

北京科海培训中心



网页设计·网站建设培训系列教材

网页核心

XML 应用技巧与实例



www.Kehaipzbook.com

www.Kehaipzbook.com

www.Kehaipzbook.com

刘 光 编著



清华大学出版社

北京科海培训中心

► 网页设计·网站建设培训系列教材

网页核心 XML 应用技巧与实例

刘 光 编著

清华大学出版社

(京)新登字 158 号

内 容 提 要

本书结合大量的实例,全面系统地介绍了下一代网页发布语言——XML 的功能及其使用。全书内容包括:XML 概述,预备知识,XML 语法,使用 CSS 格式化 XML 文档,使用 XSL,名域,使用 DTD 验证 XML 文档,XML Schema,XML 在数据库中的应用,以及 XML 的其他问题等。内容安排由浅入深,由易到难,包含丰富的实例。读者通过本书的学习,再加上一些练习,肯定能设计出页面漂亮、功能强大的网页。

JS484/06

版权所有,盗版必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得进入各书店。

书 名:网页核心 XML 应用技巧与实例

作 者:刘光

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者:北京门头沟胶印厂

发 行:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:21.75 字数:529 千字

版 次:2000 年 10 月第 1 版 2000 年 12 月第 2 次印刷

印 数:5001~10000

书 号:ISBN 7-302-04068-0/TP. 2397

定 价:28.00 元

前 言

在现代高科技中,计算机是发展最快的,而网络作为计算机中的新兴技术,发展速度更是快中之快。现在有人提出“新摩尔定律”,即光纤定律(Optical Law):Internet 频宽每 9 个月会增加一倍,而成本同时降低一半。这比摩尔定律讲的芯片容量增长一倍需 18 个月还要快。网络发展的高速度不仅体现在硬件上,从网页发布语言的发展中也可窥见一斑。短短几年间,HTML 由 1992 年的第一个版本发展到如今的 4.01 版本,包括 JAVA APPLET、JAVASCRIPT、ActiveX、CSS 以及 DOM 在内的一系列新技术被应用于网上,XML 作为 W3C 推荐的下一代网页发布语言更是风靡全球。而与此同时,又有一大批新的标记语言问世。

面对如此迅猛的发展,网页编辑器的开发者和网页设计人员可能会感到无所适从。究竟应该选择哪一种网页标记语言作为自己的 Web 页面发布语言呢?

HTML 大大丰富了主页的视觉、听觉效果,为推动 WWW 的蓬勃发展,为推动信息和知识的网上交流发挥了不可取代的作用。然而,HTML 是专门为描述主页的表现形式而设计的,它疏于对信息语意及其内部结构的描述,不能适应日益增多的信息检索要求和存档要求;再者,HTML 对形式的描述能力也是很不够的,它无法描述矢量图形、科技符号和一些其他的特殊显示效果。HTML 的标记日益臃肿,文件结构混乱而缺乏条理,导致浏览器的设计越来越复杂。这一切都使得 HTML 已不能再满足网络上新的应用需求。

为了弥补 HTML 的不足,W3C 发布了 XML 作为下一代网页发布语言。XML 是 SGML 的子集,它继承了 SGML 和 HTML 的诸多优势。XML 不再限于标记语言的范畴,它是一种元标记语言。XML 把 DTD 的定义权开放,不同行业可以根据自己的实际需求定义描述内容的 DTD,以适应本行业内部的信息交流和存档需要,还可以定义专门描述形式的 DTD,丰富信息的描述效果。XML 强调形式描述与内容描述的分离,一方面使 XML 文件的编写者集中精力于数据本身,而不受显示方式的细枝末节的影响;另一方面允许为相同的数据定义不同的显示方式,从而适合于不同应用、不同媒体,使 XML 数据得到最大程度的重用性。XML 赋予标记一定的语意,加强了对内容的描述,便于网上的信息检索。XML 将网络上传输的文档规范化,也方便了浏览器的设计。总之,XML 具有易于编辑、便于管理、适于存档、容易查询等诸多优势,成为网络发展中继 JAVA 后的又一个亮点。

