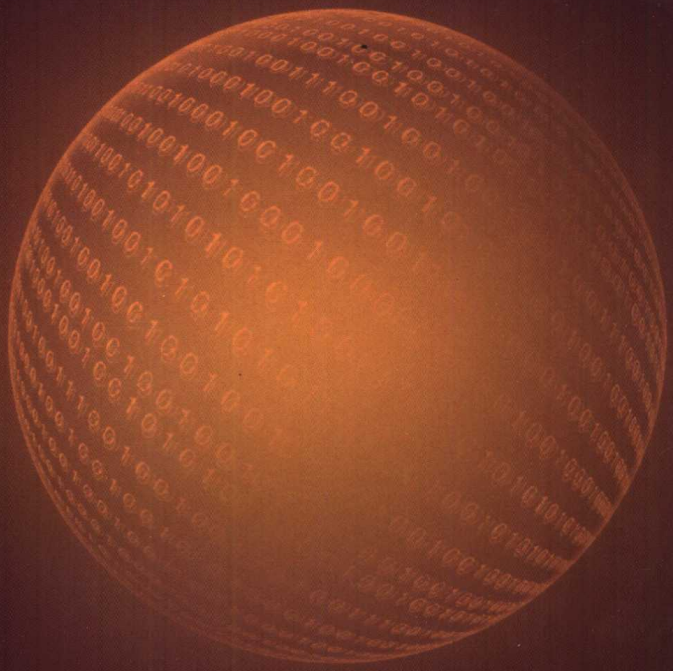

操作系统与网络技术
系列教材

XML基础与应用教程

印 旻 景建萍

高等教育出版社

操作系统与网络技术系列教材

XML 基础与应用教程

印 旻 景建萍

高等教育出版社

内 容 提 要

本书主要介绍 XML 的功能及实际应用。第一章首先概述 XML 的历史、现状和未来发展前景；第二章到第六章详细介绍了 XML 的语法及相关编程知识，包括 DTD、CSS、XSL、XPath、XPointer、XLink 等；最后，从第八章开始，结合实例讨论如何利用 XML 开发实际的 Internet 应用，包括 XML 语言与 Java 语言的结合使用，XML 技术与 Java Servlet 技术的结合使用，XML 在 ASP 中的应用等。本书的最后一章还介绍了一种专用的 XML 语言——VoiceXML 语言，一方面作为读者定义自己的 XML 的范例，同时也向广大读者介绍了 Internet 上的语音用户界面这一令人兴奋的新生代技术。

本书可作为高校本科、高职高专学校的教材或教学参考书，也适合于所有对新一代 Internet 编程技术感兴趣的读者学习、使用。

图书在版编目（CIP）数据

XML 基础与应用教程/印旻、景建萍. —北京：高等教育出版社，2001.7

ISBN 7-04-009244-1

Ⅰ.X… Ⅱ.印… Ⅲ.可扩充语言，XML－程序设计－高等学校－教材 Ⅳ.TP312

中国版本图书馆 CIP 数据核字（2001）第 036585 号

XML 基础与应用教程

印旻 景建萍

出版发行	高等教育出版社		
社　　址	北京市东城区沙滩后街 55 号	**邮政编码**	100009
电　　话	010-64054588	**传　　真**	010-64014048
网　　址	http://www.hep.edu.cn http://www.hep.com.cn		
经　　销	新华书店总店北京发行所发行		
印　　刷	中国农业出版社印刷厂		
开　　本	787×1092　1/16	**版　　次**	2001 年 7 月第 1 版
印　　张	12.75	**印　　次**	2001 年 7 月第 1 次印刷
字　　数	300 000	**定　　价**	16.00 元

