

XML实用教程

- ◆ 初识XML
- ◆ XML语法
- ◆ 在XML中使用DTD
- ◆ XML和CSS
- ◆ XSL转换
- ◆ XML Schema简介
- ◆ 数据岛
- ◆ Java中的DOM
- ◆ .NET中的DOM
- ◆ 基于XML的公共信息交流网



王 冬 陈可汤 王 龙 编著



清华大学出版社

高等学校计算机应用规划教材

XML 实用教程

王冬 陈可汤 王龙 编著



清华大学出版社

北 京

内 容 简 介

XML是一种描述数据和数据结构的语言,可以保存在任何可存储文本的文档中。本书主要介绍了XML的基本概念、语法规则、文档类型定义、模式定义、级联样式表、可扩展样式表和文档对象模型,此外还介绍了XML在Java和.Net中的编程,最后通过一个综合案例演示了XML在实际项目开发中的应用。本书内容由浅入深,在讲解基本概念和基础知识的同时给出了大量实例,每章还给出了填空题、选择题和上机题,便于读者巩固所学的知识。

本书适合具有一定软件开发基础和初步 HTML 知识的读者阅读,也可作为高等院校计算机、电子商务以及信息类相关专业的教材,还可供相关技术人员、广大 Web 应用程序开发者和用户参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

XML 实用教程 / 王冬, 陈可汤, 王龙 编著. —北京: 清华大学出版社, 2014

(高等学校计算机应用规划教材)

ISBN 978-7-302-35154-2

I. ①X… II. ①王… ②陈… ③王… III. ①可扩充语言—程序设计—高等职业教育—教材
IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 013720 号

责任编辑: 王 定 程 琪

封面设计: 牛艳敏

责任校对: 邱晓玉

责任印制: 何 芊

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62794504

印 刷 者: 北京富博印刷有限公司

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 18.5 字 数: 427 千字

版 次: 2014 年 2 月第 1 版 印 次: 2014 年 2 月第 1 次印刷

印 数: 1~3500

定 价: 30.00 元

产品编号: 049979-01

前 言

XML(Extensible Markup Language, 可扩展标记语言)不仅是一种优秀的元标记语言,同时也是一种优秀的数据交换格式。用 XML 描述数据具有结构简单、便于人和计算机阅读的双重功效,弥补了关系型数据对客观世界中真实数据描述能力的不足。XML 集 HTML 和 SGML 的优势于一身,具有易于编辑、便于管理、适于存档和容易查询等诸多优势,已经成为新一代网络标准语言。

本书以 XML 为中心,详细介绍了与 XML 相关的一些基本技术,包括 XML 语法、DTD 和 XML Schema、使用 CSS 和 XSL 格式化 XML 文档、XPath 及 DOM 等技术。通过本书的学习,使读者对 XML 语言有一个全面的了解,掌握相关的主要知识点,在软件设计中运用 XML 技术灵活地进行信息的描述、交换和显示。全书共 10 章,具体内容如下:

第 1 章 初识 XML。本章主要介绍了标记语言产生的原因和发展、SGML 语言、HTML 语言,概述了 XML 语言的概念、优势和相关技术,最后对主流的 XML 编辑器——XMLSpy 2011 的安装、主界面、主要功能和基本用法进行详细讲解。

第 2 章 XML 语法。本章主要介绍 XML 的语法,重点介绍了 XML 文档结构,XML 格式的约束规则,如何声明元素和属性以及怎样使用命名空间等基础语法内容。

第 3 章 在 XML 中使用 DTD。本章主要介绍的是确保 XML 文档有效的文档类型定义——DTD,重点介绍了 DTD 中元素、属性、实体的声明语法和格式,以及在 XML 文档中使用 DTD 的方式。

第 4 章 XML 和 CSS。本章首先介绍了 CSS 的作用,然后介绍了 CSS 语法及常见的 CSS 属性的设置,并以实例的方式讲述了如何在 XML 中使用 CSS。

第 5 章 XSL 转换。本章讲述了 XSL 的基本概念,着重讲解 XSL 转换语言、XSL 文档的编写、模板的规则及应用模板、XSL 的转换执行过程以及实现 XSL 的方式。

第 6 章 XML Schema 简介。本章主要介绍了 XML Schema(XML 模式)定义的基本语法知识,重点介绍了模式的文档结构、XML 模式中元素及属性的声明语法和使用 XML 模式的方式。

