高等学校计算机应用规划教材

XML 编程与应用教程

（第 2 版）

孙更新　李伟超　李玉玲　编著

清华大学出版社

北　京

内 容 简 介

可扩展标记语言(XML)是一种新的 Web 开发辅助语言，利用它可通过 Internet 进行信息的描述、交换和显示。本书是学习和应用 XML 语言的实用教材，书中不仅详细阐述了 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、与数据库的集成、文档对象模型，还介绍了 XML 在 Java 和.NET 中的编程，最后通过一个综合案例和 5 个课程实验演示了 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，便于读者消化吸收所学内容。每章还包括了小结和习题，便于读者巩固所学的知识。

本书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，也可供相关研究人员、广大 Web 应用程序开发者和用户参考。

图书在版编目(CIP)数据

XML 编程与应用教程/孙更新，李伟超，李玉玲 编著. —2 版. —北京：清华大学出版社，2014
(高等学校计算机应用规划教材)

ISBN 978-7-302-35352-2

Ⅰ. X…　Ⅱ. ①孙… ②李… ③李…　Ⅲ. 可扩充语言—程序设计—高等学校—教材　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 020915 号

责任编辑：刘金喜　蔡　娟
装帧设计：孔祥丰
责任校对：成凤进
责任印制：

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，010-62796045

印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张：**20　　**字　　数：**462 千字
版　　次：2010 年 4 月第 1 版　　2014 年 4 月第 2 版　　**印　　次：**2014 年 4 月第 1 次印刷
印　　数：1～4000
定　　价：32.00 元

产品编号：

前　言

XML 是 Extensible Markup Language 的缩写，译为可扩展标记语言。XML 已经在 Web 编程、新型数据库系统、计算机网络应用编程、网络数据交换和跨平台编程中发挥越来越重要的作用。XML 正在成为电子商务运营和数据管理的核心技术。

本书是一本学习 XML 基本概念和基础理论、掌握 XML 开发技术的实用性图书，较为全面地介绍了 XML 语言及其相关技术，并在详细介绍 XML 语言及其相关标准的同时，注重 XML 技术在数据库和编程方面的实际应用，重点突出了 XML 与各种常用编程语言的结合。全书共分为 11 章和 5 个课程实验案例，具体内容如下。

第 1 章 XML 概述：主要介绍了什么是 XML，以及 XML 产生的背景、XML 的优越性和 XML 的常见应用等 XML 基础知识。

第 2 章 XML 语法：主要介绍 XML 的语法，重点介绍了 XML 的文档结构、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。

第 3 章 XML 文档类型定义：主要介绍确保 XML 文档有效的文档类型定义，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 模式定义：主要介绍了 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 5 章 XML 文件的显示：主要介绍如何使用级联样式表和可扩展样式语言来显示 XML 文档，重点介绍了 XPath、XSL 模板、XSLT 语法元素等。

第 6 章 XML 和 Microsoft SQL Server 的集成：在 Microsoft SQL Server 2008 中可以应用 XML 技术方便地进行数据处理，本章将介绍 Microsoft SQL Server 2008 对 XML 的支持及其相互转换。

第 7 章 XLink 和 XPointer：主要介绍了 XLink 和 XPointer 的相关概念及具体使用方法。

第 8 章 XML DOM 编程模型：重点介绍文档对象模型结构以及如何使用该模型操作 XML 文档。

第 9 章 Java 中的 XML 编程：详细介绍了 Java 解析和操作 XML 文档的 3 种不同方式，以及在每种不同的方式下具体的编程方法。

第 10 章 .NET 中的 XML 编程：主要介绍了在.NET 平台上操作和处理 XML 文件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，以及与 XML 及 ADO.NET 技术的关系。

第 11 章 基于 XML 的在线相册：通过一个综合案例演示了在实际项目开发中 XML 的应用。

5 个课程实验案例：网上论坛、家庭财务管理系统、人事档案管理系统、影院订票系统

和在线投票管理系统。

本课程总共有 60 学时，各章学时分配见下表(供参考)。

学时分配建议表

课 程 内 容	学 时 数			
	合 计	讲 授	实 验	机 动
第 1 章　XML 概述	1	1		
第 2 章　XML 语法	3	2	1	
第 3 章　XML 文档类型定义	6	4	2	
第 4 章　XML 模式定义——XSD	6	4	2	
第 5 章　XML 文件的显示——CSS 和 XSL	8	6	2	
第 6 章　XML 和 Microsoft SQL Server 的集成	3	2	1	
第 7 章　XLink 和 XPointer	6	4	2	
第 8 章　XML DOM 编程模型	3	2	1	
第 9 章　Java 中的 XML 编程	4	3	1	
第 10 章　.NET 中的 XML 编程	6	4	2	
第 11 章　基于 XML 的在线相册	8	3	3	2
附录　课程实验	6	1	5	
合　计	60	36	22	2

本书将技术阐述与实践应用相结合，强调理论结合实际，全书始终以介绍 XML 中已成熟的标准和应用技术为主。书中的应用实例均来自实际开发工作，读者对其稍加修改后即可直接应用到实际开发中。

本书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，也可供相关研究人员、广大 Web 应用程序开发者和用户参考。

本书 PPT 教学课件可通过 http://www.tup.com.cn/downpage 下载。

本书由孙更新、李伟超（编写第 5~7 章）、李玉玲编著。此外，吕平、王坚宁、王魁、许小荣、王东、王龙、王松年、张凤琴、陈可汤、陈作聪、沈毅、周艳丽、张璐、祁招娣等同志也参与了本书的编写，在此，对他们表示衷心的感谢。

由于 XML 相关技术标准在不断发布和更新，加之时间仓促和作者水平有限，本书中难免会有纰漏和不足之处，恳请各位专家和读者批评指正。

编　者

2014 年 1 月

目　　录

第1章　XML 概 述

XML 是 Internet 环境中跨平台的、依赖于内容的技术，是当前处理分布式结构信息的选择工具，它可以简化文档信息在 Internet 中的传输。XML 不仅满足 Web 应用开发人员的需要，而且还适用于电子商务、电子政务、数据交换等多个领域。本章将介绍什么是 XML、XML 产生的背景、XML 的优越性和 XML 的应用，以及 XML 开发工具 XMLSpy 2011 的安装与使用。

本章重点：

- XML 的定义
- XML 的应用背景
- XML 开发工具

1.1　XML 的概念

XML(Extensible Markup Language，可扩展标记语言)是一套定义语义标记的规则，这些标记将文档分成许多部件并对这些部件加以标识。它也是元标记语言，可以定义其他与特定领域有关的、语义的、结构化的标记。

可扩展标记语言(XML)是 SGML(Standard Generalized Markup Language，标准通用标记语言)的子集，其目标是允许普通的 SGML 在 Web 上以目前 HTML(HyperText Mark-up Language，超文本标记语言)的方式被服务、接受和处理。XML 的定义方式易于实现，并且可以在 SGML 和 HTML 中进行操作。

XML 由 XML 工作组(原先的 SGML 编辑审查委员会)开发，此工作组由 World Wide Web Consortium(W3C)在 1996 年主持成立。工作组由 Sun Microsystems 的 Jon Bosak 负责，同样由 W3C 组织的 XML SIG(Special Interest Group，原先的 SGML 工作组)也积极参与了 XML 工作组的工作。

XML 最初的设计目标如下。

- XML 应该可以直接用于 Internet。
- XML 应该支持大量不同的应用。
- XML 应该与 SGML 兼容。
- 处理 XML 文件的程序应该容易编写。
- XML 中的可选项应无条件地保持最少，理想状况下应该为 0 个。
- XML 文件应该是人们可以直接阅读的，应该是条理清晰的。
- XML 的设计应能快速完成。

- XML 的设计应该是形式化的、简洁的。
- XML 文件应易于创建。
- XML 标记的简洁性是最后考虑的目标。

下面就是一段 XML 示例文档：

```
<myfile>
<title> XML 教程</title>
<author>张三</author>
<email>icewine@tom.com</email>
<date>20090330</date>
</myfile>
```

注意：

- 这段代码仅仅能让读者感性认识 XML，并不能实现什么具体应用。
- 其中类似<title>、<author>的语句就是自己创建的标记(Tags)，它们和 HTML 标记不一样，例如这里的< title>是文章标题的意思，而在 HTML 中<title>指页面标题。

XML 不同于 HTML，超文本标记的语言定义了一套固定的标记，用来描述一定数目的元素。例如，HTML 文档包括了格式化、结构和语义的标记。<B>就是 HTML 中的一种格式化标记，它使其中的内容变为粗体；<TD>也是 HTML 中的一种结构标记，指明内容是表中的一个单元。XML 是一种元标记语言，用户可以定义自己需要的标记。这些标记必须根据某些通用的规则来创建，但是标记的意义，具有较大的灵活性。例如，若用户正在处理与学籍有关的事情，需要描述学生的学号、姓名、年龄、家庭住址等信息，就必须创建用于每项的标记。新创建的标记可在文档类型定义(Document Type Definition，DTD)或是 XML Schema(XML 模式)中加以描述，而关于如何显示这些标记的内容则由附加在文档上的样式表提供。

例如，在 HTML 中，一首歌可能是用定义标题标记<dt>、定义数据标记<dd>、无序的列表标记<ul>和列表项标记<li>来描述的。但是事实上这些标记没有一件是与音乐有关的。用 HTML 定义的歌曲如下：

```
<dt>金曲 top1
<dd> 青花瓷
<ul>
<li>词：方文山
<li>曲：周杰伦
</ul>
```

而在 XML 中，同样的数据可能标记为：

```
<SONG>
<TITLE>青花瓷</TITLE>
<COMPOSER>周杰伦</COMPOSER>
<LYRICIST>方文山</ LYRICIST>
</SONG>
```

在这段代码中没有使用通用的标记如<dt>和<li>，而是使用了具有意义的标记，如<SONG>、<TITLE>、<COMPOSER>等。这种用法具有许多优点，包括源代码易于阅读，使人能够看出代码的含义。

1.2　XML 的产生背景

XML 最初的设计目的是为了电子数据交换，更具体地说是为电子数据交换提供一个统一的标准格式。

1.2.1　电子数据交换简介

EDI(Electronic Data Interchange，电子数据交换)是一种利用计算机进行商务处理的新方法。EDI 将贸易、运输、保险、银行和海关等行业的信息，用一种国际公认的标准格式，通过计算机通信网络，使各有关部门、公司与企业之间进行数据交换与处理，并完成以贸易为中心的全部业务过程。

EDI 不是用户之间简单的数据交换，EDI 用户需要按照国际通用的消息格式发送信息，接收方也需要按国际统一规定的语法规则，对消息进行处理，并使其他相关系统进行 EDI 综合处理。整个过程都是自动完成，无须人工干预，减少了差错，提高了效率。因此 EDI 又被人们通俗地称为“无纸贸易”。

EDI 是早期计算机网络发展的一个主要目的，而结构化信息的一个主要目的就是使数据交换成为可能。因为如果不同行业中需要交互使用的信息采用统一的模型标识，信息就能方便和高效地进行共享(对于 XML 来说，这个统一的模型就是 XSD)。

