

高等院校信息技术规划教材

XML应用教程

吴洁 编著

INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY
INFORMATION TECHNOLOGY

清华大学出版社



高等院校信息技术规划教材

XML应用教程

吴 洁 编著

清华大学出版社
北京

内 容 简 介

本书在形象而轻松的语言环境中讲述可扩展标记语言(XML)的原理、结构和相关技术,通过大量深入浅出的案例,全面介绍XML的基础知识,帮助读者了解XML如何使数据内容结构化从而应用于计算机网络、电子商务等各个领域。

在XML基本知识的基础上,本书重点介绍了XML的应用和数据接口,如XML与数据库,XML与SOAP,XML与DOM,以及基于XML的Visual Basic、ASP和.NET等应用的开发,使读者掌握基于XML的数据交换和编程技术,了解Web中的最新发展动态、应用趋势和集成化的编程风格,领略XML在跨平台、跨应用程序及跨编码语言的未来分布式解决方案中的作用。

本书可作为高校计算机相关课程的教材,而且对计算机的研究人员亦有重要的参考价值。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

XML应用教程/吴洁编著. —北京:清华大学出版社,2005.12

(高等院校信息技术规划教材)

ISBN 7-302-11363-7

I. X… II. 吴… 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(2005)第077534号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

组稿编辑: 张 龙

文稿编辑: 张为民

印 刷 者: 北京四季青印刷厂

装 订 者: 三河市金元装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 21.5 字数: 500千字

版 次: 2005年12月第1版 2005年12月第1次印刷

书 号: ISBN 7-302-11363-7/TP·7479

印 数: 1~3000

定 价: 27.00元

序

preface

在科教兴国方针的指引下,我国高等教育进入了一个新的历史发展时期,招生规模和在校生数量都有了大幅度的增长。我们在进行着世界上规模最大的高等教育。与此同时,对于高等教育的研究和认识也在不断深化。高等学校要明确自己的办学方向和办学特色,这既是不断提高高等教育水平的必然要求,更是高校不断发展和壮大必须首先考虑的问题。

教育部领导明确提出要有相当部分的高校致力于培养应用型人才,此类院校在计算机教学中如何实现自己的培养目标,如何选择适用的应用型教材,已成为十分重要和迫切的任务。应用型人才的培养不能简单照搬研究型人才的培养方案,要在丰富的实践基础上认真总结,摸索新形势下的教学规律,在此基础上设计相关课程、改进教学方法,同时编写应用型教材。这一工作是非常艰巨的,也是非常有意义的。

在清华大学出版社的大力支持和配合下,于2003年成立了应用型教材编委会。编委会汇集了众多高校的实践经验,并经过集中讨论和专家评审,遴选了一批优秀教材,希望能够通过这套教材的出版和使用,促进应用型人才培养的实践发展,为建立新的人才培养模式作出贡献。

我们编写应用型教材的主要出发点是:

1. 适应教育部对高等教育的新要求,以及市场对应用型人才需求量的不断增加。
2. 计算机科学技术不断更新,发展速度加速,教材内容和教学方式将适时更新和改进。
3. 教育技术的发展,对教材建设提出了更高的要求,教材将呈现出纸介质出版物、电子课件以及网络学习环境等相互配合的立体化形态。

4. 突出应用,增强实训,根据不同的专业要求,加强针对性,使理论与实践紧密结合。

从上述各点出发,我们将努力建设一套全新的、有实用价值的应用型计算机教材。经过参编教师的努力,第一批教材已经面世。教材将滚动式地不断更新、修正、提高,逐渐树立起自己的品牌。希望使用本系列教材的广大师生不断反馈各类意见,逐步建设具有应用型特色的精品教材。

李文忠

2004年9月

前言

Foreword

高 昊

东南大学计算机学院

可扩展标记语言(XML)由万维网协会(W3C)推出的新一代数据交换的标准,用于定义 Web 网页上的文档元素和商业文档。XML 在 Internet 中的地位已经确立,世界主流平台迅速在其最新的版本中加入 XML 的编译器,竞相与之靠拢。业界对 XML 的研究和应用正在兴起,并在 Internet 上迅速发展。

XML 最大的特点是将数据结构化,实现数据共享。作为标记语言,XML 又是一种元语言。用户可用来创建自己需要的其他标记语言,如出版社可用来定义书籍标记语言,化学家则用以定义化学标记语言(CML)等。这使得 XML 的应用能迅速渗透到各个领域,对 XML 的学习正在全面展开。

