

21世纪高等院校网络工程规划教材

21st Century University Planned Textbooks of Network Engineering



XML

网页设计实用教程

XML Web Design

蔡体健 廖志芳 汤文亮 莫佳 编著

- 参照W3C推荐标准
- 选择XML经典案例
- 展示技术优势与特点

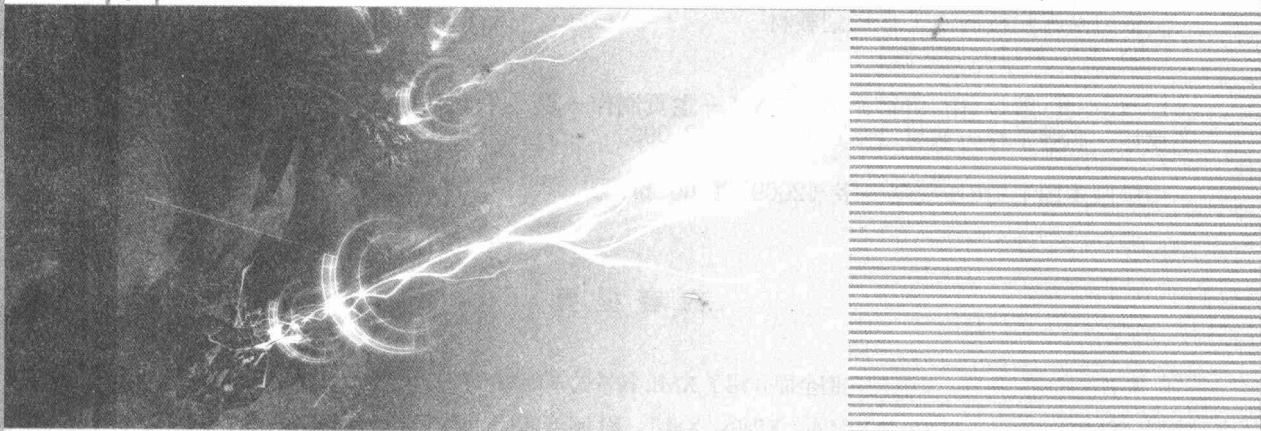


人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等院校网络工程规划教材

21st Century University Planned Textbooks of Network Engineering

TP312XM-43
724



XML

网页设计实用教程



XML Web Design

蔡体健 廖志芳 汤文亮 莫佳 编著

人民邮电出版社

北京

图书在版编目 (CIP) 数据

XML网页设计实用教程 / 蔡体健等编著. —北京: 人民邮电出版社, 2009. 3
21世纪高等院校网络工程规划教材
ISBN 978-7-115-19269-1

I. X… II. 蔡… III. 可扩展语言, XML—主页制作—程序设计—高等学校—教材 IV. TP312 TP393.092

中国版本图书馆CIP数据核字 (2009) 第002173号

内 容 提 要

本书以详实的内容、丰富的实例全面介绍了 XML 技术的基础知识和应用编程。内容包括 XML 基本语法、DTD、XML Schema、XML 命名空间、XPath、XSLT、XLink、XPointer、XQuery、DOM、SAX 等, 以及相关的编程技术, 包括 Java、JavaScript、数据岛、JSP、AJAX 等。本书参照 W3C 的最新标准, 同时兼顾当前浏览器的实际支持能力, 所有例题都经过上机调试。

本书可作为高等院校计算机科学与技术、网络工程、电子商务等专业“XML 技术基础”课程的教材, 也可供成人教育学校和培训班作为教材使用。

21 世纪高等院校网络工程规划教材

XML 网页设计实用教程

-
- ◆ 编 著 蔡体健 廖志芳 汤文亮 莫 佳
责任编辑 滑 玉
执行编辑 张 鑫
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京昌平百善印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 15.75
字数: 394 千字 2009 年 3 月第 1 版
印数: 1—3 000 册 2009 年 3 月北京第 1 次印刷