1.2.2　XML 的产生及其与 SGML、HTML 的关系

XML 同 HTML 一样，都来自 SGML(Standard Generalized Markup Language，标准通用标记语言)。早在 Web 出现之前，SGML 就已存在。正如它的名称所言，SGML 是国际上定义电子文件结构和内容描述的标准，是一种非常复杂的文档结构，主要用于具有大量高度结构化数据的防卫区和其他各种工业领域，便于分类和索引。同 XML 相比，SGML 的功能很强大，缺点是它不适用于 Web 数据描述，而且 SGML 软件的价格非常昂贵。SGML 十分庞大，既不容易学，又不容易使用，在计算机上实现也十分困难。鉴于这些因素，Web 的发明者——欧洲核子物理研究中心的研究人员根据当时(1989 年)的计算机技术，开发了 HTML。

HTML 只使用 SGML 中很少的一部分标记，例如 HTML 的早期版本中只定义了 70 余种标记。为了便于在计算机上实现，HTML 规定的标记是固定的，即 HTML 语法是不可扩展的。HTML 这种固定的语法使它易学易用，在计算机上开发 HTML 的浏览器也十分容易。正是由于 HTML 的简单性，使得基于 HTML 的 Web 应用得到极大的发展。

但是，随着 Web 应用的不断发展，HTML 的局限性也越来越明显地体现出来。

- 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。
- 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。
- 使用目前的 HTML，网页文档开发者必须要对文档进行许多的调整才能兼容各种流行的浏览器。
- 由于浏览器不去检查网页中错误的 HTML 代码，因而导致 Intemet 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。

尽管 HTML 推出了一个又一个新版本，但始终不能满足不断增长的需求。因此，有人建议直接使用 SGML 作为 Web 页面语言，这固然能解决使用 HTML 遇到的困难，但是 SGML 太庞大了，用户学习和使用不方便尚且不说，要全面实现 SGML 的浏览器就非常困难。于是自然会想到仅使用 SGML 的子集，使新的语言既方便使用又容易实现。正是在这种形势下，Web 标准化组织 W3C 建议使用一种精简的 SGML 版本，于是 XML 应运而生了。

XML 是一个精简的 SGML 子集，它将 SGML 丰富的功能与 HTML 的易用性结合到 Web 的应用中。XML 保留了 SGML 的可扩展功能，这使 XML 从根本上有别于 HTML。XML 要比 HTML 强大得多，它不再是固定的标记，而是允许定义数量不限的标记来描述文档中的资料，允许嵌套的信息结构。HTML 只是 Web 显示数据的通用方法，而 XML 提供了一个直接处理 Web 数据的通用方法。HTML 着重描述 Web 页面的显示格式，而 XML 着重描述的是 Web 页面的内容。

XML 中包括可扩展样式表语言(Extensible Style Language，XSL)和可扩展链接语言(Extensible Linking Language，XLL)两部分。

为了使 XML 易学易用，XML 精简了很多 SGML 中难得用一次的功能。正如几十万汉字中常用的只不过几千，SGML 常用的部分也只占 20%，XML 抛弃了 SGML 中不常用的部分，精简了 80%。这样一来，XML 的语法说明书就只有 30 页，而 SGML 却有 500 页。

XML 的设计中也考虑了易用性，易用性表现在两个方面：一方面，用户编写 Web 页面方便；另一方面，设计人员实现 XML 浏览器也不太困难。

总之，XML 使用一个简单而又灵活的标准格式，为基于 Web 的应用提供了一个描述数据和交换数据的有效手段。但是，XML 并非是用来取代 HTML 的。HTML 着重于如何描述才能将文件显示在浏览器中，XML 与 SGML 相近，着重于如何描述才能将文件以结构化方式表示。就网页显示功能来说，HTML 比 XML 要强，但就文件的应用范畴来说，XML 比 HTML 要超出很多。

1.3　XML 的优越性

XML 的优点主要表现在以下各方面。

- 更有意义的搜索。

数据可通过 XML 进行唯一的标识。没有 XML，搜索软件必须了解每个数据库是如何构建的。这实际上是不可能的，因为每个数据库描述数据的方法都不同。有了 XML，情况就完全不同了。

- 开发灵活的 Web 应用软件。

数据一旦建立，XML 就能被发送到其他应用软件、对象或者中间层服务器以做进一步的处理，它也可以被发送到桌面用浏览器浏览。XML 和 HTML、脚本、公共对象模式一起为灵活的三层 Web 应用软件的开发提供了所需的技术。

- 不同来源数据的集成。

XML 能够使不同来源、结构化的数据很容易地结合在一起。软件代理商可以在中间层服务器上对从后端数据库和其他应用处发来的数据进行集成。然后，数据就能被发送到客户或其他服务器上做进一步的集成、处理和分发。

- 多种应用得到的数据。

XML 的扩展性和灵活性允许它描述不同种类应用软件中的数据，从搜索到的 Web 页到数据记录。同时，由于基于 XML 的数据是自我描述的，数据不需要有内部描述就能被交换和处理。

- 本地计算和处理。

XML 格式的数据发送给客户后，客户可以用应用软件解析数据并对数据进行编辑和处理。使用者可以用不同的方法处理数据，而不仅仅是显示它。XML 文档对象模式(DOM)允许用脚本或其他编程语言处理数据。此外，数据计算不需要回到服务器就能进行。可以分离使用者查看数据的界面，使用简单、灵活、开放的格式，给 Web 创建功能强大的应用软件，而这些软件原来只能建立在高端数据库上。

- 数据的多样显示。

数据发到桌面后，能够用多种方式显示。通过以简单、开放、扩展的方式描述结果化的数据，XML 补充了 HTML，被广泛地用来描述使用者界面。HTML 描述数据的外观，而 XML 描述数据本身。由于数据显示与内容分开，XML 定义的数据允许指定不同的显示方式，使数据更合理地表现出来。

- 数据可进行粒状的更新。

通过 XML，数据可以进行粒状的更新。每当一部分数据变化后，不需要重发整个结构化的数据。变化的元素必须从服务器发送给客户，但变化的数据不需要刷新整个使用者的界面就能够显示出来。

- 在 Web 上发布数据。

由于 XML 是一个开放的、基于文本的格式，它可以和 HTML 一样使用 HTTP 进行传送，不需要对现存的网络进行改变。

- 升级性。

由于 XML 彻底把标识的概念同显示分开了，处理者能够在结构化的数据中嵌套程序化的描述，以表明如何显示数据。这是令人难以相信的、强大的机制，它使得客户计算机同使用者间的交互尽可能地减少了，同时也减少了服务器的数据交换量和浏览器的响

应时间。

- 压缩性。

XML 的压缩性能很好，因为用于描述数据结构的标签可以重复使用。XML 数据是否压缩不仅要根据应用来定，还取决于服务器与客户间数据的传送量。

1.4　XML 应用综述

如何应用 XML 呢？在介绍这个内容之前，有一点必须明确，设计 XML 的目的是用来存储数据、携带数据和交换数据的，而不是用来显示数据。

我们可以应用 XML 进行如下工作。

- 使用 XML 从 HTML 文件中分离数据。

XML 数据同样可以以“数据岛”的形式存储在 HTML 页面中。此时程序员仍然可以集中精力到 HTML 格式化的使用和数据的显示上。

- XML 用于交换数据。

传统的 EDI 的使用(Electronic Data Interchange，电子数据交换)标准缺乏灵活性和可扩展性。使用 XML，并用程序能够理解在交换数据中所表示的商务数据及概念，并且根据明确的商务规则来进行数据处理。

- 电子商务领域。

XML 的丰富置标信息完全可以描述不同类型的单据，例如信用证、保险单、索赔单以及各种发票等。结构化的 XML 文档发送至 Web 的数据可以被加密，并且很容易附加上数字签名。

- 数据库领域。

XML 文档可以定义数据结构，代替数据字典，用程序输出建库脚本。应用“元数据模型”技术，可以对数据源中不同格式的文件数据，按照预先定义的 XML 模板，以格式说明文档结构统一描述并提取数据或做进一步的处理，最后将其转换为 XML 格式输出。XML、数据库、网页或文档中的表格可以相互转换。

- Agent 智能体。

倘若送到 Agent 的是 XML 结构化的数据，Agent 就能很容易地理解这些数据的含义及它已有知识的关系。基于 XML 的数据交换对于解决 Agent 的交互性问题有重要的作用。从技术上讲，XML 语言只是一种简单的信息描述语言。但从应用角度上说，XML 的价值就远不止是一种信息的表达工具。事实上，借助 XML 语言，我们可以准确地表示几乎所有类型的数字化信息，可以清晰地解释信息的内涵和信息之间的关联，也可以在最短的时间内准确地定位我们需要的信息资源。

- 软件设计元素的交换。

XML 也可以用来描述软件设计中有关设计元素，如对象模型，甚至能描述最终设计出来的软件。另外，XML 及相关技术使得软件的分发及更新在 Web 上更容易实现。

- XML 可以用于创建新的语言。

WAP 语言用于标识运行于手持设备上的 Internet 程序，它和 WML(无线标记语言)是在 XML 的基础上产生的。

1.5　XML 开发工具

Altova XMLSpy 是一个用于 XML 工程开发的集成开发环境(Integrated Development Environment，IDE)。XMLSpy 可连同其他工具一起进行各种 XML 及文本文档的编辑和处理，进行 XML 文档(例如与数据库之间)的导入导出，在某些类型的 XML 文档与其他文档类型间作相互转换，关联工程中不同类型的 XML 文档，利用内置的 XSLT 1.0/2.0 处理器和 XQuery 1.0 处理器进行文档处理，甚至能够根据 XML 文档生成代码。Altova XMLSpy 可用于涉及 XML、XML Schema、XSLT、XQuery、SOAP、WSDL 和 Web 服务技术的企业级应用的设计、编辑和调试。它是提高 J2EE、.NET 和数据库开发人员开发效率的终极利器。

1.5.1　Altova XMLSpy 的主要功能

下面我们对 Altova XMLSpy 2011 中文企业版的主要功能作简要的概述。

1. 在多种视图格式下显示和编辑 XML 文档

使用 Altova XMLSpy，可以将一个 XML 文档以不同的视图进行显示。例如，对于下面一个简单的 XML 文档：

```
<?xml version="1.0" encoding="UTF-8"?>
<books>
  <book>
        <user>宾晟</user>
  </book>
</books>
```

