک نیز بسیار مفید می‌باشند.
- ◀ برنامه‌های مرورگر، توضیحات را نمایش نمی‌دهند و آنها را تنها می‌توان در کد برنامه مشاهده نمود.

## نوشتن پنج نماد ویژه

در اسناد HTML می‌توان با نوشتن یک علامت **&** و یک نام و علامت ؛ نمادهای ویژه را نمایش داد و در اسناد درج نمود. در XML تنها پنج موجودیت هستند که می‌توان از آنها به صورت پیش‌فرض استفاده کرد. بقیه موجودیتها پیش از آنکه بتوانند مورد استفاده قرار گیرند باید در یک DTD تعریف شوند.

### برای نوشتن نمادهای ویژه:

- ۱- برای ایجاد **&** عبارت **&amp;** را بنویسید.
- ۲- برای ایجاد علامت **<** عبارت **&lt;** را تایپ کنید.
- ۳- به‌منظور ایجاد علامت **>** عبارت **&gt;** را تایپ نمایید.
- ۴- برای ایجاد علامت **"** عبارت **&quot;** را بنویسید.
- ۵- علامت **'** با تایپ عبارت **&apos;** مشخص می‌شود.

### نکته‌ها

- ◀ به غیر از پنج نمادی که گفته شده بقیه نمادها و موجودیتها را ابتدا باید در یک DTD تعریف کنید (صفحه ۵۵) سپس در اسناد به کار برید.
- ◀ علامتهای **<** یا **&** را برای برچسبها یا موجودیتها می‌توان به کار برد. در غیر این صورت اگر بخواهید به عنوان نمادها و موجودیتهای ویژه از آنها استفاده کنید باید به ترتیبی که در نکته قبلی گفته شد عمل نمایید.
- ◀ علامتهای **"** ، **'** یا **>** را به صورت مستقیم در اسناد می‌توان به کار برد. برای سایر موارد نکته بعد و نکته آخر صفحه ۳۲ را بخوانید.
- ◀ یک دلیل خوب برای نوشتن **&quot;** یا **&apos;** به جای علامتهای **"** و **'** این است که شاید مقدار یک ویژگی از علامتهای **"** و **'** تشکیل شده باشد؛ در نتیجه شما می‌توانید مقادیر را به همراه موجودیتهای داخل آنها نمایش دهید.

```
code.html
<endangered_species>
<animal>
<name language="English">Tiger</name>
<name language="Latin">panthera tigris</name>
<threat>poachers</threat>
<weight>&lt;500 pounds</weight>
<!--the source tag references the corresponding
article on the World Wildlife Fund web site-->
<source sectionid="120"
newspaperid="21"></source>
<picture filename="tiger.jpg" x="200" y="197"/>
</animal>
</endangered_species>
```

**شکل ۱-۲۰.** هنگامی که این سند توسط تجزیه‌گر، تجزیه می‌گردد به جای موجودیت **&lt;** علامت **<** نمایش داده می‌شود.

## نمایش عناصر به صورت متن

اگر می‌خواهید در اسناد XML درباره عناصر و ویژگیها چیزی بنویسید باید کاری کنید که تجزیه‌گر از تفسیر آنها صرف‌نظر کند و در عوض آنها را به صورت متن نمایش دهد. بدین منظور باید چنین اطلاعاتی را در داخل یک بخش CDATA محصور نمایید.

### برای نمایش عناصر متنی:

- ۱- عبارت `![CDATA[` را تایپ کنید.
- ۲- عناصر، ویژگیها و محتویاتی را که می‌خواهید نمایش داده شوند ولی تجزیه نگردند ایجاد نمایید.
- ۳- علائم `>]]` را تایپ کنید.

نکته‌ها

- ◀ یکی از مزایای استفاده از قسمت CDATA محدود کردن صفحه‌های سبک می‌باشد (صفحه ۱۸۷).
- ◀ بهتر است از قسمت‌های CDATA به صورت تو در تو استفاده نکنید.
- ◀ از آنجایی که نمادهای ویژه مفهوم خاص خود را در قسمت CDATA از دست می‌دهند، در صورتی که برنامه دارای علامت `<` یا `&` باشد، به جای `&lt;` یا `&amp;` می‌توانید به صورت مستقیم علامت `<` یا `&` را بنویسید.
- ◀ قسمت‌های CDATA می‌توانند در هر جایی بین برچسب شروع و برچسب پایان عنصر ریشه قرار گیرند.
- ◀ اگر بنا به دلایلی بخواهید عبارت `>]]` را پیش از رسیدن به پایان قسمت CDATA تایپ کنید باید علامت `>` را به صورت `&gt;` بنویسید. برای کسب اطلاعات بیشتر درباره نمادهای ویژه به صفحه ۳۱ و پیوست ۳ مراجعه نمایید.

```

<xml_book>
<tags><appearance>
<![CDATA[<endangered_species>
<animal>
<name language="English">Tiger</name>
<name language="Latin">panthera tigris</name>
<threat>poachers</threat>
<weight>500 pounds</weight>
<!--the source tag references the corresponding
article on the World Wildlife Fund web site-->
<source sectionid="120"
newspaperid="21"></source>
<picture filename="tiger.jpg" x="200" y="197"/>
</animal>
</endangered_species>
]]>
</appearance></tags></xml_book>

```

**شکل ۲۱-۱.** در این مثال برای نمایش کد واقعی بدون اینکه مورد تجزیه قرار گیرد از CDATA استفاده شده است.

```

<?xml version="1.0" ?>
- <xml_book>
- <tags>
- <appearance>
- <![CDATA[
- <endangered_species>
- <animal>
- <name language="English">Tiger</name>
- <name language="Latin">panthera tigris</name>
- <threat>poachers</threat>
- <weight>500 pounds</weight>
- <!--the source tag references the corresponding ar
- <source sectionid="120" newspaperid="21"></source>
- <picture filename="tiger.jpg" x="200" y="1
- </animal>
- </endangered_species>
- ]]>
- </appearance>
- </tags>
- </xml_book>

```

**شکل ۲۲-۱.** در این مثال از برنامه Internet Explorer 5 به عنوان تجزیه‌گر استفاده شده است. ببینید که چگونه در قسمت CDATA برچسبها به صورت متن در نظر گرفته می‌شوند؛ ولی `xml_book`، `tags` و برچسبهای `appearance` توسط تجزیه‌گر مورد بررسی قرار می‌گیرند.