ISBN 978-7-115-19269-1/TP

定价: 27.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前 言

XML 是一种自描述的语义/结构化的标记语言,与 HTML 等标记语言在形式上类似。XML 和 HTML 是为不同的目的而设计的:HTML 被设计用来显示数据,而 XML 被设计用来描述数据结构和语义。XML 具有自描述性、保值性、内容和显示相分离、可扩展、独立于平台等特点。

XML 可以用于存储数据、交换数据、共享数据、分离数据,还可用于创建新的语言。XML 可以使计算机数据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,从而大大提高了用户处理数据的效率和灵活性。

自从 1998 年 2 月 XML 成为 W3C 发布的一种推荐标准以来,XML 得到快速发展,技术不断成熟,应用越来越广泛,越来越多的软件和开发商都采用了 XML 标准。XML 已成为 Internet 的“世界语”。

目前各高等院校计算机相关专业都开设了 XML 技术基础课程,对 XML 教材的需求和要求也在不断增加。由于 XML 技术还在不断发展过程中,因此需要不断推出新的、实用性强的 XML 教材。作者从事多年的 XML 课程教学,收集、整理了一些经典例题,在此将多年来对 XML 技术的探索心得与大家共享。

本书内容可分成 3 部分:XML 基础知识、XML 应用编程和实验。其中,第 1~6 章属于 XML 基础知识部分;第 7~9 章属于 XML 应用编程部分;附录是实验部分。XML 基础知识部分主要介绍了 XML 基本语法、格式良好的 XML 文档、有效的 XML 文档、文档类型定义 DTD、XML 验证技术 Schema、XML 的显示技术和 XML 的高级语法等内容。XML 应用编程部分介绍了 XML 编程接口:DOM 接口、SAX 接口以及数据岛技术,最后提供了一个综合实例:学生信息管理系统,详细介绍了 XML 在 Web 环境中的开发应用。附录提供了 8 个实验,以巩固学生所学知识,同时可供课堂练习使用。

本书可作为高等院校计算机科学与技术、网络工程、电子商务等专业“XML 网页设计”课程的教材,也可供成人教育学校和培训班作为教材使用。

本书具有以下特点。

- 知识体系完整。本书较全面地介绍了“XML 技术基础”课程的主要内容,包括 XML 基本语法、DTD、XML Schema、XML 命名空间、XPath、XSLT、XLink、XPointer、XQuery、DOM、SAX 等,并介绍了相关的编程技术,如 Java、JavaScript、数据岛、JSP、AJAX 等,知识覆盖面广。

- 示例典型。作者在多年的 XML 教学中收集、整理了大量例题,本书精选了部分典型例题,以方便读者理解 XML 技术。

- 教学资料丰富。为了方便教师的教学和学生的学习,本书每章前面都有学习要点,每章的后面都有小结,并提供了可以实际练习的习题。在附录中有本书配套的实验指导。

本书推荐学时数是 48 课时。在学完本课程后,建议再开设一到两周的 XML 应用编程课程设计,以加强学生的 XML 应用程序开发能力。具体的课时分配见下表。

课 时 分 配 表

周	节	教 学 内 容	实 验 内 容
一	1~2	XML 概述、XML 知识体系结构	
二	3~4、5~6	XML 的语法规则、XML 的文档组成	结构良好的 XML 文档
三	7~8	XML 的有效性、DTD 的元素声明和属性声明	
四	9~10、11~12	一般实体、参数实体以及 INCLUDE 与 IGNORE 条件语句	XML 文档及其 DTD
五	13~14	XML Schema 简介及 W3C XML Schema 数据类型	
六	15~16、17~18	W3C XML Schema 元素与属性的定义	XML 文档及其 Schema
七	19~20	可扩展样式单 XSL 的模板	
八	21~22、23~24	可扩展样式单 XSL 语法	为 XML 文档编写可扩展样式单 XSL
九	25~26	XML 高级语法: XPath	
十	27~28	XML 高级语法: XQuery、XLink、XPointer 介绍	
十一	29~30、31~32	XML 的 DOM 编程	DOM 编程实验
十二	33~34	XML 的 SAX 编程	
十三	35~36、37~38	DOM 与 SAX 的结合, XML 与数据库的数据交换	SAX 编程实验
十四	39~40、41~42	XML 数据岛	XML 数据岛实验
十五	43~44	JSP 基础	
十六	45~46、47~48	基于 JSP 的 XML 应用程序	XML 网页编程实验