单击文档显示区域下面的“文字”标签，可以将该 XML 文档在“文字”视图中以普通文本的形式显示，如图 1-1 所示。

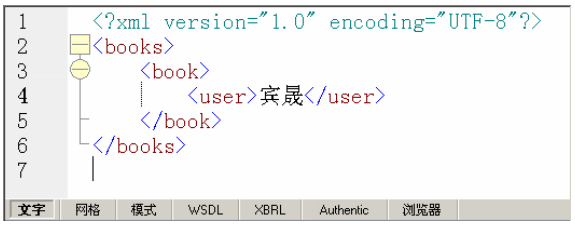

图 1-1　以文字视图显示 XML 文档

单击文档显示区域下面的“网格”标签，可以在“网格”视图中以具有层次结构的

表的形式显示 XML 文档，如图 1-2 所示。

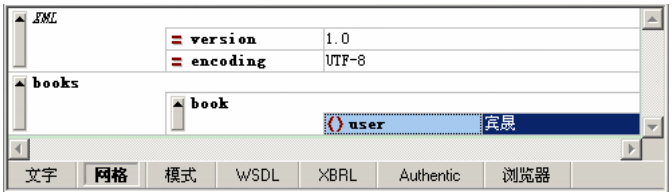

图 1-2　以网格视图显示 XML 文档

单击文档显示区域下面的“浏览器”标签，可以在“浏览器”视图中以浏览器的形式显示 XML 文档，如图 1-3 所示。

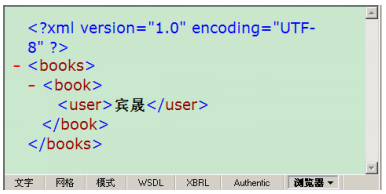

图 1-3　以浏览器视图显示 XML 文档

单击文档显示区域下面的“Authentic”标签，可以在“Authentic”视图中以图形化的所见即所得的形式显示 XML 文档，如图 1-4 所示。

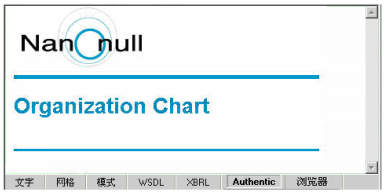

图 1-4　以 Authentic 视图显示 XML 文档

而对于 XML Schema 和 WSDL(Web Services Description Language，网络服务描述语言)文档，则可以使用“模式”或“WSDL”视图，它是以图形化用户界面的方式来创建复杂的 Schema 和 WSDL 文档，从而极大地简化了 XML Schema 和 WSDL 文档的创建过程。单击文档显示区域下面的“模式”标签，则将以图形化的形式显示 Schema 文档，如图 1-5 所示。

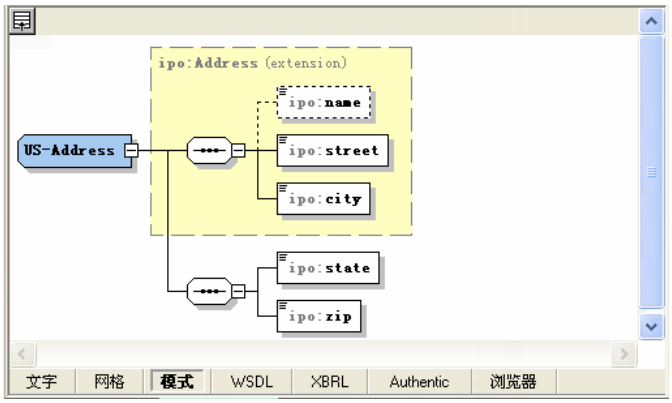

图 1-5　以模式视图显示 Schema 文档

2. 良构性检查和内置验证器

在切换视图或保存文件时，Altova XMLSpy 将会自动对 XML 文档进行良构性检查。如果是关联了 Schema(DTD 或 XML Schema)的 XML 文件，Altova XMLSpy 还会对它进行验证(Validation)。对于其他类型的文档(例如 DTD、XML Schema 等)，Altova XMLSpy 也会作语法和结构上的检查。检查的结果将在“验证”视图中显示，如图 1-6 所示。

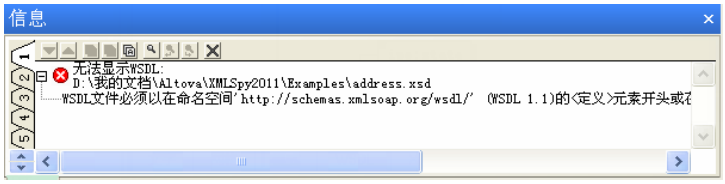

图 1-6　验证视图中显示检查的结果

3. 结构化编辑

在图 1-1 所示的“文字”视图中，行号、缩进、书签以及可展开/折叠的元素显示等功能将协助我们快速而有效地浏览文档。

4. 智能编辑

在“文字”视图中，如果正在编辑的 XML 文档已经关联了 Schema，那么自动完成功能将在编辑过程中提供极大的帮助。在敲击键盘的同时，光标所在的位置会出现一个列有元素(element)、属性(attribute)和允许出现的枚举型属性值(enumerated attribute values)的窗口。此外，在完成开始标签(start tag)的输入时，自动完成功能会自动插入相应的结束标签(end tag)，而在弹出窗口中选择的属性也会被自动插入并被引号括起来。如果一个元素下必须出现某些元素或/和属性，那么可以选择在该元素被插入时为它自动生成那些必需的成分。

5. XPath 求值

对于一个给定的 XML 文档，XPath 求值(Evaluate XPath)功能可以列出一个 XPath 表达式返回的序列(或节点集)。可以将文档节点(Document Node)或者选择一个元素作为上下文节点(Context Node)。XPath 求值的结果将显示在如图 1-7 所示的“XPath”视图中。

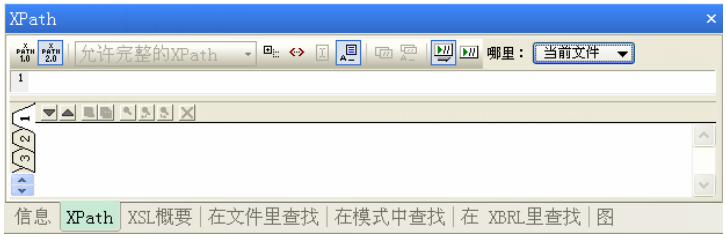

图 1-7　XPath 视图显示 XPath 求值的结果

6. XML 工程管理

在 Altova XMLSpy 中，我们可以将相关的文件组织为工程。与其他开发工具不同的是，在 Altova XMLSpy 中，工程可以是一个树状结构，即可以在一个工程下创建另一个工程。

工程中可以包含 Schema 文档、XML 文档、转换文件和输出文件等。

工程中的文件被列在“工程”窗口中，以便于访问。此外，我们还可以为整个项目或整个目录做统一的设定，例如，为整个目录的文件指定一个 Schema 文档或 XSLT 文件。

7. 数据库导入

可以将数据库中的数据导入为一个 XML 文件，并生成一个与数据库结构对应的 XML Schema 文件。Altova XMLSpy 目前支持导入数据库的有 MS Access、MS SQL Server、Oracle、MySQL、Sybase、IBM DB2。

8. 与各种常用开发工具集成

Altova XMLSpy 可以与 Visual Studio .NET 开发环境集成，也可以作为插件的形式与 Eclipse 开发环境集成。

9. 代码自动生成

如果要使用 Java、C++或 C#代码来处理 XML 文档中的数据，代码自动生成功能可以依据 XML 文档来生成包含有关 Schema 的类的定义代码。

1.5.2　Altova XMLSpy 的图形用户界面

Altova XMLSpy 的图形用户界面由下列四个主要部分组成。

- “项目”窗口：在该窗口中将文件组织为工程，并可对这些文件进行编辑。
- “信息”窗口：在该窗口中显示当前编辑项的信息。
- 主窗口：显示正在编辑中的文档的窗口。可用的文档视图数目与正在编辑的文档类型有关。可以根据需要在各种视图间切换。
- “输入助手”窗口：输入助手泛指那些在文档编辑过程中提供帮助的窗口，XMLSpy 中提供了多种不同的输入助手。可用的输入助手窗口将根据正在编辑的文档类型和主窗口的文档视图的不同而变化。

我们可以将这些窗口停靠在菜单条和工具条的下面，或者在菜单条和工具条的下方自由放置。它们的位置和大小都是可以调整的，除此之外，还可以通过 Altova XMLSpy 菜单栏中的“窗口”菜单来设置这些窗口的开关。

以上这些窗口是用户界面的主要部分，下面我们将对其进行详细介绍。

1. 项目窗口

我们可以使用项目菜单中的命令来进行工程的管理，选择“窗口”|“项目”菜单项打开和关闭“项目”窗口。“项目”窗口如图 1-8 所示。

在“项目”窗口中是以工程文件夹的形式来组织和管理项目的，工程文件夹是一种逻辑上的文件夹，表示一组文件的逻辑集合。它不是文件系统中的某个物理目录，但它可以将文件夹映射到文件系统中的某个物理目录，也可以将文件系统中不同物理路径上的多个文件加入到一个文件夹中。

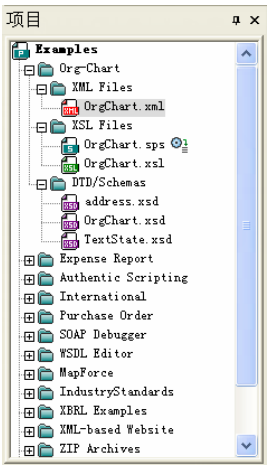

图 1-8　“项目”窗口

在“项目”窗口中可以实现以下功能。

(1) 为工程文件夹指定 XSL 转换

可以为各个文件夹指定不同的 XSL 转换参数，甚至还能使用不同的 XSL 样式表来处理同一个 XML 文档，以得到不同的输出结构。

(2) 为工程文件夹指定 DTD/Schema

通过右击“项目”窗口中的 DTD/Schemas 文件夹，并在弹出的快捷菜单中选择“添加文件”菜单项，可以为项目中的 XML 文档指定不同的 DTD 或 Schema。这样，不必修改 XML 文档即可使用不同的 DTD 或 XML Schema 对其进行验证。

2. 信息窗口

Altova XMLSpy 中提供的“信息”窗口用于显示主窗口中光标所选中的 XML 元素或属性的相关细节。我们可以选择“窗口”|“信息窗口”菜单项打开和关闭“信息”窗口，如图 1-9 所示。

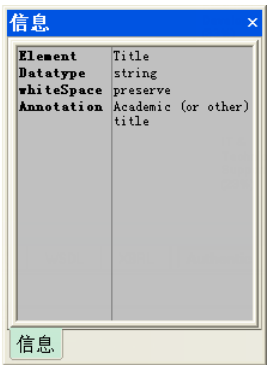

图 1-9　“信息”窗口

3. 主窗口

Altova XMLSpy 中的主窗口是用来显示和编辑文档的区域，如图 1-10 所示。

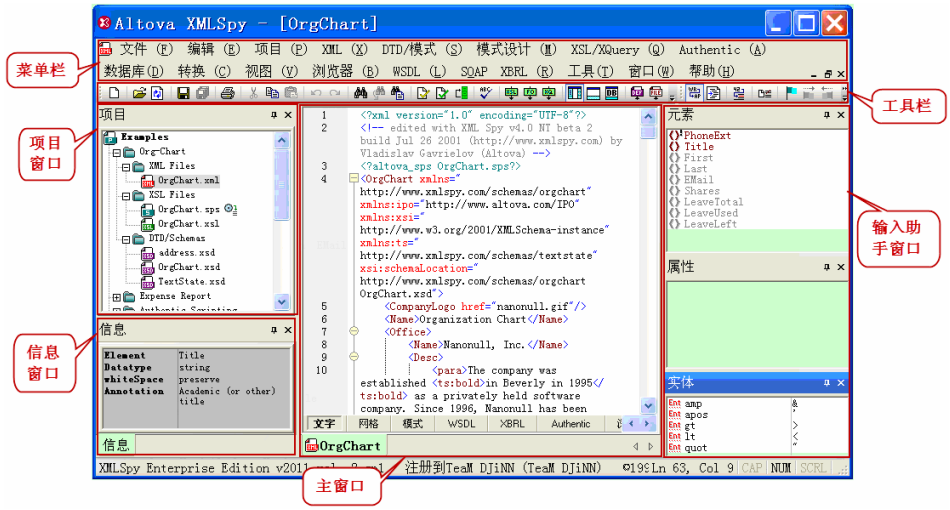

图 1-10　主窗口

在主窗口中可以同时打开/编辑任意多个 XML 文档，已打开的文件将显示在文档窗口中，并且这些文档窗口在主窗口的底部各有一个与之对应的标签。单击标题栏上的“最大化”按钮或“最小化”按钮，可以将文档窗口放到最大或最小。如果要编辑一个文件，只需单击一下该文件对应文档窗口的标题栏或窗口内部，即可进入编辑状态。