第 7 章 数据岛。本章主要介绍了数据岛技术,详细讲解了数据岛的使用方式、数据源对象 DSO 的操作,并给出了遍历和分页显示 XML 文档的实例。

第 8 章 Java 中的 DOM。本章重点介绍 Java 语言中文档对象模型 DOM 的基本概念、常用接口、常用对象和 DOM 和 XML 如何进行交互。此外,还讲述了 Java 语言中操作 XML 文档的另外一种方式——SAX,包括其工作原理、常用的事件和接口以及它对 XML 的解析。

第 9 章 .NET 中的 DOM。本章主要介绍了在 .NET 平台上操作和处理 XML 文档要用到的具体技术及相关对象,重点介绍了 .NET 平台上的 DOM 编程方法,以及 XML 与 DataSet 的关系及相互转换。

第 10 章 基于 XML 的公共信息交流网。本章通过一个综合案例演示了 XML 在实际项目开发中的具体应用,通过对案例的剖析,体会 XML 技术在 Web 应用方面的优势并进一步掌握利用 DOM 操作 XML 文档的方法和技巧。

本书是把理论和实例结合在一起的 XML 及相关技术应用的参考书,让读者先从 XML 基础知识和 XML 相关技术入手,然后明白学习这些技术的意义,从而能够运用这些技术来开发 XML 应用程序。最后能够把这些技术融合为一体,实现 XML 应用系统。

本书适合具有一定 Java 基础和初步 HTML 知识的读者阅读,也可作为高等院校计算机、电子商务以及信息类相关专业的教材,同时可供相关技术人员、广大 Web 应用程序开发者和用户参考。

本书主要由王冬、陈可汤和王龙主持编写,此外,吕平、王坚宁、王魁、许小荣、周新国、王松年、张凤琴、陈可汤、陈作聪、张璐、祁招娣、王文婷等同志在整理材料方面给予了编者很大的帮助。在此,编者对他们表示衷心的感谢。

由于作者水平有限,书中纰漏之处在所难免,欢迎广大读者、同仁批评斧正。

作 者

2013 年 8 月

目 录

第 1 章 初识 XML	1
1.1 标记语言	1
1.1.1 SGML 的诞生	2
1.1.2 HTML 的出现	2
1.2 XML 概述	4
1.2.1 为什么要使用 XML	4
1.2.2 一个简单的 XML 例子	4
1.2.3 XML 的优势	7
1.3 XML 开发工具	8
1.3.1 XMLSpy 的安装	9
1.3.2 XMLSpy 主界面	13
1.3.3 XMLSpy 的主要功能	17
1.3.4 XMLSpy 的使用	20
1.4 习题	22
1.4.1 填空题	22
1.4.2 选择题	22
1.4.3 上机题	23
第 2 章 XML 语法	24
2.1 XML 文档结构	24
2.1.1 组成与声明	24
2.1.2 注释和处理指令	26
2.2 XML 标记	28
2.2.1 定义 XML 标记	28
2.2.2 标记的分类	28
2.2.3 标记的规则	29
2.3 XML 元素	29
2.3.1 元素的命名	29
2.3.2 起始标记	30
2.3.3 结束标记	30
2.3.4 元素的类型	30
2.3.5 元素的嵌套	31

2.4 XML 属性	32
2.4.1 属性的定义	32
2.4.2 属性值的定义	34
2.5 特殊字符和 CDATA 区段	34
2.5.1 特殊字符的使用	34
2.5.2 CDATA 的格式和位置	36
2.6 XML 命名空间	37
2.6.1 命名空间的概念	38
2.6.2 命名空间的定义	38
2.6.3 默认命名空间	40
2.7 XML 文档语法小结	41
2.7.1 格式良好的 XML 文档	42
2.7.2 有效的 XML 文档	42
2.8 习题	43
2.8.1 填空题	43
2.8.2 选择题	44
2.8.3 上机题	44
第 3 章 在 XML 中使用 DTD	47
3.1 DTD 的基本概念	47
3.1.1 初识 DTD	47
3.1.2 DTD 的引用	50
3.2 DTD 中的元素	55
3.2.1 元素的定义	55
3.2.2 约束元素的子元素	56
3.2.3 约束元素的类型	57
3.3 DTD 中的属性	60
3.3.1 在 DTD 中声明属性	60
3.3.2 属性的默认值	61
3.3.3 属性的类型	63
3.4 DTD 中的实体	68
3.4.1 实体的概念和分类	69

