



国家级示范性高等院校精品规划教材

XML 程序设计

XML CHENGXU SHEJI

周从军 曾鸿 周竞鸿 王勤龙/编著



天津大学出版社
TIANJIN UNIVERSITY PRESS

要 目 录

本书主要介绍XML及其在Web中的应用。全书共分10章。第1章介绍XML的起源、应用及XML文档的表示方法。第2章介绍XML文档的解析。第3章介绍XML文档的生成。第4章介绍XML文档的传输。第5章介绍XML文档的存储。第6章介绍XML文档的查询。第7章介绍XML文档的验证。第8章介绍XML文档的加密。第9章介绍XML文档的签名。第10章介绍XML文档的集成。

国家级示范性高等院校精品规划教材

XML 程序设计

周从军 曾 鸿 周竞鸿 王勤龙/编著

图书在版编目(CIP)数据

XML 程序设计/周从军等编著. —北京:天津大学出版社, 2010.3
ISBN 978-7-304-03412-5

I. ①X... II. ①周... III. ①XML—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2010)

出版发行 天津大学出版社

地址 天津市五经路92号(邮编:300072)

电话 发行部:022-5740364

网址 www.tjup.com

印刷 河北昌泰印务有限公司

装订 全国各新华书店

开本 182mm×260mm

印张 11.75

字数 370千字

版次 2010年3月第1版

印次 2010年3月第1次

印数 1—5000

定价 39.00元



天津大学出版社

TIANJIN UNIVERSITY PRESS

凡购本书,如蒙不弃,请向本社索取样书,恕不另加邮费。

版权所有 侵权必究

内 容 提 要

本书按照 XML 及其应用的实际工作过程,通过案例详细地介绍了 XML 语言的使用方法。第一部分(第1章)是 XML 基础知识;第二部分(第2章和第3章)是 XML 数据的有效性定义,包括 DTD 技术和 XML Schema 技术;第三部分(第4章和第5章)是 XML 数据的显示,包括 CSS 和 XSL 技术;第四部分(第6章)是在 HTML 中嵌入 XML 数据及其显示技术;第五部分(第7章)是使用 DOM 访问和修改 XML 文档。

本书采用任务驱动的方式进行编写,每一部分内容都给出了相应的应用环境及解决步骤和方法,方便读者掌握 XML 在实际应用中的工作方法。本书可作为高职高专学生学习 XML 技术的专用教材,也可以作为程序员学习 XML 技术的参考资料。

王 欢 周 军 等 编 著

图书在版编目(CIP)数据

XML 程序设计/周从军等编著. —天津:天津大学出版社, 2010.3

国家级示范性高等院校精品规划教材

ISBN 978-7-5618-3415-2

I. ① X... II. ① 周... III. ① 可扩展语言, XML—程序设计—高等学校—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2010)第 034295 号

出版发行 天津大学出版社

出 版 人 杨 欢

地 址 天津市卫津路 92 号天津大学内(邮编: 300072)

电 话 发行部: 022-27403647 邮购部: 022-27402742

网 址 www.tjup.com

印 刷 河北省昌黎县第一印刷厂

经 销 全国各地新华书店

开 本 185mm×260mm

印 张 11.75

字 数 270 千

版 次 2010 年 3 月第 1 版

印 次 2010 年 3 月第 1 次

印 数 1—2000

定 价 29.00 元

凡购本书,如有缺页、倒页、脱页等质量问题,请向我社发行部联系调换

版权所有 侵权必究

前 言

XML 即可扩展标记语言 (Extensible Markup Language), 是一种与平台无关的表示数据的方法。简单地说, 使用 XML 创建的数据可以被任何应用程序在任何平台上读取, 它是标准通用标记语言 SGML (Standard Generalized Markup Language) 的一个子集, 兼有超文本标记语言 HTML (Hyper Text Markup Language) 和 SGML 二者之长, 既全面通用, 又简洁明晰。XML 的使用正越来越普及, 越来越多的领域或环境下都采用 XML 来实现需要的功能。

本书按照 XML 及其应用的实际工作过程, 采用“案例教学、任务驱动”模式来编写。每章都先提出在 XML 应用中必须解决的某一问题, 然后根据问题精心设计了案例, 并将案例分解为若干任务。针对提出的任务, 再对所需的知识点进行讲授, 最后利用相关知识解决问题。每章都有相应的实训项目和习题, 可以帮助学生进一步巩固所学知识。通过本书, 读者既掌握了 XML 的相关知识, 也熟悉了 XML 在实际工作中的应用方法。