右击位于主窗口底部的“文字”标签，将弹出一个如图 1-11 所示的包含各个文件命令的快捷菜单，使用其中的命令可以对文件进行快速操作。

4. 输入助手窗口

Altova XMLSpy 中提供的输入助手窗口将根据当前编辑的文档类型的不同而显示不同的窗口。例如，当编辑 XML Schema 文档时，将打开如图 1-12 所示的“组件”窗口。在该窗口中将显示 XML Schema 文档中的元素、基本数据类型以及复杂数据类型。

图 1-11　文件命令的快捷菜单

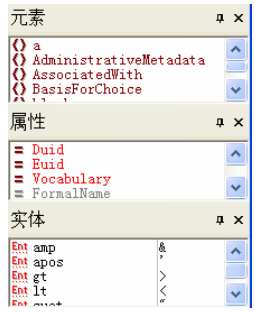

图 1-12　XML Schema 文档的输入助手窗口

1.5.3　Altova XMLSpy 2011 的安装

要使用 Altova XMLSpy 2011 企业中文版，首先必须将它安装在本地计算机上。安装 Altova XMLSpy 2011 的步骤如下。

(1) 双击 XMLSpy Ent2011.exe 的文件图标，系统将打开如图 1-13 所示的 Altova XMLSpy

2011 安装向导。

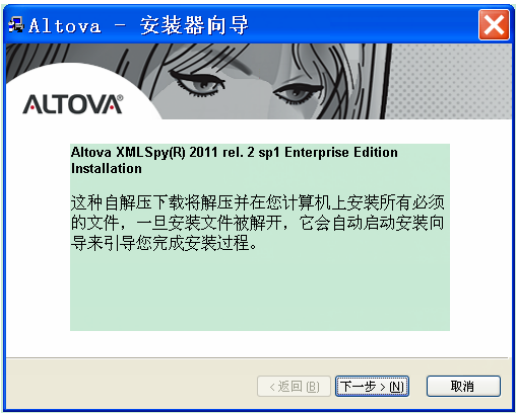

图 1-13　Altova XMLSpy 2011 安装向导

(2) 单击“下一步”按钮，将显示如图 1-14 所示的安装对话框。

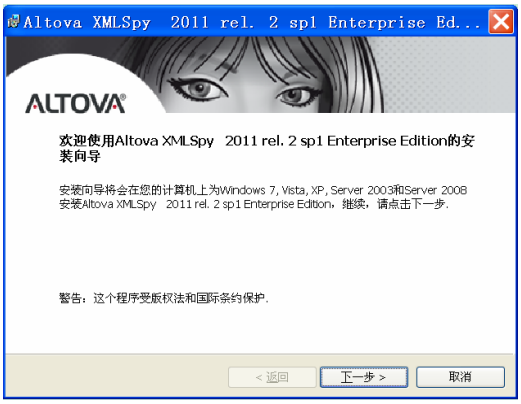

图 1-14　安装对话框

(3) 单击“下一步”按钮，将显示如图 1-15 所示的软件许可协议对话框，在该对话框中显示出了许可协议的全文，用户必须同意该协议的所有条款才可以使用该软件。选中“接受许可协议和隐私政策的条款”单选按钮。

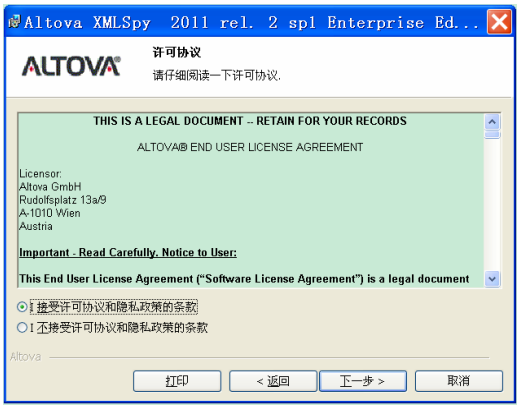

图 1-15　软件许可协议对话框

(4) 单击“下一步”按钮继续，将显示如图 1-16 所示的打开和编辑的文档类型关联对话框，在该对话框中可以选择与 Altova XMLSpy 关联的文件类型。

(5) 单击“下一步”按钮，将显示如图 1-17 所示的选择安装类型对话框，在该对话框中可以选择是完全安装还是自定义安装。对于一般用户而言，最常用的情况是选择“完成”模式，即完全安装；当然，如果要节省硬盘空间，可以选择 Custom 模式，即“自定义”模式。

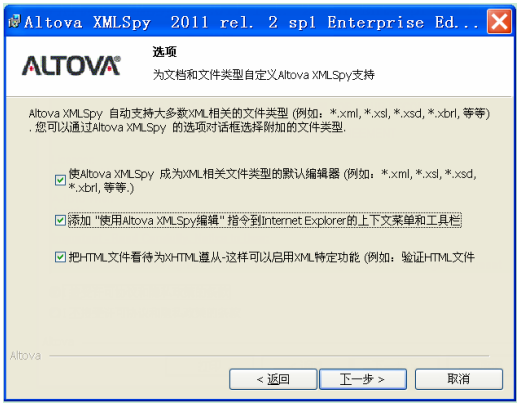

图 1-16　打开和编辑的文档类型关联对话框

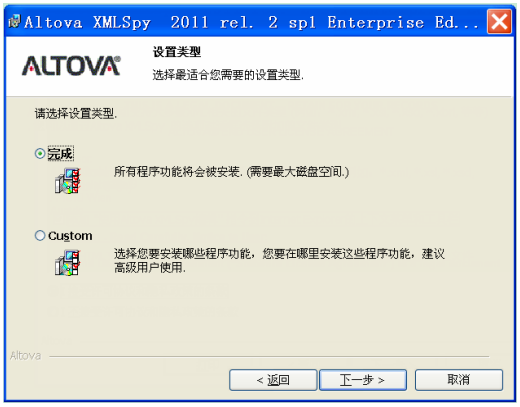

图 1-17　选择安装类型对话框

(6) 单击“下一步”按钮，将开始安装，并显示如图 1-18 所示的安装进度条。

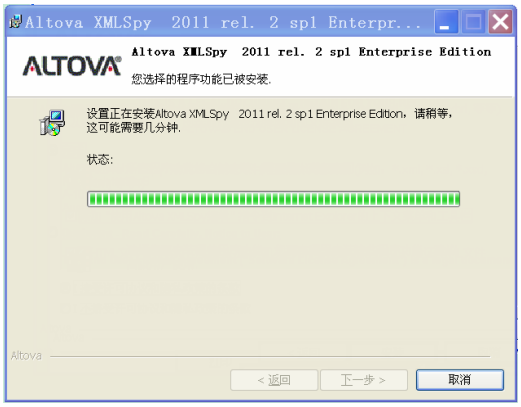

图 1-18　显示安装进度

(7) 安装完成后将显示如图 1-19 所示的“设置完成”对话框。至此，Altova XMLSpy 2011 安装完毕，单击“完成”按钮退出。

图 1-19　安装完成对话框

安装完毕后，Altova XMLSpy 2011 将被安装到指定的目录中，若操作系统装在 C 盘，则目录一般是 C:\Program Files\Altova\XMLSpy 2011。

1.5.4　Altova XMLSpy 的使用

下面通过实例介绍使用 Altova XMLSpy 创建和查看简单 XML 文档的方法。

实例 1-1　使用 Altova XMLSpy 创建和查看简单的 XML 文档

(1) 打开 Altova XMLSpy 2011，单击菜单栏中的“文件”|“新建”菜单项新建一个文档，弹出如图 1-20 所示的“创建新文件”对话框。

图 1-20　“创建新文件”对话框

(2) 在该对话框中，我们可以选择要创建的文件类型，本实例要创建的是 XML 文档，所以使用默认的设置，单击“确定”按钮，系统弹出如图 1-21 所示的“新文件”对话框。

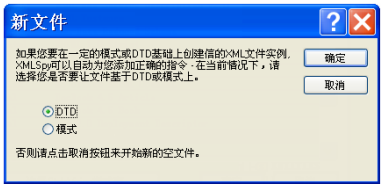

图 1-21　“新文件”对话框

(3) 在该对话框中，我们可以将新创建的 XML 文档与相应的 DTD 或 XSD 模式文件相关联，本实例中不使用 DTD 和 XSD 模式，所以单击“取消”按钮进入 XML 编辑界面，如图 1-22 所示。

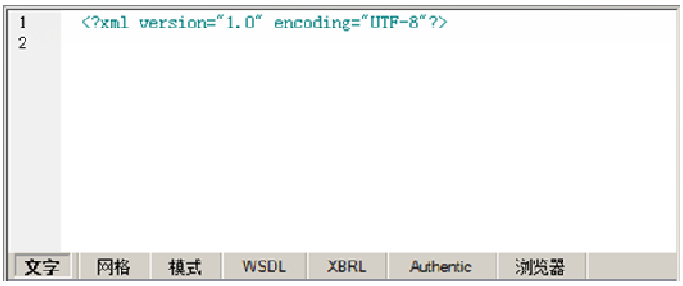

图 1-22　XML 编辑窗口

注意：此时 XML 编辑窗口最下端选择的是“文字”视图。

(4) 在 XML 编辑界面中，从第二行开始，输入以下代码：

```
<!-- JACK 给 TOM 的便条，存储为 XML -->
<note>
<to>TOM</to>
<from>JACK</from>
<heading>记住</heading>
<body>这个周末我们的约定</body>
</note>
```

代码编写完成后，在 XML 编辑界面中的显示如图 1-23 所示。

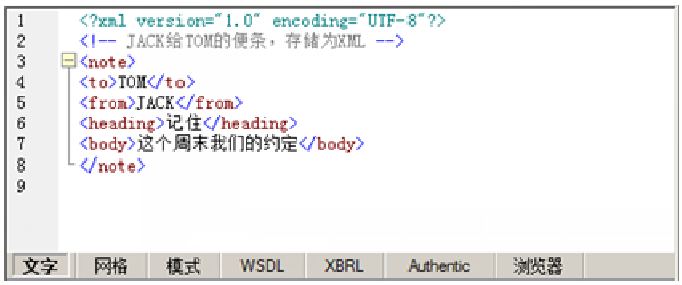

图 1-23　输入代码后的 XML 编辑界面

(5) 保存文件并查看结果。

单击工具栏中的“保存”按钮，将此文档保存为名称 Note 的 XML 文档后，在编辑窗口最下端选择“浏览器”视图，即可查看显示结果，如图 1-24 所示。

我们可以单击 note 元素左边的加号(+)或减号(－)来展开或收缩元素的结构。如果想再次查看原始的 XML 源文件，直接在编辑窗口最下端选择“文字”视图即可。

图 1-24　在“浏览器”视图中查看显示结果

若 XML 文档中存在错误，则将在 Altova XMLSpy 的验证视图中报告这个错误，例如，将本实例中代码的最后一行去掉，验证视图将显示如图 1-25 所示的错误信息。