3.4.2 普通实体	69	5.3.1 简单判断	122
3.4.3 参数实体	72	5.3.2 多条件判断	122
3.4.4 不可解析实体	74	5.3.3 循环处理	125
3.5 习题	75	5.3.4 输出内容排序	126
3.5.1 填空题	75	5.4 XPath 简介	128
3.5.2 选择题	76	5.4.1 XPath 中的节点类型和 关系	128
3.5.3 上机题	76	5.4.2 XPath 的类型和运算符	130
第 4 章 XML 和 CSS	79	5.4.3 XPath 的位置路径 表达式	131
4.1 CSS 简介	79	5.4.4 XPath 的标准函数	134
4.1.1 CSS 语法	79	5.5 习题	138
4.1.2 引用 CSS	80	5.5.1 填空题	138
4.2 CSS 选择器	82	5.5.2 选择题	138
4.3 设置文本的显示方式	87	5.5.3 上机题	138
4.4 字体	87	第 6 章 XML Schema 简介	141
4.5 文本样式	91	6.1 XML Schema 与 DTD	141
4.6 边框	94	6.2 XML Schema	142
4.7 布局	95	6.2.1 XML Schema 概述	142
4.8 颜色	98	6.2.2 XML Schema 的特点	143
4.9 注释	101	6.2.3 XML Schema 基本结构	143
4.10 使用 CSS 显示 XML 文档	102	6.3 XML Schema 的使用	144
4.10.1 内部链接样式	102	6.3.1 Schema 的作用	144
4.10.2 无需链接的内部 CSS	103	6.3.2 Schema 的引用	144
4.11 习题	105	6.3.3 XML Schema 简单示例	145
4.11.1 填空题	105	6.4 XML Schema 数据类型	149
4.11.2 选择题	105	6.4.1 简单类型	149
4.11.3 上机题	106	6.4.2 复杂类型	152
第 5 章 XSL 转换	108	6.5 XML Schema 元素的声明	159
5.1 XSL 入门	108	6.5.1 XML Schema 根元素	159
5.1.1 XSL 概述	108	6.5.2 element 元素	160
5.1.2 XML 文档结构树	110	6.5.3 空元素的定义	161
5.1.3 XSL 样式表及工作原理	110	6.5.4 设置默认值和固定值	162
5.2 XSL 模板	113	6.5.5 元素的引用	163
5.2.1 模板概述	113	6.6 XML Schema 属性的声明	164
5.2.2 输出结果内容	115	6.7 XML Schema 命名空间	165
5.2.3 模板与标记匹配	116		
5.3 XSL 控制指令	121		

6.7.1 XML Schema 命名 空间类型	165	8.2.2 DOM 中的类和方法	207
6.7.2 引用 XML Schema 命名空间	166	8.2.3 遍历 XML 文档	210
6.8 XML Schema 文档的 可视化创建	166	8.2.4 访问属性节点	212
6.9 习题	170	8.2.5 访问 XML 节点内容	213
6.9.1 填空题	170	8.2.6 删除 XML 元素和属性	215
6.9.2 选择题	170	8.2.7 创建 XML 节点	217
6.9.3 上机题	170	8.3 SAX 简介	219
第 7 章 数据岛	173	8.3.1 SAX 接口	220
7.1 数据岛概述	173	8.3.2 SAX 工作机制	221
7.1.1 数据岛定义	173	8.3.3 SAX 常用接口	223
7.1.2 数据岛绑定	174	8.4 应用 SAX 解析器	225
7.2 数据岛的应用	175	8.4.1 SAX 的创建步骤	225
7.2.1 数据岛的使用形式	175	8.4.2 SAX 解析 XML 文档	226
7.2.2 数据岛访问 XML 元素内容	178	8.5 习题	229
7.2.3 数据岛访问 XML 标记的属性	183	8.5.1 填空题	229
7.3 DSO 数据源对象	185	8.5.2 选择题	229
7.3.1 DSO 概念	185	8.5.3 上机题	229
7.3.2 访问数据岛对象	187	第 9 章 .NET 中的 DOM	232
7.3.3 操作 XML 数据岛数据	188	9.1 关于 .NET	232
7.4 DSO 和 XSL 切换样式	194	9.2 .NET 的 XML 分析器	233
7.5 习题	197	9.3 XML 阅读器处理 XML 文档	234
7.5.1 填空题	197	9.3.1 .NET 中 XML 相关类	234
7.5.2 选择题	198	9.3.2 用 XmlTextReader 类 读取和解析 XML 文档	236
7.5.3 上机题	198	9.3.3 用 XmlTextWriter 类 书写 XML 文档	239
第 8 章 Java 中的 DOM	201	9.4 DOM 操作 XML 文档	241
8.1 DOM 简介	201	9.4.1 操作 XML 的相关类	242
8.1.1 DOM 文档树模型	202	9.4.2 书写 XML 文档	243
8.1.2 DOM 基本接口	203	9.4.3 遍历 XML 文档	244
8.1.3 DOM 对象	205	9.4.4 新增 XML 节点	246
8.2 DOM 操作 XML 文档	206	9.4.5 修改 XML 节点	249
8.2.1 Java 中的 DOM	206	9.4.6 删除 XML 节点	251
		9.5 XML 与 DataSet	253
		9.5.1 XML 数据填充 DataSet	253