本书力求做到适合各种层次的读者。内容安排上由浅入深,从基本操作到高级应用,内容丰富、实例详尽。本书共分 10 章,第 1 章罗列了 HTML 的不足,引出弥补这些缺陷的 XML 及其特点、优势、应用领域等,最后介绍了 XML 和其他网页编写语言的关系,使读者对 XML 有一个总体概念,把握住其实质和方向。第 2 章是为没有网页编写知识的读者准备的,在这一章中简单地介绍了 HTML、VBScript 和 ASP 的语法和运用。第 3 章从一个 XML 文档出发,详细阐述了 XML 的语法规则,包括元素、属性、CDATA 区、规范的 XML 文件、有效的 XML 文件等内容。第 4 章详细介绍层叠样式表单的基本要素,包括 CSS 的基本概念、CSS 属性、格式化模型以及 CSS 单位等。第 5 章介绍如何使用 XSL 来格式化 XML,以及使

用 XSL 处理程序、XSL 模式语言、XSL 查询语言等。第 6 章涉及名域的必要性、名域的宣告及应用、名域的范畴以及预设的名域等。第 7 章首先概述一些有关 DTD 的问题,例如为什么要使用 DTD,DTD 到底是什么,DTD 的未来在哪里等,然后介绍了如何构造 DTD 文件,对 XML 文档进行有效性验证。第 8 章首先简介 XML Schema,使读者对它有一个整体上的把握,然后通过具体的实例介绍了如何编写 XML Schema,包括:其元素、属性类型的定义,注释的编写,使用其他大纲等。第 9 章通过两个具体实例介绍如何在应用程序中将数据库和 XML 结合起来,创建网络应用程序。第一个实例是 XML 在 VB6.0 中的应用,另一个实例是如何在 C++ Builder 5.0 中使用 XML 创建网络多层数据库。第 10 章首先详细地介绍了 XML 的一些显示语言,例如 XHTML、SVG、WML 等,并通过两个示例演示如何使用这些语言,然后介绍了 XML 的未来,以及一些有关 XML 的专业词汇。

由于本人能力和时间的限制,书中肯定有一些错误和遗漏,恳请广大读者批评指正。

编者

2000 年 5 月

目 录

第1章 XML概述	1
1.1 HTML的不足.....	1
1.1.1 HTML难以扩展.....	1
1.1.2 HTML交互性差.....	2
1.1.3 HTML语义性差.....	2
1.1.4 单向的超链接.....	5
1.1.5 HTML缺乏对双字节或多国文字的支持.....	6
1.2 XML的诞生	7
1.2.1 XML产生的背景.....	7
1.2.2 什么是XML	9
1.2.3 开发者和开发目标.....	11
1.2.4 XML的优越性	12
1.2.5 XML的应用	15
1.3 XML和其他相关语言或标准的关系.....	17
1.3.1 SGML、HTML和XML之间的联系.....	17
1.3.2 XML和Java的关系.....	18
1.3.3 RDF和XML的联系	19
1.3.4 OSD和CDF与XML的关系	19
1.3.5 Netscape浏览器中如何实现XML.....	19
1.3.6 Microsoft浏览器中如何实现XML	20
1.4 学习XML的准备工作	20
1.4.1 必备知识.....	20
1.4.2 编写XML的工具.....	21
第2章 预备知识	24
2.1 HTML简介.....	24
2.2 ASP简介.....	26
2.2.1 Active Server Pages 是什么.....	26
2.2.2 Active Server Pages可做什么.....	27
2.2.3 Active Server Pages是怎样工作的.....	27
2.2.4 在Active Server Pages中结合Scripts脚本.....	29
2.3 VBScript简介	35
2.3.1 VBScript简介.....	36
2.3.2 客户端和服务端编程语言	36

2.3.3 使用VBScript.....	36
2.3.4 把Visual Basic脚本集成到Active Sever Pages中.....	37
2.3.5 声明变量.....	39
2.3.6 注释Visual Basic脚本.....	46
第3章 XML的语法	47
3.1 XML文件的组成	47
3.1.1 实例.....	47
3.1.2 XML文件的基本组成.....	48
3.1.3 XML文档的物理结构.....	53
3.2 XML文档的类型	57
3.2.1 构造良好的 XML文档.....	57
3.2.2 有效的XML文档.....	58
3.3 表现XML	59
3.3.1 CSS.....	61
3.3.2 XSL	61
第4章 使用CSS格式化XML.....	63
4.1 开始CSS文档.....	63
4.1.1 Style Sheets简介	63
4.1.2 将样式与XML结合	65
4.2 深入学习CSS.....	69
4.2.1 定义样式.....	69
4.2.2 为对象确定位置.....	72
4.2.3 层叠顺序.....	74
4.2.4 CSS格式化模型.....	75
4.3 应用实例	86
4.3.1 样式一.....	87
4.3.2 样式二.....	94
4.4 CCS的单位.....	96
4.4.1 长度单位.....	96
4.4.2 百分比单位.....	97
4.4.3 颜色单位.....	97
4.4.4 URL单位.....	98
4.5 CSS的高级应用	98
4.5.1 CSS定位.....	99
4.5.2 CSS打印支持.....	106
4.5.3 声音层叠样式表.....	107
4.5.4 CSS滤镜和渐变.....	108
4.5.5 在CSS中使用名域.....	141

