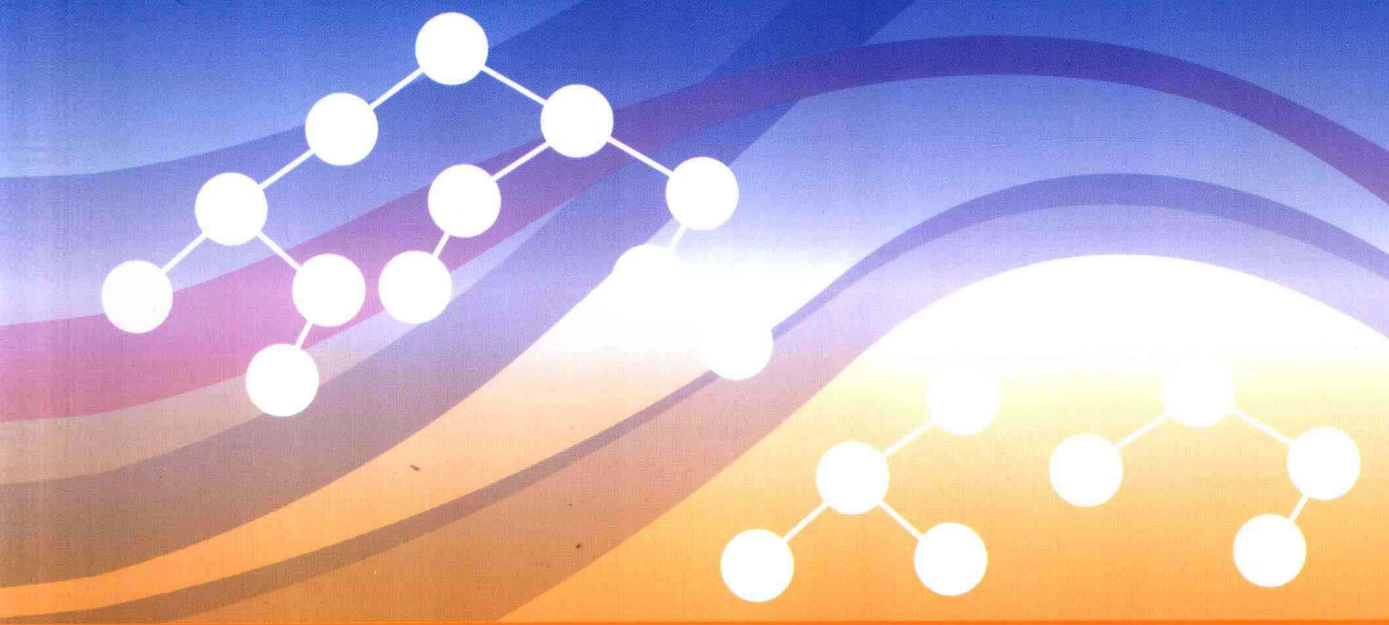
高等学校计算机专业规划教材

XML Technology and Applications

XML技术与应用



彭涛 孙连英 编著

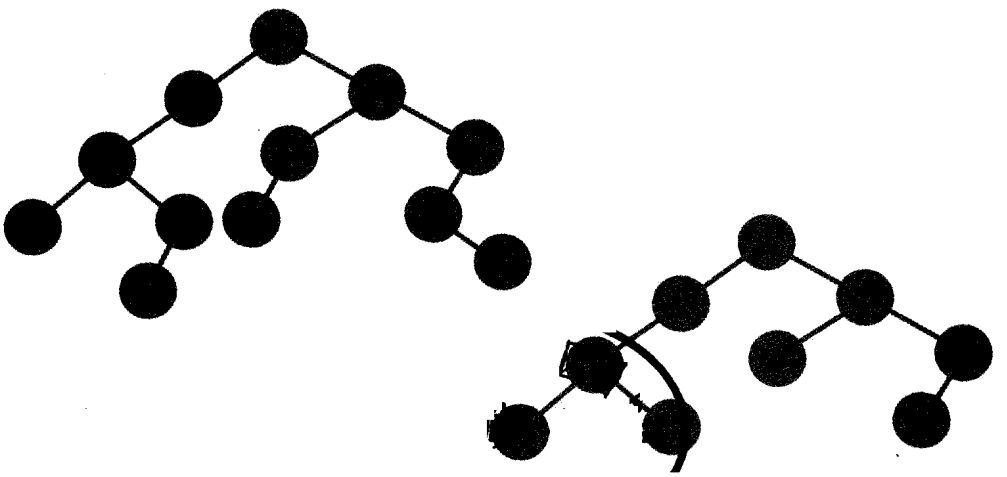
清华大学出版社

高等学校计算机专业规划教材

XML Technology and Applications

XML技术与应用

彭涛 孙连英 编著



清华大学出版社
北京

内 容 简 介

XML 是由 W3C 定义的一种语言，是表示结构化数据的行业标准。XML 在电子商务、移动应用开发、Web Service、云计算等技术和领域中起着非常重要的作用。本书不仅结合实例详细讲解了 XML 的基础知识，同时也就 XML 的主要应用领域进行了案例讲解。

本书共 12 章，内容包括 XML 简介、XML 的规范性、XML 的有效性(包括 DTD 和 XML Schema)、XML 的应用、XML 的转换 XSLT、XML 的解析(包括 DOM、SAX 和 dom4j)等，其中有 4 章用案例讲解了 XML 在不同领域和技术中的应用，这些案例容易理解且均可运行，对于读者理解 XML 的应用有很大的帮助。

本书适合作为高等院校软件工程、计算机科学与技术等相关专业的研究生参考教材，也可作为相关专业的高年级本科教材，同时也可作为初学者学习 XML、Android 移动应用开发、Java EE 开发的培训教材。

图书在版编目(CIP)数据

XML 技术与应用/彭涛，孙连英编著. —北京：清华大学出版社，2012.6
高等学校计算机专业规划教材
ISBN 978-7-302-28466-6

Ⅰ. ①X…　Ⅱ. ①彭…　②孙…　Ⅲ. ①可扩充语言，XML—程序设计—高等学校—教材　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 064917 号

责任编辑：龙启铭
封面设计：常雪影
责任校对：时翠兰
责任印制：李红英