本书详细地介绍了 XML 语言的使用方法, 从具体问题入手, 通过大量的程序实例来讲述 XML 语法, 使用 DTD 和 XML Schema 定义 XML 文档的结构, 使用 CSS 和 XSL 格式化 XML 文档, 使用数据岛显示 XML 数据, 使用 DOM 访问和修改 XML 文档。

本书由湖北国土资源职业学院的周从军、周竞鸿, 湖北职业技术学院的王斌, 襄樊职业技术学院的曾鸿、王勤龙共同编著。周竞鸿编著第 1 章, 曾鸿编著第 2 章、第 3 章, 王勤龙编著第 4 章, 王斌编著第 6 章, 周从军编著第 5 章、第 7 章。全书由周从军统稿。咸宁职业技术学院的胡新和教授审阅了全书, 并提出了许多宝贵意见和建议。

由于编者水平有限, 编写时间仓促, 书中难免有不妥之处, 敬请广大读者批评指正, 以便下次修订时完善。

《XML 程序设计》编委会

2010 年 3 月

目 录

第1章 XML 基础	1
1.1 XML 简介	1
1.2 创建一个 XML 文档	3
案例 创建 XML 文档并检查该文档的良好性	3
任务 1 确定存储数据的方法	3
任务 2 确定存储数据的文档格式	3
任务 3 创建一个 XML 文档存储数据	15
任务 4 检查文档格式的正确性	17
1.3 该你动手了	20
1.4 小结	20
1.5 习题	20
第2章 用 DTD 验证 XML	23
2.1 DTD 简介	23
2.2 创建 DTD	23
案例 1 创建 DTD 文档	23
任务 1 确定需要存储的元素和属性	24
任务 2 确定声明元素的方法	24
任务 3 确定声明属性的方法	28
任务 4 确定声明实体的方法	33
任务 5 建立 DTD 文件	37
2.3 利用 DTD 验证 XML	41
案例 2 利用 DTD 验证 XML 文档的有效性	41
任务 1 创建一个 XML 文档	41
任务 2 关联 XML 文档与 DTD	42
任务 3 下载和安装 Altova XMLSpy	44
任务 4 验证 XML 文档有效性	46
2.4 该你动手了	47
2.5 小结	47
2.6 习题	48
第3章 使用 XML Schema 验证 XML	49
3.1 XML Schema 简介	49
3.2 创建 XML Schema	50

案例 1 创建 XML Schema 文档	50
任务 1 确定需要存储的元素和属性	50
任务 2 确定元素和属性内容的数据类型	51
任务 3 声明用户自定义数据类型	53
任务 4 确定声明元素的方法	59
任务 5 确定声明属性的方法	65
任务 6 创建 XML Schema 文件	67
3.3 利用 XML Schema 验证 XML	71
案例 2 利用 XML Schema 验证 XML 文档的有效性	71
任务 1 创建一个 XML 文档	71
任务 2 安装根据 XML Schema 验证 XML 的程序	72
任务 3 验证 XML 文档的有效性	72
3.4 该你动手了	73
3.5 小结	73
3.6 习题	73
第 4 章 使用 CSS 格式化页面	75
4.1 CSS 简介	75
4.2 创建 CSS	75
案例 利用 CSS 显示 XML 文档	75
任务 1 确定要显示的元素	76
任务 2 确定每个元素显示的机制	78
任务 3 创建 CSS 文件	88
任务 4 将 CSS 应用于 XML 文档	88
任务 5 在浏览器中查看 XML 文档	90
4.3 该你动手了	90
4.4 小结	90
4.5 习题	91
第 5 章 使用 XSL 显示 XML 页面	92
5.1 XSL 简介	92
5.2 用 XSL 显示 XML	93
案例 1 用 XSL 格式化数据	93
任务 1 确定要显示的数据	94
任务 2 确定格式化要显示数据的机制	94
任务 3 创建 XSL 文件	104
任务 4 将 XSL 文件用于 XML 文件	106
任务 5 查看转换结果	107
案例 2 数据的条件显示及排序	108