4.6 演示实例	148
4.6.1 滤镜的淡出淡入效果	148
4.6.2 改变页面字体样式	150
4.6.3 艺术风格	155
4.6.4 页面的刷新	157
第5章 使用XSL	162
5.1 XSL概述	162
5.1.1 为什么要使用XSL	162
5.1.2 XSL是怎样工作的	163
5.1.3 XSL的组成	163
5.2 开始学习XSL	164
5.2.1 生成和组装一个HTML模板	164
5.2.2 书写一个格式良好的HTML	170
5.2.3 访问属性和输出属性	172
5.2.4 给XML数据排序	175
5.2.5 条件模板	176
5.2.6 XSL模式语言	182
5.3 使用XSL处理程序	198
5.3.1 使用XSL转换数据岛内容	199
5.3.2 在ASP中使用XSL	203
5.3.3 使用一个新的DOM文档得到XSL转换的结果	207
5.3.4 侦测和处理XSL错误	208
5.4 深入学习XSL	213
5.4.1 构造结果树	213
5.4.2 样式表结构	216
5.4.3 模板规则和模式	218
5.4.4 模板	222
5.4.5 联合样式表	226
5.4.6 对象格式化	227
5.5 XML查询语言XML-QL	229
5.5.1 XML-QL的结构	230
5.5.2 XML-QL的特点	233
第6章 名域	235
6.1 名域概述	235
6.1.1 为什么要使用名域	235
6.1.2 名域的宣告	239
6.1.3 名域的范畴	241
6.2 预设的名域	242

6.2.1 预设名域的概念和宣告	242
6.2.2 预设名域与范畴联合运用	244
第7章 使用DTD验证XML文档	248
7.1 DTD概述	248
7.1.1 为什么需要DTD	248
7.1.2 什么是DTD	250
7.1.3 DTD的未来	251
7.2 元素和属性类别宣告	251
7.2.1 常用量词	251
7.2.2 元素类别宣告	252
7.2.3 属性类别宣告	253
7.2.4 将DTD和XML文档连接起来	256
7.2.5 实例练习	259
第8章 XML Schema	261
8.1 XML Schema 概述	261
8.1.1 为什么需要Schema	261
8.1.2 Schema的诞生	262
8.1.3 XML Schema与DTD有什么不同	262
8.1.4 XML Schema的新特色	262
8.1.5 Schema的应用领域	263
8.2 使用Schema	263
8.2.1 Schema的框架结构	263
8.2.2 元素	264
8.2.3 名域	270
8.2.4 属性	275
8.2.5 注释 (description)	277
第9章 XML在数据库中的应用	278
9.1 在VB中的应用实例	278
9.1.1 准备工作	278
9.1.2 编写程序	281
9.1.3 调试和运行程序	284
9.2 在C++ Builder中的应用实例	288
9.2.1 应用服务程序	289
9.2.2 客户应用程序	293
第10章 XML的其他问题	309
10.1 XML的显示语言	309
10.1.1 XHTML (eXtensible Hyper Text Markup Language)	309

10.1.2	SVG.....	313
10.1.3	VML和PGML.....	324
10.1.4	SMIL、OEB和 XSL-FO	324
10.1.5	WML和HDML	325
10.2	XML的未来	330
10.2.1	支持XML的公司和它们的开发工具	330
10.2.2	XML展望.....	330
10.3	XML的专业技术词汇	332

第1章 XML 概述

在当今的IT业中，XML（Extensible Markup Language，可扩展标识语言）越来越成为一个热门的话题。我们知道，Web上的通用语言是HTML，正是由于它的出现，Internet才像现在这样如此深入人心，规模也变得异常庞大。也正因为如此以及HTML的不足，人们越发觉得HTML不够用了。1996年，W3C（万维网联盟）开始对HTML的后续语言进行研究，并于1998年2月正式推出了XML。在这一章中，首先介绍了HTML的不足，然后引出弥补这些缺陷的XML，以及它的一些特点、优势、应用领域、诞生等问题，最后介绍了XML和其他网页编写语言的关系，使读者对XML有一个总体的概念。