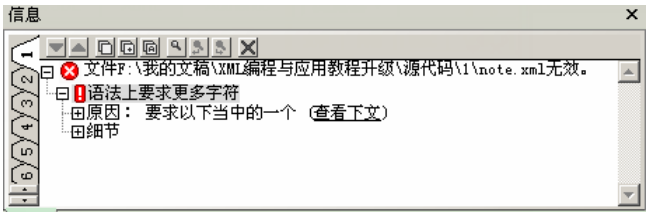

图 1-25　在验证视图中显示 XML 文档出错信息

1.6　本 章 小 结

本章详细阐述了 XML 技术的起源、发展与应用，并在此基础上详细介绍了目前应用最广泛的 XML 开发工具 Altova XMLSpy 的安装和使用。本章作为对 XML 技术的概述，可以使读者对 XML 技术有一个大概的了解，为后面各章内容的学习，打下良好的基础。

1.7　习　　题

一、填空题

1. XML 早期主要是用来进行_____。
2. 可扩展标记语言 XML 是_____的子集。
3. XML 最初的设计目的是为了_____。
4. 可使用 XML 从_____文件中分离数据。
5. 程序开发语言 WAP 和 WML 是在_____的基础上产生的。

二、选择题

1. XML 由(　　)工作组(原先的 SGML 编辑审查委员会)开发。

A.XML　　B. SGML　　C. W3C　　D.HTML

2. 下面的选项中，(　　)是使用 EDI 的优点(多选)。
   A. 降低了纸张的消费
   B. 减少了重复劳动，提高了工作效率
   C. 使得贸易双方能够以更迅速、更有效的方式进行贸易
   D. 改善贸易双方的关系
3. 下面的选项中，(　　)是 XML 的优点(多选)。
   A. 开发灵活的 Web 应用软件
   B. 数据可进行粒状的更新
   C. 在 Web 上发布数据
   D. 不同来源数据的集成
4. 下面的选项中，(　　)是我们可以应用 XML 进行的工作(多选)。
   A. 交换数据
   B. 软件设计元素的交换
   C. 创建新的语言
   D. 从 HTML 文件中分离数据
5. 下面的选项中，(　　)是 Altova XMLSpy 2011 的主要功能(多选)。
   A. 在多种视图格式下显示和编辑 XML 文档
   B. 良构性检查和内置验证器
   C. 结构化编辑
   D. 数据库导入

三、简答题

1. 什么是 XML?
2. 相对于其他应用于 EDI 的结构化信息技术而言，XML 的优势有哪些?
3. XML 的优点有哪些?

第2章　XML 语 法

XML 的语法规则既简单又严格，非常容易学习和使用。XML 文档使用了自描述的和简单的语法，熟悉 HTML 的读者将会发现它的语法和 HTML 非常相似。本章将重点介绍 XML 文档的定义规则和语法，只有掌握了 XML 文档的规则才能定义出格式良好的 XML 文档。

本章重点：

- XML 文档结构与规则
- XML 声明
- XML 文档内容与命名空间

2.1　XML 文档概述

下面我们首先来看一个 XML 文档，具体代码定义如下：

```
1  <?xml version="1.0"    encoding="GB2312"?>
2  <myfile>
3  <title> Xml 语法</title>
4  <author>张三</author>
5  <email>icewine@tom.com</email>
6  <date>20090330</date>
7  </myfile>
```

第 1 行是一个 XML 声明，表示文档遵循的是 XML 1.0 版的规范。

第 2 行定义了文档中的第一个元素 myfile，也称为根元素。这个就类似 HTML 文档里的<HTML>开头标记。注意，这个元素的名称是用户自定义的。紧接着定义了 4 个子元素：title、author、email 和 date，分别表示文章的标题、作者、邮箱和日期。第 7 行表示根元素的结束。

这就是一个基本的 XML 文档，从中读者可以看到，XML 文档是相当简单的。同 HTML 文档一样，XML 文档也是由一系列的标记组成的，不过，XML 文档中的标记是我们自定义的标记，具有明确的含义，我们可以对标记的含义作出说明。

可以用 Internet Explorer 5.0 或以上版本的浏览器来查看 XML 文档。在浏览器中打开一个 XML 文档，将显示出文档中的根和其子元素。上述 XML 文档在浏览器中的显示结果如图 2-1 所示。

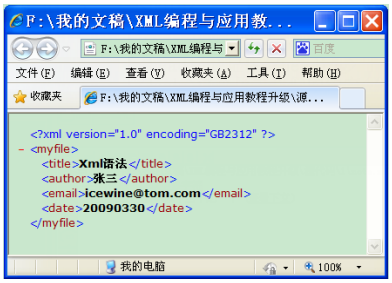

图 2-1　XML 文档在浏览器中的显示结果

可以单击元素左边的加号(+)或减号(－)来展开或收缩元素的结构。例如，单击 myfile 元素左边的减号(－)，可以收缩元素的结构，显示结果如图 2-2 所示。

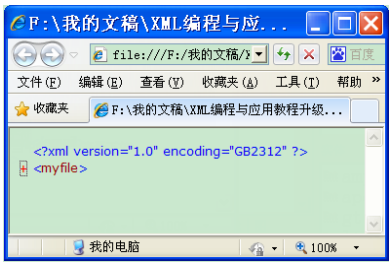

图 2-2　单击减号收缩后的文档显示结果

如果想要查看原始的 XML 源文件，就必须从浏览器菜单中选择查看源选项。显示的 XML 文档的源代码如图 2-3 所示。

图 2-3　查看 XML 源文件

若 XML 文档中有语法错误，Internet Explorer 浏览器就会报告这个错误。例如，我们将 XML 文档第一行代码中的 encoding="GB2312"去掉，再次用 Internet Explorer 浏览器打开这个文档时，将显示如图 2-4 所示的错误信息。

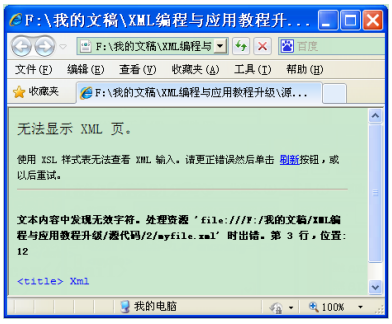

图 2-4　XML 文档出错时显示的错误信息

2.2　XML 文档结构

每个 XML 文档都分为两个部分：序言(prolog)和文档元素(或文档节点)。序言出现在 XML 文档的顶部，其中包含关于该文档的一些信息，有点像 XHTML 文档中的<head>部分。在上面的 XML 文档中，序言包含了一个 XML 声明。它也可以包含其他的元素，如注释、处理指令或是 DTD(文档类型定义)。

Well-formed (良好格式的)XML 文档必须有一个文档元素，用来包含可能有的其他内容。XML 文档中的所有内容都应该出现在文档元素的内部。在遵守 XML 命名规则的前提下，用户可以为元素和属性选择任意名字。如图 2-5 所示，显示了 XML 文档的结构。

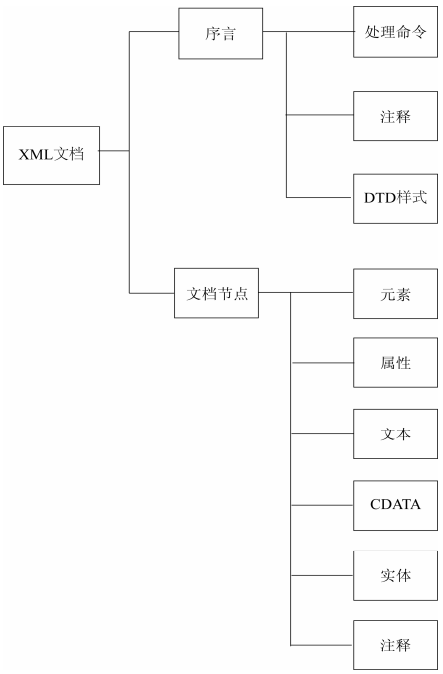

图 2-5　XML 文档的结构

我们通过下面这段代码来进行详细说明：

```
1  <?xml version="1.0" encoding="gb2312"?>
2  <!DOCTYPE filelist SYSTEM "filelist.dtd">
3  <filelist>
4  <myfile>
5  <title> Xml 语法</title>
6  <author>张三</author>
7  </myfile>
8  </filelist>
```

其中第 1 行<?xml version="1.0" encoding="gb2312"?>就是一个 XML 文档的声明，注意，有效的 XML 文件的第一行必须是 XML 文档的声明；第 2 行是序言，说明这个文档是用 filelist.dtd 这个文件来定义文档类型的；第 3 行是文档元素，以下就是包含在文档元素中的子元素、属性、注释和元素内容。

2.3　XML 文档规则

在定义 XML 文档时必须符合一定的规则，按照规则定义的 XML 文档被称为格式良好的 XML 文档。如果 XML 文档在定义时按照与其关联的 DTD 或 XML Schema 中的规则来

匹配，则这类 XML 文档被称为有效的。

2.3.1　格式良好的 XML 文档规则

吸取 HTML 松散格式带来的经验教训，XML 一开始就坚持实行“良好的格式”。我们先看 HTML 的一些语句，这些语句在 HTML 中随处可见。

- < b>< i>sample< /b>< /i>
- < td>sample< /TD>
- < font color=red>sample< /font>

在 XML 文档中，上述几种语句的语法都是错误的，上述语句在 XML 中的正确写法如下：

- < b>< i>sample< /i>< /b>
- < td>sample< /td>
- < font color="red">sample< /font>

这是因为虽然 XML 的文档和 HTML 文档类似，也是使用元素来标识内容，但是创建 XML 文档时必须遵守下列重要规则。

1. 必须有 XML 声明语句

这一点我们在上一节学习时已经提到过。XML 声明是 XML 文档的第一句，其格式如下：

```
<?xml version="1.0" standalone="yes/no" encoding="UTF-8"?>
```

XML 声明的作用是告诉浏览器或者其他处理程序：这个文档是 XML 文档。

2. 注意大小写

在 XML 文档中，大小写是有区别的，<P>和<p>是不同的标识。注意在写元素时，前后标识的大小写要保持一样。例如，把<Author>TOM</Author>，写成<Author>TOM</author>是错误的。

我们最好养成一种习惯，或者全部大写，或者全部小写，或者大写第一个字母。这样可以减少因为大小写不匹配而产生的文档错误。

3. 所有的 XML 文档必须有且只有一个根元素

良好格式的 XML 文档必须有一个根元素，就是紧接着声明后面建立的第一个元素，其他元素都是这个根元素的子元素，都属于根元素一组。

根元素是一个完全包括文档中其他所有元素的元素。根元素的起始标记要放在所有其他元素的起始标记之前，根元素的结束标记要放在所有其他元素的结束标记之后。

4. 属性值必须使用引号

在 HTML 代码里面，属性值可以加引号，也可以不加。例如，word和word都可以被浏览器正确解释。但是在 XML 中规定，所有属性值必须加引号(可以是单引号，也可以是双引号)，否则将被视为错误。

5. 所有的标识必须有相应的结束标识

在 HTML 中，标识可以不成对出现，而在 XML 中，所有标识必须成对出现，有一个开始标识，就必须有一个结束标识，否则将被视为错误。

6. 所有的空标识必须被关闭

空标识就是标识对之间没有内容的标识，比如<img>等标识。在 XML 中，规定所有的标识必须有结束标识，当然，一个标识以/>符号结尾也可以表示空标识。例如，下面两行是等效的。

```
1　＜title/＞＜br＞
2　＜title＞＜/title＞
```