任务 1 确定要显示的数据	109
任务 2 明确条件控制显示数据的机制	109
任务 3 明确数据排序显示的机制	111
任务 4 创建 XSL 文件	113
任务 5 将 XSL 文件用于 XML 文件	114
任务 6 查看转换结果	114
5.3 多模板 XSL 文档格式	114
案例 3 多模板 XSL 设计及应用	114
任务 1 确定要显示的数据	115
任务 2 明确多模板显示数据的机制	115
任务 3 设计多模板 XSL	118
任务 4 创建 XSL	121
任务 5 将 XSL 应用到 XML	122
任务 6 查看 XML 文档	122
5.4 该你动手了	122
5.5 小结	123
5.6 习题	124
第 6 章 数据岛	125
6.1 数据岛简介	125
6.2 数据岛数据显示	127
案例 1 单条记录的显示	127
任务 1 确定 HTML 文档中以数据岛的形式调用 XML 文档的方法	127
任务 2 确定在 HTML 文档中显示单条记录的机制	127
任务 3 创建 HTML 文档	128
任务 4 查看显示结果	128
案例 2 利用 TABLE 显示多条数据	129
任务 1 确定 HTML 文档中以数据岛的形式调用 XML 文档的方法	129
任务 2 确定在 HTML 文档中显示多条记录的机制	130
任务 3 创建 HTML 文档	131
任务 4 查看显示结果	131
案例 3 用 Recordset 的函数来浏览 XML 数据岛	131
任务 1 确定 HTML 文档中以数据岛的形式调用 XML 文档的方法	132
任务 2 确定在 HTML 文档中逐条显示记录的机制	132
任务 3 创建 HTML 文档	136
任务 4 查看显示结果	136
案例 4 分页显示数据岛数据	136
任务 1 确定 HTML 文档中以数据岛的形式调用 XML 文档的方法	136
任务 2 确定在 HTML 文档中分页显示数据岛记录的机制	137

任务 3 创建 HTML 文档	141
任务 4 查看显示结果	141
6.3 该你动手了	141
6.4 小结	142
6.5 习题	142
第 7 章 使用 DOM 管理 XML 文档	144
7.1 DOM 概述	144
7.1.1 DOM 简介	144
7.1.2 DOM 结构树	144
7.1.3 DOM 对象架构	145
7.2 使用 DOM 验证 XML 文档的有效性	149
案例 1 使用 DOM 按照 XML Schema 验证 XML 文档	149
任务 1 确定按照 XML Schema 验证 XML 文档有效性的机制	149
任务 2 编写按照 XML Schema 验证 XML 文档有效性的代码	151
任务 3 执行代码	153
7.3 使用 DOM 对象读取 XML 文档	153
案例 2 使用 DOM 从 XML 文档中读取数据	154
任务 1 创建用户界面	155
任务 2 明确根据节点名称搜索节点的机制	156
任务 3 明确根据 XPATH 搜索节点的机制	157
任务 4 编写 HTML 文档	158
任务 5 执行代码	160
7.4 使用 DOM 对象动态修改 XML 文档	160
案例 3 使用 DOM 对象创建并修改 XML 文档	160
任务 1 创建用户界面	160
任务 2 明确创建 XML 文档的机制	162
任务 3 明确动态修改 XML 文档的机制	164
任务 4 明确动态加载 XML 文档的机制	166
任务 5 编写客户端事件处理程序	167
任务 6 编写服务器端动态修改 XML 文档的程序	174
任务 7 部署并运行程序	176
7.5 该你动手了	176
7.6 小结	177
7.7 习题	177
参考文献	178

第1章 XML 基础

本章将介绍以下内容:

1. XML 文档的基本结构
2. 创建 XML 文档的方法
3. XML 文档的编码问题
4. 检查 XML 文档良构性的方法

1.1 XML 简介

XML 是 Extensible Markup Language 的缩写, 是由万维网联盟 (World Wide Web Consortium, W3C) 定义的一种语言, 意为可扩展的标记语言。所谓可扩展, 是指 XML 允许用户按照 XML 规则自定义标记。标记语言, 则特指用一系列约定好的标记来对电子文档进行标记, 以实现电子文档的语义、结构、格式的定义。与 XML 相关的标记语言有 SGML 和 HTML。

SGML 是 Standard Generalized Markup Language 的缩写, 意为标准通用标记语言。该语言从 1986 年通过国际标准化组织 (ISO) 的认证开始而被大家普遍接受。SGML 是一种元语言 (Meta Language), 具有良好的扩展性, 可以用于定义新的语言。该标准定义独立于平台和应用的文本文档的格式、索引和链接信息, 为用户提供一种类似于语法的机制, 用来定义文档的结构和指示文档结构的标签。但由于这种语言的复杂程度和难度都很高, 因而这种语言还没有被广泛应用。