1.1 HTML的不足

网络技术发展至今，已渗透到社会生活的每一个角落。而令网络如此平易近人，为大众所接受，单靠TCP/IP是远远不够的，由此HTML——超文本标记语言便应运而生，并以其简单精练的语法、极易掌握的通用性与易学性，使Web网页可以接近于每一个普通人，互联网也才得以普及发展以至今日的辉煌。然而，电子商务、电子图书、远程教育等全新领域如异军突起，迅猛发展并逐渐成为互联网世界必不可少且愈发重要的组成部分，随之而来的是Web文件的复杂化、多样化、智能化，于是高容量、高信息量、高效率便成为网络信息传输技术发展的追求。与此同时，还有另一种需求变得愈发广泛而迫切，那便是同样的数据能否根据不同用户不同需求而以不同的效果、不同的形式表达再现出来。而这一切是HTML所不能满足的。本节主要介绍了HTML在今天网络时代所面临的困境，它在机器之间互相交流、传递信息等方面的不足，以及XML是如何弥补这些缺陷的。

1.1.1 HTML难以扩展

HTML是符合SGML语法的一种固定格式的超文本标记语言，因其格式固定，故难以扩展。浏览器供应商通过非标准的扩展来获得市场份额，这就破坏了Web的交互操作性。HTML中有一个大问题就是它的tag集合是固定的，用户不能新增有意义的并能供他人使用的tag。

XML的诞生正是为了解决这个问题，从其全称“可扩展标记语言”（Extensible Markup Language）就可以知道这一点。XML是SGML的一个子集，一个摒弃了其中一些复杂性并考虑到适合Web特性的一个子集。所以它和SGML一样，都是元语言，可以定义其他的语言。HTML4.0现在由SGML描述，将来可能由XML重新定义。HTML的标记是固定的，只有如H1、P、Table等在HTML标准DTD（文档类型定义）中定义的标记。而XML的标记是用户定义的，所以从理论上讲，其类型的数量可以是无限的。

1.1.2 HTML交互性差

用户需要与应用进行交互,而使用HTML却并不容易做到。频繁地请求处理已经使现在的Web服务器不堪重负了,动态HTML可以把一些处理交给浏览器,但在这方面它远不及XML。更为重要的是,当我们想对同一数据以不同的视角进行观察时,HTML的办法是请求服务器重传另一个页面,这无疑更加重了服务器的负担。而如果使用XML,我们就可以在本地排序、过滤和进行其他的数据操作。

网络浏览器是一个应用平台,以HTML作为数据标准,网络上的应用程序依赖服务器上的CGI(公共网关接口)脚本来处理网页上的数据。这样的结构是非常清晰的,但HTML却使在服务器的一端要处理的数据量过大,这使得网络速度降低,效率不高。

1.1.3 HTML语义性差

HTML最初是被设计用来标记文档以反映其结构的(如Title、H1、L1、P、Table等),同时它也包含一些用于显示格式的标记(如B、I),再加上超文本链接,HTML在Web上获得了巨大的成功。HTML对布局、外观方面很擅长,却极度缺乏对内容,也就是信息含义的表达能力,除了极少数几个用来表达内容或文义的标签,如<p>、<address>、<title>、外,几乎全都是用来设计网页格局的。所以HTML是设计来做人机交流用的。

HTML只是一种表达的技术,它并不一定能揭示HTML tag中所隐含的含义。举一个最简单的例子,<h2>Apple</h2>这句话在网络浏览器中有特定的表现,但是HTML却没有告诉我们它到底是什么。Apple只是一个英文单词罢了,它在不同的环境之下可能会有不同的意义,是一个计算机公司,一个水果,还是一个姓氏? HTML并没有告诉我们Apple具体的内容。

由于HTML没有考虑语义性,因此它至少在以下三个方面存在极大的不足。

(1) 电子数据交换(EDI)