凡购买高等教育出版社的图书，如有缺页、倒页、脱页等质量问题，请与当地图书销售部门联系调换

前　言

XML 是继 Java 之后的又一新兴技术热点，目前其开发热潮不亚于甚至有过于 Java。与 Java 及其他一切新兴技术相似，XML 正处于逐步摸索和逐渐成熟的过程，其语言标准在不断地变化，Web 开发者和程序员也在不断地探索使用 XML 开发应用程序的各种方法和模式。随着各方面经验的积累，人们愈来愈清楚地意识到，XML 正在成为 Internet 编程新一轮具有革命性的代表技术之一，使用 XML 替代或部分替代 HTML，将会大大简化和方便 Internet 应用的开发和使用，从而实现真正的网上交易和事物处理，使 Internet 从静态的全球性信息集散地发展成为动态的全球性交互平台。

本书首先概述 XML 的历史、现状和未来发展前景；再详细介绍 XML 语法及相关编程知识；最后结合实例，讨论如何利用 XML 开发实际的 Internet 应用，包括如何开发 XML 与 Java 结合的应用等。使用本书不需要特别的基础，在学习第二章到第六章所介绍的 XML 语法过程中，读者可以随时尝试创建并丰富自己的 XML 文档，并逐步了解如何创建、管理和维护正确的文档结构，如何定义和编写 XML 的 DTD 文件。从第七章开始介绍如何使用不同的 XML 解析器，从而真正运行自己编写的 XML 程序文档。随后的几个章节将由简至繁地通过实例介绍如何在实际的 Internet 应用中选择使用不同的 XML 文档结构开发 XML 程序，对于较为复杂的应用，还讨论了结合 Java 等编程语言开发 XML 应用程序的方法。

本书适合于所有对新一代 Internet 编程技术感兴趣的读者学习、使用或参考。了解 HTML 编程的读者会发现，XML 与 HTML 系出同源，易于学习，令学习过程事半功倍。熟悉 Java 编程的读者也会发现，XML 与 Java 独到的结合之处，是开发 Internet 应用的又一个好帮手。即使对于没有上述编程基础的读者，本书也可以作为学习 Internet 编程的参考书，帮助读者尽早对这个领域建立更全面的了解，从一个较高的起点开始，早日加入到日益壮大的 XML 开发人员的队伍中来。

本书的第四章和第六章的第二节由景建萍编写，其余部分由印旻编写。

由于作者水平所限，书中难免遗漏和疏误之处，欢迎各位同行专家和广大读者批评指正。

编　者

2001.3

目　录

第一章 概 述

本章将简要介绍 XML 的有关背景知识及其应用前景，包括 XML 的设计目的，它所针对的问题，它同 SGML 和 HTML 的关系，以及 XML 对 Internet 编程、网页创作、数据库设计甚至整个网络结构构建的深远影响。最后简要列举一些使用 XML 的网络应用类型。

1.1 XML 简 介

XML 是 eXtensible Markup Language 的缩写，其全称为“扩展标记语言”。它定义了一种文件格式，一种保存数据的方法，使用这种格式和保存方法的计算机数据，可以在不同的计算机平台和不同的计算机程序之间方便、平稳、快速和无障碍地转移和流动，从而大大提高了我们处理数据的效率和灵活性。这就是 XML 的主要设计目的。

为达到这个目的，XML 的基本原理规定数据按照其固有的结构以层次化的格式存储，从而形成一种基于内容的格式，使得无论是人、还是各种计算机程序，都能方便地了解、掌握和维护 XML 文档的内在结构信息，快速、准确地定位所需信息。比较下面两段 HTML 与 XML 的程序。

例 1.1 HTML 程序。

```
<!--显示飞机票订单的 HTML 程序-->
   <h1>TICKET_ORDER
   <p>ID：SD1803
   <p>FROM：北京
   <p>TO：上海
   <p>DATE：2000 年 5 月 20 日
   <p>TIME：8：30AM
   <p>PRICE：700.00
   <p>AMOUNT：2
   <p>TOTAL：1400.00
```

例 1.2 XML 程序。

```
<!--保存飞机票订单的 XML 程序-->
<TICKET_ORDER>
        <ID>SD1803</ID>
        <FROM>北京</FROM>
        <TO>上海</TO>
        <DATE>
```

```
            <YEAR>2000</YEAR>
            <MONTH>5</MONTH>
            <DAY>20</DAY>
        </DATE>
        <TIME>
            <HOUR>8</HOUR>
            <MINUTE>30</MINUTE>
        </TIME>
        <PRICE>700.00</PRICE>
        <AMOUNT>2</AMOUNT>
        <TOTAL>1400.00</TOTAL>
</TICKET_ORDER>
```