HTML 是 Hyper Text Markup Language 的缩写, 意为超文本标记语言。它是一种专门用于格式化 Web 数据的语言。该语言简单明了, 在 Web 技术的推广方面, 起了很大的作用, 几乎使所有的人都可以将自己创建的文档和网页通过 Internet 共享给其他人。超文本指的是, 因为它可以加入图片、声音、动画、影视等内容, 并且它可以从一个文件跳转到另一个文件, 与世界各地主机的文件连接, 通过 HTML 可以表现出丰富多彩的设计风格, 所以它的每个标记都有特定的含义, 代表一种页面的设置方法, 规定数据显示的格式。HTML 是基于 SGML, 而 W3C 将 HTML 规定为在互联网上发布信息的标记语言, 它的所有标记被作为标准而固定了下来, 即是使用固定标签集的一种 SGML 文档, 被称为一种“SGML 应用”。因此, HTML 不属于元语言, 它不能用于定义其他标记语言。

为了使万维网技术扩展到文档处理和数据交换的领域, W3C 推出了 XML。而 XML 是 SGML 的一个子集, 设计它的目的是便于能够在 Internet 上进行数据交换。它有如下优点。

1) 具有可扩充性。XML 最明显的特点在于它可以创建标记和文法结构。正是这种优点使得用户能够根据自己的特殊需要制订出适用于自身的一套标记和文法结构, 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。所以才会有那么多其他语言可以用 XML 来制订, 例如矢量标记语言 (VML)、音乐标记语言 (MML)、资源描述语言 (RDF)、无线标记语言 (WML) 等。

2) 具有自我描述数据的功能。除了可以使用标记与属性来描述数据的意义之外, XML 还提供了可用来描述文件结构的 DTD 与 XML Schema。可根据 DTD 和 XML Schema 对 XML 文档的内容进行检验。

3) 具有一种数据多种显示的功能。XML 的一个优点就在于它将信息的数据部分和信息样式显示部分进行了区分。在一个 XML 文档中, 只保存表示信息的数据, 并不决定这些数据的显示样式, 数据的样式显示部分是由其他语言来解决的 (通常是 CSS 和 XSL), 这样就可以给同一份数据添加多种样式, 从而得到多种显示效果。这是使用 HTML 不可能做到的。

4) 实现数据的分布式处理。XML 是一种针对 Internet 而设计的标记语言, 一个保存有数据的 XML 文档可以在 Internet 上自由传送。当 XML 格式的数据被发送给客户端后, 客户可以通过应用软件从 XML 文档中提取出这些数据, 进而对它进行编辑和处理, 而不仅仅是显示结果。XML 文件对象模型 (Document Object Model) 允许用脚本和其他编程语言处理 XML 格式的数据。该种情况下的数据处理可以在客户端完成, 而没有必要回到 WWW 服务器上, 此种方法节省了 Internet 上的数据带宽。而且将原来必须由服务器处理的许多负载都分配到了客户端上, 从而降低了服务器的负担, 优化了服务器的性能。

5) 实现不同数据源之间的数据交换。XML 是以文本形式来描述的一种文件格式, 因此具有跨平台的能力。XML 的跨平台特性还在于它提供了一种不同的数据源之间进行数据交换的公共标准, 是一种公共的交互平台。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, 从而可以在各种不同的平台环境下实现数据交换。

6) 简单易学且功能强大。HTML 的成功很大程度上在于它的简单性, 这种简单性对于 HTML 的推广, 甚至对于 Internet 的普及都起到了不可忽视的作用。同样, XML 标记语言也具有这种简单易学的特性, 这也是 XML 标记语言创立之初的一个重要目标, 另一方面, 它的功能也十分强大。XML 本身就是 SGML 的一个子集, 是一种专门应用在 Internet 上的 SGML 文档, 因而 XML 充分继承了 SGML 的优点, 也克服了 SGML 极其复杂的缺点。XML 和 HTML 都是 SGML 的一部分, 都是 SGML 在特殊形式下的特殊表现。但 HTML 过于简化, 只适用于信息的样式描述, 而不适用于信息的结构化表述, XML 的出现正是为了弥补这种缺陷。XML 的出现尽管要替代 HTML 在 Internet 上的主角地位, 但 XML 并不能完全取代 HTML, 这是因为 XML 和 HTML 在 Internet 上充当的角色并不是完全一样的。HTML 关心的是信息在浏览器中的显示效果, 而 XML 关心的则是信息的保存方式, 不但 HTML 不能被 XML 所取代, 从某种意义上讲, XML 还离不开 HTML, 因为 XML 的数据显示往往要被转变成为 HTML 文档, 才能被浏览器识别。因此 XML 并不能取代 HTML, 它们各有所长, 但从长远意义上讲, XML 定将在 Internet 上占据主导地位。