在HTML中,产品名和产品价格可能隐藏在文档的任何地方,没有任何特别之处,计算机无法将其从文档识别出来,因此也就谈不上对其进行更深入的处理。而XML可以将产品名和产品价格用类似PRODUCT_NAME和PRODUCT_PRICE的标记将它们标记出来,这样种种不同的应用就能理解它们的含义并进行相应的处理。XML实际带来的是一场EDI技术革命,因为它彻底弥补了商业规则和数据内容与结构密不可分的传统EDI的缺陷。

(2) 数据库

要在HTML中展示数据库的数据,一般的过程是:通过某种手段从数据库取出数据,然后将它们写到用于展示的HTML文档中,加上各种标记,提交给用户。这样做的结果就是用户得到一些“死”数据。即使用户仅仅想以另外一种方式浏览数据,也只能重来一次。而XML提供了极大的灵活性,我们可以把数据库中的每个字段和XML中的相应标记对应起来,如此带来的好处是不言而喻的。

(3) 搜索引擎

相信每个使用过搜索引擎的人都有过同样的体会,返回的结果太多,你需要足够的耐心和时间去检查哪些是你需要的结果。问题在于现在的搜索引擎只能检索单词的频率、文

档的标题以及某些情形下描述页面内容的元标记等，这是HTML的缺陷带来的限制。而在XML中，像〈AUTHOR〉这样的标记很明确地表达了它包含的意义，如果我们查找张某某写的书，而张某某又是被〈AUTHOR〉所标记，搜索引擎就可以很清楚地知道我们表达的语义而返回正确的结果。

下面以一个例子来说明HTML在语义方面的不足和XML在这方面的优势。HTML段代码图书信息以表格的方式排列：

```
<HTML>
<H1><center>图书系列</H1>

<TABLEborder="1"cellpadding="5">
  <TR>
    <TH>名称</TH>
    <TH>作者</TH>
    <TH>出版社</TH>
    <TH>价格</TH>
  </TR>
  <TR>
    <TH>学习XML</TH>
    <TH>贺小飞</TH>
    <TH>机械工业出版社</TH>
    <TH>30.00</TH>
  </TR>
  <TR>
    <TH>数字地球导论</TH>
    <TH>周文其</TH>
    <TH>科学出版社</TH>
    <TH>35.00</TH>
  </TR>
  <TR>
    <TH>中国土壤系统分类</TH>
    <TH>陈文名</TH>
    <TH>科技出版社</TH>
    <TH>28.00</TH>
  </TR>
  <TR>
    <TH>C++Builder4.0多媒体编程</TH>
    <TH>雷大侠</TH>
    <TH>人民邮电出版社</TH>
    <TH>48.00</TH>
  </TR>
</TABLE>
</HTML>
```