可以看出，例 1.1 中的 HTML 程序把所有的飞机票订单信息罗列出来，其中用尖括号括起的标记仅仅说明了这些信息的显示格式。而例 1.2 中的 XML 程序则清晰地显示出了订单的逻辑结构，不但方便程序阅读和不同信息的抽取，还可以直接与存储订单的数据库的各个字段对应起来。如果网上订票程序使用了 XML 语言，用户输入各条相应信息子项后，数据就可以以例 1.2 的格式存储并一一对应地存入数据库。这样一来，统计乘客人数的程序就可以迅速利用标记<ID>和<AMOUNT>更新班机上的空余座位数；统计一年或一天中不同时段乘客流量的程序也可以利用<DATE>和<TIME>标记查询到自己所需要的信息片段。而如果使用例 1.1 中的 HTML 程序，就没有这些基于结构和内容的标记，要完成相同的数据处理和操作，其复杂性将大大增加。

1.2　XML 的历史与未来

XML 的历史可以追溯到 1969 年，美国 IBM 公司一组研究人员开始设计一种名为 GML（Generalized Markup Language）的语言。在印刷、统计等需要大规模数据处理的行业和部门的支持下，这项研究工作持续了十几年，于 1980 年推出了 SGML（Standard Generalized Markup Language）语言，并于 1986 年获得国际标准化组织（ISO）的批准。其后，SGML 的发展较为平稳，并不为其领域之外的人们所广泛了解，直至 1991 年，当大名鼎鼎的 HTML（Hypertext Markup Language）问世之后，大家才对 SGML 有所耳闻。

HTML 是目前 Web 上最常用也是使用人数最多、范围最广的工具，尽管 HMTL 与 SGML 看起来相似，而且它也借用了 SGML 的很多元素，但是 HTML 的处理方法却与 SGML 有很大的不同。最主要的一点是 HMTL 使用固定的、仅与外观显示格式有关的标记，它虽源于 SGML，但相对于 SGML 却简单得多，因此也有一定的局限性。随着 Web 的飞速发展，HTML 的固有局限性愈来愈凸现出来，逐步成为 Web 发展的障碍。

1996 年，W3C（World Wide Web Consortium）成立了一个专门的委员会，研究如何开发一种新的标记语言，既能够利用 SGML 的强大数据处理能力，又适合于在 Web 上使用。1996 年底，这个委员会推出了 XML 语言的第一稿，同时将自身更名为 XML 工作组，并于 1998

年 2 月 10 日正式向 W3C 提交了 XML 的最终推荐标准。这个标准就是 XML1.0，其中详细说明了 XML 的设计目的，主要包括：

1. XML 可以直接应用于 Internet。
2. XML 应支持尽可能多种类的应用。
3. XML 应与 SGML 兼容。
4. XML 的解析程序应易于编写和处理。
5. XML 文档应便于用户阅读、理解。
6. XML 文档应易于创建。
7. XML 的设计过程应简单、迅速。

经过五年的发展，XML 已经被广大编程人员、数据库开发者和文档编写人员所接受，其发展速度之惊人不亚于当年的 HTML 或 Java。例如 W3C 所支持的另外两种语言 SMIL（Synchronized Multimedia Integration Language，用于创建 Web 上的复杂的多媒体文档）和 MathXL（用于显示和处理数学公式）都是以 XML 为基础，XML 已经逐渐成为整个 Web 的基本结构和未来各种发展的基础，它将把 Web 从静态的信息中心扩展成为互动的世界性的商贸和事务处理中心，对整个世界产生深远的影响。

1.3 SGML、HTML 和 XML

我们已经知道，HTML 与 XML 都源自 SGML，三者都是标记语言，其中 SGML 定义了标记语言的基本概念，奠定了标记语言的技术基础，在 HTML 和 XML 中随处可见 SGML 的影子。与 HTML 和 XML 相比，SGML 的应用和涵盖范围、语言定义集合及其具体的实现方法，都要复杂得多，感兴趣的读者可以参考 Charles Goldfarb 所著的 The SGML Handbook（牛津大学出版社，1990），这里就不再详细讨论了。