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
印 刷 者：三河市君旺印装厂
装 订 者：三河市新茂装订有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：19.5　　**字　　数**：454 千字
版　　次：2012 年 6 月第 1 版　　**印　　次**：2012 年 6 月第 1 次印刷
印　　数：1～3000
定　　价：29.50 元

产品编号：047641-01

前　言

背景

XML是由万维网联盟(W3C)制订的一种标记语言，是表示结构化数据的事实上的工业标准，广泛应用于结构化数据的存储和交换。XML提供了直接在数据上工作的通用方法，其优势在于将结构化数据的存储和显示相分离，允许不同来源数据的无缝集成和对同一数据的多种处理。

目前在软件开发、电子商务等领域，几乎处处都能看见XML的身影。很多人可能觉得XML非常简单，因为他们每天都会接触XML文档，但是许多人仅仅知道XML可以用作配置文件，至于更多的、更深的内容就知之甚少了。

其实，XML还可以作为一种轻量级的持久化解决方案，就像数据库一样。XML也可以直接作为表现层来代替传统的HTML。实际上，XML无论对于Java平台，还是其他如.NET平台都具有非常重要的价值。XML在电子商务、移动应用开发、Web Service、云计算等技术和领域中也起着非常重要的作用。

随着云计算的兴起，本地和云端之间通过Internet进行数据传输和数据交换成为必然的趋势，这其中XML仍然承担着数据存储和数据交换的重任。

本书特色

任何技术的目的都是为了应用。本书不仅结合实例详细讲解了XML的基础知识，同时也就XML的主要应用领域进行了实例讲解。全书共分12章，内容包括XML简介、XML的规范性、XML的有效性(包括DTD和XML Schema)、XML的转换XSLT、XML的解析(包括DOM、SAX以及dom4j)、XML的应用(包括数据存储与数据交换、AJAX、Web Service、Android应用开发等)，另外，对JSON这种数据交换的新方法也进行了讲解。