在浏览器中显示结果如图1-1所示。

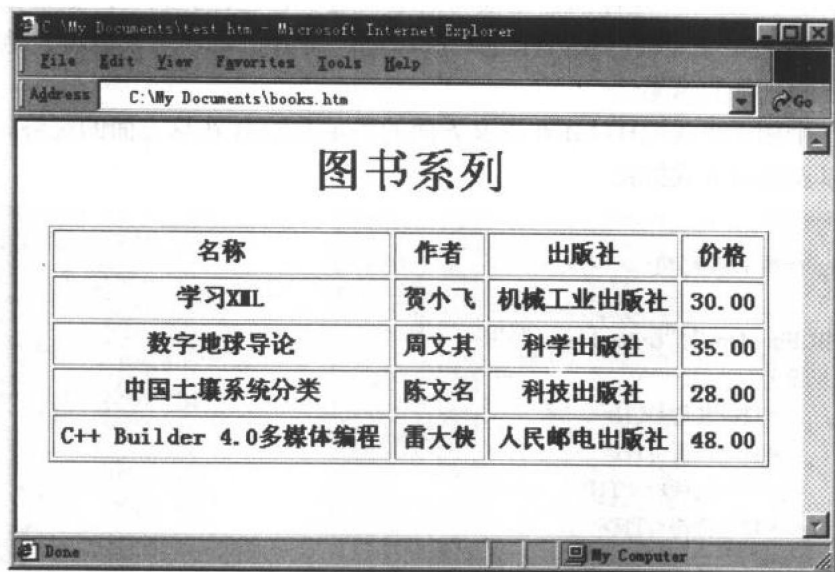


图 1-1 以 HTML 表格显示的图书信息

在上面的例子中，HTML存在的问题是它的标签大多是设计来呈现文章格局和外观的，比如<table>、<tr>、<td>等等到处都是，语义不强。XML和HTML的一大不同之处，就在于在XML里可以自由地定义标签，定义出来的标签可以按照自己的意思充分地表达文件的内容，比如我们可以定义<name>、<author>、<title>等这样充分表达我们意思的标签。在XML中，我们只注重内容，这和HTML强调布局的做法是截然不同的，至于XML文件的外观显示，则可以通过和CSS或XSL结合起来来做XML到HTML格式或其他格式的转换，这点在后面详细介绍。这里举例说明XML强调语义：

```
<?xmlversion="1.0"?>
<!-- 此XML文档包含了图书的信息-->
<books>
  <book>
    <title>C++ Builder program</title>
    <author>John York</author>
    <publisher>Science Publish house</publisher>
    <money>30.00</money>
  </book>
  <book>
    <title>C++ Builder4.0</title>
    <author>Mary</author>
    <publisher>Science Publish house</publisher>
    <money>80.00</money>
  </book>
</books>
```

```
<book>
  <title>Visual C++ Program</title>
  <author>Park</author>
  <publisher>Science Publish house</publisher>
  <money>45.0</money>
</book>
</books>
```

将此文件保存为books.xml，然后在浏览器中打开，显示如图1-2所示。

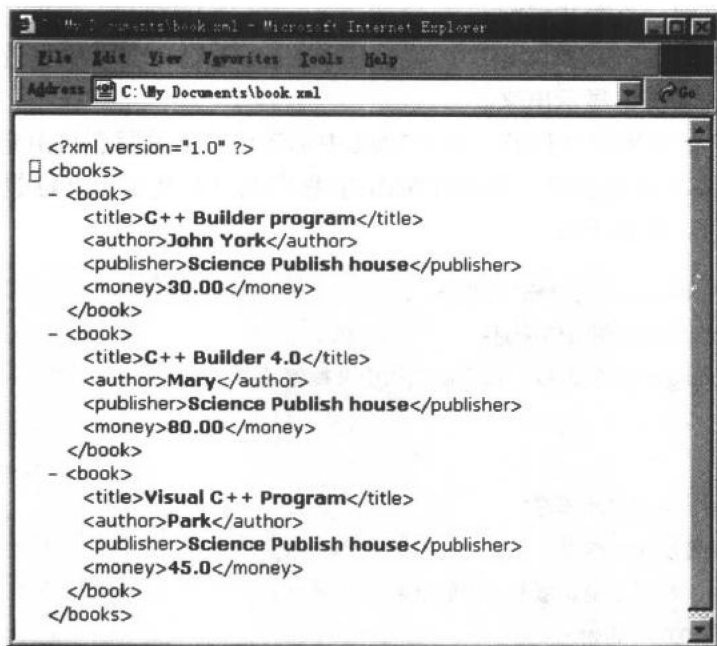


图 1-2 以 XML 显示的图书信息

1.1.4 单向的超链接

虽然现在HTML提供的超文本链接机制被证明是很有用的并取得了巨大的成功，但这种机制的如下特征注定了它的局限性：

- (1) 超链接在它的某一端(源端)定义。
- (2) 超链接确定了它的另一端(目的端)，虽然服务器可以有很大的自由度来寻找和创建它的目的端。
- (3) 用户只能从源端出发走到目的端。
- (4) 超链接的效果由浏览器而不是由超链接本身来决定。举例来说，当用户打开一个链接时，一般的情形是用目的端的文档来取代源端的文档，浏览器可能提供一个选择让用户在一个新窗口中打开。

实际上，在早些的若干专用超媒体系统中已经有一些更灵活的机制来表达更复杂的链接关系，如多个目的端、多向链接等，XML可以满足这样的需求。

1.1.5 HTML 缺乏对双字节或多国文字的支持

HTML 缺乏对双字节或多国文字的支持,或者说支持不够。例如,中文信息页面在不同的平台下会出现格式不齐等问题。

XML 将使 Web 建立者创建真正的国际性站点变得比以前更容易。比如 Java, 它由 Unicode(ISO10646) 定义, Unicode 是一个国际接受的标准, 用于描述世界上所有的字母、字型、字符和表意符, 它包括 ASCII ISO 拉丁字符、日本、韩国、中国、印度、希腊、阿拉伯和其他字符, 它甚至允许字符集的混合, 例如一篇以日文显示的 XML 文档可以通过一个元音变音参照到一个德文词汇。