HTML 是标记语言家族中非常著名的一员，它与 HTTP 协议共同组成了 WWW 的最基本标准，对人们的工作、生活和娱乐方式都产生了革命性的影响。简单地说，HTML 文件包含了文档数据和显示格式两部分，其中文档数据是显示在 WWW 浏览器中的数据内容，显示格式则规定了这些内容以何种格式、样子呈现给用户。HTML 可以看作是 SGML 的一个应用，它的所有标记都有固定的含义，编程者不能随意更改，其中大部分标记都用来规定信息的显示格式。例如
规定换行，<FONT>规定字体等。

近年来，WWW 已经成为公认的世界性的信息中心，HTML 就是这个信息中心的核心。而新近崛起的 XML 正在把 WWW 从静态的信息中心扩展成为互动的世界性的商贸和事务处理中心。HTML 仅能定义数据的外观和表现形式，而 XML 则是 HTML 的扩展，它定义了数据的真正物理含义。相对于 HTML 而言，XML 具有灵活、简单、易读的优点，开发者可以在 XML 中自定义与特定事务或处理相关的标记，无论是最终用户还是程序设计人员，都能轻松地读懂这些标记。XML 还具有很好的可扩展性，其简单性使之成为开发小应用程序的好帮手，其清晰的结构又使之在处理大型应用时游刃有余。XML 的格式设计充分考虑了可重用性，开发者可以方便地揉和、处理、剪辑和重建已有的应用，快速、可靠地开发应用，建立系统，实现真正的纯计算机交易和无纸应用。

1.4　XML 的 Web 应用

严格地说，XML 仅规定了一种文件格式，要在 Web 中使用 XML 并充分发挥其优势，还需要与其他相关技术，如网络技术、数据库技术、网络发布技术等结合使用。由于 XML 是数据的载体，它在 Web 中的主要作用是实现 Web 的各个层次、各个节点、各个平台间更有效的数据传输、交换和处理。本节将讨论 XML 在 Web 应用中的几种常用方式。

1.4.1　XML 与 HTTP 协议

在 Web 中使用 XML 最简单的方法就是借用目前 HTML 语言在 Web 中的使用模式，参见图 1.1。

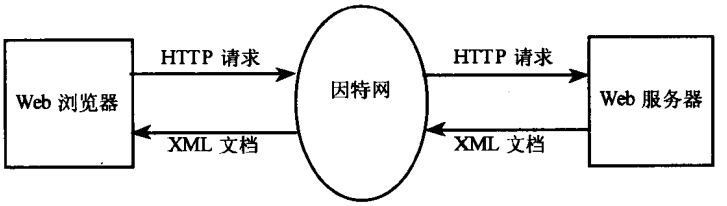

图 1.1　使用 HTTP 协议传递 XML 文档

如图 1.1 所示，当 Web 浏览器通过网络向 Web 服务器提出 HTTP 请求时，Web 服务器根据此请求调出对应的 XML 文档并经网络传回 Web 浏览器，由 Web 浏览器解释该 XML 文档，完成相应的执行或显示操作。

这种模式实际是 Web 上最基本的工作模式，只是使用 XML 文档替代了 HTML 文档。需要特别注意的是，这种模式中的 Web 浏览器必须能够读懂 XML 文档，即必须包含 XML 解析器。最初的 Web 浏览器主要用来阅读 HTML 文档，仅仅包含 HTML 解析器。HTML 语言较为简单，其中的各种标记基本固定不变，而 XML 文档中包含用户自己定义的标记，XML 解析器要想读懂这些标记，就需要比 HTML 解析器更复杂的结构。