9.5.2 将 DataSet 的数据 写入文件	254	10.2.1 Users.xml	262
9.5.3 DataSet 和 XML 架构	255	10.2.2 Information.xml	263
9.6 习题	257	10.3 前台模块的设计	264
9.6.1 填空题	257	10.3.1 前台首页的显示	264
9.6.2 选择题	257	10.3.2 信息列表的显示	267
9.6.3 上机题	257	10.3.3 信息详情的显示	270
10.3.4 查询信息	271	10.3.4 查询信息	271
第 10 章 基于 XML 的公共信息		10.4 后台模块的设计	272
交流网	260	10.4.1 用户登录	273
10.1 系统分析与设计	260	10.4.2 管理信息	274
10.1.1 系统需求分析	260	10.4.3 发布信息	281
10.1.2 系统模块设计	261	10.4.4 删除信息	283
10.2 系统 XML 文档设计	262		

第1章 初 识 XML

XML是W3C(万维网联盟)提出的一种可扩展标记语言,它是随着人们对信息传递要求的不断提高而产生的一种新技术。XML的出现使得标记语言可以由编程人员根据需要进行扩展,这就给Web编程带来了革命性的进步。本章介绍XML的基础知识,包括标记语言的发展历史、XML的特点和优势以及XML开发工具XMLSpy的安装和主要功能,让读者为以后章节内容的学习做好充分的准备。

1.1 标 记 语 言

标记语言也被称为“标签语言”或“置标语言”,它是英文 Markup Language 的中译。这类语言不同于我们所熟知的一般“程序语言”。程序语言是设计应用软件的工具,而标记语言主要是用来制作电子文档和存储数据。

标记语言由一些标识码(code)或控制标签(tag)组成,这些标识码或标签可以让电子文档变得更具结构性,而这种结构性有利于应用软件解析和应用。但是,这些标识码或标签如果独立存在则是无意义的,它们必须结合数据后才能形成一份有用的电子文档。而所谓有用的电子文档则是针对应用软件来说的,也就是说应用软件能够解读这份电子文档中的标记语言,并且根据标记语言的意义来对该份电子文档做特定的处理。

例如,HTML 中可以使用
标签,
可以被应用软件解释为换行,但是如果前后没有数据,那么这个换行是没有意义的。

目前所使用的标记语言可以分为专门用途的标记语言(Specialized Markup Language)和一般通用的标记语言(Generalized Markup Language)两类。

(1) 专门用途的标记语言。专门用途的标记语言是为某种应用软件特别制定的,或是为了某种特别的应用领域所发展出来的。例如,HTML 就是一种专门用途的标记语言,制定 HTML 是为了将文档数据以不同的方式显示在浏览器这个应用软件中。另外像大多数文字处理软件,也是专门用途的标记语言的应用。例如,OOXML 专门用于微软公司的 Office 系列软件。因为这类标记语言的应用范畴较窄,所以其标记语言本身的复杂性较低,当然也就简单易学。

(2) 一般通用的标记语言。一般通用的标记语言仅描述文档中数据内容的意义与结构,但并没有定义如何应用这些数据,也就是说它并不是为某一个特别的软件量身订制的。文档如果使用一般通用的标记语言来标记,那么因为所有软件都可以使用,其制作出来的电子文档的可移植性当然比专门用途的标记语言要好。例如,SGML就是属于一般通用的标记语言,SGML有严谨复杂的语法,可以用来标记文档中的数据以便于应用