目前国内有关 XML 的书籍开始热销,但作者注意到其大部分是面向专业人群的,技术上有所侧重,而适合本专科生以上(或有兴趣读者)系统了解和学习 XML 有关知识的教材还不多见。

本书将从底层构造到与各类平台的接口,从各类 XML 的应用到相关技术,向读者推出一本既有理论性,又侧重案例的较为全面的一本读物或是教材。

我们希望在一种形象而轻松的语言环境中向读者讲述 XML 的原理、结构和相关技术,通过大量深入浅出的案例,全面而通俗地介绍 XML 的基础知识,帮助读者了解 XML 如何使数据内容结构化,从而应用于计算机网络、电子商务的各个领域。

XML 的应用是读者更关注的内容,本书在展示 XML 基本知识的基础上,重点介绍了 XML 的应用和数据接口,如 XML 与数据库,XML 与 SOAP,XML 与 DOM,以及基于 XML 的 Visual Basic、ASP 和 .NET 等应用的开发,使读者掌握基于 XML 的数据交换和编程技术,了解 Web 中的最新发展动态、应用趋势和集成化的编程风格,领略 XML 在跨平台、跨应用程序及跨语言的未来分布式解决

方案中的作用。

本书不仅可被相关课程的学生作为教材使用,而且对从事计算机和其他与计算机相关的研究人员亦有重要的参考价值。

本书由吴洁编著,吕云屏、李鹏飞参与了部分章节的编写和整理工作。由于水平有限,加之时间仓促,缺点和错误在所难免,恳请专家和广大读者批评指正。

吴 洁

2005 年 5 月于南京

目录

Contents

第 1 章 XML 概述	1
1.1 HTML 的局限性	1
1.2 从 SGML 到 XML 的产生	2
1.3 XML 的定义和特点	3
1.4 一个简单的 XML 案例	3
1.5 XML 文档的编写	4
1.5.1 XML 标记的编写	4
1.5.2 XML 的规则	4
1.5.3 注释的添加	8
1.5.4 五个特殊的符号	8
1.5.5 CDATA 段	9
1.5.6 工具的使用	10
第 2 章 XML 数据的底层结构——DTD 和 XML Schema	11
2.1 XML 文档类型定义文件 DTD	11
2.1.1 DTD 的编写语法	13
2.1.2 XML 文档和 DTD 的匹配	26
2.2 XML 模式定义文件 Schema	36
2.2.1 Schema 与 DTD 的区别和优势	36
2.2.2 局部和全局定义	36
2.2.3 Schema 的基本结构和使用	36
2.2.4 简单类型	37
2.2.5 复杂类型	45
第 3 章 XML 文件的设计	55
3.1 XML 文档与其结构的匹配	55
3.1.1 “濒危动物”案例	55

3.1.2 书籍借阅信息案例	63
3.2 命名空间	65
3.2.1 命名空间的概念	65
3.2.2 如何声明命名空间	66
3.2.3 如何使用命名空间	67
3.3 XML 文档的有效性验证	73
3.3.1 格式良好的 XML 文档	73
3.3.2 Schema 验证 XML 文档结构正确性	78
第 4 章 XML 的样式单 XSLT	84
4.1 XSLT 概述	84
4.2 XSLT 样式表的创建和输出	85
4.3 模板的创建与应用	89
4.4 批量处理	91
4.5 各节点的条件处理	93
4.6 节点的排序	98
4.7 语法注意点	101
第 5 章 XML 的编译、编辑环境 and 设计工具的运用	102
5.1 几种流行的 XML 编译器	102
5.1.1 Java™ XML Pack	102
5.1.2 JAXB 与 XML 绑定的 Java 构架	102
5.1.3 Microsoft XML Parser	103
5.2 XMLSPY	103
5.2.1 用户界面	103
5.2.2 创建和编辑 Schema	104
5.2.3 创建和编辑 XML	105
5.2.4 有效性检验	106
5.2.5 XMLSPY 与各数据平台的接口	107
5.3 样式单的设计工具 Stylesheet Designer	107
5.3.1 用户界面和图标	107
5.3.2 设计步骤	109
5.3.3 书签和超链接的插入	110
5.3.4 XML 元素在动态表格中的生成	114
第 6 章 层叠样式表 CSS	118
6.1 CSS 的内容	118