本书特点如下：

(1) 目前市面上大多数XML书籍偏重于介绍XML技术本身，关于其应用的篇幅较少。在本书有关XML的应用领域选取了有代表性的以下内容：数据存储(包括配置文件)、数据交换、AJAX、Web Service、Android移动应用开发。这些都是目前软件开发中的热门领域和技术。

(2) 应用案例贯穿全书，应用案例容易理解并均可运行，对于读者理解XML的应用有着很大的帮助。

(3) 全书主要使用表示股票行情信息的XML文档作为案例，部分需要使用命名空间的章节使用包含书籍和酒店预订的案例。

(4) 对于XML文档的解析和处理贯穿全书，每章都有相关的处理XML文档的Java

应用程序案例，包括普通应用程序、基于 HTTP 协议的网络应用程序、Servlet、Android 应用程序等。

读者对象

本书可作为高等院校软件工程、计算机科学与技术等相关学科的研究生参考教材，也可作为相关专业的本科教材，同时也可作为初学者学习 XML、Android 移动应用开发、Java EE 开发的培训教材。

本书作者

本书由北京联合大学信息学院彭涛、孙连英编写，其中，第 3、4、5、6、9、11 章由彭涛编写，第 1、2、7、8、10、12 章由孙连英编写，全书由彭涛统稿。在编写过程中，得到了华阳、章磊、闫鑫、杨海涛、王满、郭强等的帮助，在此表示感谢。

本书受到北京市教育委员会科技发展计划项目（KM201211417002）、北京联合大学人才强校计划的资助。

对于本书实例开发中的程序源代码，读者可以在清华大学出版社网站上免费下载。

书中遗漏或错误之处，敬请读者批评指正。

目　录

第 1 章　XML 简介

1.1　什么是 XML

1.1.1　XML 的产生背景

1996 年，万维网相关技术的主要设计组织 W3C(World Wide Web Consortium，万维网联盟)开始创建一种可扩展的标记语言，它要能够结合 SGML(Standard Generalized Markup Language，标准通用标记语言)的灵活性并能像 HTML(Hypertext Markup Language，超文本标记语言)一样可以被广泛接受，这种语言就是 XML 语言，其全称是 Extensible Markup Language，称为可扩展标记语言。所谓可扩展是指 XML 允许用户按照 XML 规则自定义标签。XML 文件是由标签及其所包含的内容构成的纯文本文件，与 HTML 文件不同的是，这些标签可自由定义，其目的是使得 XML 文件能够很好地体现数据的结构和含义。W3C 推出 XML 的主要目的是让数据的内容更加容易理解，使基于 Internet 的数据交换更方便。W3C 的主页是 http://www.w3c.org，关于 XML 的网页在 http://www.w3c.org/XML，大部分的技术文档可以在 http://www.w3c.org/XR 找到。XML 1.0 版本是由 W3C 在 1998 年 2 月的推荐标准中定义的，W3C 的推荐标准就像 Internet 的 RFC(Request for Comments)一样，是一种非正式的“标准”。文档中的许多小问题和基础标准的一些变化导致了 2000 年 10 月第 2 版的出版，第 2 版修正和更新了文档，但没有改变 XML 本身。

W3C 为 XML 制定了 10 个设计目标，具体内容包括：

(1) XML 应该可以在 Internet 上直接使用；

(2) XML 应该广泛地支持不同的应用；

(3) XML 要和 SGML 兼容；

(4) 处理 XML 文档的程序应该容易编写；

(5) XML 的可选特征应该保持绝对最低限，最好是零；

(6) XML 文件要易读、清晰；

(7) XML 的设计应该可以快速预备；

(8) XML 应设计得正规、简洁；

(9) XML 文件应该很容易创建；

(10) XML 标签的简洁性应该是无关紧要的。

1.1.2　一个 XML 文档示例

示例 1.1 是一个简单的 XML 文件，该文件包含了 NASDAQ 市场股票行情的有关数据。

示例 1.1：Chapter01_01.xml

```
<?xml version="1.0"?>
<Stock>
    <Symbol>SINA</Symbol>
    <Last>59.92</Last>
    <Date>1/13/2012</Date>
    <Time>4:00pm</Time>
    <Change>-0.20</Change>
    <Open>59.16</Open>
    <High>60.95</High>
    <Low>58.41</Low>
    <Volume>3991991</Volume>
    <PreviousClose>60.12</PreviousClose>
    <AnnRange>46.86 -147.12</AnnRange>
    <Name>Sina Corporation</Name>
</Stock>
```