程序处理。同一份SGML文档可以应用到各种不同的应用软件,而且该份文档不需要做任何修改或调整。

目前,标记语言广泛应用于网页和网络应用程序中。使用最多的标记语言是超文本标记语言(HTML)和可扩展标记语言(XML)。

1.1.1 SGML 的诞生

SGML(Standard Generalized Markup Language)即标准通用标记语言。SGML的思想最初是从IBM的一个信息管理项目中产生的,称为GML(通用标记语言)。它是一种IBM格式化文档语言,用于对文档组织结构、各部件及其之间的关系进行描述。由于在当时的信息交换过程中,经常会发生数据格式不同的问题,随着网络技术的不断发展,这一问题日益严重,制约了人们的信息交流。1986年,国际标准化组织(ISO)采纳了IBM的这一思想,并将其整理为SGML。SGML是一种基于文档标记语言的元语言,它不仅具有良好的扩展性,而且可移植性强,在任何一种环境下都可以正常使用。

SGML通过SGML文档的形式来表现。SGML文档定义独立于应用平台和所使用的文本文档的格式、索引和链接信息。它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。SGML文档由三个部分组成,即语法定义、文档类型定义和文档实例。

- (1) 语法定义:定义文档类型和文档实例的语法结构。
- (2) 文档类型定义:定义文档实例的结构和组成结构的元素类型。
- (3) 文档实例:文档的主体部分。

SGML文档的结构相当严谨,其中文档类型定义(DTD)是它的核心所在。DTD为组织文档的文档元素提供了一个框架,同时为文档元素之间的相互关系制定了规则。SGML文档具有极强的完整性和稳定性,其可适用的范围也相当广。然而,提供如此完整和稳定功能的文档语言,也导致其自身相当复杂,难以让人掌握。

1.1.2 HTML 的出现

HTML(Hyper Text Markup Language,超文本标记语言)是目前网络上应用最为广泛的语言,也是构成网页文档的主要语言。

1989年,欧洲粒子物理实验室的研究人员开发了一个基于SGML的超文本简化版本,被称为超文本标记语言(HTML)。HTML是SGML的一个子集,继承了SGML的许多重要特点。HTML只使用了SGML中很小一部分标记,并且为了便于在计算机上实现,HTML可使用的标记是固定的,也不需要包含DTD,因而其语法是不可扩展的。HTML这种固定的语法使其易学易用,开发用来处理HTML文档的浏览器也十分容易。正是由于HTML的简单性和易用性,使Web技术及其应用得到了空前的发展,并很快从计算机界走向全社会,最终走向千家万户。

HTML使用各种定义好的标记来分别标识和格式化不同的网页元素,使网页制作与Web信息发布变得非常简单。现在,普通人无需了解HTML的语法就可以使用各种所见即

所得的HTML编辑器,如Dreamweaver、FrontPage等,进行Web页面的创作。HTML已经成为最主要的Web页面语言。

下面就是一个简单的 HTML 网页代码:

```
1.  <!DOCTYPE html>
2.  <html>
3.  <head>
4.  <title>表格标记</title>
5.  </head>
6.  <body>
7.  <div >
8.    <div >
9.      <h2>DJ/摇滚/翻唱</h2>
10.     <table width="100%" border="0" cellpadding="0" cellspacing="1">
11.       <tr >
12.         <td width="20%"><a href="#">DJ 前卫音乐</a></td>
13.         <td width="20%"><a href="#">中国 DJ 嗨嗨网</a></td>
14.         <td width="20%"><a href="#">久嗨 DJ 音乐网</a></td>
15.       </tr>
16.       <tr >
17.         <td><a style="color: #FF0000" href="#">嗨嗨吧 DJ 音乐</a></td>
18.         <td><a href="#">百度 MP3-热门 DJ</a></td>
19.         <td><a href="#">DJ400 舞曲网</a></td>
20.       </tr>
21.     </table>
22.   </div>
23. </div>
24. </body>
25. </html>
```

上面的 HTML 网页代码在页面中显示了一个三行三列的表格,表格中列出了六个音乐网站的名称和链接。

作为互联网的一个组成部分,HTML 以简单精练的语法、极易掌握的通用性与易学性,使互联网得以普及发展以至今日辉煌。然而,从目前的发展来看,HTML 却存在着相当大的局限性。