本书由蔡体健主编。各章编写分工如下:蔡体健编写第 1、3、4、5、6、8、9 章;汤文亮编写第 2 章;莫佳编写第 7 章;廖志芳负责统稿和审读工作。

参与本书编写的还有樊晓平、丁振凡、郑轶等,在此表示感谢。由于编者水平有限,加之编写时间仓促,书中难免出现错误和不妥之处,敬请读者批评指正。作者的 E-mail: lao_cai68@126.com。

另外,书中例题的源代码、习题答案等可以到人民邮电出版社教学服务与资源网下载,网址为: <http://www.ptpedu.com.cn>; 或在 <http://cai.ecjtu.jx.cn> 网站以 guest 用户名登录下载。

编者

2008 年 11 月

目 录

第 1 章 XML 概述1	
1.1 XML 简介.....1	
1.1.1 XML 的发展历史.....1	
1.1.2 XML 与 HTML 的比较.....2	
1.1.3 XML 简介.....4	
1.2 XML 的用途.....5	
1.3 XML 的技术规范.....7	
1.3.1 XML 基本规范.....7	
1.3.2 XML 关联技术规范.....7	
1.4 XML 解析器与浏览器.....10	
1.5 XML 编辑软件.....12	
1.5.1 XMLSpy.....12	
1.5.2 XMLWriter.....12	
本章小结.....13	
思考与练习.....13	
第 2 章 XML 语法规则15	
2.1 一个结构完整的 XML 文档.....15	
2.2 XML 标记语法.....16	
2.2.1 标记的命名规则.....16	
2.2.2 标记的使用规则.....17	
2.3 XML 文档的结构.....18	
2.3.1 XML 声明.....19	
2.3.2 XML 注释.....20	
2.3.3 处理指令.....20	
2.3.4 文档类型声明.....21	
2.3.5 XML 元素.....21	
2.3.6 XML 属性.....22	
2.3.7 实体.....23	
2.3.8 CDATA 节.....24	
本章小结.....25	
思考与练习.....25	
第 3 章 文档类型定义28	
3.1 XML 文档的有效性.....28	
3.2 DTD 概述.....29	
3.2.1 文档类型声明与文档类型定义的区别.....29	
3.2.2 文档类型声明.....29	
3.2.3 DTD 与 XML 实例文档之间的关系.....32	
3.2.4 DTD 的用途.....32	
3.2.5 DTD 所包括的内容.....33	
3.3 DTD 对元素的声明.....33	
3.3.1 元素声明语法.....33	
3.3.2 元素内容模式.....34	
3.3.3 元素声明中的正则表达式.....36	
3.3.4 元素声明综合示例.....36	
3.4 DTD 对属性的声明.....37	
3.4.1 属性声明的语法.....38	
3.4.2 属性类型.....39	
3.4.3 属性默认值.....43	
3.4.4 属性声明综合示例.....44	
3.5 实体的声明与引用.....45	
3.5.1 实体的概念与分类.....45	
3.5.2 普通实体.....46	
3.5.3 参数实体.....51	
本章小结.....54	
思考与练习.....54	
第 4 章 XML 验证技术 Schema57	
4.1 命名空间.....57	
4.1.1 命名空间的概念.....57	
4.1.2 使用命名空间解决命名冲突问题.....58	
4.1.3 默认的命名空间.....58	
4.1.4 使用命名空间.....59	
4.2 XML Schema 介绍.....60	
4.2.1 XML Schema 与 DTD 的关系.....61	
4.2.2 XML Schema 的内容.....61	
4.2.3 XML Schema 两种模型及 Schema 引用.....61	
4.3 W3C XML Schema 完整的示例.....63	
4.4 W3C XML Schema 的数据类型.....64	