利用 HTTP 协议传送 XML 文档到 Web 浏览器的模式有很多优点。首先能确保文档在不同的浏览器中有相同的显示效果。目前，不同公司的浏览器（主要是 Netscape Navigator 和 Microsoft IE）在基本的 HTML 语言标准之外，又分别扩充定义了若干互不相同的新标记，使得在一种浏览器中能够执行的 HTML 程序不能被另外一种浏览器理解，或者相同的 HTML 文档在不同的浏览器中显示的效果不同。使用 XML 替代 HTML 则可以避免这个移植性的问题，因为每个 XML 文档都会自带其自定义标记的说明供浏览器使用，浏览器基于这些说明来解释、执行和显示 XML 文档，就不会产生一致性问题。

其次，使用 XML 文档会成倍地提高搜索引擎的效率。因为搜索引擎可以直接参考 XML 中的标记作为搜索的依据，迅速缩小搜索范围，而不必扫描整个文档内容。另外，在 Web 服务器和 Web 浏览器之间传递 XML 文档还可以扩展浏览器的功能。

1.4.2 基于 Web 服务器的 XML 应用

基于 Web 服务器的 XML 应用在把 XML 文档送回 Web 浏览器之前，首先在 Web 服务器端进行若干附加的处理操作，例如数据库操作等。这种模式通常需要更复杂的网络结构，同时也会有更强大的功能。

这种模式通常会将 XML 程序与其他一些 Web 服务器端的技术结合起来使用，例如 CGI、Java Servlet、JSP 等。在 XML 文档被传送回 Web 浏览器之前，Web 服务器先要利用上述技术对其进行一定的加工处理，从而保证 Web 浏览器能够获得动态更新的数据。

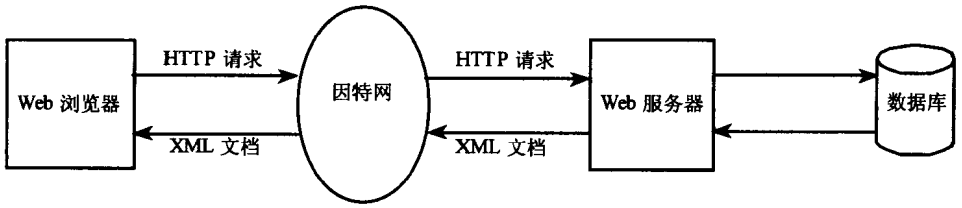

图 1.2 基于 Web 服务器的 XML 应用

图 1.2 显示了基于 Web 服务器的 XML 应用中较为常用的一种，当服务器接收到浏览器发出的 HTTP 请求时，就调用并执行保存在服务器中的 Java Servlet 程序，由该 Servlet 程序通过 JDBC 与数据库连接并完成必要的查询、插入、删除等数据库操作，然后利用获得的数据产生一个新的包含动态数据的 XML 文档，并将其发送回 Web 浏览器。

需要特别指出的是，使用关系型数据库可以保存 XML 文档中的数据，但是由于 XML 所固有的基于内容的层次性结构，使用对象型数据库或对象-关系型数据库保存 XML 数据效果会更好。

1.4.3 XML 与网络中间件

网络应用通常包含不同的组件和层次，各组件和层次之间需要交换数据，于是一个网络应用中就会有各种不同的数据源，如数据库、文本文件、多媒体文件、对象等。对于这些多种多样的数据源来说，XML 是一个很好的数据中间存储格式，所以 XML 在网络中间件中有着广泛的应用，实际上，最早的 XML 商业应用就与网络中间件有关。

XML 可以在使用中间件的网络应用中扮演多种不同的角色，从最简单的由单一数据源取得数据并直接传回浏览器，到复杂应用中从多个数据源中取得多个文本及多媒体数据项加以同步、裁减、合成、统计等操作，都可以将 XML 作为数据中转、交换的载体和工具。另外，XML 还可以用于浏览器到服务器之间的交换数据，或者在不同服务器之间交换数据。

XML 可以简化上述种种发生在网络应用中的数据流动，就像使用相同语言则易于交流沟通一样，网络应用的各组件和层次如果使用 XML 作用公共的统一的数据格式，就能顺畅、方便地实现数据交换。对于那些不方便直接使用 XML 数据格式的组件，例如关系型数据库，可以在其与其他组件接口处构造一个 XML 与其内部数据结构相互转换的层次，把从其他组件输入的 XML 格式的数据转化成其内部数据格式，当操作完毕需要向其他组件输出数据时，先做反方向的转换，再将 XML 格式的数据输出给其他组件。