(1) 由于标准的 HTML 标记已经由 W3C 预先确定,不能根据需要自行定义,所以当描述具有各种复杂内容的文档时,HTML 就显得力不从心。

(2) HTML 面向的是数据格式的描述,而非面向数据对象本身,因此,HTML 标记并没有给出数据内容的含义。

(3) 使用目前的 HTML,网页文档开发者必须对文档进行许多调整才能兼容各种流行的浏览器。

(4) 由于浏览器不检查网页中的错误 HTML 代码,因而导致 Internet 上有大量的文档

包含了错误的 HTML 语法，这个问题越来越严重。

由于 HTML 存在着以上这些局限性，因此 W3C 开始寻找解决方案。

1.2 XML 概述

XML(Extensible Markup Language, 可扩展标记语言)是在简化 SGML 的基础上发展而来的，它是 SGML 的一个子集，保留了 SGML 标准的很多优点，能更加容易操作和在互联网环境下实现。XML 是为文档交换所设计的，以一种开放的、自我描述的方式定义了数据结构。

1.2.1 为什么要使用 XML

1998 年 2 月，W3C 协会正式推出了 XML，这是一种可以用来创建标记语言的元语言，它可以用来标记任何一种所能想见的事物。数学公式、化学分子结构、音乐符号这些行业信息都在 XML 中得以结构化地表示，跨平台的信息交换也可以制订基于 XML 的通信协议。就像 HTML 在 Web 和全球数据发布中所起的作用一样，XML 在电子商务和数据交换中起着重要的作用。XML 不仅已广泛用于与计算机和网络有关的各个方面，还可用于数学、机械、生物、化学、物理、音乐制作等各个领域，它在跨平台、跨应用程序和跨语言中的使用，使它在分布式解决方案中发挥着重要作用。

XML 保留了 SGML 的一些特点，并克服了 HTML 的局限性。其主要特点是：

- XML 可用于现有的 Web 协议(如 HTTP 和 MIME)和机制(如 URL)。
- XML 支持 Web 的各种不同的应用，并使用了一种类属的方法使其具有可扩展性。HTML 则不支持，它需要借助用于链接 HTML 和应用的脚本语言才能支持 Web 的各种不同的应用。
- XML 与 SGML 兼容，所以大多数 SGML 应用可以向 XML 转化。
- 与 HTML 文档一样，XML 文档易于创建。
- XML 文档内容和结构清晰易懂，所以即使对非专业人员来说也易于阅读和使用。
- XML 的设计严谨，所以 XML 中标记的信息可以很容易地被计算机程序所处理。
- XML 标准定义精练，这就保证了其下载和处理的速度。

1.2.2 一个简单的 XML 例子

在了解了 XML 的历史、基本概念后，为了让读者对 XML 有一个初步的了解，我们通过一个简单的 XML 文档来揭开 XML 的面纱，并对其中的代码及其意义加以简单的分析。

实例 1-1 XML 的简单例子

本例中的 XML 文档是一个描述三个火车车次信息的 XML 文档。具体实现步骤如下。

(1) 打开 Windows 中的记事本，创建一个文件后缀名为.xml 的文件，命名为 trainschedule，

在文件中输入如下代码:

```
1.  <?xml version="1.0" encoding="GB2312" ?>
2.  <!-- 文件名: trainschedule.xml -->
3.  <火车时刻表>
4.      <车次>T222
5.          <起始站>北京</起始站>
6.          <终点站>成都</终点站>
7.          <开车时间>18:20</开车时间>
8.          <到达时间>08:18</到达时间>
9.      </车次>
10.     <车次>T223
11.         <起始站>上海</起始站>
12.         <终点站>北京</终点站>
13.         <开车时间>22:30</开车时间>
14.         <到达时间>09:58</到达时间>
15.     </车次>
16.     <车次>T224
17.         <起始站>广州</起始站>
18.         <终点站>武汉</终点站>
19.         <开车时间>09:20</开车时间>
20.         <到达时间>23:20</到达时间>
21.     </车次>
22. </火车时刻表>
```

这是一个典型的 XML 文档,第 1 行语句用来声明该文档为一个符合 XML 规范的文档。其中, <?xml 表示 XML 文档声明的开始,说明该文档是一个 XML 文档; ?>表示文档声明的结束; version="1.0"表示的是必须声明的 XML 版本信息,用来声明该文档遵循 W3C 的 XML 1.0 标准; encoding="GB2312"用来声明该文档使用 GB2312 简体中文字符集。