6.1.1	CSS 介绍	118
6.1.2	CSS 可以分别与 XML 和 XHTML 结合	119
6.1.3	在 XML 文件上直接应用 CSS	119
6.1.4	在 XML 上使用 CSS 和 XSLT	123
6.1.5	在 XSLT 样式表中设置样式	125
6.2	用 CSS 设计页面	128
6.2.1	定义元素	128
6.2.2	元素定位	134
6.2.3	设置元素大小	136
6.2.4	边框、空格填充和边距	138
6.2.5	环绕文本	141
6.2.6	图片	142
6.3	用 CSS 样式化文本	148
6.3.1	字体	148
6.3.2	文本特性	149
第 7 章	XML 与数据库	150
7.1	数据存储	150
7.1.1	关系数据库中信息的存储与查询	150
7.1.2	XML 数据中信息的存储与查询	152
7.2	XML 与数据库的转换	155
7.2.1	数据库到 XML 文档的信息传送	155
7.2.2	XML 文档到数据库的信息传送	159
7.3	SQL Server 对 XML 的支持	163
7.3.1	从 SQL Server 数据库中提取 XML 数据	164
7.3.2	XML 数据插入到数据库	172
7.4	Oracle 对 XML 的支持	173
第 8 章	简单对象访问协议	175
8.1	SOAP 的解决方案概述	175
8.2	SOAP 的框架	176
8.2.1	SOAP 的头元素 Header	177
8.2.2	SOAP 的体元素 Body	178
8.2.3	SOAP 的 Fault 元素	179
8.2.4	SOAP 消息的 XML 模式文件	180
8.3	HTTP 头和 SOAP	181
8.3.1	请求头	182



8.3.2	响应头	183
8.3.3	发送 SOAP 响应和请求消息的信封元素的区别	186
8.3.4	用于 SOAP 消息的 Body 内容的模式 Schema	187
第 9 章	XML 编程技术	189
9.1	XML 数据接口 DOM	189
9.1.1	DOM 接口中的主要对象	190
9.1.2	XML 文档的创建和加载	194
9.1.3	XML 文档的遍历	195
9.1.4	XML 元素的添加、删除和内容修改	196
9.2	XML 与 ASP 结合的实例	197
9.2.1	ASP 的概念	197
9.2.2	ASP 中应用 XML	197
9.3	XML 与 .NET	203
9.3.1	关于 .NET	203
9.3.2	.NET 框架	203
9.3.3	.NET 框架对 XML 文档的支持	204
9.3.4	.NET 框架的 XMLDOM 分析器	205
9.3.5	.NET 框架的 XML 阅读器	207
9.3.6	.NET 框架中的 XML 数据管理	220
9.3.7	XML 的串行化	232
9.3.8	XSLT 转换	241
第 10 章	一个典型的案例	252
10.1	需求分析	252
10.1.1	功能设计	252
10.1.2	设计思想	252
10.2	程序架构	253
10.3	数据架构	254
10.3.1	源文件的建立	254
10.3.2	数据库设计	257
10.4	子程序的实现	257
10.4.1	浏览	257
10.4.2	回复	261
10.4.3	发布	263
10.4.4	删除	266
10.4.5	归档	266

10.4.6	管理员登录	273
10.4.7	Web 服务	274
10.5	完整的程序清单	275
10.5.1	程序清单 1	275
10.5.2	程序清单 2	291
10.5.3	程序清单 3	293
10.5.4	程序清单 4	294
10.5.5	程序清单 5	296
10.5.6	程序清单 6	297
10.5.7	程序清单 7	298
10.5.8	程序清单 8	300
10.5.9	程序清单 9	300
10.5.10	程序清单 10	301
10.5.11	程序清单 11	303
10.5.12	程序清单 12	305
10.5.13	程序清单 13	307
10.5.14	程序清单 14	308
10.5.15	程序清单 15	310
10.5.16	程序清单 16	313
10.5.17	程序清单 17	314
10.5.18	程序清单 18	317
10.5.19	程序清单 19	318
10.5.20	程序清单 20	320
附录 A	XML Schema 中的简单数据类型	323

XML 概述

可扩展标记语言(eXtensible Markup Language, XML)在 Internet 中的地位已经确立,对 XML 的研究和应用正在兴起并在 Internet 时代背景下迅速发展。

HTML 作为 Internet 上传统的描述语言,随着网络应用的深入而渐显疲态,其种种缺陷如难以扩展、缺乏交互性、缺乏语义性定义等,都促使一个新的标记语言的诞生。