4.4.1 W3C Schema 内置数据类型	6.1.1 XPath 基本概念	112
类型	6.1.2 XPath 定位路径	115
4.4.2 简单数据类型	6.1.3 XPath 函数	119
4.4.3 复杂数据类型	6.1.4 XPath 1.0 与 XPath 2.0	
4.5 元素和属性的定义	的区别	121
4.5.1 元素的定义	6.2 XQuery	123
4.5.2 属性的定义	6.3 XLink 和 XPointer	128
4.5.3 元素、属性定义的综合	6.3.1 XLink 简介	128
示例	6.3.2 XPointer 简介	130
4.6 引用 W3C XML Schema 模式	本章小结	131
的方法及相关知识	思考与练习	132
4.6.1 XML 文档引用 W3C XML	第 7 章 XML 应用编程	135
Schema 模式的方法	7.1 XML 编程接口与 XML 解析器	135
4.6.2 elementFormDefault 属性	7.2 DOM 接口	136
的作用	7.2.1 DOM 与 XML DOM	136
本章小结	7.2.2 DOM 节点树	136
思考与练习	7.2.3 DOM 常用接口	138
第 5 章 可扩展样式单语言	7.2.4 DOM 树的遍历	143
5.1 在 XML 中使用 CSS 和 XSL	7.2.5 动态创建 XML 文档	150
5.2 层叠样式单简介	7.2.6 DOM 节点树的修改与	
5.2.1 CSS 语法	删除	154
5.2.2 CSS 的选择器	7.3 SAX 接口	157
5.2.3 CSS 的属性集	7.3.1 SAX 接口概述	157
5.3 XSL 概述	7.3.2 SAX 工作原理	159
5.3.1 CSS 与 XSL 的区别	7.3.3 使用 SAX	160
5.3.2 XSL 的组成及工作原理	7.3.4 SAX 错误的处理	167
5.3.3 XML 文档结构树	7.3.5 DOM 接口与 SAX 接口	
5.3.4 XSL 文档的基本结构	的结合使用	167
5.4 XSL 的模板	7.4 XML 与数据库	170
5.4.1 定义模板元素	7.4.1 DOM 与数据库	170
5.4.2 调用模板元素	7.4.2 SAX 与数据库	174
5.5 XPath 初步介绍	本章小结	177
5.6 XSL 的元素语法	思考与练习	177
5.6.1 选择模式元素	第 8 章 XML 数据岛	180
5.6.2 测试模式元素	8.1 数据岛基本概念	180
5.6.3 其他模式元素	8.1.1 数据岛定义	180
本章小结	8.1.2 数据岛的引入方式	181
思考与练习	8.2 数据岛的数据绑定	182
第 6 章 XML 高级语法	8.2.1 能与 XML 数据岛绑定	
6.1 XPath	的 HTML 标记	183