1.2 创建一个 XML 文档

在实际应用中,XML 发挥着巨大的作用,在很多领域和环境都采用 XML 来实现需要的特定功能。下面就通过一个实例来说明 XML 的作用。

案例 创建 XML 文档并检查该文档的良好性

问题陈述:学校需要开发一个教材管理系统,希望该管理系统能很方便地共享使用学校其他管理系统(如学籍管理系统、教务管理系统)中的数据,比如系部信息、班级信息、学生信息、课程信息等,该管理系统的数据库也可以提供给其他系统共享使用,比如教材信息。现有的这些管理系统使用的硬件和软件平台可以不相同。根据这一需要,确定班级信息的存储方案。班级信息包括班号、班级名称、班级人数、所属专业编号、所属系部编号。

解决上述问题,需要完成以下任务。

任务 1: 确定存储数据的方法。

任务 2: 确定存储数据的文档格式。

任务 3: 创建一个 XML 文档存储数据。

任务 4: 检查文档格式的正确性。

任务 1 确定存储数据的方法

若在不同平台交换数据信息,常会发生数据格式不同的问题,从而给信息交换带来困扰,XML 文档可以用来解决这一问题。

由于 XML 是以文本形式来描述的一种文件格式,因此具有跨平台的能力,可以在各种不同的平台环境下实现数据交换。XML 无处不在、发展迅速,众多软件开发商都采用了 XML 标准。

因此,在本章的问题陈述中,当教材管理系统、学籍管理系统、教务管理系统等均要共享班级信息、学生信息、课程信息、教材信息等数据时,可以将这些数据信息存储为 XML 文档的格式。

任务 2 确定存储数据的文档格式

1. XML 文档结构

在确定需要将数据存储到一个 XML 文档之后,首先我们来了解 XML 文档结构,然后针对本节提出的问题,确定存储数据的文档格式。

XML 的文档结构可以从逻辑结构和物理结构两个方面来说明。

从逻辑结构上讲,一个 XML 文档通常以一个 XML 声明开始,通过 XML 元素来组织 XML 数据。XML 元素包括标记和字符数据。为了组织数据更加方便、清晰,我们还可以在字符数据中引入 CDATA 段,并可以在文档中引入注释。此外,由于有时需要给 XML 处理程序提供一些指示信息,XML 文档中可以包含处理指令。

例如,一个包含班级信息的 XML 文档结构如下:

```
<?xml version="1.0" encoding="GB2312" ?>
```

```
<所有班级>
```

```
  <班级信息 班号="0806130301">
```

```
    <班级名称>高网 0301 班</班级名称>
```

```
    <班级人数>44</班级人数>
```

```
  </班级信息>
```

```
  <班级信息 班号="0806130302">
```

```
    <班级名称>高网 0302 班</班级名称>
```

```
    <班级人数>45</班级人数>
```

```
  </班级信息>
```

```
</所有班级>
```

如果将 XML 文档看成一个逻辑整体,就可以将 XML 文档看成是由若干个节点组成,每个节点都是一个文档的逻辑组成部分,同时,一个节点还可以再包含若干个节点。因此,可以说文档的结构如同一个树型结构,上例对应的文档结构如图 1-1 所示。其中,DOCUMENT 部分称为文档元素,是文档的根,但它不同于文档的根标记,所有 XML 文档的根都是一样的,它是一个 XML 文档的入口。它包含两个子节点——XML 声明和整个文档的根标记(所有班级)。根标记下面包含的元素称为子元素,子元素又可以包含子元素,包含子元素的元素(根元素除外)称为枝,不包含子元素的元素称为叶子。同时,可以为元素添加说明信息,用来说明元素特征的,称为属性。