7. 标识必须正确嵌套

标识之间不得交叉。在以前的 HTML 文件中，可以这样写：

```
〈B〉〈H〉XXXXXXX〈/B〉〈/H〉
```

<B>和<H>标记之间有相互重叠的区域，而在 XML 中严格禁止这样标记交错的写法，标识必须以规则性的次序出现。

8. 处理空白字符

XML 处理空白字符和 HTML 不一样。HTML 标准规定，不管有多少个空白，都当作一个空白来处理，例如，在 HTML 中，" Hello my name is TOM "将会被显示成“Hello my name is TOM”，因为 HTML 解析器会自动把句子中的空白部分去掉。

而在 XML 中规定，对于所有标记以外的空白，解析器都要忠实地交给应用程序来处理，即解析器会保留内容中所有的空白字符并不加修改地传递给应用程序，但元素标记和属性值中的空白会被删除。对于如下代码：

```
1　<?xml version="1.0" encoding="gb2312"?>
2　<作者>张三</作者>
```

显示结果如图 2-6 所示。

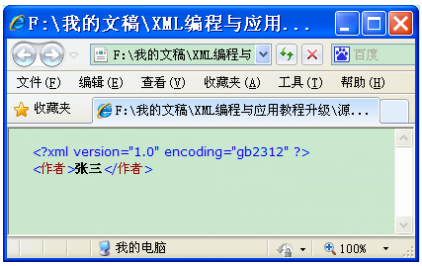

图 2-6　显示结果

而对于下列代码：

```
1  <?xml version="1.0" encoding="gb2312"?>
2  <作者>
3      张三
4   </作者>
```

在上述代码的第 3 行中，在字符串“张三”的前后都包含有空白字符。使用浏览器打开该文件将显示如图 2-7 所示的运行结果。通过运行结果我们可以看到，源文件中的空白字符被原样地显示出来。

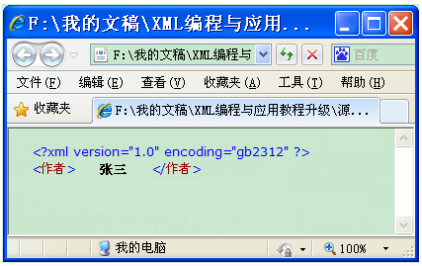

图 2-7　空白字符原样显示

9. 处理特殊字符

在 XML 文件中，如果要用到表 2-1 中的特殊字符，必须用相应符号来代替。一些字符有特殊含义，例如，“<”已当作标签使用，不能出现在 XML 文件中。XML 提供了一些预先定义的实体，称为内部实体。

表 2-1　特殊字符的替代符号

特 殊 字 符	替 代 符 号
用来显示小于(<)符号	<
用来显示大于(>)符号	>
用来显示 and(&)符号	&
用来显示双引号(")符号	"
用来显示单引号(')符号	'

另外，XML 标记必须遵循下面的命名规则。

- 名字中可以包含字母、数字。
- 名字不能以数字或"_" (下划线)开头。
- 名字不能以字母 XML (或者 xml，Xml，xMl...)开头。
- 名字中不能包含空格。

在 XML 文档中，任何的差错都会得到同一个结果：网页不能被显示。各浏览器开发商已经达成协议，对 XML 实行严格而挑剔的解析，任何细小的错误都会被报告。例如，下面的例子就不符合 XML 语法。

```
＜title＞物种起源＜sub＞动物类
＜/title＞ 约翰・邦纳＜/sub＞
```

改正后符合语法的形式如下：

```
＜title＞物种起源
＜sub＞动物类＜/sub＞
＜author＞约翰・邦纳＜/author＞
＜/title＞
```

2.3.2　格式良好的 XML 文档

一个遵守 XML 语法规则并遵守 XML 规范的文档称之为 Well-formed XML(格式良好的 XML)。XML 必须是 Well-formed 的，才能够被解析器正确地解析出来，并显示在浏览器中。一个格式良好的 XML 文档包含一个或多个元素(用起始和结束标记将其分隔开)，并且它们相互之间正确地嵌套。其中有一个元素，即文档元素(或称根元素)，包含了文档中的其他所有元素。所有的元素构成一个简单的层次树，所以元素和元素之间唯一的直接关系就是父子关系。文档内容可能包括标记和/或字符数据。

总之，XML 文档中的数据对象如果满足下列条件，那就是格式良好的文档。

- 语法合乎 XML 规范。
- 元素构成一个层次树，只有一个根节点。
- 除非提供了 DTD，否则没有对外部实体的引用。

下面的例子就是一个格式良好的 XML 文档。

```
1  <?xml version="1.0" encoding=" GB2312"?>
2  <Shop>
3  <BOOK BOOKID="B001">
4  <BOOKNAME>英语</BOOKNAME>
5  <WRITERFIRSTNAME>汤姆</WRITERFIRSTNAME>
6  <WRITERLASTNAME>张三</WRITERLASTNAME>
7  <PRICE>$20.00</PRICE>
8  </BOOK>
9  </Shop>
```

该文档的语法符合 XML 文档规范，第 1 行是 XML 文档的声明，第 2 行是文档的根元素，第 2～8 行是对根元素中子元素的定义，第 9 行是根元素的结束标签。

2.3.3　有效的 XML 文档

我们看到，在 XML 文件中，用的大多都是自定义的标记。但是如果两个同行业的公司 A 和 B 要用 XML 文件相互交换数据的话，他们之间必须有一个约定，即编写 XML 文件可以用哪些标记、母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。这样他们在用 XML 交换数据时才能够畅通无阻。这种约定可以是 DTD(Document Type Definition，文档格式定义)，也可以是 XML Schema(XML 模式)。

一个格式良好的 XML 文档应该遵守 XML 语法规则，而一个有效的 XML 文档应该既是一个格式良好的 XML 文档，同时还必须符合 DTD 或 XML 模式所定义的规则。因此，

格式良好的 XML 文档不一定是有效的 XML 文档，但有效的 XML 文档一定是格式良好的 XML 文档。

DTD 定义了 XML 文档中可用的合法元素。它通过定义一系列合法的元素来决定 XML 文档的内部结构。XML Schema 是基于 XML 的 DTD 的替代品，而且 DTD 和 Schema 可以相互替代。

例如，DTD 文件 product.dtd 的内容如下所示。

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT PRODUCTDATA (PRODUCT)+>
<!ELEMENT PRODUCT (PRODUCTNAME,DESCRIPTION,PRICE,QUANTITY)>
<!ELEMENT PRODUCTNAME (#PCDATA)>
<!ELEMENT DESCRIPTION (#PCDATA)>
<!ELEMENT PRICE (#PCDATA)>
<!ELEMENT QUANTITY (#PCDATA)>
<!ATTLIST PRODUCT PRODUCTID ID #REQUIRED CATEGORY (BOOKS|TOYS) "TOYS">
```

在该 DTD 文件中定义了 XML 文档中能够使用的元素的名称和类型，关于 DTD 的详细内容将在第 3 章中详细说明。

下面的例子就是一个有效的 XML 文档。

```
1  <?xml version="1.0"?>
2  <!DOCTYPE PRODUCTDATA SYSTEM "product.dtd">
3  <PRODUCTDATA>
4  <PRODUCT PRODUCTID="P001" CATEGORY="TOYS">
5  <PRODUCTNAME>乱世佳人</PRODUCTNAME>
6  <DESCRIPTION>以美国内战为背景进行叙事</DESCRIPTION>
7  <PRICE>25.00</PRICE>
8  <QUANTITY>35</QUANTITY>
9  </PRODUCT>
10 </PRODUCTDATA>
```

该 XML 文档中的定义就是按照 product.dtd 中的规范进行的，因此被称为是有效的。其中，第 2 行代码用来指定用于该 XML 文档的 DTD 文件。

2.4　XML 声 明

XML 文档以 XML 声明作为开始，向解析器提供了关于文档的基本信息。同所有的处理指令一样，XML 声明也是由“<?”开始，以“?>”结束。“<?”表示该行是一个命令；在“<?”后面紧跟“xml”，表示该文件是一份 XML 文件(注意，必须小写)，这是处理指令的名称，用于声明 XML 的版本和采用的字符集；在“<”和“?”之间、“?”和“>”之间以及第一个“?”和“xml”之间不能有空格，这就是 XML 语法严格性的一个体现。

在第二个“?”之前可以有一个或多个空格，也可以没有空格。

声明语句中的 version 属性表示 XML 的版本，因为解析器对不同版本的解析肯定会有区别。注意，version 必须小写，版本号用引号引起。在 XML 声明中要求必须指定 version 的属性值。同时，声明中还可以有两个可选属性，分别是 encoding 和 standalone。

encoding 属性表示该文档所使用的字符集。该声明中引用的 ISO-8859-1 字符集包括大多数西欧语言用到的所有字符。W3C 的 XML1.0 规范里规定，所有的 XML 解析器必须接受 UTF-8 和 UTF-16 编码的 Unicode 字符，所以，符合 XML 规范的软件工具一定都支持这两种 Unicode 编码。如果 XML 声明中没有设置 encoding 属性来明确指定文档所用的字符编码方式，则一律以 Unicode 编码看待。XML 解析器通过寻找 XML 文档开始处的字节顺序标记，能够自动检测出文档中的 Unicode 编码是 UTF-8、还是 UTF-16。

我们也可以通过在 XML 文档声明中指定 encoding 属性来说明该 XML 文档所使用的字符编码方式。在 XML 规范中，列出了一大堆编码类型，但一般我们用不到这么多编码，只要知道下面几个常见的编码就可以了。

- 简体中文码：GB2312。
- 繁体中文码：BIG5。
- 西欧字符：UTF-8。

例如，使用下面的语句来指明文档中的字符编码方式为 GB2312 编码。

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

最后，standalone 属性(可以是 yes 或 no)定义了是否可以在不读取其他任何文件的情况下处理该文档。例如，如果 XML 文档没有引用其他任何文件，则可以指定 standalone="yes"；如果 XML 文档引用了其他描述该文档可以包含什么的文件，则可以指定 standalone="no"。因为 standalone="no"是默认的，所以很少会在 XML 声明中看到 standalone。

注意：

如果同时设置了 encoding 和 standalone 属性，则 standalone 属性要位于 encoding 属性之后。

对于含有中文字符的 XML，其中的字符可以采用 Unicode 来编码或采用 GB2312(简体中文码)编码来表示。如果文档中的字符使用的是 GB2312 编码，则必需设置 encoding 属性值为 GB2312(也可设为 UTF-8)，下面通过一个实例来说明这个问题。

实例 2-1　含有中文字符的 XML

(1) 用 Windows 自带的记事本程序创建一个名为 BOOK. XML 的文件，文件内容如下所示。

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

