

# 项目一 员工信息的存储——XML 概述

- 能力与知识目标

1. 掌握 XML 的基本结构；
2. 掌握 XML 元素和属性的命名规则；
3. 掌握 XML 元素和属性的规范格式；
4. 掌握 XML 元素与属性的差异；
5. 掌握名称空间的声明和使用方法；
6. 掌握 XML 文档中的相关知识要点。

- 项目任务

本项目主要完成员工信息的存储,将员工的信息进行结构化设计,并运用标记的方式描述信息数据的含义,使数据的显示与数据的存储相分离。

- 项目分析

要完成员工信息的存储,首先必须将信息结构化,运用 XML 可扩展的特性自定义标记来描述员工的信息并格式化信息。

## 任务 1 员工信息数据的存储

### 1.1.1 问题情境及其实现

将表 1-1-1 中的员工基本信息存储在文件中。

表 1-1-1 员工基本信息表

职工号	姓名	性别	年龄	住址	部门
202	李四	男	33	大东区	采购部
203	王佳	女	25	皇姑区	采购部
204	李娜	女	29	大东区	采购部
103	袁豆豆	男	33	大连	运输部
102	郭晓峰	男	44	广州	运输部
101	王小强	男	33	重庆	运输部

(续表)

职工号	姓名	性别	年龄	住址	部门
301	李鹏达	男	23	和平区	售后服务部
401	刘芳芳	女	29	大东区	质检部
501	齐大胜	男	33	大连	销售部
302	李小明	女	32	沈阳	售后服务部
502	李爱	女	23	山东	销售部

**说明:**要将表 1-1-1 中的员工基本信息存储在文件中的方法很多,但是数据的存储具有一定的结构,在数据库中很容易实现数据的存储,但是在没有安装数据库的情况下,怎样将这些简单的信息数据按一定的结构存储起来,方便我们对其进行操作,这就需要用到 XML 文件,那什么是 XML 文件呢?它具有什么样的特点和技术?带着这些问题我们来了解一下 XML 的发展历史,初步认识一些 XML 的相关概念。

## 1.1.2 相关知识:XML 的发展、XML 语言应用、XML 相关技术、XML 编辑解析工具

### 1. XML 的发展

XML(eXtensible Markup Language,可扩展标记语言)是 W3C(全球信息网协会)于 1998 年 2 月制定出统一的标准来规范 XML 的使用,即 XML1.0 规范。这种规范是计算机业界制定的一种协议,它使某种软件的应用在全世界范围内达到统一,以方便人们的使用和相互交流。规范的应用在于软件开发商将根据规范开发出合乎规范要求的软件。

XML 的核心是标记,XML 的强大功能来源于它的可扩展性。那么什么是标记语言呢,XML 可扩展性的含义是什么,为什么要创建 XML?

#### (1) 标记语言

标记语言(Markup Language),也称置标语言,就是使用某种“记号”来表示特殊信息的语言。例如,在读书时,人们习惯用彩色笔在书本上的某些句子下面划线等标记,如图 1-1-1 所示,用下划线来表示这些内容是很重要的这一信息,这就是一种标记语言,只是这种标记语言并不能被计算机理解并执行。

计算机能够理解的“记号”就是我们所说的标记,用于指明文档中数据的显示格式或数据含义。标记语言就是使用标记来界定和描述这些数据的语言。换句话说,所谓的标记语言就是它们所定义的标记,来给予电子文件除了内容之外的其他特殊含义。

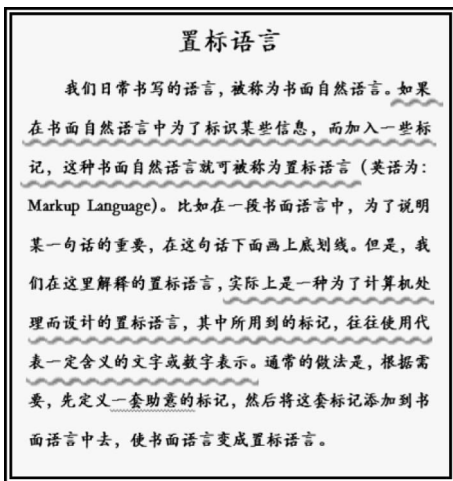


图 1-1-1 书中的标记

标记语言是由一些格式标记或控制标记组成的,这些标记决定了数据信息的显示格式或含义描述,如何定义标记和使用标记是标记语言的核心。例如:

```
<p>我喜欢<bold>唱歌</bold></p>
```

```
<人物>我</人物> <爱好>喜欢唱歌</爱好>
```

总结:标记语言就是有关使用什么样的记号、如何使用这些记号的一种语言,通过它可以和计算机进行自由的交流。

根据标记语言的具体使用范围分为专用标记语言和通用标记语言。

#### ①专用标记语言

专用标记语言是为某种特殊用途而产生的标记语言,通常被某一种或几种软件所支持。例如:WORD、HTML、写字板、WPS。

#### ②通用标记语言

通用标记语言的设计并不针对某一特殊的应用软件,或某一特殊的用途,它通常只描述文件中文字的内容与结构,是一般化的语言(是元语言,即定义语言的语言)。例如:SGML、GML。

特点:具有被国际标准化组织通过并被全世界普遍接受的标准。

#### (2)标准通用标记语言(SGML)

1969年,IBM公司开发了第一种文档描述语言,用来解决不同系统中文档格式不同的问题。IBM公司把这种标记语言称为通用标记语言(Generalized Markup Language, GML)。

1986年,GML被国际化标准组织(ISO)采纳为数据存储和交换的国际标准:ISO8897和WebSGML,称为SGML(Standard Generalized Markup Language,标准通用标记语言)。SGML经过了二十多年的发展,变成了一套相当完整且被大家所接受的标记语言。

SGML是一种可以定义其他标记语言的元语言,因为其利用DTD(Document Type Definition,文档类型定义)来指定文件的逻辑结构和所使用的标记,如此一来,人们就可以将文件自由地转换成所要的格式了。

SGML的功能非常强大,它为描述电子文档提供了一套必要的通用框架,同时也为电子文档信息结构化提供统一的法则。通过该种语言做出的电子文档,具有很好的跨平台性,便于计算机之间的通信和信息处理,有了SGML就不再有两个单位间的电子文件不兼容的结果。由SGML所提供的功能已被美国一些大型企业所采用,如汽车公司、电信公司、科技信息出版商和航空业界等,同时也作为一种文档标准而被美国政府及其合同商广泛使用。

总的来说,SGML的优点主要有列三点:

- ①规范完整。
- ②可移植性好。
- ③稳定性高。

接下来我们看看SGML有哪些缺点,为何至今没有被广泛应用。SGML的缺点主要有列三点:

①规范过于详尽。SGML规范发展的非常完整,稳定性高,在开发时没有模糊的地方,凡事按照规范就行了,这是它的优点;可是如此一来也被规范束缚住了手脚,有任何地方不

符合规范就会导致工作无法进行下去。

②非常复杂。SGML 规范的完整性也导致了其非常复杂,要完全了解这种标准,没有几年的时间,几乎是不可能的。

③开发成本高。SGML 的高复杂性使得相关软件开发的费用太高。

SGML 的复杂性和实现起来的高成本意味着大多数的企业和个人无法拥有这一强大的技术,因而在很长一段时间没有得到推广。鉴于这些原因,欧洲粒子物理实验室的研究人员 Tim Berners Lee 于 1989 年,在 SGML 的基础上提出了超文本标记语言 HTML。

### (3) 超文本标记语言(HTML)

HTML(Hypertext Markup Language,超文本标记语言)是一种专门用于格式化 Web 数据的语言。HTML 免费、简单,用它编写的文件(文档)的扩展名是“.html”或“.htm”,它们是可供浏览器解释浏览的文件格式。

HTML 是 SGML 的一个子集,实际上是 SGML 语言的一种应用。HTML 使用各种定义好的标记分别标识和格式化不同的网页元素,HTML 可使用的标记是固定的、预先定义好的元素标记,用来标识一般用途的网页元素。

**【例 1-1-1】** HTML 文件举例。如图 1-1-2 所示。

```
<html>
  <head>
    <title>你好,HTML</title>
  </head>
  <body>
    <h3>你好,欢迎来到 HTML 的世界! </h3>
  </body>
</html>
```

HTML 的优点:

①标记有限。所有标记不到 100 个,常用的只有几十个。

②语法要求宽松。大小写不敏感,控制标记也不一定配对使用。

③支持软件很多。所有文本编辑器都可以用,另外如 Frontpage、HTML ABC、Dreamwaver 等。

HTML 以简单易学、灵活通用的特性,使人们发布、检索、交流信息都变得非常简单,从而使 Web 成了最大的环球信息资源库。

HTML 的缺点:

①标记确定。用户无法设计出适合自己领域的特殊标记,不能描述矢量图形、数学公式、化学符号等特殊对象,在数据显示方面的描述能力也不尽如人意。

②HTML 标记语言的标准不统一。不同的浏览器厂商为了各自利益,都来制造符合自



图 1-1-2 例 1-1-1. html 的浏览效果

身利益的标记,导致了国际上 HTML 标记语言混乱。

③只着重于数据的显示样式,无法区分数据和样式。它只能显示内容而无法表达数据内容,无法处理结构化信息。而这一点恰恰是电子商务、智能搜索引擎所必需的。例如:

```
<h3>Apple</h3>
```

这段代码表示了“Apple”这段信息,还表示了这段文字在浏览器中是以 h3 的格式显示的,但是 HTML 却没有告诉我们“Apple”到底是什么。Apple 只是一个英文单词,它在不同的环境下会有不同的含义:一家电脑公司,一种水果,抑或是一幅图片? HTML 并没有也不能揭示数据的含义。

HTML 语言最重要的缺点是:HTML 只是 SGML 的一个实例化的子集,可扩展性差,用户根本不能自定义有意义的标记供他人使用。这一切都成为 Web 技术进一步发展的障碍。

在这种情况下,开发一种兼具 SGML 的强大功能、可扩展性以及 HTML 的简单性的语言势在必行。由此诞生了 XML 语言。

#### (4)可扩展标记语言(XML)

XML(eXtensible Markup Language,可扩展置标语言)是由 W3C 于 1998 年 2 月发布的一种标准。XML(规范 40 页)是 SGML(规范 500 页)的一个简化子集,是专门用于在 Internet 中储存和传送信息的标记语言,XML 确保数据结构是一致的并独立于应用程序或供应商。作为 SGML 的简化版本,XML 具有了更广泛的易用性。简单明了的语法结构使初学者很容易就可以掌握 XML,此外,XML 使得现有的 Internet 协议和软件更为协调,从而简化了数据处理和传输。

特点:将 SGML 的丰富功能与 HTML 的易用性结合到 Web 的应用中,以一种开放的自我描述方式定义了数据结构,在描述数据内容的同时能突出对结构的描述,从而体现出数据之间的关系。

创建 XML 的目的:尽量简化 SGML,并继承其优点,同时又尽可能弥补 HTML 的缺点。

**【例 1-1-2】** 一个 XML 文件的示例。如图 1-1-3 所示。

```
<? xml version="1.0" encoding="gb2312"? >
```

```
<Example>
```

```
  <title>你好,HTML</title>
```

```
  <Greeting>你好,欢迎来到 HTML 的
```

```
世界! </Greeting>
```

```
</Example>
```

#### 说明:

①第一行是有关 XML 的声明部分,这部分必不可少并且必须写在程序的第一行。

②最外层的标记 < Example > </Example>称为根标记,包含了所有的下层子元素,根标记存在且唯一。



图 1-1-3 例 1-1-2.xml 的浏览效果

③其他标记对如<title></title>、<Greeting></Greeting>必须配对使用,标记名都是用户根据描述数据的需要而自己定义的,title表示“标题”,Greeting描述“欢迎语”。

XML文档在IE浏览器中的显示效果与源文档的格式与内容类似,而且通过IE更容易看出,XML文档是一种树状的层次结构,并能清楚地标明文档中各个元素的结构,这也充分体现了XML文件保存数据的特性。因为对于XML文档而言,通常人们只关心它的数据内容和数据结构,而它的显示形式是由其他语言来定义的(通常是CSS或XSL,后面会有这两种语言的详细描述)。标记的描述数据结构的作用可以使XML很容易地转换成关系数据库管理系统格式,或者XML数据的其他表现形式,比如表格形式等。

#### (5)XML与SGML、HTML的关系

##### ①XML与SGML的关系

从本质上讲,XML和HTML都是SGML的子集。因此,这三种语言都有一些共同点,如相似的语法和标记符的使用。但是,XML从根本上讲就是SGML的一个精简子集,而HTML是SGML的一种应用。

XML简单易学,但功能却十分强大。XML使用了SGML中约20%的标记,却实现了SGML近80%的功能。XML是一种专门应用在Internet上的SGML文档,它对SGML进行了“大刀阔斧”的简化,将一些和Internet无关的,或是很少用到的规范部分全部删掉,并增加了部分互联网的特殊成分。因而XML在充分继承了SGML功能丰富优点的同时,也克服了SGML极其复杂的缺点。

与XML和HTML不同,SGML可能永远不会在Internet上被广泛接受,因为它从来没有为某个网络协议的需求而设计或优化过。然而,作为所有标记语言的母语言,SGML的强大的功能和规范的完整性仍使它在高端复杂结构的应用中将继续发挥作用。