8.2.2 数据绑定与 DSO 对象	185	9.4.1 显示查询提示信息的 页面	211
8.2.3 数据绑定方式	185	9.4.2 在数据库中查询信息	214
8.2.4 绑定多层 XML 文档	188	9.4.3 采用不同的样式显示 查询结果	218
8.2.5 XML 标记属性的处理	190	9.5 管理信息模块	220
8.3 DSO 对象	191	9.5.1 在客户端对数据岛的 操作	221
8.3.1 DSO 对象对 XML 文档 的操作	192	9.5.2 AJAX 简介	224
8.3.2 DSO 的 ADO 记录集	194	9.5.3 在服务器端的数据保存	227
8.4 动态选择 XML 文档的显示 样式	196	本章小结	230
本章小结	199	附录 实 验	232
思考与练习	200	实验 1 结构良好的 XML 文档	232
第 9 章 XML 的 Web 应用实例	202	实验 2 XML 文档及其 DTD	233
9.1 应用实例需求概述	202	实验 3 XML 文档及其 Schema	235
9.2 XML 与数据库	204	实验 4 为 XML 文档编写可扩展 样式单 XSL	237
9.2.1 实例中所使用的数据库 及数据源	205	实验 5 DOM 编程实验	239
9.2.2 实例中所使用的 XML 文档	206	实验 6 SAX 编程实验	241
9.3 JSP 介绍	207	实验 7 XML 数据岛实验	242
9.3.1 JSP 的配置	208	实验 8 XML 网页编程实验	243
9.3.2 JSP 的内置对象简介	210	参考文献	244
9.4 查询信息模块的实现	210		

第 1 章 XML 概述

随着因特网技术的发展，现有的 HTML 及其脚本技术呈现出其局限性。可扩展标记语言（XML）的出现，把网络表达语言向前推进了一大步，使得标记语言可以由编程人员按照需要进行扩展，其扩展能力是无限的，这就给 Web 编程带来了革命性进展。XML 作为一种 Internet 的“世界语”，已受到越来越多技术人员的重视。本章将介绍 XML 的引导知识，包括 XML 的发展历史、特点、用途、开发环境、知识体系结构等。

本章学习要点：

- 了解 XML 的发展历史；
- 理解 XML 与 HTML 的关系；
- 了解 XML 的用途；
- 了解 XML 的技术规范；
- 掌握编写一个简单的 XML 文档的方法。

1.1 XML 简介

可扩展标记语言（eXtensible Markup Language, XML）是由万维网联盟（World Wide Web Consortium, W3C）于 1998 年 2 月发布的推荐标准。它定义了一种文件格式，一种描述数据、保存数据的方法，使用这种格式的计算机数据，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，从而大大提高了处理数据的效率和灵活性。这就是设计 XML 的主要目的。

XML 基于文本格式，以结构化的方式描述各种类型的数据，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 文档的内容结构信息，快速、准确地定位所需的信息。

1.1.1 XML 的发展历史

XML 的历史可追溯到 1969 年，美国 IBM 公司一组研究人员开始设计一种通用标记语言（Generalized Markup Language, GML），来描述文件及其格式。这项研究工作持续了十几年，于 1980 年推出 SGML，并于 1986 年获得国际标准化组织（ISO）的批准。SGML 是一种关于电子文献格式描述的计算机语言。这里的“文献”不仅包含传统意义的文本式文献，而且包括图像、音频、视频、动画等一切电子格式的文档。SGML 从两个层次——结构和内容来

描述文献，其核心是文档类型定义（Document Type Definition, DTD）。

SGML 是一个庞大复杂的系统，学习和使用起来比较复杂，价格非常昂贵，而且 SGML 并不适合 Web 数据描述，因此没有在互联网上流行起来。

而源于 SGML 的超文本标记语言（HyperText Markup Language, HTML）却在互联网上得到广泛的应用。HTML 借用了 SGML 的很多元素，但相比 SGML 来说简单得多，且比较适合 Web 页面的开发。但 HTML 只有固定的、与外观显示格式有关的标记，且标记相对较少，缺少 SGML 的柔性和适应性，不能支持特定领域的标记语言，如对数学、化学、音乐等领域的表示支持较少，即开发者很难在 Web 页面上表示数学公式、化学分子式和乐谱等。

1996 年，W3C 成立了一个专门委员会，研究如何开发一种新的标记语言，既具有 SGML 的强大数据处理能力，又适合在 Web 上使用。为此该专门委员会重新定义了 SGML 的一些内部值和参数，去掉了大量的很少用到的繁杂功能，保留了 SGML 的结构化功能，使得网站设计者可以定义自己的文档类型。1996 年底，这个委员会推出了 XML 的第一稿，同时将自身更名为 XML 工作组，并于 1998 年 2 月正式向 W3C 提交了 XML 的最终推荐标准。这个标准就是 XML 1.0。2004 年 2 月 4 日 XML 1.1 推荐标准正式发布，但目前推荐遵循的是 W3C 于 2000 年 10 月 6 日发布的 XML 1.0 标准。图 1-1 所示的是几种置标语言的发展关系。