示例 1.1 中的 XML 文件包含了一个 XML 声明(有关 XML 声明将在第 2 章介绍)：

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

和 13 个标签。每个标签都必须包括开始标签和结束标签，开始标签和结束标签之间的内容称为该标签所标记的内容，简称“标签的内容”。一个标签的内容可以包含文本或者其他标签，其中包含的标签称为该标签的子标签。XML 文件有且仅有一个根标签，其他标签都必须被根标签所包含，整个 XML 文件形成一个树状结构。在示例 1.1 的 XML 文件中，根标签的开始标签是＜Stock＞，结束标签是＜/Stock＞，该根标签有 12 个子标签。XML 文件必须符合一定的语法规则，只有符合这些规则，XML 文件才可以被 XML 解析器解析，以便利用其中存储的内容。第 2 章会详细地讲解 XML 的语法规则。

下面的示例 1.2 和示例 1.3 中的 XML 文件都是错误的，其中示例 1.2 没有根标签，而示例 1.3 中标签 Symbol 与 Last 之间存在交叉，没有形成树状结构。

示例 1.2：Chapter01_02.xml

```
<?xml version="1.0"?>
<Stock>
    <Symbol>SINA</Symbol>
    <Last>59.92</Last>
</Stock>
<Stock>
    <Symbol>SINA</Symbol>
    <Last>59.92</Last>
</Stock>
```

示例 1.3：Chapter01_03.xml

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

```
<Stock>
    <Symbol>SINA
    <Last></Symbol>59.92</Last>
</Stock>
```

1.2 XML 与 HTML 的关系

XML 的语法规则非常严格，这一点和 HTML 有很大的不同，HTML 本身语法十分不严格，这在一定程度上影响了网络信息的传输和共享。W3C 吸取了 HTML 发展的经验和教训，对 XML 指定了严格的语法标准。例如，标签都必须要有一个开始标签和结束标签，所有的标签都必须合理嵌套，即形成树状结构。也就是说，XML 文件必须符合一定的语法规则，只有符合这些规则，XML 文件才可以被 XML 解析器解析，以便利用其中存储的数据。XML 文件分为格式良好的(well-formed)XML 文件和有效的(validated)XML 文件。符合 W3C 制定的基本规则的 XML 文件称为格式良好的 XML 文件，格式良好的 XML 文件如果再符合额外的关于标签的约束则称为有效的 XML 文件。格式良好的 XML 文件的内容参见第 2 章、第 3 章和第 4 章将讲解有效的 XML 文件。

为了检查一个 XML 文件是否是格式良好的，一个简单的方法就是使用浏览器(例如 IE)打开 XML 文件。如果 XML 文件是格式良好的，浏览器将显示 XML 文件的内容，否则，将显示错误信息。图 1.1 显示了使用 Microsoft Internet Explorer(以下简称 IE)打开示例 1.1 的结果。图 1.2 显示了使用 IE 打开示例 1.2 时浏览器的报错信息，错误信息为“XML 文档只能有一个顶层元素”，即根元素，报错在第 6 行，即第二个 Stock 标签的开始标签处，因为浏览器发现作为文档根元素出现的元素 Stock 出现了两次，违背了“XML 文档只能有一个顶层元素”的规则。