##### ②XML与HTML的关系

SGML功能强大,但却复杂难以掌握;HTML简单易学,却功能有限;而XML继承了HTML简单易学特性的同时,也具有了SGML的强大功能。那么人们不禁要问:XML能够取代HTML吗?就目前来说,这个问题的答案是否定的,两者根本谈不上替代的关系。XML不是HTML的替代品,XML和HTML是两种不同用途的语言。

首先,XML与HTML在Internet上承担的责任并不完全一样。由上述HTML和XML的两个实例1-1-1和1-1-2可以看出,XML文档和HTML文档虽然非常相似,但他们在浏览器中显示的结果却差别较大,主要原因是HTML重视的是信息在浏览器中的显示效果,展示文件的内容,而XML只关心数据本身,是描述数据及其结构的语言。也就是说一个XML文档并不决定数据的显示样式,数据显示效果都是由被称为样式表的文档来实现的。所以,XML在使用上,很多地方还离不开HTML的帮助。因为XML数据的显示往往要被转变成HTML文档,才能被浏览器识别。

在Internet环境中跨平台的依赖于内容的技术,是这个时代中处理分布式结构信息的选择工具。在XML中和HTML一样,使用标签来标记文本正文。XML把信息的结构与他的现实方式分离开,这样相同的XML文档可以在不同的环境中使用和重用。

其次,由于HTML目前还是Internet上表示数据的主流语言,这部分要被转变为XML格式需要很长时间。但可以肯定的是,在未来的Web开发中,XML将和HTML一样受到

重视,它们都是 Web 技术的基础。XML 将成为最普遍的数据操纵和数据传输的工具。可以说,二者各有所长,“HTML 主外,XML 主内”。

SGML、HTML 和 XML 将凭借各自的优势继续适用于其合适的地方。SGML 在高端复杂结构的应用以及在规范制定方面将发挥其强大的功能。HTML 仍是在 Web 上显示数据的最简单的方法。而对于数据的广泛性和灵活性有特定要求的广大用户,XML 将是最好的选择。

## 2.XML 语言及应用

XML 是专门用于在 Internet 中储存和传送信息的标记语言。这只是它表面的定义,那么 XML 有什么优点,XML 到底能做什么,为什么要使用 XML 等问题,希望学习这一节后读者能对 XML 的整体概念有一个直观的认识。

### (1)XML 的特点

XML 有两个显著的特点(也可说是两大优点):良好的可扩展性和 XML 数据内容与显示样式的分离。根据这两方面的特点,我们来回答上述问题。

#### ①XML 到底能做什么

XML 最主要的特点体现在可扩展性方面,就是它允许用户根据自己的特殊需要制定出适用于自身的标记和文档结构,以便于结构化地描述自己领域的信息,从而提供一种处理数据的最佳方式。因此,可以说 XML 就是用来处理数据的,而且是专门用于在 Internet 中储存和传送信息的标记语言。

#### 【例 1-1-3】 影片信息表。

```
<? xml version="1.0" encoding="gb2312"? >
```

```
<! --文件名:例 1-1-3.xml-->
```

```
<影片信息表>
```

```
<影片>
```

```
<片名>宫锁沉香</片名>
```

```
<类别>古装</类别>
```

```
<主演>周冬雨/陈晓/赵丽颖</主演>
```

```
<导演>潘安子</导演>
```

```
<简介>清朝康熙年间,兆佳沉香入宫遇了同龄的宫女琉璃,结为好友。琉璃一心想出人头地,企图邂逅的阿哥变凤凰,不惜背叛陷害沉香;而沉香则偶然与十三阿哥相爱,经过曲折坎坷,二人终于归隐远去,相爱厮守</简介>
```

```
</影片>
```

```
<影片>
```

```
<片名>变身超人</片名>
```

```
<类别>喜剧</类别>
```

```
<主演>陈柏霖/郭雪芙/邱彦翔</主演>
```

```
<导演>张时霖</导演>
```

```
<简介>红极一时的超人剧男主角铁男由于收视率下滑严重,遭新任电视台董事长苏盈盈换角,歌手 FACE 成为新世代超人,而铁男则瞬间下岗。为改变现状,他不仅接了一堆类似路人乙的角色,终于在转战电视购物时,铁男下岗再就业打开了事业第二春。</简介>
```

```
</影片>
```

</影片信息表>

本例使用元素内容保存数据信息。例如,将影片的具体信息(片名、类别、主演、导演和简介)作为<影片>元素的内容储存在 XML 文档里。这样,一个层次和语义都非常清晰的信息存储结构就形成了,然后就可以使用其他技术(如 XSL 技术)对数据进行查询、显示等各种操作。

## ②XML 的优点

XML 的优势十分显著,主要表现在下列几个方面:

### a. 实现不同数据源之间的数据交换和传输

使用 XML 可以方便地在不同系统之间进行交换。当今的计算机世界中,不同企业、不同部门中存在着许多不同的系统和数据库,两个系统之间的数据交换通常需要一些转换。XML 是一种跨平台的、与硬件无关的处理信息的工具。一种数据源只要将它的数据表示成 XML 格式就能被另一种数据源有效地识别,如图 1-1-4 所示。



图 1-1-4 不同数据源之间的数据交换

把数据转换为 XML 格式存储将大大减少交换数据时的复杂性,并且还可以使得这些数据能被不同的程序读取。XML 的这项使用超出了 Internet 的范围,适用于任何传统的用任何方式进行数据交换的计算机。

当前,所有主要的商业 RDBMS 产品,如 Oracle、SQL Server 等都成为了支持 XML 的数据库,一些对 XML 数据的处理已经被集成为常规数据库的一部分。

### b. 数据内容与其形式的分离

XML 优势在于,它将信息的数据部分和数据显示部分分开,使用户可通过使用不同的样式表和应用程序,根据需要显示和处理数据。例如,要更换数据显示的方式,仅需改变样式表的设置,用户就可以将同一数据制作成 PDF、WML (Wireless Markup Language)、HDML (Hand-Held Device Markup Language) 等不同格式,供不同的硬件显示。

XML 数据内容与形式分离还可以使搜索简单高效地进行。搜索引擎不需要再去遍历整个 XML 文档,而只需要寻找相关标记的内容。例如,要寻找“XML 高级编程”信息,只要看看是不是与<片名>这个标记下的字符串匹配。

### c. 支持不同文字不同语种间的信息交互

XML 所依赖的 Unicode 标准,是一个支持世界上所有主要语言的混合文字编码系统。使用根据 XML 规则编写的浏览器来阅读网页,可以处理各种语言的任意组合。这就是说,XML 技术不但能使各种信息在不同的计算机系统之间交互,还能跨语种、跨文化进行交互。

总之,XML 是用于网络上数据交换的语言,是一种公用的数据载体,它可以包含数据然后传递到其他任何地方(通常是在 Internet 上)。任何一个可以读取数据的物体只要支持 XML 都可以读懂来自 XML 的数据。而不同格式的 XML 通过 XSLT 也可以转换成相同的格式。



## (2)XML 的设计目标

由于 SGML 太复杂,而 HTML 又过于简单,如何简化 SGML 同时又弥补 HTML 的缺点呢?针对这个问题,XML 的研究者们提出了几个目标,而这些目标的提出也从另一个侧面反映了 XML 的特点。下面是在 W3C 网站(网址为:<http://www.w3.org>)上公布的有关 XML 所要实现的 10 个官方目标。

- ①XML 应该可以直接在因特网(Internet)上使用。
- ②XML 应该支持大量不同的应用。
- ③XML 应该与 SGML 兼容。
- ④处理 XML 文件的程序应该容易编写。
- ⑤XML 中的可选项应无条件地保持最少,理想状况下应该为 0 个。
- ⑥XML 文件应该能够让人直接阅读,而且应该有足够的可读性。
- ⑦XML 的设计应快速完成。
- ⑧XML 的设计应该是形式化的,简洁的。
- ⑨XML 文件应易于创建。
- ⑩XML 标记的简洁性是最不重要的设计目标。

## (3)XML 的应用

XML 在语法结构上的优越性,使得它可以广泛地应用于信息描述、信息交换、信息存储、Web 服务、电子商务、网络出版等多个领域,可以说 XML 已经渗透到了互联网的各个角落。总体来看,XML 的用途主要有两方面:一是作为元标记语言,定义各种实例标记语言标准;二是作为标准交换语言,担负起描述交换数据的作用。

### ①创建新的标记语言

作为元标记语言,XML 可以让用户自定义适合本行业的标记语言。下面简单介绍几个由各标准组织定义推广的本领域的 XML 标记语言:

a. XHTML(eXtensible Hyper Text Markup Language,可扩展超文本标记语言),由 W3C 于 2000 年 1 月制定。XHTML 实际上是用 XML 重新定义的 HTML,可以看作是 HTML 的 XML 版本。

b. MathML(Mathematical Markup Language,数学标记语言),用于描述 Web 上的科学内容和数学方程式的标记语言。

c. CML(Chemical Markup Language,化学标记语言),应用在化学领域的描述化学符号和分子结构信息的语言。

d. VML(Vector Markup Language,矢量标记语言),用于描述和存储各种矢量图形信息。

e. WML(Wireless Markup Language,无线标记语言),是 WAP(Wireless Application Protocol,无线应用协议)论坛制定的,特别适用于在性能方面受到严重限制的手持设备上传递和显示 Web 页面信息。这种标记语言实现了手机用户网上信息浏览和信息发送。

f. MusicML(Music Markup Language,音乐标记语言),用于编写和传递音乐乐谱。

g. RDF(Resource Description Framework,资源描述框架),是描述数据的保存、分布、产生、使用等情况的 XML 应用程序。

## ②数据交换和数据存储

计算机技术从诞生至今已经半个多世纪,各种类型的操作系统平台和数据库不胜枚举,运行其上的各种应用程序更是五花八门,要实现机器之间无歧义性的信息识别和处理,一直是困扰技术界的难点。

由于 XML 使用元素和属性来描述数据,而不提供数据的显示方法。这样,XML 就提供了一个优秀的方法来标记独立于平台和语言的内容,因此使用 XML 可以将数据在不同的系统之间进行交换。而且使用象 XSLT 这样的语言能够轻易地将 XML 文件转换成各种格式文件,比如 HTML、WML、PDF、EDI 等。

XML 交换数据能力,也促使了 Web 服务的产生和发展。Web 服务是最令人激动的革命之一,Web 服务使应用程序能够通过 Internet 进行通信,不管操作系统或编程语言是什么。它们可以在任何平台上实现并且可以由诸如 W3C 的公共标准组织定义。利用 Web 服务,应用程序不仅可以共享数据,还可以调用其他应用程序的功能,而不考虑其他应用程序是如何生成的。

总之,XML 语言是一种 Web 开发的新技术,使用它可以开发出绚丽多彩的 Web 应用,XML 的广泛使用和发展,必将推动 Web 的不断前进,开创 Web 应用的新时代。

## 3.XML 相关技术

XML 是用于描述、组织数据的,但如何恰当地构造具有嵌套、符合基本语法规则的 XML 数据,以及如何对 XML 数据进行显示、查找、排序等操作都需要一系列的 XML 相关技术作支撑。这些技术主要包括:用于规范和验证 XML 文档的 DTD 或 XML Schema 技术,解决标记重名的名称空间技术,用于格式化显示 XML 文档的 CSS 和 XSL 样式表技术、用于访问 XML 文档的 DOM 技术,以及 XML 的超链接理论——Xpath、Xlink 和 Xpointer 等。下面是对这些技术的简单介绍。

### (1)建立正规有效的 XML 文档

#### ①正规的 XML 文档

XML 作为一种计算机语言,有其简单而严格的语法,XML 也要求文档符合一定的规则才可以让解析程序正确处理。所谓正规的 XML 文档(也称为格式良好的文档),不太严格的说就是满足基本语法要求的 XML 文档。在这种文档中,文档结构和标记的使用几乎都是很随意的。

#### ②有效的 XML 文档

XML 的可扩展性方面表现在用户可以根据自己的需要,自己制定标记和文档结构用以描述、存储和共享各种数据。但是问题在于,我们经常要确保这些数据必须遵循一些特殊的结构和一组值。比如,可能规定某个标记必须有两个子标记,且这两个子标记又必须按指定的顺序出现等。

所谓有效的 XML 文档,是指遵循一组比格式正规的文档更严格的语法规则,用来规定 XML 文档的逻辑结构,并规定文档中所使用的元素、属性、实体,以及它们之间的关系等。这种语法规则被称为文档类型定义(DTD)或者模式(XML Schema)。前面我们提到过的 XML 可扩展性就体现在 DTD 或 XML Schema 的功能上,即 DTD 或 XML Schema 是定义标记语言的语言。也就是说,通过 DTD 或 XML Schema 来描述什么是有效的标记和正确

的数据结构,但二者最重要的区别有下列两点:

i. DTD 标准的制定历史悠久,稳定性高,它的语法和 XML 不同,是有独自语法结构的文件,而 Schema 文档是一种特殊的 XML 文档,它遵循 XML 语法要求。

ii. DTD 的数据类型相当有限,在 DTD 中根本不提供数值型数据类型,而 Schema 则提供了丰富的数值类型,而且还提供了自定义数据类型的机制。

一个正规的 XML 文档如果和某个 DTD 文档或 Schema 文档相关联,并遵守相应的限制条件,就称为有效的 XML 文档。

#### a. XML 的文档类型(DTD)

**【例 1-1-4】** 与影片信息表对应的 DTD 文档。

```
<? xml version="1.0" encoding="GB2312"? >
<! --文件名:例 1-1-4. dtd-->
<! ELEMENT 影片信息表 (影片+)>
<! ELEMENT 影片 (片名, 类别, 主演, 导演, 简介)>
<! ELEMENT 片名 (#PCDATA)>
<! ELEMENT 类别 (#PCDATA)>
<! ELEMENT 主演 (#PCDATA)>
<! ELEMENT 导演 (#PCDATA)>
<! ELEMENT 简介 (#PCDATA)>
```