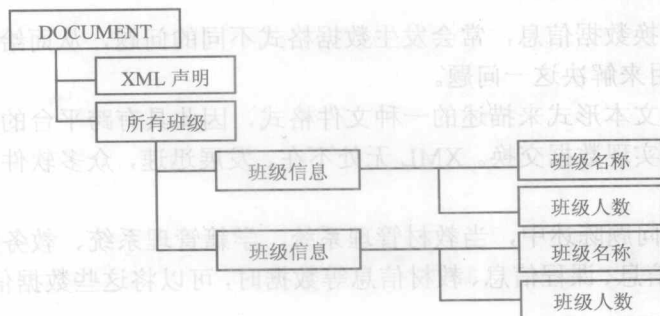


图 1-1 文档结构图

从物理结构上讲,XML 文档是由一个或多个存储单元构成的,这些存储单元就是所谓的实体。每个 XML 文档都有一个“文档实体”。这个实体是由 XML 本身给出的,无需显式定义就可以使用,它指的其实就是整个文档的内容,是 XML 语法分析器处理的起点。除此之外,在 XML 文档中还可以定义其他实体,每个实体都是有内容的,并且每个实体都有自己的实体名称。采用实体的方式有利于对一个大型的 XML 文档进行分解,从而降低编辑 XML 文档的软、硬件要求,例如可以将文档分解成可处理的字节块,从而在编辑时降低对内存大小的要求。由此看出,XML 文档的物理结构降低了文档管理和维护的复杂性。

2. 良构性和有效性

XML 文件是由标记及其所标记的内容构成的文本文件,与 HTML 文件不同的是,这

些标记可自由定义,其目的是使得 XML 文件能够很好地体现数据的结构和含义。但是,XML 文件必须符合一定的语法规则,只有符合这些语法规则,XML 文件才可以被 XML 解析器解析。XML 文件分为良构的 XML (Well-Formed XML) 和有效的 XML (Validated XML)。

具有良构性的 XML 文档必须满足以下规则。

- 1) XML 文档必须以 XML 声明作为开始。
- 2) 每个 XML 文档必须有一个根标记,也只能有一个根标记。
- 3) 除了内容为空的标记之外,标记的开始和标记的结束必须配对使用。
- 4) 内容为空的标记结尾必须加上“/”,这种标记称为“空标记”。
- 5) 标记名称与属性名称必须合法,且区分大小写。
- 6) 属性的值必须加引号。
- 7) 标记可以嵌套子标记,但是两个标记之间不能有重叠域。
- 8) 特殊字符必须使用 XML 中特定的编号来表示。
- 9) 各种指令,如注释、处理指令,必须正确编写和放置在文档的正确位置。

►【例 1.1】非良构的 XML 文档

```
<?xml version="1.0" encoding="GB2312" ?>
```

```
<所有班级>
```

```
<班级信息 1>
```

```
<班级名称>高网 0301 班</班级名称>
```

```
<班级人数>44</班级人数>
```

```
<班级信息 2><!--错误二-->
```

```
</班级信息 1>
```

```
<班级名称>高网 0302 班</班级名称>
```

```
<班级人数>45</班级人数>
```

```
</班级信息 2>
```

```
</所有班级>
```

```
<所有班级></所有班级><!--错误一-->
```

在例 1.1 中,出现了两个根元素<所有班级>,而良构性规定每个 XML 文档只能有一个根元素,这是错误一;另外,在元素<班级信息 1>中又包含了元素<班级信息 2>的一部分声明,导致了元素之间产生重叠域,这是错误二。

一个有效的 XML 文档首先必须是一个结构良好的 XML 文档,同时还必须符合额外的一些约束。这些额外的约束是利用 DTD 验证或 XML Schema 验证。我们可以利用 DTD 或 XML Schema 来检查 XML 文档是否与规定的格式一致,若是一致的,则该文档是有效的 XML 文档,否则该文档就不具备有效性。有关 DTD 和 XML Schema 相关内容,将在第 2 章和第 3 章讲解。

3. XML 声明

一个良构的 XML 文档应当以 XML 声明作为 XML 文档的第一行,前面不能有空白、注释或其他处理指令。它包括 XML 版本信息、编码属性和独立属性等。

一个完整的 XML 声明语法格式如下:

`<?xml version="1.0" encoding="编码" standalone="yes/no" ?>`

1) **version** 属性: 不能省略且必须在属性列表中排在第一位, 指明所采用的 XML 版本号, 值为“1.0”。该属性用来保证对 XML 未来版本的支持。目前 XML 最新的版本为 1.1, 但推荐使用 W3C 于 2000 年发布的 XML1.0 版。

2) **encoding** 属性: 可选, 指定了文档的编码系统, 即规定了 XML 文档采用哪种字符集进行编码。如果未指定该属性的值, 其默认值为 UTF-8。