开发者不需要学习任何脚本语言, 就能使 Unicode 在 XML 文档中生效, 文档在用户的浏览器中用合适的字符集显示出来。

在 HTML 中不能使用中文标签, 而在 XML 中标签可以由编写者自由定义, 定义出的标签可以充分反映编写者的意思, 因此在 XML 中我们就可以使用中文标签, 这样就能充分表达我们的意思。参见下例。

```
<?xml version="1.0" encoding="gb2312"?>
<!-- 此XML文档包含了图书的信息-->
<!-- 加入encoding="gb2312"后, 就可以使用中文标签了-->
<图书信息>
  <书>
    <书名>学习XML</书名>
    <作者>贺小飞</作者>
    <出版社>机械工业出版社</出版社>
    <定价>40.00</定价>
  </书>

  <书>
    <书名>数字地球导论</书名>
    <作者>周文其</作者>
    <出版社>科学出版社</出版社>
    <定价>48.00</定价>
  </书>

  <书>
    <书名>中国土壤系统分类</书名>
    <作者>陈文明</作者>
    <出版社>铁道出版社</出版社>
    <定价>53.50</定价>
  </书>
</图书信息>
```


将此文件保存为books.xml，在浏览器中显示如图1-3所示。

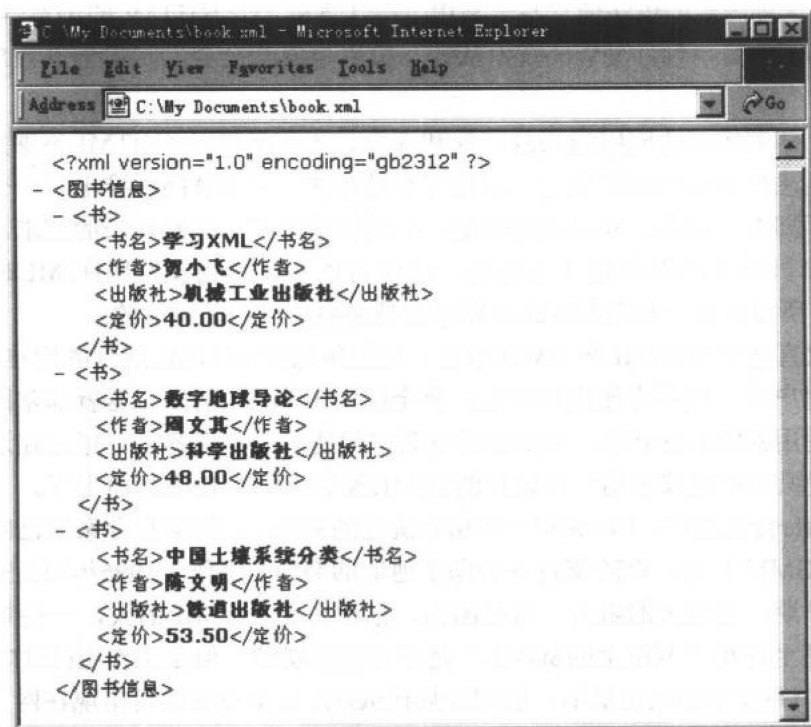


图 1-3 使用中文标签

1.2 XML的诞生

正是由于HTML存在前一节中所说的缺点，人们已经开始研究能改进或替代HTML的Web页面制作语言。其中最有成效的，并已初步投入使用的有：可扩展的标记语言（Extensible Markup Language），简称XML；叠式页面（Cascading Style Sheets），简称CSS；以及动态HTML，即DHTML。这些语言从不同角度解决了HTML存在的问题：XML有利于信息的表达和结构化组织，从而使数据搜索更有效；CSS解决Web页面的继承和显示问题；DHTML则主要用于Web页的动态显示。

本节介绍XML产生的背景，以及它的优越性与应用。

1.2.1 XML产生的背景

XML同HTML一样，都来自Standard Generalized Markup Language（标准通用标记语言）简称SGML。早在Web未发明之前，SGML就已存在。顾名思义，SGML是一种用标记来描述文档资料的通用语言，它包含了一系列的文档类型定义，简称DTD，DTD中定义了标记的含义，因而SGML的语法是可以扩展的。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。鉴于这些原因，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)计算机技术的能力，提出了HTML语言。

HTML只使用SGML中很小一部分标记，例如HTML3.2定义了70种标记。为了便于在