这个 DTD 文档说明 XML 文档实例必须具有下列数据结构:

- i. <影片信息表> 标记必须有一个或多个 <影片> 标记;
- ii. 每个 <影片> 标记必须有五个子标记,且这五个标记为 <片名>、<类别>、<主演>、<导演> 和 <简介>,缺一不可,次序颠倒了也不行;
- iii. <片名>、<类别>、<主演>、<导演> 和 <简介> 五个标记为叶子标记,其内容是普通的文本。

根据 DTD 定义的结构来编写的 XML 文档,通常要在文档中声明这些标记的出处(也就是定义标记的 DTD 文档)。

声明文档出处的程序语句为:

```
<! DOCTYPE 影片信息表 SYSTEM "例 1-1-4. dtd">
```

这里,“<! DOCTYPE”是声明语句的开始部分,“影片信息表”是文档的根标记名称,“SYSTEM”是系统关键字,“例 1-1-4. dtd”是 DTD 文件的 URL。带有文档类型定义的 XML 文档清单如例 1-1-5 所示。

**【例 1-1-5】** 与 DTD 文档相关联的 XML 文档。

```
<? xml version="1.0" encoding="gb2312"? >
<! --文件名:例 1-1-5. xml-->
<! DOCTYPE 影片信息表 SYSTEM "例 1-4. dtd">
<影片信息表>
  <影片>
    <片名>宫锁沉香</片名>
    <类别>古装</类别>
    <主演>周冬雨 陈晓 赵丽颖</主演>
```

```
<导演>潘安子</导演>
```

<简介>清朝康熙年间,兆佳沉香入宫遇了同龄的宫女琉璃,结为好友。琉璃一心想出人头地,企图邂逅的阿哥变凤凰,不惜背叛陷害沉香;而沉香则偶然与十三阿哥相爱,经过曲折坎坷,二人终于归隐远去,相爱厮守</简介>

```
</影片>
```

```
<影片>
```

```
<片名>变身超人</片名>
```

```
<类别>喜剧</类别>
```

```
<主演>陈柏霖 郭雪芙 邱彦翔</主演>
```

```
<导演>张时霖</导演>
```

<简介>红极一时的超人剧男主角铁男由于收视率下滑严重,遭新任电视台董事长苏盈盈换角,歌手 FACE 成为新世代超人,而铁男则瞬间下岗。为改变现状,他不仅接了一堆类似路人乙的角色,终于在转战电视购物时,铁男下岗再就业打开了事业第二春。</简介>

```
</影片>
```

```
</影片信息表>
```

在实际应用中,有效的 XML 文档优于简单的格式正规的 XML。只有有效的文档才能符合交换标准和简化有效性验证解析器对这些数据的处理,而对于正规的 XML 文档,读者容易接受但其应用范围非常有限。

#### b. XML 的文档模式(Schema)

功能:同 DTD,都是用来定义标记和标记用法的语言。是为 XML 单独开发的,功能更强,数据类型丰富。而 DTD 只提供文本型。

**【例 1-1-6】** 与 Schema 文档相关联的“对联.xml”文档。

```
<? xml version="1.0" encoding="gb2312" standalone="no"? >
```

```
<对联 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="对联.xsd">
```

```
<上联>书中自有黄金屋</上联>
```

```
<下联>书中自有颜如玉</下联>
```

```
<横批>读书有用</横批>
```

```
</对联>
```

**【例 1-1-7】** 与对联对应的 Schema 文档“对联.xsd”。

```
<? xml version="1.0" encoding="gb2312"? >
```

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
<xsd:element name="对联">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="上联" type="xsd:string"/>
```

```
<xsd:element name="下联" type="xsd:string"/>
```

```
<xsd:element name="横批" type="xsd:string"/>
```

```
</xsd:sequence>
```

```
</xsd:complexType>
```

```
</xsd:element>
```

```
</xsd:schema>
```

## (2)XML 的名称空间

目的:给全球的所有标记起一个唯一的名字,以便区分。在标记的前面加上标记出处。

方法:通过使用 URI(统一资源标识)限定元素和属性的方法,解决 XML 文档中标记重名问题。

## (3)XML 的样式语言

为了显示 XML 数据,W3C 发布了两个建议规范:CSS(层叠样式表)和 XSL(可扩展样式语言)。下面通过例子分别论述这两种样式语言的风格。

### ①使用 CSS 格式化 XML 文档

CSS 最初是用于制定 HTML 文档显示格式的,现在用来对 XML 文档进行简单的样式规划。CSS 的主要作用是定义一个标记或一类标记的显示样式,以便实现对于一个页面样式的独立管理。

一个 CSS 样式表就是一组规则。每个规则给出此规则所适用的标记的名称,和一组适用于此规则要应用到的标记样式。使用 CSS 显示 XML 文档的第一步是编写好作为数据的 XML 文档,由于目前 CSS 不支持中文,因此,在下面的例 1-1-8 中使用英文字符串作标记名描写文本信息,在例 1-1-9 中为其编写样式表(文件名为“例 1-1-9.css”),来获得有关 CSS 显示 XML 文档的一般认识。

**【例 1-1-8】** 使用英文字符串作标记名的诗词表。

```
<? xml version="1.0" encoding="gb2312"? >
<! --文件名:例 1-1-8.xml-->
<Literature>
  <Title>人生易老天难老</Title>
  <Subtitle>----日子都到哪里去了? </Subtitle>
  <Athor>作者:未名</Athor>
  <body>
    <paragraph>
      <sentence>燕子去了,有再来的时候;</sentence>
      <sentence>杨柳枯了,有再青的时候;</sentence>
      <sentence>桃花谢了,有再芬芳的时候。</sentence>
      <sentence>但是,聪明的你,请告诉我,日子一去怎么不复返呢? </sentence>
    </paragraph>
    <paragraph>
      <sentence>是有人偷了他们吧,那是谁? 又藏在何处? </sentence>
      <sentence>是他们自己逃走了吧? 现在又到了哪里? </sentence>
    </paragraph>
  </body>
</Literature>
```

**【例 1-1-9】** 应用于“例 1-1-8.xml”的样式表文件。

```
/* 文件名:例 1-1-9.css */
@charset "GB2312";
/* Literature 标记以块显示,占据宽度 650px,字体为宋体,字体大小 24px,文档显示左对齐 */
Literature {display:block;width:650px;font-family: "宋体";font-size: 24px;text-align: left; }
/* Title 标记以块方式显示,字体大小为 36px,加粗,显示为居中 */
Title {display: block; font-size: 36px;font-weight: bold;text-align: center;}
```

```

/* Subtitle 标记以块方式显示,显示为居右 */
Subtitle {display:block;text-align: right;}
/* Athor 标记以块方式显示,显示为居中 */
Athor {display: block;text-align: center;}
/* body 标记以块方式显示 */
body {display: block;}
/* 文档 paragraph 标记以块方式显示,加载背景图片 bbl. jpg,段落间间距 20px */
paragraph{display: block;background-image:url(bbl. jpg);margin-top:20px;}
/* 文档 sentence 标记以块方式显示,句子高度为 35px,字体为隶书 */
sentence{display: block;height: 35px;font-family: "隶书";}

```

这个 CSS 文件定义了各个标记的内容应以什么样式显示。创建好样式后,为了在例 1-1-8. xml 中应用该样式表,就需要在 XML 文档中添加一条语句用以指明 XML 文档样式文件的路径。来告诉 XML 文档所使用的样式单文件的 URL 和类型。添加了样式单后的 XML 文档如下所示:

```

<? xml version="1.0" encoding="gb2312"? >
<! --文件名:例 1-1-8. xml-->
<? xml-stylesheet type="text/css" href="例 1-7. css"? >
<Literature>
.....
</Literature>

```


上面程序段中的第二行语句是将样式单嵌入 XML 文档的处理指令。其中, type 属性指定所用的样式语言类型, href 则指定样式的文件路径,通常为 URL,可以是绝对的,也可以是相对的。

“例 1-1-8. xml”文档在关联了 CSS 样式表文档“例 1-1-9. css”之后的信息就具有了其显示样式,如图 1-1-5 所示。



图 1-1-5 CSS 格式化 XML 文档显示效果

CSS 技术早已为广大网页设计人员所熟悉,在 HTML 网页中得到了大量的应用。CSS 的标准由 W3C 制定,到目前为止有 CSS1 和 CSS2 两个标准。CSS2 是 CSS1 的升级版,它完全覆盖了 CSS1 的功能,并且在结构性数据的显示方面做得更好。

 **提示:**为了统一性,本书中的 XML 文档尽量使用中文字符串作标记名称,与层叠样式表相关的例外。

## ②使用 XSL 格式化 XML 文档

XSL 是专门为了格式化 XML 页面而发展起来的一种标记语言,XSL 提供的格式化规则可以非常全面和精确的描述要显示的 XML 画面,同样可以通过在 XML 文档中链接 XSL 样式表而实现丰富多彩的文档格式。

就显示 XML 文档而言,XSL 样式表要远比 CSS 样式表功能更强大而且更灵活。CSS 显示 XML 文档的方法是依次扫描 XML 中的所有标记,给各个标记添加样式并显示出来,所以一个标记只能显示一次。XSL 提供了完备的输出机制,可以根据需要精确地挑选出要显示的 XML 数据,同时套上特定的格式然后显示,这就意味着 XML 文档中同一个数据可以显示多遍,并可根据需要对数据进行排序、查询等操作。

XSL 目前的真正功能是实现不同标记语言之间的转换以及和各种应用程序之间的数据交换。由于 XSL 当前的主要应用是将 XML 文档代码转变成 HTML 代码,实现 XML 文档在 IE 浏览器中的显示,所以称其为一种样式语言。

### (4)XML 的数据源对象(DSO)

XML 文档是存储数据的,从某种意义上来说也就是一个数据库。因此 XML 文档可以作为 HTML 网页中数据的来源。这种与 HTML 网页绑定的 XML 数据源对象(Data Source Object,DSO),通常也被称作数据岛(Data Islands)。数据岛的实现机制是将一个 XML 文档或一段 XML 代码当作一个类似于数据库的对象,使用传统操作数据库的方法来实现 HTML 网页中操作 XML 文档中的数据,例如数据的添加、删除、显示和更新等。

XML 文档只包含一系列数据,其本身并不决定这些数据的显示样式,而 HTML 是一种理想的网络信息显示语言。使用数据岛技术可以把 HTML 和 XML 两种技术相结合,优势互补,使得原始数据在能够保持本来意义和结构的同时,使 XML 文档中的元素和属性数据与 HTML 文档中特定的元素进行绑定,并充分利用 HTML 丰富多变的数据表示技巧,实现将 XML 文档中的数据信息通过对应的 HTML 元素展示出来。这种借助 HTML 网页来显示 XML 文档内容的方式,既保持了 XML 文档数据与显示格式分离的特点,使得数据的存储与应用具有极大的灵活性;同时又运用了 HTML 网页丰富的显示格式,具有一定的可编程特性。

### (5)XML 的文档对象模型(DOM)

文档对象模型(Document Object Model,DOM),是 W3C 制定的一套应用程序标准接口的规范。DOM 是与平台和语言都无关的,因而可以用各种语言在各种平台上实现。任何语言可以按照 DOM 规范去实现这些接口,给出解析文件的解析器。

XML DOM 是由一组代表 XML 文档中不同组成部分的程序对象组成,这些对象提供各自的属性和方法,使得应用程序开发者能够通过编写脚本程序来显示和操作 XML 文档中的相应的组件。换言之,DOM 技术为处理 XML 文档提供了一个编程接口,实现从其他应

用程序中调用或管理分析 XML 文档的方法。

方法:将一个 XML 文档看成一个对象,通过固定的方法和属性对 XML 文档中不同的标记或节点进行读和写。

#### (6)XML 的链接和定位语言

超文本链接是 Web 的关键技术。在 HTML 语言中超链接只是一条简单的语句,但在 XML 语言中,超文本链接已经不是 XML 标记语言的一部分,而是一种独立的链接语言。XML 的链接语言目前主要由三部分构成,分别为:XPath、XLink 和 XPointer。这里只是简单地介绍一下,有关它们的详细介绍,请上网参看 <http://www.w3.org/TR>。

##### ①XLink

XLink(XML Linking Language)就是 XML 链接语言,它的功能比 HTML 超链接强大许多。XLink 分为简单链接和扩展链接两种。简单链接的链接目标为某一单一文件或资源,且链接为单一方向,与 HTML 中的超链接类似;而扩展链接的目标则可以是多个文件或资源,并可定义这些目标之间的关联性。可以说 XLink 就是描述 Internet 上任一页面上的任一部分与 Internet 上的其他页面中的某些部分之间关系的一种语言。

##### ②XPath

XPath(XML Path Language)是一种有关如何在 XML 文档中进行定位,即如何快速找出 XML 文档中个别部分的语言。XPath 使用路径表达式识别 XML 文档中的节点,这些路径表达式看起来很像计算机的文件系统。XPath 包含处理字符串、数字和布尔功能,具有匹配机制,用于在 XML 文档中查找某个节点。

在 XPath 数据模型中,将一个 XML 文档中的 XML 元素作为一个层次结构的各个节点。XPath 数据模型可以看成是用节点层次来表示 XML 的信息集。XPath 共定义 7 种节点类型:根节点(文档节点)、元素节点、属性节点、文本节点、注释节点、处理指令节点和名称空间节点。

##### ③XPointer

XPointer(XML Linking Language)就是 XML 的指针语言,它是建立在 XPath 语言基础之上,提供一种比 XPath 所允许的更精细的访问控制制度,其目的是允许基于元素类型、属性值、字符内容和相对位置访问 XML 文档各个部分。用 XPointer 找到的文档部分可以用作链接目标或其他用途。

例如,给定一属性名字,可以利用 XPointer 来寻找对应的属性值。又例如,利用 XPointer 可以轻松找到某一 XML 文档中的第 5 个元素的第 3 个子元素下面的第 10 个子元素。因此,XPointer 所提供的 XML 文档内容的指向和定址功能,大大增强了 XML 对信息的搜索、提取和再利用的功能。

## 4.XML 编辑解析工具

我们在编辑和使用 XML 时,需要相关工具(软件)作支持。本节主要介绍与 XML 相关的软件:XML 编辑器和 XML 解析器。

### (1)XML 编辑器

XML 编辑器就是编写 XML 文档的软件,可以是任何一款文本编辑器。例如,可以使用记事本、写字板以及 Word 等编辑器,XML 在创建完成后用“.xml”作为扩展名加以保存。



因此,我们在此之前都是用“记事本”作为编辑工具进行编辑的。

但是,使用记事本编辑 XML 很容易出现书写上的错误,而且无法对我们编辑 XML 文档提供帮助。随着 XML 的快速发展,专门为 XML 量身定做的编辑器如雨后春笋般冒了出来。XML 编辑器应该能够非常容易地创建和修改符合用户需要使用的 DTD 或 W3C XML 模式的有效 XML 文档。

一个专业 XML 程序编辑器应具有下列功能:

- ①起始标记输入完成后,可以自动生成结束标签;
- ②根据 XML 不同的语法给不同的代码标记不同的颜色;
- ③通过 DTD 验证所书写的 XML;
- ④通过一个有效的格式验证所编写的 XML。

下面介绍两款常用的专业 XML 程序编辑器:XMLWriter 与 XMLSpy,它们体积小,界面简单清楚,广为初学者及专业 XML 开发者喜爱。不过这两个程序并非免费,只有 30 天的试用期。

#### ① Wattle Software 的 XMLWriter

XMLWriter 是一款适合于初学者及专业 XML 开发者的 XML 编辑工具,可以在 Windows 环境下支持 XML、XSL、DTD、CSS、HTML 及文本格式文件的编辑和调试。XMLWriter 有一个直观性、个性化的用户界面让使用者编辑,还具有书签功能,自动查找并替代功能。其他的功能还有 XML 联机帮助、插件管理、即时色彩编码转换、树型结构查看、批量及命令行处理、可扩充的能力及更多功能。

#### ② Altova XMLSpy

Altova XMLSpy 是所有 XML 编辑器中做得非常好的一个软件,专门用于开发 XML 工程的综合集成开发环境 (IDE)。其企业版的功能是最强的,它为 DTD、XML Schema、XML 文件以及 XSLT 样式表的创建和编辑提供了高效而灵活的环境。它具有强大的编辑功能、支持多种文档视图和 XSLT 转换。它可以在 XML 文档与文本文件/数据库之间进行导入和导出。另外,它还提供对工程中 XML 文档的管理、WSDL 编辑器和代码的自动生成等功能。图 1-1-6 是 Altova XMLSpy 2012(汉化版)操作界面。

#### (2) XML 解析器

XML 解析器是分析 XML 文档是否有语法错误,文档的结构和标记的使用是否合法的一种应用程序。在此之前,我们使用 Microsoft 公司 IE 浏览器中的 MSXML 作为 XML 解析器。

在解析 XML 文档时,解析器依据 XML 语法,对 XML 文档做检查,若文档不符合 XML 基本语法规范,就会反馈相应的错误信息给用户,替用户省下不少检查的功夫,这也就是为什么大多数专门的 XML 编辑器将解析器的功能融入其中的缘由。例如,我们前面介绍的 XMLWriter 和 XMLSpy 都同时具有解析器功能。

根据 XML 文档分为良好格式的文档和有效的文档两类,解析器也分为两种:

①非验证的解析器(non-validating):确保数据是格式良好的 XML,但不需解析任何外部资源。例如,IE 是一种非验证的解析器,并不检查文档的有效性。

②验证的解析器(validating):不但检测 XML 文档的语法和元素的层次结构,而且还验证所使用的元素和属性是否遵循 DTD 规则或 XML Schema 规则。

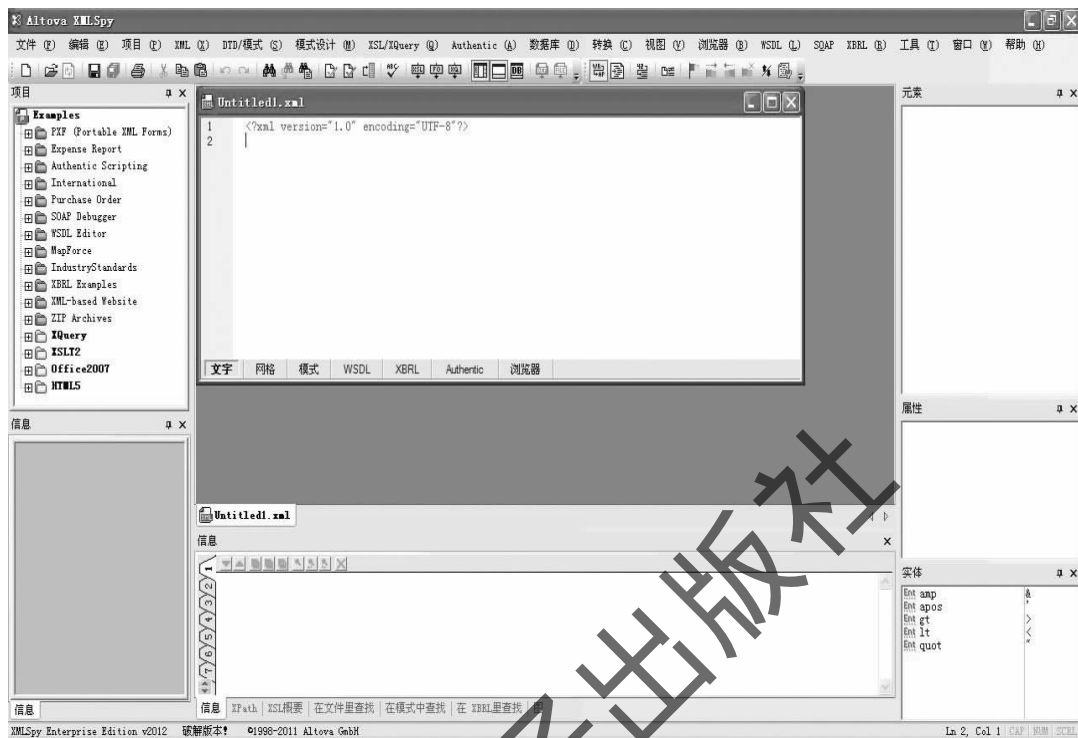


图 1-1-6 Altova XMLSpy 2012 操作界面

### 1.1.3 知识扩展:HTML 与 XML 的比较

表 1-1-2 是它们之间的比较。

表 1-1-2

HTML 与 XML 的比较

比较内容	HTML	XML
可扩展性	不允许用户定义自己的扩展标签	允许用户定义自己的扩展标签
结构描述	不支持深层次的嵌套表达	支持深层次的嵌套表达,适合表示结构复杂的文档数据
可读性及可维护性	难以阅读、维护	结构清晰,便于阅读、维护
数据和显示的关系	内容描述和表现形式合为一体	内容描述和表现形式相分离

## 任务 2 员工信息数据的结构化描述

### 1.2.1 问题情境及其实现

编写一个简单的 XML 文件,存储表 1-2-1 中的数据信息。

表 1-2-1

员工信息表

姓名	EMAIL	编号	上司
张三	Zhangsan@126.com	A10001	B10002
赵阳	Zhaoyang@163.com	B10002	/
王高翔	Wanggx@126.com	A10012	A10234
李蕾	654323126@qq.com	A10234	/
屈飞	Qufei@sina.com	A10345	A10234

编写的 XML 文档如下：

```
<? xml version="1.0" encoding="gb2312"? >
<! --employee.xml-->
<员工信息表>
  <员工 编号="A10001" 上司="B10002">
    <姓名>张三</姓名>
    <EMAIL>Zhangsan@126.com</EMAIL>
  </员工>
  <员工 编号="B10002">
    <姓名>赵阳</姓名>
    <EMAIL>Zhaoyang@163.com</EMAIL>
  </员工>
  <员工 编号="A10012" 上司="A10234">
    <姓名>王高翔</姓名>
    <EMAIL>Wanggx@126.com</EMAIL>
  </员工>
  <员工 编号="A10234">
    <姓名>李蕾</姓名>
    <EMAIL>654323126@qq.com</EMAIL>
  </员工>
  <员工 编号="A10345" 上司="A10234">
    <姓名>屈飞</姓名>
    <EMAIL>Qufei@sina.com</EMAIL>
  </员工>
</员工信息表>
```

从上面编写的 XML 文档可以看到，相关的数据信息在 XML 中可以用元素来描述，也可以用属性来存放。那么 XML 文档的具体结构的构成形式怎样，其语法如何规范呢？带着这些问题我们先来学习 XML 基本的语法知识及其名称空间概念。

## 1.2.2 相关知识:XML 基本语法、名称空间

### 1.XML 基本语法

#### (1)XML 基本结构解析

创建一个简单的 XML 文档。

**【例 1-2-1】** 影片信息表。

```
<? xml version="1.0" encoding="gb2312"? >
```

```
<!--文件名:例 1-2-1.xml-->
```

```
<影片信息表>
```

```
<影片>
```

```
<片名>宫锁沉香</片名>
```

```
<类别>古装</类别>
```

```
<主演>周冬雨 陈晓 赵丽颖</主演>
```

```
<导演>潘安子</导演>
```

<简介>清朝康熙年间,兆佳沉香入宫遇了同龄的宫女琉璃,结为好友。琉璃一心想出人头地,企图邂逅的阿哥变凤凰,不惜背叛陷害沉香;而沉香则偶然与十三阿哥相爱,经过曲折坎坷,二人终于归隐远去,相爱厮守</简介>

```
<剧照 file="gscx.jpg"/>
```

```
</影片>
```

```
<影片>
```

```
<片名>变身超人</片名>
```

```
<类别>喜剧</类别>
```

```
<主演>陈柏霖 郭雪芙 邱彦翔</主演>
```

```
<导演>张时霖</导演>
```

<简介>红极一时的超人剧男主角铁男由于收视率下滑严重,遭新任电视台董事长苏盈盈换角,歌手 FACE 成为新世代超人,而铁男则瞬间下岗。为改变现状,他不仅接了一堆类似路人乙的角色,终于在转战电视购物时,铁男下岗再就业打开了事业第二春。</简介>

```
<剧照 file="bscr.jpg"/>
```

```
</影片>
```

```
</影片信息表>
```



**提示:** 解析器是依据 XML 语法,分析 XML 文档是否有语法错误,文档的结构和标记的使用是否合法的一种应用程序。当所编写的 XML 文档存在语法错误时,IE 将给出相应的错误提示信息。一般用户可使用 Microsoft IE 5.0 或更新的版本作为 XML 解析器。

归纳一下,一个 XML 文档是由文档头和文档体两个主要部分,以及一个可选的文档尾组成的。其中:

文档头:XML 声明、处理指令(可选)和注释(可选)等;

文档体:文档中内容信息所在的部分,即根元素的所有内容;

文档尾(可选):可包括注释、处理指令或空白。不正确使用文档尾会带来不必要的麻烦,所以大多数 XML 文档都没有文档尾。

### ①XML 声明

XML 规定每个正规的 XML 文档应当由一个 XML 声明开始,不允许在前面再有任何其他字符,甚至空格或者注释,就是说 XML 声明(如果有的话)必须是 XML 文档的第一个内容。例如:

```
<? xml version="1.0" encoding="gb2312" standalone="yes" ? >
```


<? xml:表示 XML 文档声明的开始,说明该文档是一个 XML 文档。

version="1.0":必须声明的 XML 版本信息,表示该文档遵循 W3C 的 XML 1.0 规范。

encoding="gb2312":如果是用简体中文编写 XML,就必须加上这项声明。encoding 是可选项,它的默认值是 UTF-8。

standalone="yes":表示该文档是一个独立的文档,没有引用其他外部文件。Standalone 是可选项,其值必须是 yes 或 no,如果 standalone="no",就意味着需要外部的 DTD,standalone 属性的默认值为 yes。

? >:表示文档声明结束。

 **注意:**如果使用 XML 声明,version 是必需的。如果使用任何非 UTF-8 或 UTF-16 的编码,则显式的 encoding 声明也是必需的。

虽然 XML 声明在一些 XML 文档中不是必需的(即声明可以省略,而文档仍然是正规的)。但是,我们建议读者养成一个好习惯,即应该总是将带有 version 和 encoding 的 XML 声明进行显式的定义,这样有助于清晰地把文档标识为 XML。

## ② 处理指令

处理指令(Processing Instruction,PI)是包含在 XML 文档中的一些命令性语句,其用途是为 XML 解析器提供传送给应用程序或脚本程序的信息,也可以利用处理指令为 XML 文档指定应用软件或显示文档的方法。

处理指令一般具有下列语法格式:

```
<? 处理指令名 处理指令信息? >
```


**说明:**

- 处理指令必须以“<?”开始,以“? >”结束。
- “处理指令的名称”用于指明哪个应用程序(或其他对象)需要 PI 来控制。指令名的大小写必需注意,并且这些指令名已被保留,不能再作为其他处理软件或脚本程序的名称。
- “处理指令信息”是指被传送到处理软件或脚本程序的信息,可由除了“? >”之外的任何连续字符组成。

例如:

```
<? xml-stylesheet type="text/css" href="例 1-7.css"? >
```

又如,要想通知 XML 解析器某篇 XML 文档所使用的编码字符集或有关版本信息等,就必须通过处理指令来实现。因此,XML 声明就是处理指令的一个特例,后面的样式表中我们还会讲到。

 **提示:**从严格意义上来说 XML 声明不是一条处理指令,它是由 XML 1.0 定义的唯一的声明。

## (2) XML 文档的元素

元素是 XML 内容的基本容器,可以包容字符数据、其他元素以及其他标记(注释、PI、实体引用等)。大多数 XML 数据(除了注释、PI 和空格)都必须包含在元素中。也就是说,XML 文档的元素包含了真正的数据信息,整个文档的数据内容就是由各种元素按照一定的逻辑结构组织而成的。

### ① 元素的组成

元素由起始标记、元素内容和结束标记组成。用户把要描述的数据对象放在起始标记

和结束标记之间。元素的一般语法格式为：

```
<标记名>元素内容</标记名>
```

说明：

a. XML 元素的标记名也就是元素的名称,故有时元素与标记二者不严格区分。  
b. 特殊字符必须被正确地转义。例如,“<”字符必须使用实体引用“&lt;”进行转义后代替。

c. 元素的起始标记和结束标记必须配对使用,元素可以正确地嵌套,但不能交叉。

d. 空元素标记(没有元素内容,可以省略结束标记)必须以“/”符号结束。

例如：

```
<片名>宫锁沉香</片名>
```

```
<类别>古装</类别>
```



**注意：**在开始的“<”与标记名之间不能有空格,结束标记中的斜杠只能紧跟在“<”开始的尖括号后,并且在标记中只能有一个斜杠“/”。同样,如果是空元素,在“/”和“>”之间不能有任何空格。

## ② 元素的命名规则

在 XML 文档中,可以按自己的需要来定义和使用元素名称,但需要遵守以下规则：

a. 英文名称必须以英文字母或者下划线“\_”开头,中文名称可以以汉字或者下划线“\_”开头。

b. 在使用默认编码集的情况下,名称可以由英文字母、数字、下划线“\_”、连接符“-”和句点“.”构成。

c. 在指定了编码集的情况下,则名称中除上述字符外,还可以出现该字符集中的合法字符。

d. 元素名称中不能含有空格。

e. 元素名称中的英文字母对大小写是敏感的。例如<name>、<NAME>与<Name>是三个不同的标记。

f. 另外,根据 XML 规范的定义,冒号“:”用作命名空间的分隔符,因此在普通标记名称中不应该使用。

g. 标记名称中不要以字符串 xml、XML 或任何按此顺序排列的这三个字母的各类组合(例如,xmL 或 Xml 等)开始。W3C 保留对这三个字母开头的使用权。



**提示：**元素名的命名规则也适用于属性和其他实体的命名规则。

## ③ 元素的类型

元素是 XML 文档的主要内容,大多数 XML 数据都必须包含在元素中。根据元素是否包含内容,XML 元素可分为非空元素和空元素两种类型。

### a. 非空元素

绝大多数 XML 元素在起始标记和结束标记之间都包含有元素内容,元素内容称为数据或信息,它们可以是文本,也可以是子元素。起始标记和结束标记将文档的数据进行结构化组织,并确定元素的范围和相互关系(父子关系或兄弟关系)。

**【例 1-2-2】** 非空元素举例。

```
<? xml version="1.0" encoding="gb2312"? >
```

```
<! --文件名:例 1-2-2.xml-->
```

```
<! --非空元素举例-->
```

```
<影片信息表>
```

```
<影片>
```

```
<片名>宫锁沉香</片名>
```

```
<类别>古装</类别>
```

```
<主演>周冬雨 陈晓 赵丽颖</主演>
```

```
<导演>潘安子</导演>
```

```
<简介>清朝康熙年间,兆佳沉香入宫遇了同齡的宫女琉璃,结为好友。琉璃一心想出人  
头地,企图邂逅的阿哥变凤凰,不惜背叛陷害沉香;而沉香则偶然与十三阿哥相爱,经过曲折坎坷,二人终于归  
隐远去,相爱厮守</简介>
```

```
<剧照 file="gscx.jpg"/>
```

```
</影片>
```

```
</影片信息表>
```

本例中,每个元素都有起始标记、结束标记和各自的内容,<影片信息表>是根元素。其中<片名>、<类别>、<主演>、<导演>和<简介>五个元素的内容是文本,它们是兄弟关系,<影片>元素虽然没有自己直接的文本内容,但它是上述五个元素的父元素,因而它们都是非空元素。

**b. 空元素**

空元素是指没有任何数据内容的元素,即在开始标记和结束标记之间既没有数据内容也没有子元素。例如下面的元素:

```
<image> </image>
```

人们通常使用空元素的简写形式,即仅使用一个单独的标记,而在标记名称的后面添加一个斜杠"/",其优点是既简洁又明确地指出该元素不应当有任何内容。例如上面的元素可简写为:

```
<image/>
```

空元素的常见应用是包含一个或多个属性。例如:

```
<剧照 file="gscx.jpg"/>
```

**c. 混合型元素**

在 XML 文档中,如果元素既有子元素又有自己的字符串内容,这种元素称为混合型元素,这种元素的内容称为混合型元素内容。

**【例 1-2-3】** 混合型内容元素举例。

```
<? xml version="1.0" encoding="gb2312"? >
```

```
<! --文件名:例 1-2-3.xml-->
```

```
<! --混合型内容元素举例-->
```

```
<影片信息表>
```

```
<影片>
```

```
新片基本信息
```

```

<片名>宫锁沉香</片名>
<类别>古装</类别>
<主演>周冬雨 陈晓 赵丽颖</主演>
<导演>潘安子</导演>

```

<简介>清朝康熙年间,兆佳沉香入宫遇了同齡的宫女琉璃,结为好友。琉璃一心想出人头地,企图邂逅的阿哥变凤凰,不惜背叛陷害沉香;而沉香则偶然与十三阿哥相爱,经过曲折坎坷,二人终于归隐远去,相爱厮守</简介>

```
<剧照 file="gscx.jpg"/>
```

```
</影片>
```

```
</影片信息表>
```

本例中,<影片>元素就是混合型元素。原则上讲,混合型元素破坏了文档的高度结构化,不利于应用软件对 XML 文档的处理,在实际开发过程中应该尽量避免这种情况,所以本教材不对这类元素及相关技术进行着重介绍。

#### ④ 元素的嵌套

一个格式正规的 XML 数据被定义为一棵简单的层次结构树形式,有且仅有一个顶层元素,称为文档元素或根元素,其他所有元素都必须被包含在这个根元素中。例如下面的例 1-2-4 表明了这一事实。

#### 【例 1-2-4】 影片信息表。

```
<? xml version="1.0" encoding="gb2312"? >
```

```
<! --文件名:例 1-2-4.xml-->
```

```
<影片信息表>
```

```
<影片>
```

```
<片名>宫锁沉香</片名>
```

```
<类别>古装</类别>
```

```
<主演>周冬雨 陈晓 赵丽颖</主演>
```

```
<导演>潘安子</导演>
```

<简介>清朝康熙年间,兆佳沉香入宫遇了同齡的宫女琉璃,结为好友。琉璃一心想出人头地,企图邂逅的阿哥变凤凰,不惜背叛陷害沉香;而沉香则偶然与十三阿哥相爱,经过曲折坎坷,二人终于归隐远去,相爱厮守</简介>

```
<剧照 file="gscx.jpg"/>
```

```
</影片>
```

```
<影片>
```

```
<片名>变身超人</片名>
```

```
<类别>喜剧</类别>
```

```
<主演>陈柏霖 郭雪芙 邱彦翔</主演>
```

```
<导演>张时霖</导演>
```

<简介>红极一时的超人剧男主角铁男由于收视率下滑严重,遭新任电视台董事长苏盈盈换角,歌手 FACE 成为新世代超人,而铁男则瞬间下岗。为改变现状,他不仅接了一堆类似路人乙的角色,终于在转战电视购物时,铁男下岗再就业打开了事业第二春。</简介>

```
<剧照 file="bscr.jpg"/>
```

```
</影片>
```



</影片信息表>

本例中,根元素为<影片信息表>,它由两个<影片>子元素组成。其中,每个<影片>子元素又包含<片名>、<类别>、<主演>、<导演>和<简介>共五个子元素。这些最里层的子元素只含有文本内容,没有子元素,称为叶子元素。所有元素共同组成一个树形结构,各元素间或是并列关系,或者是包含关系,没有出现交叉的情况。

在 IE 浏览器中,不同类型的内容采用不同的颜色来显示,单击左边的减号“-”可以将元素折叠,折叠后将在其前面显示为一个加号“+”。这一点很像我们在 Windows 系统中使用的“文件夹”。

XML 对元素有一个非常重要的要求:它们必须正确的嵌套。对于元素嵌套,有下列规则:

a. 根节点:一个格式正规的 XML 文档有且仅有一个根节点,称为文档的根,该根节点代表该文档本身,是一个 XML 文档的入口,根节点包含一个根元素。

b. 一个包含其他元素的元素称为父元素,而直接包含在父元素之下的元素称为该父元素的子元素。

c. 根元素:根元素是树中其他所有元素的父元素,XML 文档中其他所有元素都是根元素的后代。子元素还可以包含子元素。

d. 叶子元素:没有子元素的元素。

例如,下面两个元素的嵌套格式就是错误的。

<影片>

<片名>宫锁沉香</影片>

</片名>

### (3)XML 文档的属性

在 XML 文档中,数据主要以元素形式来描述和存储,而属性主要用来保存那些与元素及其内容相关的额外信息。元素的每个属性是一个名称-数值对,名称和数值分别为一个字符串。

属性必须在标记的开始标记中声明,语法格式为:

非空元素:<标记名 属性列表>内容</标记名>

空元素:<标记名 属性列表/>

例如:

<班级 上课时间="星期四第二大节" 教室="3#202">开发 04</班级>

<班级 上课时间="星期四第二大节" 教室="3#202" 班级名="开发 04"/>

说明:

①属性的命名规则与标记的命名规则相同。

②同一个元素中不能有两个相同名称的属性。

③属性值必须用(')或(")作为其界定符。

④用于分隔符的字符不能出现在字符串中,如果字符串中包含单引号,分隔符必须使用双引号,反之亦然。

⑤如果属性值同时包含单引号和双引号,则其中的引号就要使用预定义实体引用进行转义。

可以看出,数据既可以存储在元素中也可以用属性来表示。请看下面的例子:

**【例 1-2-5】** 数据存储在元素中。

```
<? xml version="1.0" encoding="gb2312"? >
<学生>
  <学号>20130114101</学号>
  <姓名>李倩</姓名>
  <性别>女</性别>
  <英语>85</英语>
  <数学>88</数学>
  <C语言>92</C语言>
</学生>
```

**【例 1-2-6】** 相关数据存储在属性中。

```
<? xml version="1.0" encoding="gb2312"? >
<学生>
  <姓名 学号="20130114101" 性别="女">李倩</姓名>
  <英语>85</英语>
  <数学>88</数学>
  <C语言>92</C语言>
</学生>
```

什么时候用属性,什么时候使用元素没有一个现成的规则可以遵循。一般情况下,尽量使用元素来描述数据,仅使用属性来描述那些与数据关系不大的附加信息。

#### (4) 字符数据

字符数据是一种纯文本,它不包含元素标记或其他标记。文本由字符组成,XML 使用 Unicode 字符集(统一的字符编码标准集“Universal Multiple-Octet Coded Character Set”,简称为 UCS,也可以看作是“Unicode Character Set”的缩写。),而不仅仅是 ASCII 码。

对于一些特殊的字符,比如,“&”和“<”都用作 XML 的起始定界符,因此绝不能以字符串形式出现(在 CDATA 节中例外)。如果要使用这些特殊的字符,就必须使用实体引用对它们进行转义。

另外,有些字符无法直接从键盘输入到文档中,比如希腊字母、象形汉字等,我们可以使用字符引用来解决这一问题。

##### ① 预定义实体引用

实体引用允许在元素内容或属性值中插入任何字符串。XML 预先定义了 5 个内建的实体引用(表 1-2-2),它们用作 XML 标记定界符的转义序列。预定义实体用于 XML 文档中的特殊字符,否则这些字符将被解释为标记的组成部分。

表 1-2-2

XML 预定义的实体引用

| 字符 | 实体引用   | 说明   |
|----|--------|--|
| <  | &lt;   | 通常用来替换,在 CDATA 节中除外(参见后面)                  |
| &  | &amp;  | 通常用来替换,在 CDATA 节中除外                        |
| >  | &gt;   | 可能用来替换,在 CDATA 节中,如果“>”后紧接着字符串“]”,就必须使用该实体 |
| "  | &quot; | 可用来替换字符串中的双引号(")                           |
| '  | &apos; | 可用来替换字符串中的单引号(')                           |

这些实体引用必须以“&.”开始,以“;”结束。解析器在处理文档时,将自动使用符号“>”来代替“&gt;”,用“&.”来代替“&amp;.”等。


**【例 1-2-7】** 预定义实体的引用。