有时一个网络应用系统中会有不只一个的组件需要使用自己的内部数据结构，需要构建XML与内部数据结构的格式转化层，此时它们之间可以共享相同的XML解析器和有关该应用XML自定义标记的说明。

1.4.4　文档对象模型（DOM）

由于XML以文档的内在结构组织数据，其数据结构与面向对象的语言中封装了数据和操作的对象非常相似，所以W3C又专门为XML应用提出了文档对象模型（Document Object Model，简称DOM）的概念。

动态HTML语言（Dynamic HTML）是为了突破HTML的静态数据处理局限而提出的语言，推出后很受欢迎。但是由于不同厂商推出的动态HTML相互之间不能兼容，限制了这种语言的推广使用。DOM就是W3C为了统一不同的动态HTML语言而作出的努力，同时DOM的使用还不仅仅局限于动态HTML语言，它在XML技术中显得更为重要。

DOM提供了一组与语言无关的标准对象模型，使用这些标准对象模型编写XML或HTML文档为开发人员或最终用户提供了一个获取、修改或操作文档内容的方便界面。

使用DOM API界面能够将数据存储与数据表示分离开来，是开发XML应用的有效工具，目前Microsoft等公司都提供有DOM API工具。

习　题　一

1. 什么是XML语言？XML语言由什么语言发展而来？XML语言与HTML语言的关系如何？
2. XML的设计目的是什么？
3. XML与HTML相比有何优点？
4. 简述XML在Web中的几种主要应用方式。

第二章　XML 基本语法

本章将详细介绍 XML 的基本语法知识，读者在学习完本章之后就可以尝试编写 XML 小程序。由于 XML 还处于飞速上升期，有些技术和标准还在研究和探索之中，但是其核心概念和语法已经确立。本章首先介绍标记语言的一般性基础知识，然后具体讨论 XML 的语法。

2.1　标记语言的基本概念

这里先简单介绍一下标记语言及其基本概念。

标记语言的基本特点是：在表述数据内容的基础上，在其中插入各具含义的标记，起到对数据内容解释、说明和规范、限制的作用。

所有的标记都用尖括号“<”和“>”括起来，以区分标记与数据内容。标记一般成对使用，例如<DATE>和<HTML>是开始标记，</DATE>和</HTML>是结束标记，但是也有标记可以单独使用，例如<P>等。

成对使用的标记将数据内容划分为不同的元素，开始标记标志着元素的开始，而结束标记则标志着元素的结束。元素的名称在开始标记和结束标记中加以说明，通常的格式是：

```
<元素名称 属性名称=属性值>
</元素名称>
```

例如，前面提到的 DATE 和 HTML 都是元素的名称。在开始标记内，除了元素名称之外，通常还包括属性项，用以说明该元素的各种属性。例如：

```
<a href="http://www.yahoo.com">
```

在这个开始标记中，a 是元素名称，href="http://www.yahoo.com"是元素的属性项，其中 href 是属性名称，“http://www.yahoo.com”是属性值。一个元素可以同时说明多个属性，例如开始标记：

```
<table BORDER=0 CEFFSPACING=5 WIDTH="%100">
```

就说明了 BORDER、CEFFSPACING 和 WIDTH 三个属性的值。

除了属性，元素还可以定义更复杂的语法结构，甚至包含其他的元素，例如例 1.2 中的程序段：

```
<DATE>
        <YEAR>2000</YEAR>
        <MONTH>5</MONTH>
        <DAY>20</DAY>
</DATE>
```

其中，DATE 元素中又包含了 YEAR、MONTH 和 DAY 三个子元素。而且根据 DATE 元素

的语法，它必须包括上述三个子元素。

元素也可以没有任何属性，甚至没有任何内容，例如下面的语句就定义了一个没有属性，没有内容的空元素：

```
<Empty_Element>
</Empth_Element>
```