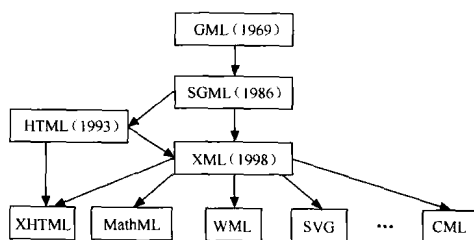


图 1-1 几种置标语言的发展关系

1.1.2 XML 与 HTML 的比较

XML 和 HTML 都来源于 SGML，它们都是标记语言，两者在语法上有很大的联系，学习 XML 可以从 HTML 入手。比较下面两段 HTML 和 XML 的程序。

例 1.1 notel.1.html，一个显示便条信息的 HTML 文档。

```

<html>
<head>
  <title>订单信息</title>
</head>
<body>
  <h1>订单号：1001</h1>
  <h2>商品名称：跑步鞋</h2>
  <h2>单价：150 元</h2>
  <h2>数量：2 双</h2>
</body>
</html>

```

例 1.2 notel.2.XML，一个存储便条信息的 XML 文档。

```

<?xml version="1.0" encoding="gb2312"?>

```

```
<订单>
```

```
  <订单号>1001</订单号>
```

```
  <商品名称>跑步鞋</商品名称>
```

```
  <单价>150</单价>
```

```
  <数量>2</数量>
```

```
</订单>
```

比较以上两个例题，分析 XML 和 HTML 存在的主要区别如下。

(1) HTML 的标记是固定的，预定义的，不可扩展的；而 XML 的标记是可扩展的，是可以由用户自定义的。在例 1.1 的 HTML 文档中，标记<html>、<head>、<title>、<body>、<h1>、<h2>都是固有的，具有固定的含义。HTML 文档创建者只能使用当前 HTML 标准所支持的标记。与 HTML 不同，XML 标记由架构或文档的作者自定义，可无限扩展。在例 1.2 的 XML 文档中，标记<订单>、<订单号>、<商品名称>、<单价>、<数量>都是由用户自定义的，用户可根据自己的需要创建标记。

(2) HTML 的标记说明了信息的显示格式；而 XML 标记表示了数据的逻辑结构。在例 1.1 的 HTML 文档中，将订单信息按规定的格式显示罗列出来，如<h1>标记控制了“订单号：1001”数据按一级标题的方式显示。而 XML 标记表示了数据的逻辑结构，例 1.2 的 XML 文档将订单数据以结构化的方式保存起来，<订单>是一个大的数据对象，它包括<订单号>、<商品名称>、<单价>、<数量>这几个数据对象，XML 标记并没有显示控制信息。

图 1-2 所示的是 HTML 文档在浏览器中打开的效果，可以发现 HTML 文档中的数据按照各标记的设置显示出来；图 1-3 所示的是 XML 文档在浏览器中打开的效果，可以看到 XML 文档中的数据是有层次结构的，但数据在浏览器中是按原样显示，没有任何显示控制。

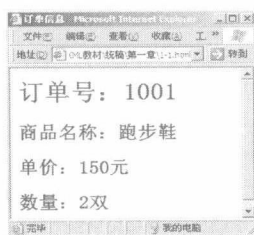


图 1-2 HTML 文档在浏览器中打开的效果

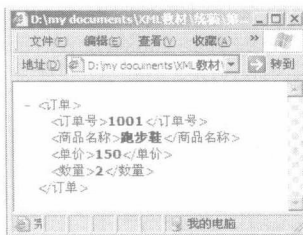


图 1-3 XML 文档在浏览器中打开的效果

(3) XML 和 HTML 为不同的目的而设计。XML 被设计用来描述数据，其焦点是数据的内容；HTML 被设计用来显示数据，其焦点是数据的外观。HTML 旨在显示信息，而 XML 旨在描述信息。XML 不是要替换 HTML，实际上 XML 和 HTML 是两种不同用途的语言。