```
<? xml version="1.0" encoding="GB2312"? >
<代码>if score &gt;= 90 then
    print "恭喜,优秀!"
else if score >= 60 then
    print "考试通过"
else
    print "哈哈,&lt; 60,重修了!"
endif
</代码>
```

预定义实体也能用于属性值中。

**【例 1-2-8】** 预定义实体在属性值中的应用。

```
<? xml version="1.0" encoding="gb2312"? >
<预定义实体 描述="预定义实体有 &amp; amp;、&amp; lt;、&amp; gt;、&amp; quot;和 &amp; apos;共 5
个,分别代表 &amp;、&lt;、&gt;、&quot;和 &apos;"/>
```

 **注意:** 只有“<”字符和“&.”字符对于 XML 来说是严格禁止使用的。剩下的都是合法的,为了减少出错,使用实体是一个好习惯。

## ② 字符引用

字符引用就是用 Unicode 字符值来引用该字符。字符引用通常用于表示一个可显示的字符,该字符不能被方便的输入到文档中。

字符引用由十进制或十六进制的数字在前面分别加上“& #”或“& # x”字符串,后面紧跟分号“;”组成。字符引用格式为:

&# NNNNN; 或 &# xXXXX;

其中,NNNNN 和 XXXX 字符串可以是 1~5 个十进制数字,或者是 1~4 个十六进制数字。例如:

```
<test>&# x 6C49;是“汉”字;&# 123;是左大括号;&# 169; is the same as &# xA9; and &# 174; is
the same as &# xAE;
</test>
```

通过使用相应的数字 Unicode 值,字符引用能够表达单一字符或者符号。例如:

```
&#x0394; &#x0393; &#x0396;
```

### 【例 1-2-9】 字符引用示例。

```
<? xml version="1.0" encoding="gb2312"? >
<字符引用>
  <女装>价格为:456&#xFFE5;</女装>
  <玩具>价格为:200&#xFE69;</玩具>
  <邮箱>email:wyj&#xFF20;zvti.edu.cn</邮箱>
</字符引用>
```

### (5)CDATA 节(CDATA 标记)

CDATA 节是一种用来包含文本的方法,它的作用是通知 XML 解析器,把 CDATA 节中的所有内容都当作纯文本来解析,而不把它们解释为标记、实体引用等,并原封不动的将这段代码传递给下一个 XML 应用程序。

CDATA 标记的基本格式为:

```
<! [CDATA[文本内容]]>
```

这里的“文本内容”可以是除了“]]>”之外的任何字符串。

CDATA 节主要用于需要把整个文本解释为纯字符数据而并不是标记的情况。当有一个包含许多“<”、“>”、“&.”或字符而非标记的庞大文本时,这是非常有用的。例如,对于大部分 C、VB 和 Java 等程序代码,最好把它们都放到 CDATA 节中。

### 【例 1-2-10】 带有 CDATA 节的 XML 文档。

```
<? xml version="1.0" encoding="gb2312"? >
<CDATA 节练习>
  <代码>
    <! [CDATA[
      while (p && j<i)
      { p=p->next; ++j; }
    ]]>
    <!--这是一段 C++ 代码-->
  </代码>
  <代码>
    <! [CDATA[
      <? xml version="1.0" ? >
      <Command1 Width="700" Height="475">
        <Caption>welcome</Caption>
        <Name>cmdwelcome</Name>
      </Command1>
    ]]>
    <!--这是一个 XML 文档-->
  </代码>
</CDATA 节练习>
```



**注意:**CDATA 节可以出现在字符数据能够出现的任何地方,CDATA 节不能嵌套。

## (6) 注释

在 XML 文档中适当地添加注释语句,有助于提高文档的可读性。XML 解析器通常忽略这些注释部分,不会对它们作任何处理。在注释部分中,标记不会被解释,实体引用也不会被展开。

注释语句格式为:

```
<!--注释文字-->
```

**【例 1-2-11】** 带有注释的 XML 文档。

```
<? xml version="1.0" encoding="gb2312" ? >
```

```
<!--有关本书的信息-->
```

```
<影片>
```

```
<片名>变身超人</片名>
```

```
<类别>喜剧</类别>
```

```
<!--
```

```
<主演>陈柏霖 郭雪芙 邱彦翔</主演>
```

```
<导演>张时霖</导演>
```

```
-->
```

<简介>红极一时的超人剧男主角铁男由于收视率下滑严重,遭新任电视台董事长苏盈盈换角,歌手 FACE 成为新世代超人,而铁男则瞬间下岗。为改变现状,他不仅接了一堆类似路人乙的角色,终于在转战电视购物时,铁男下岗再就业打开了事业第二春。</简介>

```
</影片>
```

在使用注释语句时,注意下列事项:

① 注释不可以放在 XML 声明之前;

② 注释不能放在标记中。例如,下面写法是非法的:

```
<price <!--price 默指为人民币-->26.00</price>
```

③ 两个连字符号"--"除了作为注释起始和结束标记的一部分外,不能再出现在注释中。

例如,下面写法是非法的:

```
<!--注释以连字符"--"结束-->
```

## (7) XML 文档类型

### ① 格式良好的 XML 文档(Well-formed)

所有遵守 XML 基本语法规则的文档都称为格式良好的(Well-formed)文档。这类文档在使用时可以不使用 DTD(文档类型定义)或 XML Schema(模式)来描述它们的结构。

对于一个 XML 文档来说,只有是格式良好的,才能够被解析器正确地解析,或是交给有关的应用程序处理。要保证所编写的 XML 文档是格式良好的,必须遵守下面几个准则:

XML 文档的第一行必须是声明,用以指明该文档是 XML 文档,它所使用的是 XML 规范版本。在文档的前面不能够有任何其他元素或者注释等信息。

元素构成一棵层次树,在一个 XML 文档中有且只有一个根元素。起始标记都有匹配的结束标记(空元素标记除外)。元素可以正确地嵌套,但不能交叉。元素名称、属性名称以及指令名称的大小写敏感。一个元素不能有两个同名的属性,且属性值必须用(')或(")括起来。特殊字符必须被正确地转义。没有对外部实体的引用(除非提供了 DTD)。

通常情况下,符合上述规定的 XML 文档就是格式良好的文档。这是编写 XML 文档的最基本要求。如果一个 XML 文档不是格式正规的,那么 XML 解析器在解析时就会给出相应的错误提示信息,并且不再会以正常的方式向应用程序传递字符数据和(或)XML 结构。

## ②有效的 XML 文档

如果一个格式良好的 XML 文档能够满足某些进一步有效性约束和匹配文档内容的语法,那么这个文档就被称为是有效的 XML 文档。这就是说,一个有效的 XML 文档不仅是格式良好的,而且要对所使用的标记和文档结构进行明确的定义。XML 能以 DTD 或 XML Schema 的形式提供文档结构的描述。

DTD 文档的主要功能是定义文档的逻辑结构,让用户知道所使用的 XML 文档需要包含哪些元素,元素中能够包括哪些子元素,各个元素出现的顺序,元素中的属性怎样定义等有效性约束。

Schema 文档和 DTD 文档的功能一样,都是用来定义标记和标记用法的语言。

有效的 XML 文档在实际应用中地位很重要,因为只有有效的 XML 文档才能被人很好地理解和被应用程序有效地处理。

## 2.XML 名称空间

在 XML 语言中允许自定义标记,那么在大型而复杂的 XML 文档中不可避免地会遇到标记名称相同但其代表的意义却不同的现象。尤其是在将多个 XML 文档组合使用时,名字冲突的情况会更容易发生。为了解决标记重名带来的冲突问题,XML 规范提供了名称空间的机制。当两个标记的名字相同时,可以通过隶属于不同的名称空间相互区分。通过名称空间的学习,要了解名称空间的基本概念,掌握名称空间的声明方法和使用方法,熟悉默认名称空间的声明方法和使用方法,掌握名称空间的作用范围。

### (1)名称空间的基本概念

名称空间就是在逻辑上相关的任何一组名称,在名称空间中,每个名称都必须是惟一的,并且是按照名称空间的规则来构建的。

在具体介绍名称空间之前,让我们看一个简单的 XML 文档。

**【例 1-2-12】** 存储员工信息和家属信息的 XML 文档。

```
<? xml version="1.0" encoding="gb2312"? >
```

```
<员工资料>
```

```
<员工>
```

```
<姓名>王小强</姓名>
```

```
<性别>男</性别>
```

```
<电话>13912345678</电话>
```

```
<部门>运输部</部门>
```

```
<家属>
```

```
<姓名>孙静</姓名>
```

```
<电话>023-67654321</电话>
```

```
</家属>
```

```
</员工>
```

```
</员工资料>
```

在这个文件中,<姓名>和<电话>两个元素在不同的上下文中使用了两次,一次是作为员工的资料,另一次是作为家属的,这两个标记应用的范围不同,而标记名却一样,尽管我们可以轻松地识别出它们的区别,但却会造成电脑解析时的混淆。

### ①名称空间的名字和前缀

名称空间的目的是有效地区分名字相同的标记。名称空间在 XML 文档中的具体应用是在标记名前添加一个全球唯一的标识符,在可能发生混淆的最外层标记中声明,这样相同的标记就因为有个唯一的标识符而可以区分了。为了保证这个标识符的唯一性,W3C 推荐使用统一资源标识符(Uniform Resource Identifier,URI)作为名称空间的名字。

在实践中,一般把网址作为名称空间的名字,因为网址总是独一无二的。但是,由于一般网址中的合法字符不能用于 XML 名称中,如字符“/”;而且,Internet 上的网址一般都很长,使用和阅读都不方便。为了解决这个问题,W3C 制订的解决方法是为 URI 创建一个别名(例如“tea”和“fam”),这就是所说的名称空间前缀。这个别名限定为一个合法的 XML 名称字符串,与网址一起声明于最外层的标记内,在文档中使用这个别名代替网址,对元素进行标识。

#### 【例 1-2-13】 使用名称空间的员工资料信息。

```
<? xml version="1.0" encoding="gb2312"? >
<员工资料 xmlns:tea="http://www.aaa.com.cn/ygxx" xmlns:fam="http://www.aaa.com.cn/jsxx">
  <tea:员工>
    <tea:姓名>王小强</tea:姓名>
    <tea:性别>男</tea:性别>
    <tea:电话>13912345678</tea:电话>
    <tea:部门>运输部</tea:部门>
    <fam:家属>
      <fam:姓名>孙静</fam:姓名>
      <fam:电话>023-67654321</fam:电话>
    </fam:家属>
  </tea:员工>
</员工资料>
```

本例中,在根元素<员工资料>的起始标记中声明了“tea”和“fam”两个标识符,分别代表员工名称空间(<http://www.aaa.com.cn/ygxx>)和家属名称空间(<http://www.aaa.com.cn/jsxx>)。在定义了名称空间后,在文档中应用元素时,都添加了相应的名称空间标识符(前缀),从而保证了每个元素仍然是唯一的名称。这样,处理程序就能明确地区分哪一个是员工的姓名和电话,哪一个又是家属的姓名和电话。

### ②名称空间的声明

XML 的名称空间提供了将名称空间 URI 和前缀相联系的机制,也就是名称空间声明。声明名称空间的基本语法为:

```
<元素名 xmlns:名称空间前缀="URI" >
```

例如:

```
<员工资料 xmlns:tea="http://www.aaa.com.cn/ygxx">
```

说明:


a. xmlns:指定名称空间的关键字。

b. 名称空间前缀:为了使用方便而随便起的名字,用来标识字符串 URI。例如,这里的 tea 就代表 http://www.aaa.com.cn/ygxx。XML 解析器看到一个名称空间声明后,就会把名称空间前缀与对应的 URI 绑定在一起。前缀的命名规则与元素的命名规则相同。

c. 名称空间声明时,如果不包含名称空间前缀,称为默认的名称空间声明。

d. URI:是一个有一定的语法、用来标识资源的字符串。一个 URI 可以是一个 E-mail 地址、一个 Internet 主机的域名、一个文件的绝对路径等。一定要记住,URI 代表的是定义标记的单位或组织,并不代表一种资源,它可以指向一盒饼干或一个玩具,甚至可以根本就不存在。

e. 虽然实例编写人员可以任意选择前缀,但对于某些名称空间,有一些约定俗成的前缀。例如,xsl 前缀通常映射到可扩展样式表语言(XSL)名称空间。具有相同情况的还有 HTML 等。

 **注意:**不允许把前缀映射到空字符串名称空间。在例 1-2-13 中,如果指定 xmlns:tea="" 将是非法的。

### ③ 名称空间的使用

XML 文档使用名称空间前缀来限定元素和属性的本地名称。在例 1-2-13 中,根元素 <员工资料> 中的每个子元素都有一个由两部分组成的名称:一个名称空间前缀和一个本地名称。例如:

```
<tea:姓名>
```

在声明了名称空间之后,其中的属性也可以添加名称空间前缀进行限定。

**【例 1-2-14】** 修改例 1-2-13,在其中添加“身份证号”和“单位”两个属性。

```
<? xml version="1.0" encoding="gb2312"? >
```

```
<员工资料 xmlns:tea="http://www.aaa.com.cn/ygxx" xmlns:fam="http://www.aaa.com.cn/jsxx">
```