实体是标记语言中另一个非常重要的概念，实体通常用来指向一个字符或数据、一段文本甚至是一个文件。当包含实体的标记语言被解释执行或显示时，实体所指向的真正内容将替代实体符号被执行或显示。在 XML 中实体还可以指向其他应用，例如 Java 程序等，使得 XML 程序与 Java 程序能够结合起来使用。

元素、属性和实体是所有标记语言都使用的最基本的概念。XML 中的元素、属性和实体将在后面陆续做详细介绍。

2.2　创建简单的 XML 文档

XML 文档大致可以分为有效 XML 文档和无效 XML 文档两类。所谓有效 XML 文档，是指其内容结构严格遵守它的标记说明，即符合自身语法规定的 XML 文档；无效 XML 文档则指不能通过自身语法检查的文档。无效并不等于错误，无效 XML 文档一样可以正常运行。很多情况下，正是因为文档本身包含了很多复杂的结构，才不能通过语法检查，但是这些复杂的结构同时也带来了更为强大的功能。我们首先来看最简单的有效 XML 文档。

与其他的标记语言一样，XML 文档由内容和标记两部分组成，其中内容是文档中的数据部分，而标记则说明了该文档数据的组织结构。标记由尖括号括起以便与数据区分开来。尖括号中的内容并非要传送或储存的数据，而是一些说明性的指令，告知 XML 解析器和其他处理 XML 的程序如何解释文档的内容数据，应对这些数据做哪些处理。

2.2.1　声明

XML 文档的标记中包含了很多相当严格的声明性信息，如 XML 声明、版本声明、字符编码声明等。首先来看看 XML 声明。

1. XML 声明

所有的 XML 文档（或程序）都必须以 XML 声明作为文档的第一条语句，XML 文档可以包含非内容数据的注释信息，但即使是注释信息也不能出现在 XML 声明之前。将 XML 声明写在文档的最前面可以及早通知 XML 解析器该文档的性质。

最简单的 XML 声明如下：

```
<?xml?>
```

这个标记以问号开始，以问号结束，两个问号之间包括了关键字 XML 或 xml。XML 是 XML 语言的保留字，用户在定义自己的标记时，不能将元素命名为 XML 或 xml。XML 中的很多标记都是成对出现，而 XML 声明则只有一个，即只在文档开始处出现一次。

需要注意的是，虽然 XML 声明中 XML 三个字母可以任意大小写，但 XML 语言本身却是区分大小写的语言。当用户定义自己的标记时，一定要注意这一点。例如，如果定义了元

素的开始标记：

<MyElement>

则结束标记必须是：

</MyElement>

</Myelement>或</myElement>等都是错误的。

2. 版本声明

XML 声明中还可以加入其他的声明信息进一步说明当前 XML 文档的性质。版本声明就是其中之一。

版本声明也是 XML 文档必须定义的部分，它必须紧跟再 XML 声明之后，例如：

<?xml version='1.0'?>

说明当前 XML 文档使用的是 XML1.0 标准。

版本声明的作用在于通知 XML 解析器使用何种 XML 标准解释、处理当前的文档。由于 XML 还处在飞速地发展变化中，当新标准推出时，版本声明可以帮助解析器区分用新、旧两种标准书写的 XML 文档，从而保证使用老版本、旧标准的文档也能够被正确地解释、执行，而不必将新标准推出之前所写的 XML 文档全部重写。

具体的版本号可以用单引号括起，也可以用双引号括起，例如：

<?xml version="1.1"?>

也是正确的书写方法。

3. 字符编码声明

字符编码声明说明当前 XML 文档中的字符采用何种字符编码方式，字符编码声明并非强制性声明，一个 XML 文档中可以没有字符编码声明，但是如果定义了字符编码声明，就必须把它放在 XML 声明之中，版本声明之后。例如：