第 2 行是一个注释语句,第 3 行~第 22 行是 XML 元素。其中,第 4 行~第 9 行是定义第 1 个火车车次的具体信息,包括了车次、起始站、终点站、开车时间和到达时间这样 5 个元素(节点)。以下第 10 行~第 15 行,第 16 行~第 21 行又定义了其他两个火车车次的元素信息。

正如大家所看到的,XML 文档并不复杂。同 HTML 一样,XML 文档也是由一系列的元素组成。但是,XML 文档中元素的标记是自定义的,并具有明确的含义。可以看出,XML 文档中数据元素的组织是一种树状的层次结构,其根元素包含了所有下层子元素,而每个下层子元素除了可以有自己的文字内容和属性之外,又可包含若干更下层的子元素。如本例中根元素是“火车时刻表”,下层子元素是“车次”,更下层的子元素是“起始站”、“终点站”、“开车时间”和“到达时间”。这些元素的标记名称并不是 XML 规范预先定义好的,而是由用户自己根据描述数据的需要而命名的。由此可以看出,XML 文档结构清晰易读,而且它是根据开放的标准建立的。

和 HTML 文档一样,XML 文档实际上是用纯文本撰写的文本文件,可以使用任何一

种文字编辑软件来创建和修改。例如,可以使用 Microsoft Windows 提供的“记事本”编辑器。XML 文档创建完成后应该用.xml 作为文件扩展名加以保存。

(2) 通常可以直接在 IE 浏览器中打开 XML 文档,就像打开 HTML 网页一样。如果 XML 文档并未链接有关的样式表,IE 浏览器将会显示文档中的全部文字内容,包括所有标记及数据的文字内容。在这种情况下,IE 浏览器将以不同颜色的代码来表示文档的不同组成部分,并且可以收缩/伸展的树状结构来清楚地标明文档中各个元素的层次结构。比如,将上面的 trainschedule.xml 范例文档在 IE 浏览器中打开,将显示如图 1-1 所示的结果。

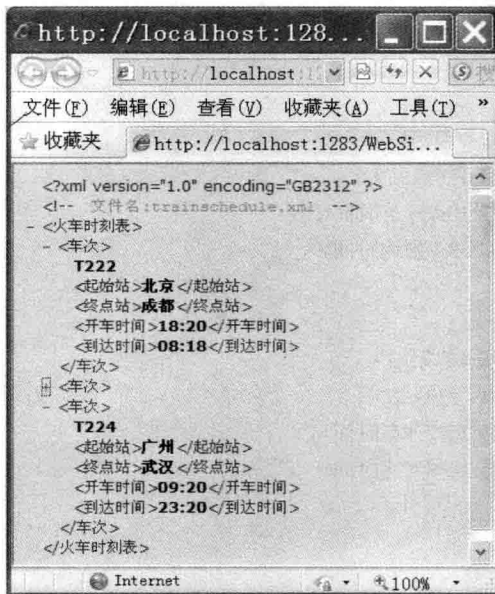


图 1-1 XML 文档显示结果

此时,在IE浏览器窗口中,如果单击某元素开始标记左端的减号“-”,将会使该元素的内容收缩起来;而单击元素标记左端的加号“+”,将会使该元素已收缩的内容再次展开。如果单击根元素开始标记旁的减号“-”,将会使文档中的全部元素都收缩起来。

大家可以发现,在IE浏览器中显示的XML文档几乎和源文档的格式与内容一样。这是因为对于XML文档本身而言,人们通常只关注它的数据内容,而它的显示形式是由CSS(层叠样式表)或XSL(可扩展样式表)定义的。如果XML文档包含了所链接的某个样式表,IE浏览器将只显示XML文档中的数据,同时根据这个样式表中指明的显示规则来安排这些数据的显示格式。实际上,对于电子数据的存储和交换,仅需要一个XML文档即可,只有在需要将数据以某种形式显示出来时,才必须为其编写并链接相应的CSS或者XSL样式表文件。

在浏览器中打开某个XML文档时,如果在文档中出现任何差错,都会得到文档不能被显示的结果。各浏览器软件开发厂商已经达成协议,对XML文档实行严格的解析,任何细小的错误都会被报告。

1.2.3 XML 的优势

XML 受到业界的广泛欢迎不仅因为它的应用前景十分广泛,更重要的是它基于文本,非常容易理解和使用。比起之前的几种标记语言,XML 具有十分明显的优势。

1. 良好的可读性

在HTML文档中,数据是没有含义的,人们无法知道某个数据具体代表什么。然而在XML文档中,人们可以使用特定的标记为数据定义相关的语义。例如,可以规定某个数据代表物体,某个数据代表人名等。这样一来,XML文档数据的可读性就大大提高了。