```
  <tea:员工 tea:身份证号="510103198005082666">
```

```
    <tea:姓名>王小强</tea:姓名>
```

```
    <tea:性别>男</tea:性别>
```

```
    <tea:电话>13912345678</tea:电话>
```

```
    <tea:部门>运输部</tea:部门>
```

```
    <fam:家属 单位="市民政局财务处">
```

```
      <fam:姓名>孙静</fam:姓名>
```

```
      <fam:电话>023-67654321</fam:电话>
```

```
    </fam:家属>
```

```
  </tea:员工>
```

```
</员工资料>
```

说明:

a. 名称空间应用于它所声明的元素和属性中。


b. 限定的属性名称(例如“身份证号”)属于相关的名称空间。

c. 未限定的属性名称(例如“单位”)属于默认名称空间。



d. 通常情况下,属性存在于元素的起始标记内,只要该元素的标记名称能够被识别就可以了,很少有不同含义的同名元素中又有归属于不同名称空间的同名属性。因此,在实际文档中一般不需要为属性指定名称空间。

e. 很容易看出在每个独立的元素和属性名称上使用名称空间前缀相当乏味和麻烦,特别是在一个文档中的大多数名称都属于相同的名称空间时。对于这个问题,可以通过默认名称空间来解决。

 **注意:** 限定名称不会匹配使用相同的本地部分的未限定名称。例如, <tea:姓名> 不会和 <姓名> 相同。

## (2) 默认名称空间

在 XML 文档中,可将文档中使用较多的名称空间声明为默认名称空间。这种情况下,如果为某个元素声明了默认名称空间,则该元素及其所有未加名称空间前缀的子元素(不包括属性)均将属于这个默认名称空间的控制范围,除非它们受到另一个名称空间的限制。

默认名称空间声明格式为:

```
<元素名 xmlns="URI" >
```

**【例 1-2-15】** 在根元素中使用默认名称空间。

```
<? xml version="1.0" encoding="gb2312"? >
<员工资料 xmlns="http://www.aaa.com.cn/ygxx" xmlns:fam="http://www.aaa.com.cn/jsxx">
  <员工>
    <姓名>王小强</姓名>
    <性别>男</性别>
    <电话>13912345678</电话>
    <部门>运输部</部门>
    <fam:家属>
      <fam:姓名>孙静</fam:姓名>
      <fam:电话>023-67671321</fam:电话>
    </fam:家属>
  </员工>
</员工资料>
```

如前所述,一个默认名称空间的作用范围仅限于定义这个名称空间的元素及其子元素。在本例中,由于在根元素<员工资料>的起始标记处声明了默认名称空间,因此在整个文档中,凡未加名称空间前缀“fam”的元素(不包括属性)都属于默认名称空间“http://www.aaa.com.cn/ygxx”。

如果在一个元素中声明了默认名称空间,则在该元素的子元素中仍然可以再声明其他默认名称空间,这时,在子元素上声明的默认名称空间的作用范围就在该子元素上。

**【例 1-2-16】** 在子元素中再声明默认名称空间。

```
<? xml version="1.0" encoding="gb2312"? >
<员工资料 xmlns="http://www.aaa.com.cn/ygxx">
  <员工>
    <姓名>王小强</姓名>
    <电话>13912345678</电话>
  </员工>
```

```

<员工 xmlns="http://www.aaa.com.cn/teshuygxx">
  <姓名>袁豆豆</姓名>
  <电话>15323456766</电话>
</员工>
<员工>
  <姓名>齐大胜</姓名>
  <电话>13823415888</电话>
</员工>
</员工资料>

```

本例中,第 2 个<员工>元素及其子元素都在名称空间“http://www.aaa.com.cn/teshuygxx”的控制中,在这个作用域之外的其他元素都属于名称空间“http://www.aaa.com.cn/ygxx”的作用范围内。

另外,如果一个默认名称空间声明的 URI 的属性值是空的,那么这个元素就不在任何名称空间内。换句话说,可以通过在默认的名称空间声明中使用一个空属性值,来取消默认的名称空间。

**【例 1-2-17】** 修改例 1-2-15,取消默认名称空间。

```

<? xml version="1.0" encoding="gb2312"? >
<员工资料 xmlns="http://www.aaa.com.cn/ygxx" xmlns:fam="http://www.aaa.com.cn/jsxx">
  <员工>
    <姓名>王小强</姓名>
    <性别>男</性别>
    <电话>13912345678</电话>
    <部门>运输部</部门>
    <fam:家属 xmlns="">
      <姓名>孙静</姓名>
      <电话>023-67654321</电话>
    </fam:家属>
  </员工>
</员工资料>

```

本例中,在<家属>元素内任何未限定的元素名称都不会属于任何的默认名称空间。在这个作用域之外的元素名称则继续属于前面声明的默认名称空间。

### (3) 名称空间的作用范围

一个名称空间的作用范围是声明该名称的元素及其子元素,除非是在该元素的某一个子元素上又声明了相同的名称空间前缀。

出于上述原因,一个很常用的 XML 名称空间的使用就是利用继承的规则,将名称空间在根元素中声明,这样的名称空间的作用域将是整个文档,虽然这会丧失一部分的灵活性,但却能大大减少出错的机会。

### (4) 混合使用 XML 和 HTML 标记

利用名称空间机制,可以实现在 XML 文档中直接使用 HTML 标记,就像在 HTML 文档中一样,当前主要的 XML 浏览器都支持这种用法,在显示 XML 文档时,会将其中的

HTML 标记正确表示出来。

在 XML 文档中使用 HTML 标记之前,首先必须声明它的名称空间。例 1-2-18 就是一个 XML 和 HTML 混合使用的例子。

**【例 1-2-18】** 使用英文字符串作标记名的影片信息表。

```
<? xml version="1.0" encoding="gb2312"? >
<! --文件名:例 1-2-18.xml-->
<? xml-stylesheet type="text/css" href="例 1-2-19.css"? >
<Film-info xmlns:html="http://www.w3.org/1999/xhtml">
  <film >
    <html:img src="gscx.jpg"/>
    <name>宫锁沉香</name>
    <ToStar >主演:周冬雨 陈晓 赵丽颖</ToStar >
  </film>
  <film>
    <html:img src="bscr.jpg"/>
    <name>变身超人</name>
    <ToStar >主演:陈柏霖 郭雪芙 邱彦翔</ToStar >
  </film>
</Film-info>
```

**【例 1-2-19】** 应用于“例 1-2-18.xml”的样式表文件。

```
/* 文件名:例 1-2-19.css */
/* 影片信息以块方式显示,左页边距为 0.5 厘米,上页边距为 0.2 厘米 */
Film-info, film{display:block;margin-left:0.5cm; margin-top:0.2cm;}
/* 片名以块方式显示,字体大小为 16 磅,红色,粗体,其他字体为 10 磅 */
name, ToStar{display:block;font-size:16pt;color:red; font-weight:bold; }
```

其中使用 HTML 标记时要严格遵守 XML 的语法规则,每个元素必须正确关闭,其显示效果如图 1-2-1 所示。







图 1-2-1 XML 文档混合 HTML 元素显示图

### 1.2.3 知识扩展:菜单信息表

表 1-2-3

菜单信息表

| 菜品名   | 价格     | 制作描述  | 图片  |
|-------|--------|---|---|
| 肥肠粉   | ¥12.00 | 先将泡好的红薯放进沸腾的肥肠汤锅里来回荡几下,捞起来后倒入已放好肥肠和调料的大口碗里,最后从汤锅里舀一大勺沸汤沿粉丝浇淋下去,再往碗里放几粒花生和几抹葱花即可。                              | <br>fcf.jpg    |
| 水煮肉片  | ¥28.00 | 先将芹菜、蒜苗切段,青笋切片;再将肉片加少量淀粉和匀,旺火下少许油,加豆瓣炒至油成红色,加少量汤,待汤烧滚,蔬菜、肉片通通丢进去,略煮(不可久煮)然后倒入盆中;最后将花椒面、辣椒面大把大把地撒下去,再淋上一层滚油即可。 | <br>szrp.jpg   |
| 麻婆豆腐  | ¥15.00 | 选石膏豆腐切四方丁放碗中,用开水泡去涩味。烧热炒锅下菜油,烧至六成热,将剁细的牛肉末炒散至呈黄,加盐、豆豉、郫县豆瓣再炒,加鲜肉汤,下豆腐,用中火烧至豆腐入味,再下青蒜苗节、酱油,略烧片刻即勾芡收汁,撒葱末即成。    | <br>mpdf.jpg   |
| 鱼香牛肉丝 | ¥32.00 | 将牛肉丝用鸡蛋白、干菱粉、精盐搅拌均匀,下锅炒一下取出。另将笋丝、泡辣椒丝入猪油锅炒一下,再将牛肉丝加入,用旺火炒十多秒钟,沥去油,将准备好的姜末、蒜泥、糖、黄酒、醋、辣油、菱粉、葱花、酱油、味精淋上。         | <br>yxnr.s.jpg |

将上表的数据信息用 XML 文档存储。XML 文档编辑如下:

```
<? xml version="1.0" encoding="gb2312"? >
<! --Edited with XML Spy2012 :menu-info.xml-->
<菜单>
```

```
<菜品>
```

```
<名称>肥肠粉</名称>
```

```
<价格>¥12.00</价格>
```

```
<图片 file="fcf.jpg"/>
```

```
<制作方法>先将泡好的红薯放进沸腾的肥肠汤锅里来回荡几下,捞起来后倒入已放好肥肠和调料的大口碗里,最后从汤锅里舀一大勺沸汤沿粉丝浇淋下去,再往碗里放几粒花生和几抹葱花即可。</制作方法>
```

```
</菜品>
```

```
<菜品>
```

```
<名称>水煮肉片</名称>
```

```
<价格>¥30.00</价格>
```

```
<图片 file="szrp.jpg"/>
```

```
<制作方法>先将芹菜、蒜苗切段,青笋切片;再将肉片加少量淀粉和匀,旺火下少许油,加豆瓣炒至油成红色,加少量汤,待汤烧滚,蔬菜、肉片通通丢进去,略煮(不可久煮)然后倒入盆中;最后将花椒面、辣椒面大把大把地撒下去,再淋上一层滚油即可。</制作方法>
```

</菜品>

<菜品>

<名称>麻婆豆腐</名称>

<价格>¥15.00</价格>

<图片 file="mpdf.jpg"/>

<制作方法>选石膏豆腐切四方丁放碗中,用开水泡去涩味。烧热炒锅下菜油,烧至六成热,将剁细的牛肉末炒散至呈黄,加盐、豆豉、郫县豆瓣再炒,加鲜肉汤,下豆腐,用中火烧至豆腐入味;再下青蒜苗节、酱油,略烧片刻即勾芡收汁,撒葱末即成。</制作方法>

</菜品>

<菜品>

<名称>鱼香牛肉丝</名称>

<价格>¥32.00</价格>

<图片 file="yxnrs.jpg"/>

<制作方法>将牛肉丝用鸡蛋白、干菱粉、精盐搅拌均匀,下锅炒一下取出。另将笋丝、泡辣椒丝入猪油锅炒一下,再将牛肉丝加入,用旺火炒十多秒钟,沥去油,将准备好的姜末、蒜泥、糖、黄酒、醋、辣油、菱粉、葱花、酱油、味精淋上。</制作方法>

</菜品>

</菜单>

 **注意:**这里的图片信息是以其图片文件名形式存储,在后面的样式表课程中将会讲解其图片的显示方法。

## 综合实训:影碟出租管理系统

本节我们利用前面学习的XML基本语法知识做一个小项目:影碟出租管理系统。由于我们还只是讨论了基本的XML语法,所以下面的例子是相当简单的。

在系统设计中,将要用到“影碟”、“租用者”和“租用”3个表,系统目前有表1-2-4~表1-2-6的测试数据。

表 1-2-4

“影碟”信息表

| 影碟编号      | 碟名    | 类别  | 分类 | 导演  | 主演      | 碟数 | 在库碟数 | 影片图片   |
|-----------|-------|-----|----|-----|---------|----|------|--|
| DSJ100103 | 天龙八部  | 电视剧 | 武侠 | 赖水清 | 钟汉良 张檬  | 38 | 0    | <br>TLBB.JPG  |
| DSJ200202 | 真爱黑白配 | 电视剧 | 爱情 | 张普荣 | 胡宇崴 陈庭妮 | 21 | 21   | <br>ZAHBP.JPG |

(续表)

| 影碟编号      | 碟名    | 类别  | 分类 | 导演    | 主演      | 碟数 | 在库碟数 | 影片图片  |
|-----------|-------|-----|----|-------|---------|----|------|---|
| DSJ200213 | 咱们结婚吧 | 电视剧 | 爱情 | 刘江    | 高圆圆 黄海波 | 50 | 50   | <br>ZMJHB.JPG  |
| DSJ300112 | 精忠岳飞  | 电视剧 | 历史 | 鞠觉亮   | 黄晓明 林心如 | 69 | 39   | <br>JZYF.JPG   |
| DSJ400131 | 老有所依  | 电视剧 | 家庭 | 赵宝刚   | 刘涛 张铎   | 41 | 41   | <br>LYSY.JPG   |
| DY100124  | 非常幸运  | 电影  | 爱情 | 丹尼·戈登 | 章子怡 王力宏 | 1  | 1    | <br>FCXY.JPG   |
| DY100134  | 小时代 1 | 电影  | 爱情 | 郭敬明   | 杨幂 柯震东  | 1  | 1    | <br>XSD1.JPG  |
| DY200164  | 小时代 2 | 电影  | 爱情 | 郭敬明   | 杨幂 柯震东  | 1  | 1    | <br>XSD2.JPG |

表 1-2-5

“租用者”信息表

| 租用证号        | 姓名  | 性别 | 押金(元) | 租金(元) | 办证时间       |
|-------------|-----|----|-------|-------|------------|
| YS201207023 | 王如画 | 女  | 100   | 100   | 2012.07.25 |
| YS201301005 | 刘云帆 | 男  | 100   | 120   | 2013.01.20 |
| YS201301013 | 李明峰 | 男  | 100   | 150   | 2013.01.23 |
| YS201304006 | 程欣  | 女  | 100   | 80    | 2013.04.27 |
| YS201310013 | 曾国林 | 男  | 100   | 50    | 2013.10.28 |

表 1-2-6

“租用”信息表

| 租用证号        | 影碟编号      | 租用碟数 | 租用明细    | 租用/归还日期    | 归还否 |
|-------------|-----------|------|---------|------------|-----|
| YS201207023 | DSJ200202 | 11   | 11~21 集 | 2013-10-12 | 是   |
| YS201301005 | DSJ100103 | 20   | 1~20 集  | 2014-02-16 | 否   |
| YS201310013 | DSJ100103 | 18   | 21~38 集 | 2014-02-12 | 否   |
| YS201304006 | DY100134  | 1    | /       | 2014-02-15 | 是   |
| YS201301013 | DSJ300112 | 30   | 1~30 集  | 2014-02-17 | 否   |

系统要求:设计一套该系统特有的标记语言来保存上面表中的信息。对于不同类型的信息,我们使用元素和属性的混合形式。这里有一个原则:信息应尽可能地使用元素方式来保存,比如碟名、导演、类别等,因为属性方式的信息破坏了 XML 文件中信息保存的数据结构特性。而对于标识性信息,例如影碟编号、租用证号等标志类信息则通常使用属性的方式来表示。

**【例 1-2-20】** 影碟租用管理系统 dvd-lend-system.xml。