本章介绍了 HTML 的局限性,XML 的产生和特点,以及 XML 文档的编写方法。

1.1 HTML 的局限性

HTML 取得的巨大成功有目共睹,但随着时间的推移和 Web 应用需求的增长,HTML 的局限性逐渐显现,表现在:

(1) HTML 缺乏语法检查。HTML 文档的结构具有规范化的定义,每一个 HTML 版本有一个文档类型定义(Document Type Definition, DTD)。也有一些开发工具和 Web 站点可用于检查 HTML 文档的语法。然而,现有浏览器可以接受任何看上去类似 HTML 的文本。

(2) HTML 缺乏结构。HTML 具有有序的头标签,在 DIV 标签中也可以嵌套信息块。问题是开发人员通常并不把结构置于他们的 HTML 文档中。

(3) HTML 对内容不敏感。HTML 中的标签与其间的内容没有任何关联。

(4) HTML 不是面向对象的。HTML 落后于程序设计语言向面向对象方法转换的步伐。

(5) HTML 缺乏健壮的链接机制。HTML 常常会遇到网页链接的中断的问题。链接是被手工编码到 HTML 中的,如果一个目标文件改变了地址的话,其链接也必须手工进行修改。

(6) HTML 是不可重用的。HTML 代码片段是可以重用的,样式单也可以重用。然而,从程序设计意义上来说,它不具备可重用性。

(7) HTML 是不可扩展的。

作为分析 HTML 优缺点的例子,考虑如下代码:

```
<body>
<center><img src=pix/door1.jpg></center>
<hr width=50% size=5 noshade>
Open the door to your future.
<P>Learn about rewarding careers
in door painting.
```

上述代码混合使用了标签的大小写形式,标签的值可以放在引号中,也可以不放在引号中,<html>标签和<body>的关闭标签都缺失了,然而该页仍然能被显示。这种能力既是一个优点又是一个缺点。浏览器需要对标签应是什么作出最合理的猜测。但由于 HTML 代码是非标准化和结构化的,所以它又不能用于任何其他应用的处理过程。

随着 Internet 应用的深入,HTML 的这些局限性越发显现出来。人们急需一种能够结构化地表示信息并支持网络跨平台应用和数据交换的标准网络语言,来满足各种应用和通信的需要。XML 就在这样的背景下诞生了。

1.2 从 SGML 到 XML 的产生

HTML 源于标准通用标记语言(Standard Generalized Markup Language, SGML)。SGML 是一种元语言,即描述标记语言的语言。HTML 是其中的一种,被称为一种“SGML 应用”。

“Markup”称为标记,又称为标记符或标签等。通过 SGML 声明的方式,SGML 应用可以指定哪些字符可被解释为数据,哪些应被解释为标签的部分。就像 HTML 中的标签一样,习惯于用字符“<”和“>”来标明那些“Markup”。它们也可以被设置为“{”和“}”。DTD 是 SGML 应用的核心,在 DTD 中定义了信息的结构,即标记元素的名称和关系,实际的信息就根据 DTD 的结构而相应地被组织成标记文档。DTD 的设计须仔细进行,如果它被修改,则已经标记的文档将会无效。

SGML 可用来建立任何类型的标记语言,这使得它功能非常强大。但是它的设计周期较长,同时也非常复杂。于是从 SGML 演化出各种各样的其他标记语言。XML 也由此从 SGML 中衍生出来。

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,还可用于如数学、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,使它在未来分布式解决方案中发挥重要作用。

1.3 XML 的定义和特点

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

- (1) XML 可用于现有的 Web 协议(如 HTTP 和 MIME)和机制(如 URL)。
- (2) 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,它需要借助使用用于链接 HTML 和应用的脚本语言才支持 Web 的各种不同的应用。
- (3) XML 与 SGML 兼容,所以大多数 SGML 应用可以向 XML 转化。
- (4) 与 HTML 文档一样,XML 文档易于创建。
- (5) 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。
- (6) 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。
- (7) XML 标准定义精练,这就保证了其下载和处理的速度。

1.4 一个简单的 XML 案例

下面来看一个简单的 XML 文档,有助于对 XML 产生一个初步的印象。

```
<? xml version="1.0"?>
<contacts>
  <contact>
    <name>
      <first>David</first>
      <last>Calder</last>
    </name>
    <address>
      <street>56 McMahons Rd</street>
      <city>Frankston</city>
      <state>Vic</state>
      <zip>3199</zip>
    </address>
    <tel>9903 4567</tel>
    <fax>9903 4444</fax>
    <mobile>040 8887 7777</mobile>
    <email>david.calder@nowhere.com.au</email>
  </contact>
</contacts>
```