2. 方便的可扩展性

XML 允许各个组织或个人建立适合他们自己需要的标记集合或标记库,并且这些标记集合可以迅速地投入到互联网的使用中。实际上,现在许多行业和机构都在利用 XML 制订业内使用的标记语言标准。比较典型的有:化学标记语言(CML)、数学标记语言(MathML)、矢量图形标记语言(VML),以及无线通信标记语言(WML)等。

3. 数据内容与形式的分离

在 XML 文档中,数据的显示样式已从文档中分离出来,放入相关的样式表文件中。这样一来,如果要改动数据的表现形式,就不需要改动数据本身,而只需改动控制数据显示的样式表文件即可。XML 的优势正是在于能够确保同一网络站点的数据信息在不同设备上成功地显示。

4. 便于信息的检索

由于 XML 通过给数据内容贴上标记来描述其含义,并且把数据的显示格式分离出去,因此对 XML 文档数据的搜索可以简单高效地进行。在此情况下,搜索引擎没有必要遍历整个文档,而只需查找指定标记的内容即可。此外,XML 文档所附加的结构和语义信息,允许计算机立刻在本地处理大量数据,从而减轻了网络服务器的负担,极大地降低了网络流量。

5. 可以轻松地跨平台应用

XML文档是基于文本的,所以很容易被人和机器阅读,也非常容易使用,便于不同设备和不同系统间的信息交换。XML可以定义自己的数据格式,很容易使用软件工具去读、写、交互这些格式的文档。不仅如此,XML可以支持各种通道的数据传输,不同系统或不同设备之间完全可以采用XML作为信息交换的平台和媒介,建立起更为有效的应用。

6. 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符号编码系统。所以,XML 技术使各种信息不但能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。

7. 为科学实践与交流提供捷径

XML为科学家进行科学实践,交流理论、计算机及实验结果提供了一条特别便利的途径。例如:MathML(数学标记语言)支持数学家在网页中插入表达式和几行相关的简单文字,便于浏览者把这些表达式直接插入相关的数学软件供计算和作图使用;CML(化学标记语言)和AML(天文学标记语言)能帮助研究人员检索大量杂志引文,并迅速筛选出恰好适用于他们研究课题的论文;BSML(生物序列标记语言)则可以支持遗传学家交流和处理基因图谱绘制和基因序列分析工程所产生的信息,使研究者能轻易搜索巨大的遗传密码数据库,并把最终片段显示成有意义的图谱而不是杂乱无章的字母序列。

8. 大大改善网页的超链接性能

使用XML来描述网页数据,就如同为互联网制作了一个图书馆卡片目录。目前已经完成的RDF(资源描述框架)将使数十亿通过超链接联系起来的网页,能够在XML的支持下发挥更大的作用。

9. 便于信息的长期保存

现代社会的绝大部分资料都是以电子文档形式保存的,并且以不同格式保存的文档需要有相应的不同软件来将其打开。若干年后,很可能某些电子文档还在,但能够打开这些文档的软件则已遭淘汰而无法找到。以XML格式保存的文档就不会有上述问题,因为XML文档是基于文本的,并且文档中的每项数据都有清晰的语义,非常容易打开和阅读。此外,XML文档很容易转换为其他格式的文档,所以非常适合用来作为信息的长期保存形式。

10. 适合面向对象的程序开发

XML文档是非常容易阅读的,不仅对人,对机器也是如此。XML文档数据的逻辑结构是一种树形的层次结构,文档中的每一个元素都可以影射为一个对象,同时也可以有相应的属性和方法,因而非常适合于使用面向对象的程序设计方式来开发处理这些XML文档的应用程序。

总之,XML的技术优势十分明显,其应用前景必将十分广阔。

1.3 XML 开发工具

Altova XMLSpy是一个符合行业标准的XML集成开发环境(Integrated Development Environment,简称IDE)。它为用户提供最佳的可扩展标记语言编辑器、模式设计器、代码生成器、文件转换器、调试器和配置器,并支持XSLT(可扩展样式表语言转换)、Xquery(XML查询语言)、WSDL(Web服务描述语言)、SOAP(简单对象访问协议)技术、.NET和Eclipse等集成开发环境。这里介绍的版本是XMLSpy 2011 中文企业版。