```
<书名>Java 培训教程</书名>
<作者>张力</作者>
<售价>39.00 元</售价>
</书>
<书>
<书名>JavaScript 网页开发</书名>
<作者>张力</作者>
<售价>28.00 元</售价>
</书>
</书架>
```

(2) 用 Internet Explorer 5.0 以上的浏览器打开 BOOK. XML 文件，看到的结果如图 2-8 所示。

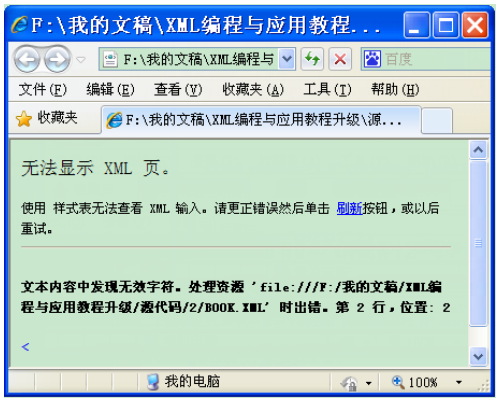

图 2-8　显示结果

错误提示的信息是“文本内容中发现无效字符”，这就是因为在 BOOK.XML 的 XML 文档声明语句中没有明确指定文档中的字符编码方式，导致浏览器用默认的 Unicode 编码来解析该文档，而该文档中的字符实际上使用的是 GB2312 编码，而非 Unicode 编码。

(3) 选择浏览器的“查看”|“源文件”菜单项，将打开的 BOOK. XML 文件内容中的第 1 行修改成如下形式：

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

保存修改后，刷新显示 BOOK. XML 文件的浏览器窗口，看到的结果如图 2-9 所示。单击某个标签前面的减号(-)，嵌套在该标签中的所有内容将被折叠起来，标签前面的减号(-)也将变成加号(+)。单击某个标签前面的加号(+)，嵌套在该标签中的所有内容将被展开，标签前面的加号(+)也将变成减号(-)。

(4) 在上面打开 BOOK. XML 文件的记事本程序中，选择“文件”|“另存为”菜单项，在打开的“另存为”对话框中，选择“保存类型”为“所有文件”，“编码”为“UTF-8”，如图 2-10 所示。

图 2-9　正确的显示结果

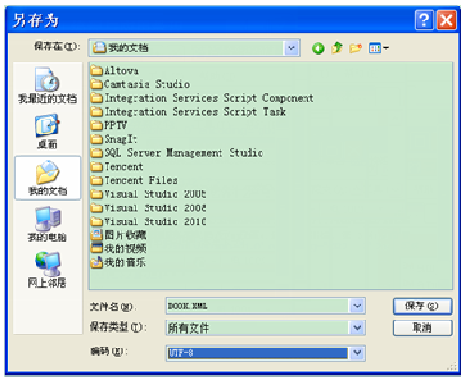

图 2-10　“另存为”对话框

以 UTF-8 编码保存 BOOK.XML 文件后，尽管在记事本程序窗口中显示的效果没有任何变化，但是 BOOK.XML 文件内部存储的数据却已经改变，通过比较 BOOK.XML 保存前后的文件大小就可以看出来。刷新显示 BOOK.XML 文件的浏览器窗口，看到的结果如图 2-11 所示。

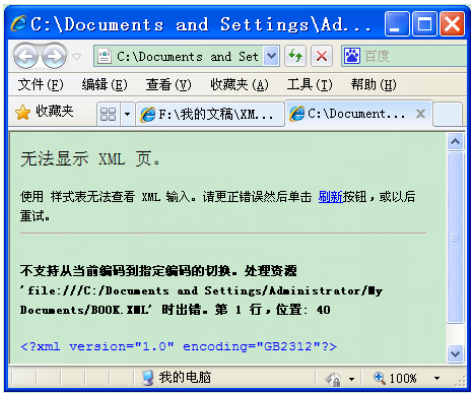

图 2-11　出错的显示结果

因为现在的 BOOK.XML 文件的字符编码已经转换为 UTF-8，而文档的起始声明中指定的 encoding 属性为 GB2312，所以，浏览器在解析 BOOK.XML 文件时会发生错误。

(5) 将 BOOK. XML 文档起始声明的 encoding 属性修改为 UTF-8，保存后刷新显示 BOOK.XML 文件的浏览器窗口，就又可以看到类似如图 2-9 所示的正确显示结果了。

2.5　XML 文档内容

XML文档中除了上一小节中介绍的XML声明之外，还包括XML文档内容，它表示XML文档中包含的信息，是 XML文档的主要组成部分。

2.5.1　XML 元素

元素是 XML 文档的基本组成部分。它们可以包含其他的元素、字符数据、字符引用、实体引用、PI、注释以及 CDATA 部分，这些合在一起被称做元素内容。所有的 XML数据(除了注释、PI 和空白)都必须包容在元素中。下面我们将对 XML元素、注释、字符引用、实体引用等做详细介绍。

1. XML 元素的命名规范

在 XML中，基本上没有什么保留字，所以我们可以随心所欲地用任何词语来为元素命名，但是 XML元素的命名必须遵守下列规范。

- 元素的名字可以包含子母、数字和其他字符。
- 元素的名字不能以数字或者标点符号开头。
- 元素的名字不能以 XML(或者 xml、Xml、xMl. . .)开头。
- 元素的名字不能包含空格。
- 尽量避免使用“-”、“.”，因为这些符号有可能引起混乱。
- 元素的命名应该遵循简单易读的原则，例如，<book_title>是一个不错的名字，而<the_title_of_the_book>则显得啰嗦了。
- XML 文档往往都对应着数据表，应该尽量让数据库中字段的命名和相应的XML 文档中元素的命名保持一致，这样可以方便数据变换。
- 非英文/字符/字符串也可以作为 XML 元素的名字，例如<歌曲>、<文章>等，这都是完全合法的名字。但是有一些软件不能很好地支持这种命名，所以我们最好使用英文字母来命名。
- 在 XML元素的命名中不要使用“:”，因为 XML 的命名空间需要用到这个十分特殊的字符。

2. 起始标记

一个表示元素开始的分隔符被称做起始标记。起始标记是一个包含在尖括号里的元素类型名。我们也可以把起始标记看作是“打开”一个元素。下面是一些合法的起始标记：<book>、<BOOK>、<BOOk>、<ABC>。

再次强调，由于 XML 对大小写敏感，所以前三个例子是不等同的标记；而且元素类

型名可以使用任何合法字母，而不一定是 ASCII 码字符。

3. 结束标记

代表元素结束的分隔符被称作结束标记。结束标记由一个反斜杠和元素类型名组成，并被围在一对尖括号中。每一个结束标记都必须与其对应的起始标记相匹配，我们可以把结束标记理解为关闭了一个由起始标记打开的元素。下面是一些合法的结束标记，它们与前面列举的起始标记相对应：</book>、</BOOK>、</BOOk>、</ABC>。

所以，带有完整的起始、结束标记的元素应该是如下形式。

```
<某个标记> 包含的内容</某个标记>
```

4. XML 元素的类型

在 XML 文档中，元素有很多作用，它们可以标记内容，可以为它们标记的内容提供一些描述，也可以为数据的顺序和相对重要性提供信息，还可以展示数据之间的关系。

XML 文档中一共有四类元素：空元素，仅含文本的元素，仅含子元素的元素，含子元素、文本或混合元素的元素。下面一一进行介绍。

(1) 空元素

如果元素中不包含任何文本，那么它就是个空元素。空元素有两种书写方式，下列两行代码的作用是完全相同的。

```
1  <book></book>
2  <book/>
```

(2) 仅含文本的元素

有些元素仅含文本内容。从下面的例子中可以看到，<title>和<format>都是仅含文本的元素。

```
<title>红楼梦</title>
<format>电视剧</format>
```

(3) 含其他元素的元素

一个元素可以包含其他的元素。容器元素称为父(parent)元素，被包含的元素称为子(child)元素。例如，<DVD>元素就是一个包含子元素的元素。

```
<DVD id="1">
<title>红楼梦</title>
<format>电视剧</format>
</DVD>
```

(4) 混合元素

混合元素既含有文本也含有子元素。下面的代码片段显示了一个混合元素。

```
<DVD id="1">经典电视剧
<title>红楼梦</title>
```

```
<format>电视剧</format>
</DVD>
```

5. 元素的嵌套

XML 对元素有一个非常重要的要求——它们必须正确地嵌套。也就是说，如果一个元素(通过起始标签和结束标签来进行分隔)在另一个元素内部开始，那么也必须在同一个元素内部结束。例如，下列这些元素是格式良好的(Well-formed)。

```
< BOOK >
<TITLE>红楼梦</TITLE>
<AUTHOR>曹雪芹</AUTHOR>
< /BOOK>
```

不过，下列这些元素的格式不正确，因为它没有采用正确的元素嵌套格式。

```
<BOOK><TITLE>红楼梦</BOOK></TITLE>
```

在第一次遇到没有被正确使用的嵌套标记时，XML 解析器会立刻报告一个“not well-formed(非格式良好的)”的错误报告，而且通常情况下会退出处理过程。

综上所述，在 XML 文档中使用元素时应注意以下几点要求。

- 元素必须含有开始标签和结束标签。
- 在没有内容的情况下，才可以使用省略写法。
- 标签名称必须符合 XML 命名规则。
- 元素必须正确地嵌套。

2.5.2　XML 属性

XML 元素可以拥有属性。什么是属性？我们先来看下面这段 HTML 代码。

```
<IMG SRC="computer.gif ">
```

SRC 是 IMG 元素的属性，提供了关于 IMG 元素的额外信息。

属性是对标识进行进一步的描述和说明，一个标识可以有多个属性。XML 中的属性与 HTML 中的属性是一样的，每个属性都有它自己的名称和数值，属性是标识的一部分。例如：

```
<author sex="female">马丽</author>
```

XML 中的属性也是由用户自己定义的，属性由名称和数值组成，其中数值是包含于单引号或双引号中的；另外，一个元素可以有多个属性。它的基本格式如下：

```
<元素名 属性名="属性值">
```

特定的属性名称在同一个元素标记中只能出现一次；另外，属性值不能包括<、>、&符号。例如，对于一个人的性别，person 元素可以这样写：

```
<person sex="female">
```

也可以这样写:

```
<person sex="female">
```

上面的两种写法在一般情况下是没有区别的，而双引号的应用更普遍一些。

当元素包含属性时，常称为复合类型(Complex Type)元素，这在书写 XML 模式文档时是很重要的。

我们可以将属性改写为嵌套的子元素。例如，对于下列代码

```
<DVD id="1">
<title>红楼梦</title>
<format>电视剧</format>
</DVD>
```

我们可以改写为：

```
<DVD>
<id>1</id>
<title>红楼梦</title>
<format>电视剧</format>
</DVD>
```

关于哪种写法更好并没有一个明确的规则，两种写法都是可接受的。属性在 HTML 中可能十分便利，但在 XML 中，我们最好避免使用属性。因为使用属性时会引发以下一些问题。

- 属性不能包含多个值(而子元素可以)。
- 属性不容易扩展。
- 属性不能够描述结构(而子元素可以)。
- 属性很难被程序代码处理。
- 属性值很难通过 DTD 进行测试。

如果使用属性来存储数据，那么所编写的 XML 文档一般比较难以阅读和操作。所以我们最好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额外信息。

下面我们总结一下与属性相关的规则。

- 属性由名称和数值组成。
- 属性值必须封装在单引号或双引号中。
- 属性中不能含有 XML 标签。
- 属性名称必须遵循 XML 命名规则。

2.5.3　注释

注释可以出现在XML文档的任何位置。注释以 <!-- 开始，以 --> 结束。注释内的任

何标记都被忽略。如果希望除去XML 文档的一块较大部分，只需用注释标记括住那个部分即可。要恢复这个注释掉的部分，只需除去注释标记即可。

注释的语法如下：