XML 的常用编码属性值如下。

① “UTF-8”: 标记和内容可以使用汉字、日文、英文。

② “GB2312”: 标记和内容可以使用 ASCII 字符和汉字。

③ “ISO-8859-1”: 标记和内容只能是 ASCII 字符。

采用 UTF-8 编码的 XML 文件必须选择 UTF-8 编码来保存, 其他编码属性的 XML 文件必须选择 ANSI 编码来保存。

3) **standalone** 属性: 可选, 值为“yes”或“no”, 以说明 XML 文件是独立的, 还是与其他文档有关联的。该属性的默认值是“no”。

若以班级信息(班级信息中包含中文标记)模板为例, 可以设置其 XML 声明为:

`<?xml version="1.0" encoding="UTF-8" ?>`或`<?xml version="1.0" encoding="GB2312" ?>`

4. 处理指令

注意, XML 声明的标签用“<?”开始, 用“?”结束, 称为处理指令。除了 XML 声明这一特殊的处理指令外, 在 XML 文档中还可以包含其他处理指令。处理指令是包含在 XML 文档中的一些命令性语句, 目的是告诉 XML 处理一些信息或执行一定的动作, 例如 XML 声明就是告诉 XML 解析器, 该文档遵守 XML 1.0 规范, 就按照 XML 1.0 的要求来检查。关于其他方面的处理指令, 将在后面进一步说明。

处理指令的格式为:

`<?...?>`

5. 标记

XML 是基于文本的标记语言, 可以在 XML 文档中添加标记, 使文档数据结构清晰而变得容易处理。XML 标记提供和描述 XML 文件或实体的内容结构, 其结构与 HTML 基本相同。

标记的种类有很多, 如开始标记、结束标记、空元素标记、实体引用标记、字符引用标记、处理指令标记、注释标记、文档类型声明标记、CDATA 标记等。

XML 标记的基本格式为: 使用“<”和“>”来界定标记, 例如<姓名>、</姓名>、<name>、</name>都是合法的标记。

XML 标记也可以分为非空标记和空标记两种类型。

非空标记必须由开始标记与结束标记组成, 两者之间是标记内容。开始标记以“<”开始、“>”为止, 中间是标记名称; 结束标记以“</”开始、“>”为止, 中间是标记名称。

空标记中不包含任何内容, 将所有信息全部存储在属性中, 而不存储在内容中。它告诉 XML 应用程序执行某个动作或显示对象。

空标记的语法格式为：<标记名 属性列表/>

例如：<班级 名称="高网 0301 班"/>

值得注意的是，在标记“<”和标记名之间不能有空格，但在标记“/>”前面可以有空格和换行符。例如：< 班级 名称="高网 0301 班" />就是错误的。

【例 1.2】非空标记和空标记

```
<?xml version="1.0" encoding="GB2312"?>
```

```
<班级信息>
```

```
  <班级名称>高网 0301 班</班级名称>
```

```
  <班级人数/>
```

```
</班级信息>
```

对于标记有以下语法规则。

- 1) 标记必不可少，任何一个格式良好的 XML 文档中至少要有有一个元素。
- 2) 区分大小写。
- 3) 要有正确的结束标记。结束标记除了要和起始标记在拼写和大小写上完全相同，还必须在前面加上一个斜杠“/”。当一对标记之间没有任何文本内容时，可以不写结束标记，而在起始标记的最后冠以斜杠“/”来确认，这样的标记称为“空标记”。
- 4) 标记要正确嵌套。
- 5) 标记命名要合法。
- 6) 有效使用属性。

与 HTML 不同的是，XML 标记区分大小写，其中包括元素的标记名和属性。例如，<name>≠<NAME>，<name>≠<Name>。

另外，在 HTML 中使用<P>，不加</P>是合法的。XML 严格要求标记需要配对出现，每个开始标记都必须跟随着出现结束标记，但如下四种标记例外。

- 1) 空元素标记：空元素标记不包括任何内容，它将所有信息全部存储在属性中，而不存储在内容中，它告诉 XML 应用程序执行某个动作或显示对象。
- 2) 处理指令：处理指令的格式是<?PI-target instruction ?>，它不需要结束标记。
- 3) 注释标记：与 HTML 的注释指令相同，格式是<!--注释内容-->，不需要结束标记。
- 4) CDATA 标记：它用于处理让浏览器或应用软件在对 XML 文档进行解析时，将 CDATA 标记出来的数据视为纯字符数据而不解析它。