<?xml version='1.0' encoding='UTF-8'?>

声明当前的 XML 文档使用 UTF-8 字符编码标准，UTF-8 是 Unicode 编码标准中的一种。

XML 的字符编码标准是 Unicode，因此所有的 XML 解析器都应该提供对 Unicode 编码标准的支持。

Unicode 是正在普及使用的新的字符编码标准，在 Unicode 中每个字符都用 16 比特表示，可以表示 65 536 个不同的字符。与 Unicode 之前被广泛使用的 ASCII 码相比，Unicode 码最大的好处是在涉及到多语种字符处理，尤其是东方字符处理时，可以使用相同的程序。ASCII 码规定每个字符用 8 比特表示，所以总共只能表示 128 个字符，对于字符集较大的东方字符，例如汉字，就必须一个字符用两个 ASCII 码表示。这样，完成同样的功能，对于西方字符和东方字符要使用不同的处理程序，而使用 Unicode 编码则没有这个问题。目前越来越多的系统、语言和标准开始支持 Unicode 编码，Java 也是其中之一。

Unicode 码中，最常用的就是 UTF-8，这种编码标准能够兼容 ASCII 码，若 XML 文档的内容数据为英语，使用 UTF-8 最为方便。其他常用的 Unicode 码还有 USC-2，一般的 XML 解析器都应该支持这两种 Unicode 码。

4. 独立文档声明

独立文档声明通知 XML 解析器在解释当前 XML 文档时，是否需要从其他的外部资源获得有关其自定义标记的说明并检查当前 XML 文档的有效性。

独立文档声明也不是强制性声明，若一个 XML 文档需要定义独立文档声明，就必须把这个声明写在 XML 声明之中，版本声明和字符编码声明（若该文档存在字符编码声明）之后。例如，下面的声明就说明当前 XML 文档是一个需要外部资源检查其语法有效性的文档：

```
<?xml version='1.0' standalone= 'yes'?>
```

如果 XML 文档不定义独立文档声明，XML 解析器就缺省认为该 XML 文档的独立文档属性为“no”，即不需要特别检查其语法有效性。

5. 注释

XML 语言中的注释与 HTML 语言的注释非常相似，使用方法也基本一致。XML 的注释以左尖括号（即小于号）紧接着一个惊叹号和两个减号开始，以两个减号紧接着一个右尖括号（即大于号）结束，中间包含着具体的注释信息，对当前的 XML 文档或文档片段加以解释说明。例如：

```
<!--THIS IS MY XML COMMENT-->
```

注释可以提高 XML 文档的可读性，XML 解析器在解释、执行 XML 程序时会跳过所有的注释。需要特别注意的是，不能在一个标记的内部定义注释，例如下面的标记定义就是非法的：<COMMENT-INCLUDED <!--this is my comment-->/>

下面是一个简单的 XML 程序，已经在 IE 5.0 上调试通过。

```
<?xml version = "1.0"?>
<!-- Simple introduction to XML markup -->
<myMessage>
    <message>Welcome to XML!</message>
</myMessage>
```

2.2.2　元素

声明之后，就是 XML 文档的具体内容，这些内容必须包含在一个唯一的主元素中，这个主元素中通常还会按照层次结构关系定义一些其他元素，这些元素包含在主元素之中，它们自己也可以再包含各自的子元素，从而形成整个 XML 文档的树状层次结构。例如例 1.2 中的 XML 文档就包含了如图 2.1 所示的层次结构。

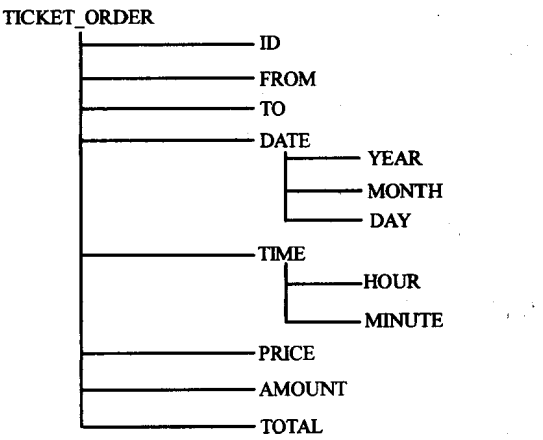

图 2.1　例 1.2 的元素层次关系树