```
<!-- 这里是注释信息 -->
```

例如，<!--THIS IS A COMMENT-->。

注释并不影响 XML 文档的处理，通常是为了便于阅读和理解。在添加注释时需要遵循以下规则。

- 注释里不能包含文本"-->"。
- 注释不能包含于标签内部。
- 元素中的开始标签或结束标签不能被注释掉。

养成良好的注释习惯将使我们的文档更加便于维护和共享。

2.5.4　字符引用和实体引用

与 SGML 和 HTML 一样，XML 为显示非 ASCII 码字符集中的字符提供了两种方法：字符引用和实体引用。我们先来了解一下字符引用。

1. 字符引用

有时一些字符是不能加入到 XML 文档中的，也许因为这些字符不在键盘上或者因为它是图形字符。在这种情况下，我们可以使用 Unicode 或十六进制数字将它们以字符引用的形式加入。比如，可以将版权符号 ã 编码成“&#169；”或“&#xA9；”。

以&#开始并以分号结束的引用都是字符引用。中间的数字是所需字符的Unicode编码。如果编码写成十六进制形式，那么它的前面有一个 X 作为前缀，如：

```
& # NNNNN;
& # XXXXX;
```

上面的字符串“NNNNN”和“XXXXX”可能是一个或多个数字，它们对应着任何 XML 允许的统一代码字符值。虽然在 HTML 中十进制数字更加通用，但XML 还是偏向于使用十六进制的形式，因为统一代码就是使用十六进制进行编码的。

下面我们通过一个实例来学习如何使用字符的引用。

实例 2-2　字符的引用

(1) 打开 XMLSpy，新建一个 XML 文档。

(2) 在编辑窗口中，从第 2 行开始，键入以下代码。

```
1  <data>
2  <data1>&#169；</data1>
3  <data2>&#xA9；</data2>
```

```
4  <data3>&#174;</data3>
5  <data4>&#xAE;</data4>
6  </data>
```

(3) 保存文件并查看结果。

将此文档保存为 CHAR1.xml 文档后，在编辑窗口最下端选择“浏览器”标签，即可查看显示结果，如图 2-12 所示。

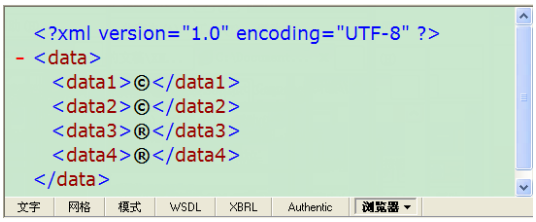

图 2-12　显示结果

2. 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，这就为字符引用提供了一种助记的替代方式。实体引用方式是在一个合法的XML 名字前面加上一个符号“&”，后面加上一个分号“；”，如下所示。

```
&name;
```

其中，有 5 个实体被定义为 XML 的固有部分，它们通常用作 XML 标记分隔符号的转义序列，如表 2-2 所示。

表 2-2　XML 中的实体定义

实　　体	用　　途
&amp;	通常用来替换字符&
&lt;	通常用来替换字符小于号
&gt;	用来替换字符大于号(>)
&apos;	可用来替换字符串中的单引号(')
&quot;	可用来替换字符串中的字符双引号(")

除了上述 5 个实体，其他所有的实体都必须在文档使用前，在 DTD 文档中予以定义。关于自定义实体的定义和使用方法，我们将在下一章中进行详细讲解。

2.6　命 名 空 间

XML 命名空间提供了一种避免元素命名冲突的方法。

2.6.1　命名冲突

因为 XML 文档中使用的元素不是固定的，那么两个不同的 XML 文档使用同一个名字来描述不同类型元素的情况就可能发生，而这种情况又往往会导致命名冲突。我们来看下面两个例子。

下面这个 XML 文档在 table 元素中包含了水果的信息:

```
<table>
<tr>
<td>苹果</td>
<td>香蕉</td>
</tr>
</table>
```

下面这个 XML 文档在 table 元素中包含了桌子的信息:

```
<table>
<name>餐桌</name>
<width>80</width>
<length>120</length>
</table>
```

如果上面两个 XML 文档片断碰巧在一起使用的话，那么将会出现命名冲突的情况。因为这两个片断都包含了 table 元素，而这两个 table 元素的定义与所包含的内容又各不相同。

2.6.2　解决命名冲突的方法

我们可以使用前缀解决命名冲突问题。

下面的 XML 文档在 table 元素中包含了水果的信息：

```
<fruit:table>
<fruit:tr>
<fruit:td>苹果</fruit:td>
<fruit:td>香蕉</fruit:td>
</fruit:tr>
</fruit:table>
```

下面的 XML 文档在 table 元素中包含了家具的信息：

```
< furniture:table>
<furniture:name>餐桌</furniture:name>
<furniture:width>80</furniture:width>
<furniture:length>120</furniture:length>
</furniture:table>
```

现在如果把上面的两段代码合为一体的话，就不会再有元素命名冲突的问题了，因为这两个文档对各自的 table 元素使用了不同的前缀，table 元素在两个文档中分别是<fruit:table>和< furniture:table>。

通过使用前缀，我们创建了两个不同的 table 元素。这种冲突解决方法的实现就要借助于命名空间了。下面我们介绍如何使用命名空间(NameSpaces)。

2.6.3　命名空间的使用

W3C 名称规范声明命名空间本身就是一个统一资源标识符(Uniform Resource Identifier，URI)。命名空间是阻止具有相同名字元素间的冲突的一种方法。在 XML 中，命名空间(NameSpaces)是被统一资源标识符分配或识别的一个虚拟空间。

命名空间通过给标识名称加一个网址(URL)以定位的方法来区别这些名称相同的标识。命名空间需要在 XML 文档的开头部分声明，命名空间的声明一般放置在元素的开始标记处，其使用语法如下所示：

```
xmlns:prefix="URI"
```

prefix 为定义的命名空间的前缀，是可选的，对应于两种命名空间的声明：默认的和明确的。

1. 默认声明

默认命名空间，不需要指定前缀，并且使用默认声明命名空间的所有元素和属性不需要任何前缀。代码示例如下：

```
<schema xmlns=" http://www.w3.org/2001/XML schema ">
        ......
        <element name="book" type="string"/>
</schema>
```

2. 明确声明

xmlns 关键字与一个命名空间 URI 的一个前缀相关联，代码示例如下：

```
<xsd:schema xmlns:xsd=" http://www.w3.org/2001/XML schema ">
      ......
      <xsd:element name="book" type="string"/>
 <xsd:/schema>
```

当在元素的开始标记处使用命名空间时，该元素所有的子元素都将通过一个前缀与同一个命名空间相互关联。

需要注意的是：用来标识命名空间的网络地址并不被 XML 解析器调用，XML 解析器不需要从这个网络地址中查找信息，该网络地址的作用仅仅是给命名空间一个唯一的名字，因此这个网络地址也可以是虚拟的，也就是说，设置URL 并不是说这个标识真的要到那个

网址去读取，仅仅作为一种区别的标志而已。我们可以分配任何名称或字符串作为一个URI。为了确保其一致性，应注意以下两点。

- 使用一个被开发者控制的 URI。
- 一般地，每个 XML 模式必须包括命名空间 URI ——http://www.w3.org/2001/XMLschema，使用它就可以确保按照 W3C 规定的标准来使用基本的 XSD 文档了。

我们通过实例来进行讲解。下面的 XML 文档在 table 元素中包含了水果的信息:

```
<h:table xmlns:h="http://www.w3.org/TR/fruit">
<h:tr>
<h:td>苹果</h:td>
<h:td>香蕉</h:td>
</h:tr>
</h:table>
```

下面的 XML 文档在 table 元素中包含了桌子的信息：

```
<f:table xmlns:f="http://www.w3sch.com/furniture">
<f:name>餐桌</f:name>
<f:width>80</f:width>
<f:length>120</f:length>
</f:table>
```

在上面两个例子中，两个 table 元素都使用了 xmlns 属性，使元素和不同的命名空间关联到一起，这是命名空间的明确声明方式。

而下面的两段代码就是命名空间的默认声明方式。

代码段 1，XML 文档在 table 元素中包含了水果的信息:

```
<table xmlns="http://www.w3.org/TR/fruit">
<tr>
<td>苹果</td>
<td>香蕉</td>
</tr>
</table>
```

代码段 2，XML 文档在 table 元素中包含了桌子的信息：

```
<table xmlns="http://www.w3schools.com/furniture">
<name>餐桌</name>
<width>80</width>
<length>120</length>
</table>
```

2.7 本 章 小 结

在本章中，我们介绍了所有 XML 文档中必需的基本语法，包括 XML 的文档结构及文档规则、XML 声明语句的写法、元素及属性的定义、字符和实体的引用、XML 中的命名空间的定义及使用等。其中最为重要的是 XML 的文档规则，因为只有符合 XML 文档规则的 XML 文档才是 Well-formed，即格式良好的 XML 文档，在书写 XML 文档时一定要特别注意这一点。

2.8 习　　题

一、填空题

1. 每个 XML 文档都分为两个部分：_____和_____。
2. <?xml version="1.0"　encoding="gb2312"?>就是一个_____。
3. XML 文档内容的主体部分，一般由_____、_____、_____、注释和内容组成。
4. 代表一个元素开始的分隔符被称做_____，代表一个元素结束的分隔符被称做_____。
5. XML 文档中一共有四类元素，分别为_____，仅含文本的元素，_____，含子元素、文本或混合元素的元素。

二、选择题

1. 属性(　　)用来表示 XML 文档所使用的字符集。
   A. version　　B. encoding　　C. standalone
2. XML(　　)提供了一种避免元素命名冲突的方法。
   A. 命名空间　　B. DTD　　C. XSD　　D. XSL
3. 下列哪段代码描述的是空元素？(　　)
   A.

```
<title>gone with the wind</title>
<format>movie</format>
<genre>classic</genre>
```

   B.

```
<DVD id="1">
<title>gone with the wind</title>
<format>movie</format>
```

```
<genre>classic</genre>
</DVD>
```

C.

```
<book/>
```

4. 含有中文字符的 XML 文档中，encoding 的属性值应设为(　　)。

A. BIG5　　B. GB2312　　C. UTF-8

5. 实体引用是一种合法的 XML 名字，前面带有一个符号(　　)。

A. &　　B. ；　　C. +

三、简答题

1. 写出格式良好的 XML 文档规则。
2. XML 元素的命名规范是什么？

四、上机题

1. 设想有这样一本书。

书名：XML 指南

第一章　XML 入门简介

1.1 节　什么是 HTML

1.2 节　什么是 XML

第二章　XML 语法

2.1 节　XML 元素必须有结束标签

2.2 节　XML 元素必须正确地嵌套

试使用 XML 文档进行描述，并上机实现。

2. 创建一个格式良好的 XML 文档，存储员工(employee)的信息，包括：员工号 id(属性)、姓名 name(元素)、年龄 age(元素)、性别 sex(元素)、住址 address(元素)。上机实现并在浏览器中进行查看。

3. 创建一个格式良好的 XML 文档，存储学生成绩的信息，包括：学号 number(属性)、姓名 name(元素)、成绩 score(元素)。上机实现并在浏览器中进行查看。