6. 元素

元素是 XML 文档最基本的单位，它包含了标记、元素内容、属性等内容。元素以树形分层结构排列，可以嵌套在其他元素中。一个元素包含一个起始标记、一个结束标记以及标记之间的数据内容。开始标记和结束标记用来描述标记之间的数据内容，标记之间的数据内容是元素的值。一个文档可以包含多个标记名相同的元素。

在 XML 文档中，元素又可以分为非空元素和空元素两种类型，其语法格式分别如下。

(1) 非空元素

<开始标记 属性列表>数据内容</结束标记>

或

<开始标记>数据内容</结束标记>

(2) 空元素

<空标记名 属性列表>

或

<空标记名/>

例如, <班级名称>高网 0301 班</班级名称>就是一个非空元素, 如果把非空元素的内容放到属性中, 则变成一个空元素, 如<班级 名称="高网 0301 班"></班级>。

XML 元素的标记名可以任意命名, 但命名时必须遵守以下规则。

1) 标记名必须是以字母或者下划线 (_) 开始, 后继字符可以是一个或者多个数字、字母、句点 (.) 和连字符 (-), 名称不允许为空, 至少含有一个字母。

2) 标记名不能包含冒号 (:), 因为冒号被用于定义名称空间。

3) 尽量不要使用以 xml (任意大小写字母组合) 为前缀的命名方式。

4) 标记名中不能包含空格。

以下作为元素的标记名是正确的:

name M _name name_ name1 name.n1 姓_名

以下标记名是错误的:

na me (出现空格)

na:me (出现冒号)

xmlname (不能使用 xml 开头)

5m (数字开头)

姓 名 (出现空格)

XML 处理器会将开始标记和结束标记之间出现的字符数据, 包括可见和非可见的字符视为元素的内容, 注意以下三个元素:

<班级名称>高网 0301 班</班级名称>

<班级名称>

高网 0301 班

</班级名称>

<班级名称>

高网 0301 班

</班级名称>

它们的元素内容是不同的。第二个元素内容比第一个元素多了两个换行符。第三个元素比第二个元素多了几个空格。

7. 特殊字符

在 XML 文档中, 所有文本都会被解析, 但在实际运用中, 需要在 XML 文档内引用 “<”、“>”、“'”、“”” 和 “&” 这些特殊符号时, 必须使用它们的实体引用。假如在 XML 文档中使用 “<” 的字符, 那么解析器将会出现错误, 因为解析器会认为这是一个新元素的开始, 这与标记发生混淆。所以, 适当的实体引用是必须的, 而且实体引用中的字母必须小写, 不能使用大写。为了减少出错, 使用实体是一个好习惯。

这些特殊的 XML 实体引用如下。

- 1) 用 “<” 代替 “<” (小于号)。
- 2) 用 “>” 代替 “>” (大于号)。
- 3) 用 “&” 代替 “&” (和)。
- 4) 用 “'” 代替 “'” (单引号)。
- 5) 用 “"” 代替 “”” (双引号)。

使用实体时, 注意实体必须以符号 “&” 开头, 以符号 “;” 结尾。

【例 1.3】实体引用举例

```
<?xml version="1.0" encoding="GB2312"?>
<班级信息>
  <班级名称>高网 0301 班&高网 0302 班</班级名称>
  <班级人数>44 人<45 人</班级人数>
</班级信息>
```

在 XML 解析器处理该例时会显示出错信息, 应将其中的 “&” 和 “<” 字符修改为实体引用, 效果如下。

```
<?xml version="1.0" encoding="GB2312"?>
<班级信息>
  <班级名称>高网 0301 班&amp;高网 0302 班</班级名称>
  <班级人数>44 人&lt;45 人</班级人数>
</班级信息>
```

8. CDATA 段

在 XML 文档中的所有文本都会被解析器解析, 只有在标记 CDATA 中包含的所有标记、实体引用都被解析器忽略, 而被 XML 处理程序当作字符数据看待。CDATA 标记的语法格式为:

```
<![CDATA[ 需要不被 XML 程序处理的文本内容 ]]>
```

CDATA 段以 “<![CDATA[” 开始, 以 “]]>” 结束, 之间为所要包含的不想被 XML 解析器处理的文本内容。例 1.4 给出了在一段 XML 文档中显示另外一个 XML 文档的方法。

【例 1.4】CDATA 段举例

```
<?xml version="1.0" encoding="GB2312"?>
<FileOfXML>
  <![CDATA[
    <?xml version="1.0" encoding="GB2312"?>
    <所有班级>
      <班级信息>
        <班级名称>高网 0301 班</班级名称>
        <班级人数>44</班级人数>
      </班级信息>
```