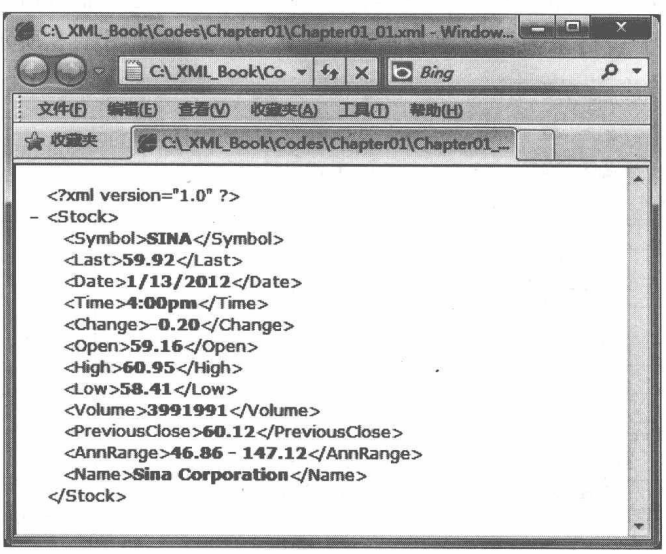


图 1.1 示例 1.1 文档在浏览器中的显示结果

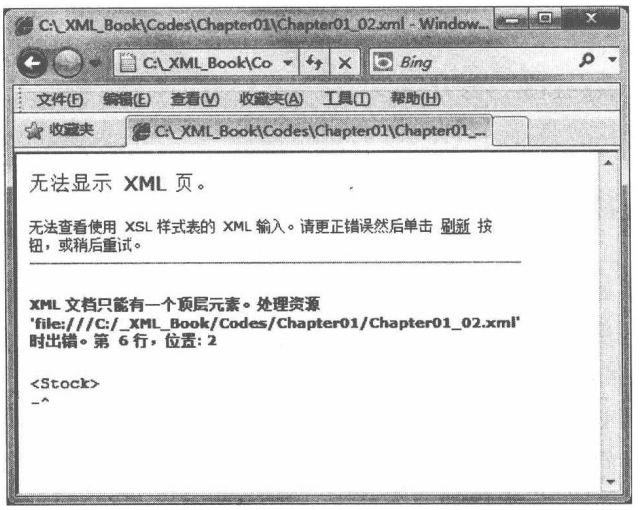


图 1.2　示例 1.2 文档在浏览器中显示的错误信息

XML 可以很好地描述数据的结构，有效地分离数据的结构和数据的显示，可以作为数据交换的标准格式，而在 AJAX（Asynchronous JavaScript And XML，异步的 JavaScript 与 XML，详见第 10 章）、Web Service（Web 服务，详见第 11 章）、云计算等相关技术中，XML 已经是数据交换领域事实上的行业标准。而 HTML 是用来编写 Web 页面的语言，HTML 同时存储了数据的内容和数据的显示外观，如果只想使用数据而不需要显示，则需要对 HTML 进行专门的处理，例如在 Internet 上广泛使用的搜索引擎，在抓取得到 Web 页面之后，就需要去除页面中包含的标签，保留页面中有用的数据并用于建立索引。另外，HTML 不允许用户自定义标签，目前的 HTML 大约有 100 多个标签。HTML 不是专门用于存储数据的结构，它主要用于描述数据的显示格式。示例 1.4 中的 HTML 文档使用了表格来显示股票行情数据，其显示结果如图 1.3 所示。