如上所示,XML 文件是由标签组成的 Mark-up 文件,XML 是针对信息内容,由开发

人员自行定义标签(以“<”开头和“>”结束),用来组织信息的一种置标语言(Mark-up Language)。

可以看出,首先,XML 充分适应用户的需要,由开发人员决定需要多少标签,如何称呼,以及它们将表示什么。其次,如果说 HTML 中包括的标签仅仅表示内容如何显示,而不表示显示什么的话,那么 XML 正好相反,在 XML 各标签的头尾之间可以包含各种丰富的信息,但不表示它们如何显示。

1.5 XML 文档的编写

1.5.1 XML 标记的编写

XML 是一种可扩展标记语言,以结构化的方式描述各种类型的数据。XML 的基本语法和 HTML 很相似,但 HTML 只允许文档制作人员使用固定的标签,而 XML 允许文档制作人员通过创建新标签,来更准确地描述数据。

下面分别是一个 HTML 和一个 XML 语句的例子。

例 1.1

```
<a href="newpage.html">Click here</a>
<person species="human">Michael Calder</person>
```

可以看到 XML 与 HTML 的结构基本相同,均可分为:元素、属性、值。

XML 元素用一个标签开始,并用一个相应的标签结束。标签是由尖括号来界定——以小于号(<)起始、大于号(>)结尾。作为标签的名称,在例 1.1 中为 person,应该具有一定的含义,能够描述该标签的内容。另外,XML 是区分字母大小写的,也就是说 person 与 Person 是不同的。

XML 标签中可以包含属性,如上例中的 species,并且给它赋值 human。属性是对元素信息的补充,在后面将对属性进行详细介绍。

1.5.2 XML 的规则

例 1.2 是一个简单的 XML 文档,它描述的是 Email 的信息。

例 1.2

```
1: <?xml version="1.0"?>
2: <message>
3:   <header>
4:     <date>25th December 2010</date>
5:     <from>Me</from>
6:     <to>You</to>
7:     <subject>Happy Christmas</subject>
8:     <banner> </banner>
9:   </header>
```

```

10:    <body>Best wishes for Christmas </body>
11:    <signature>Ho Ho Ho</signature>
12:    <footer>Brought to you by Santa</footer>
13:</message>

```

1. XML 声明

例 1.2 的 XML 文档第 1 行为 XML 声明:

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

声明接下来的内容为 XML,指出 XML 的版本号为 1.0(一些 XML 文档因为向后兼容的原因省略这个版本号,但是除非特殊的原因,一般必须包含版本的定义)。实际上,除了 1.0 之外,XML 没有其他版本,这仅仅是为了将来进行修改。

在声明中还可以指出该 XML 文档是独立的(例 1.2 即为一个独立的 XML 文档),还是与其他文档相关联的。

如果 XML 文档不独立,那么声明要写成如下形式:

```
<? xml version="1.0" standalone="no"?>
```

注意到这里的标签用“<?”开始,用“?”结束,称之为处理指令。除了上面看到的处理指令外,还可以在 XML 文档中包含其他的处理指令,比如用来指定 XML 文档样式的样式表(样式表将在后面的章节中介绍)。

下面的例 1.3 包含了较复杂一些的处理指令,并在其中嵌入了一些 JavaScript 和 Perl 脚本代码。XML 处理器忽略处理指令,只有在将 XML 文档用到应用程序中去时,它们才会被应用程序加以处理。不过,处理指令的名称(如 javascript 和 perl)必须符合 XML 命名规范。

例 1.3

```

<? xml version="1.0"?>
<home.page>
  <para>This is text containing two processing instructions,
    <? javascript I can put whatever I like here?>
    <? perl And I can put whatever I like here too?>
    one for each interpreter.</para>
</home.page>

```

2. 根元素

每个 XML 文档都有一个且只能有一个根元素,而其他元素则都要包含在这个根元素中。在例 1.2 中,根元素为 message,文档第 2 行是根元素的开始,第 13 行是根元素的结束。根元素中可以包含空元素。需要注意的是,开始标签与结束标签中的名称必须完全一致,包括字母的大小写。只有注释和处理指令可以不包含在根元素中,如 XML 声明。

3. 元素

表 1-1 列出了 XML 元素标签的组成部分。