```
<? xml version="1.0" encoding="GB2312"? >
<! --文件名:dvd-lend-system.xml-->
<影碟租用管理系统>
<! --这是一个影碟租用管理系统的部分数据信息-->
<影碟信息>
  <影碟 影碟编号="DSJ100103">
    <碟名>天龙八部</碟名>
    <类别>电视剧</类别>
    <分类>武侠</分类>
    <导演>赖水清</导演>
    <主演>钟汉良 张檬</主演>
    <碟数>38</碟数>
    <在库碟数>0</在库碟数>
    <剧照>TLBB.JPG</剧照>
  </影碟>
  <影碟 影碟编号="DSJ200202">
    <碟名>真爱黑白配</碟名>
    <类别>电视剧</类别>
    <分类>爱情</分类>
    <导演>张普荣</导演>
    <主演>胡宇崴 陈庭妮</主演>
    <碟数>21</碟数>
    <在库碟数>21</在库碟数>
    <剧照>ZAHBP.JPG</剧照>
  </影碟>
  <影碟 影碟编号="DSJ200213">
```

```
<碟名>咱们结婚吧</碟名>
<类别>电视剧</类别>
<分类>爱情</分类>
<导演>刘江</导演>
<主演>高圆圆 黄海波</主演>
<碟数>50</碟数>
<在库碟数>50</在库碟数>
<剧照>ZMJHB.JPG</剧照>
</影碟>
<影碟 影碟编号="DSJ300112">
  <碟名>精忠岳飞</碟名>
  <类别>电视剧</类别>
  <分类>历史</分类>
  <导演>鞠觉亮</导演>
  <主演>黄晓明 林心如</主演>
  <碟数>69</碟数>
  <在库碟数>39</在库碟数>
  <剧照>JZYF.JPG</剧照>
</影碟>
<影碟 影碟编号="DSJ400131">
  <碟名>老有所依</碟名>
  <类别>电视剧</类别>
  <分类>家庭</分类>
  <导演>赵宝刚</导演>
  <主演>刘涛 张铎</主演>
  <碟数>41</碟数>
  <在库碟数>41</在库碟数>
  <剧照>LYSY.JPG</剧照>
</影碟>
<影碟 影碟编号="DY100124">
  <碟名>非常幸运</碟名>
  <类别>电影</类别>
  <分类>爱情</分类>
  <导演>丹尼戈登</导演>
  <主演>章子怡 王力宏</主演>
  <碟数>1</碟数>
  <在库碟数>1</在库碟数>
  <剧照>FCXY.JPG</剧照>
</影碟>
<影碟 影碟编号="DY100134">
  <碟名>小时代 1</碟名>
  <类别>电影</类别>
```



```
<分类>爱情电影</分类>
<导演>郭敬明</导演>
<主演>杨幂 柯震东</主演>
<碟数>1</碟数>
<在库碟数>1</在库碟数>
<剧照>XSD1.JPG</剧照>
</影碟>
<影碟 影碟编号="DY200164">
  <碟名>小时代 2</碟名>
  <类别>电影</类别>
  <分类>爱情电影</分类>
  <导演>郭敬明</导演>
  <主演>杨幂 柯震东</主演>
  <碟数> 1</碟数>
  <在库碟数>1 </在库碟数>
  <剧照>XSD2.JPG </剧照>
</影碟>
</影碟信息>
<租用者信息>
  <租用者 租用证号="YS201207023">
    <姓名>王如画</姓名>
    <性别>女</性别>
    <押金>100</押金>
    <租金>100</租金>
    <办证时间>2012-07-25</办证时间>
  </租用者>
  <租用者 租用证号="YS201301005">
    <姓名>刘云航</姓名>
    <性别>男</性别>
    <押金>100</押金>
    <租金>120</租金>
    <办证时间>2013-01-20</办证时间>
  </租用者>
  <租用者 租用证号="YS201301013">
    <姓名>李明峰</姓名>
    <性别>男</性别>
    <押金>100</押金>
    <租金>150</租金>
    <办证时间>2013-01-23</办证时间>
  </租用者>
  <租用者 租用证号="YS201304006">
    <姓名>程欣</姓名>
```

```
<性别>女</性别>
<押金>100</押金>
<租金>80</租金>
<办证时间>2013-04-27</办证时间>
</租用者>
<租用者 租用证号="YS201310013">
  <姓名>曾国林</姓名>
  <性别>男</性别>
  <押金>100</押金>
  <租金>50</租金>
  <办证时间>2013-10-28</办证时间>
</租用者>
</租用者信息>
<租用信息>
  <租用 租用证号="YS201207023" 影碟编号="DSJ200202">
    <租用碟数>11</租用碟数>
    <租用明细>11-21 集</租用明细>
    <租用或归还日期>2013-10-12</租用或归还日期>
    <归还否>是</归还否>
  </租用>
  <租用 租用证号="YS201301005" 影碟编号="DSJ100103">
    <租用碟数>20</租用碟数>
    <租用明细>1-20 集</租用明细>
    <租用或归还日期>2014-02-16</租用或归还日期>
    <归还否>否</归还否>
  </租用>
  <租用 租用证号="YS201310013" 影碟编号="DSJ100103">
    <租用碟数>18</租用碟数>
    <租用明细>21-38 集</租用明细>
    <租用或归还日期>2014-02-12</租用或归还日期>
    <归还否>否</归还否>
  </租用>
  <租用 租用证号="YS201304006" 影碟编号="DY100134">
    <租用碟数>1</租用碟数>
    <租用明细></租用明细>
    <租用或归还日期>2014-02-15</租用或归还日期>
    <归还否>是</归还否>
  </租用>
  <租用 租用证号="YS201301013" 影碟编号="DSJ300112">
    <租用碟数>30</租用碟数>
    <租用明细>1-30 集</租用明细>
    <租用或归还日期>2014-02-17</租用或归还日期>
```

```
<归还否>否</归还否>
</租用>
</租用信息>
</影碟租用管理系统>
```

## 练 习

### 一、选择题

- XML 文档中一般可以包括( )。  
A. XML 文档的声明  
B. XML 文档类型定义  
C. XML 文档注释  
D. XML 标识及其内容  
E. 以上答案都不对
- 以下关于 XML 文档的声明的语句正确的是( )。  
A. `<xml version="1.0" encoding="UTF-8">`  
B. `<? XML version="1.0" encoding="UTF-8"? >`  
C. `<? xml Version="1.0" encoding="UTF-8"? >`  
D. `<? xml version="1.0" encoding="UTF-8" ? >`  
E. `<? xml version="1.0" Encoding="UTF-8"? >`
- 以下说法符合 XML 语法规则的是( )。  
A. 标记头和标记末的大小写一致  
B. 元素之间要正确的嵌套  
C. 结束标记可有可无  
D. 每个 XML 文档只能有一个根元素  
E. 每个 XML 文档可以有多个根元素
- 以下关于属性的写法正确的是( )。  
A. `<root attribute="first"/>`  
B. `<root attribute=first />`  
C. `<root attribute/>`  
D. `<root attribute="first" value="aaa"/>`  
E. `<root attribute="firs" value="aaa">`
- 以下说法中哪些是正确的( )。  
A. 结构良好的 XML 文档就是有效的 XML 文档  
B. 结构良好的 XML 文档就是符合 XML 语法规范的文档  
C. 有效的 XML 文档就是结构良好的 XML 文档  
D. 有效的 XML 文档是指通过了 DTD 验证的 XML 文档  
E. 以上说法都不正确
- 如果需要在 XML 文件中显示简体中文,那么 encoding=( )。  
A. GB2312  
B. BIG5  
C. UTF-8  
D. UTF-16
- XML 是( )。  
A. 一种标准泛用标记语言  
B. 一种扩展性标识语言

- C. 一种超文本标记语言  
D. 一种层叠样式表单是 .NET 托管程序的执行引擎
8. ( ) 注释是正确的形式。  
A. `<! 这个注释是正确的>`                      B. `<!--这个注释是正确的-->`  
C. `<!--这个注释是正确的-->`                      D. `<! --这个注释是正确的! -->`
9. 对 XML 进行验证的前提条件是( )。  
A. XML 文档的数据模型是完整的                      B. 该 XML 文档的定义是正确的  
C. 该 XML 文档的数据是正确的                      D. 该 XML 文档的格式是正确的
10. 下面( )不是 HTML 的优点。  
A. 跨平台                      B. 强大的信息展示能力  
C. 标记可自定                      D. 编写简单
11. 一个标记可有多个属性, 下列关于服装的属性, ( ) 表示是错误的。  
A. `<服装 颜色="红色" 单位="件">`  
B. `<服装 颜色="红色" 颜色="黑色">`  
C. `<服装 颜色="红色" 规格="L">`  
D. `<服装 类别="冬衣" 颜色="黑色">`
12. 名称空间声明有直接定义和缺省定义两种方式, 其中直接定义的格式是( )。  
A. `xmlns:名称空间前缀="名称空间名"`  
B. `xmlns:名称空间名="名称空间前缀"`  
C. `xmlns 名称空间前缀="名称空间名"`  
D. `xmlns:名称空间前缀=名称空间名`
13. 在 XML 文件中, 包含文件中所有数据信息的元素称根元素。一个 XML 文件有( ) 个根元素。  
A. 1                      B. 2                      C. 3                      D. 4
14. 在使用名称空间(namespace)的时候, 与标签`<企业经理:电话>`相对的结束标签为( )。  
A. `</企业经理:电话:>`                      B. `<企业经理/:电话:/>`  
C. `<企业经理:/电话:>`                      D. `<企业经理:电话:/>`
15. 元素属性的名称空间是( )。  
A. 应与元素的名称空间一致                      B. 可以与元素的名称空间不一致  
C. 可使用默认的名称空间                      D. 都不是
16. 下面有几个空标记写法, 其中正确的是( )。  
A. `<肉/>`                      B. `</肉>`                      C. `<肉>`                      D. `</肉/>`

## 二、填空题

- 一个元素开始的分隔符被称作\_\_\_\_\_, 一个元素最后的分隔符被称作\_\_\_\_\_。
- 一共有四类元素: \_\_\_\_\_; 仅含文本的; \_\_\_\_\_; 含子元素和文本的混合元素。
- XML 文档内容的主体部分, 一般由\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, 注释和内容组成。
- XML 对属性的命名作了严格的要求, 但属性值却非常灵活。XML 要求属性值必须

用\_\_\_\_\_括起来的一个字符串。

### 三、改错题

1. <? xml ? >

```
<books>
```

```
<book>
```

```
  <name>xml</name>
```

```
  <auther>sss</auther>
```

```
</book>
```

```
</books>
```

2. <! --这是有关一本书的信息-->

```
<? xmlversion="1.0" ? >
```

```
<books>
```

```
  <book>
```

```
    <name>xml</name>
```

```
    <auther>sss</auther>
```

```
</books>
```

3. <? xml version="1.0" encoding="gb2312"? >

```
<book>
```

```
  <name>xml 程序设计</name>
```

```
  <auther>栗松涛</auther>
```

```
  <publish>清华大学出版社</publish>
```

```
  <price>40.00</price>
```

```
</book>
```

```
<book>
```

```
  <name>ASP 应用</name>
```

```
  <auther>李杰</auther>
```

```
  <publish>清华大学出版社</publish>
```

```
  <price>50.00</price>
```

```
</book>
```

### 四、按要求编写 XML 文档

编写一个 XML 文档,其中有两个班级子元素(开发 031、开发 032),在班级子元素中又有若干个学生子元素,其中必包含学生个人的真实信息(学号、姓名、性别、爱好等)。

1. 什么是 XML? XML 的特点有哪些?

2. 写出 XML 的文档规则。

3. XML 元素命名规范是什么?