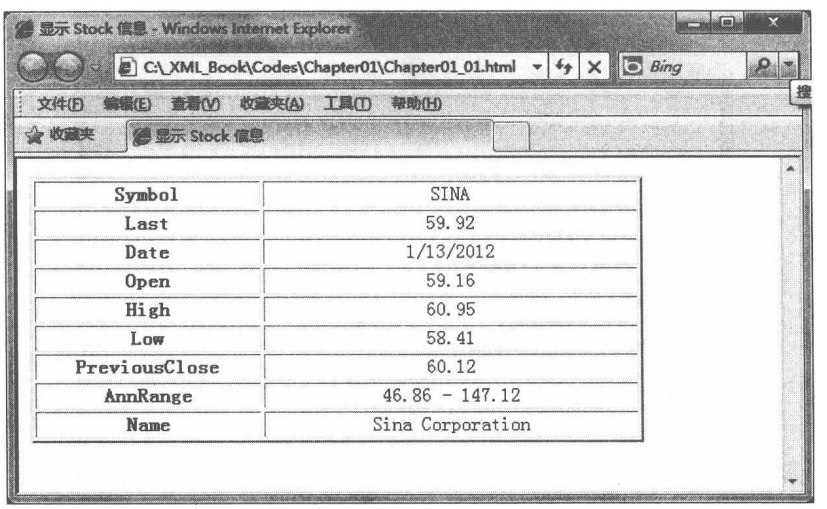


图 1.3　示例 1.4 中 HTML 文档在浏览器中的显示结果

示例 1.4：Chapter01_01.html

```
<html>
    <head>
        <title>显示 Stock 信息 </title>
    </head>
    <body>
        <table border="2">
            <tr>
                <th align="center" width="180">Symbol</th>
                <td align="center" width="300">SINA</td>
            </tr>
            <tr>
                <th align="center">Last</th>
                <td align="center">59.92</td>
            </tr>
            <tr>
                <th align="center">Date</th>
                <td align="center">1/13/2012</td>
            </tr>
            <tr>
                <th align="center">Open</th>
                <td align="center">59.16</td>
            </tr>
            <tr>
                <th align="center">High</th>
                <td align="center">60.95</td>
            </tr>
            <tr>
                <th align="center">Low</th>
                <td align="center">58.41</td>
            </tr>
            <tr>
                <th align="center">PreviousClose</th>
                <td align="center">60.12</td>
            </tr>
            <tr>
                <th align="center">AnnRange</th>
                <td align="center">46.86-147.12</td>
            </tr>
            <tr>
                <th align="center">Name</th>
                <td align="center">Sina Corporation</td>
            </tr>
        </table>
    </body>
</html>
```

上述 HTML 文档从文档的结构上无法体现数据之间的逻辑关系，浏览器仅仅是能将上述标签中所包含的文本采用表格显示方式显示在浏览器中。和 HTML 不同的是，XML 可自定义标签，标签名称是对所包含数据内容含义的抽象，而不是数据的显示格式。XML 只关注数据的组织结构，以便 XML 解析器能够按照其结构对其进行访问，XML 本身不提供数据的显示格式。浏览器不能直接显示 XML 文件的标签内容，如果想让浏览器显示 XML 文件中标签的内容，就必须以某种方式告诉浏览器如何显示，一种方式是使用级联样式单(Cascading Style Sheet，CSS)，另一种方式是使用 XSLT(eXtensible Stylesheet Language Transformations)，与 XML 数据显示有关的内容将在第 5 章讲述。对于没有指定任何显示方式的 XML 文档(例如示例 1.1 中的 XML 文档)，大多数浏览器(包括 IE、Google Chrome 等)会以树状结构来显示，该树状结构与 XML 文档本身形成的树状结构是完全一致的，如图 1.1 所示。

1.3 XML 解析器

XML 解析器是 XML 和应用程序之间的一个中介程序，其目的是为应用程序从 XML 文件中解析出所需要的数据。例如，应用程序需要读取 XML 文件中存储的股票价格信息，并准备对价格进行数据分析处理。目前的 XML 解析器支持许多编程语言，本书主要采用 Java 语言(JDK 版本为 1.6)来处理 XML 文档。以下通过一个简单的示例程序 1.5 来说明 XML 解析器的用法，被解析的 XML 文档为示例 1.1 中的 XML 文件，程序运行结果如图 1.4 所示。

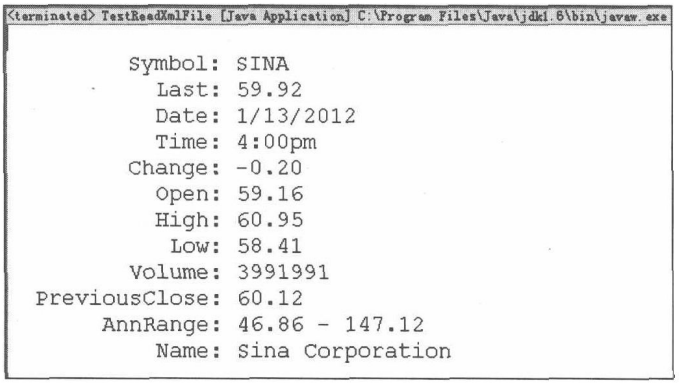


图 1.4 股票行情 XML 文档的解析结果

1.3.1 解析步骤

为了解析出示例 1.1 中 XML 文档存储的数据，需要包含如下步骤。

(1) 使用 javax. xml. parsers 包中的类 DocumentBuilderFactory，调用其静态方法 newInstance()来实例化一个 DocumentBuilderFactory 对象：