(4) XML 的数据可重复使用；而 HTML 的数据难以重用。XML 的数据可以像数据库中的数据一样被反复调用，开发者可以利用 XML 标记，提取 XML 文档中存放的特定数据，如应用程序可以利用标记<订单号>、<单价>、<数量>来统计销售情况。而且 XML 文档的标记还可以直接与存储订单的数据库的各个字段对应起来，方便 XML 与数据库之间的数据交换。

(5) HTML 的数据与排版、表现形式混在一起；而 XML 文档将数据与显示格式完全分离，能实现同样的数据以不同的方式显示输出，为信息的开放、共享和交互提供基础。

1.1.3 XML 简介

XML 是 SGML 的一个简化子集，它将 SGML 的丰富功能与 HTML 的易用性结合在 Web 应用中。XML 继承了 SGML 的许多特性，它有 SGML 20% 的复杂性，却有 SGML 80% 的功能。

1. XML 描述的是结构和语义，而不是格式化

XML 是一种语义/结构化语言，XML 标记描述的是文档的结构和意义。它不描述页面元素的格式化。文档本身只说明文档包括什么标记，而不是说明文档看起来是什么样的。

作为对照，HTML 文档包括了格式化、结构和语义的标记。就是一种格式化标记，它使其中的内容变为粗体显示；是一种语义标记，意味着其中的内容特别重要；<td>是结构标记，指明内容是表格中的一个单元；另外，某些标记可能具有所有这 3 种意义，<h1>标记可同时表示 20 磅的 Helvetica 字体的粗体、第一级标题和页面标题。

在 XML 中，使用了具有语义的、可以描述数据结构的标记。

例 1.3 music1.3.xml，一个存储了歌曲信息的 XML 文档。

```
<?xml version="1.0" encoding="gb2312"?>
<歌曲集>
  <歌曲>
    <歌名>生死不离</歌名>
    <作词>王平久</作词>
    <作曲>舒楠</作曲>
    <演唱>成龙</演唱>
    <发表时间>2008</发表时间>
  </歌曲>
</歌曲集>
```

在例 1.3 的 XML 文档中，没有使用通用的标记如<td>和，而是使用了具有意义的标记，如<歌名>、<作词>、<作曲>、<演唱>和<发表时间>等。这种用法可以使源码易于阅读，使人能够看出作者的意愿。

XML 的标记也描述了文档的数据结构。在例 1.3 的 XML 文档中，<歌曲集>是一个大对象，它包括若干个<歌曲>元素，每个<歌曲>元素下又有若干个子元素，用于存放歌曲的歌名、作词、作曲等信息。这种数据结构不仅便于人们阅读，而且可以方便机器找出文档中的所有歌曲及信息。

XML 标记描述的是文档的结构和语义。但它并没有描述页面元素的格式化表现形式。为此 XML 文档需要配套的样式单文档，来控制 XML 文档的显示输出样式，这部分内容将在第 5 章介绍。

2. XML 将数据内容和显示格式相分离

由于 XML 描述的是数据的结构和语义，而不是格式化，所以将数据内容和显示格式相分离，使 XML 文档具有很强的灵活性，Web 用户所追求的许多先进功能在 XML 环境下更容易实现。例如，同样的数据可以使用不同的显示格式显示输出。

3. XML 是元标记语言

XML 是一种元标记语言，“元标记”就是开发者可以根据自己的需要定义自己的标记，如开发者可以定义如下标记：<作词>、<作曲>、<演唱>等，任何满足 XML 命名规则的名称都可以作为标记。

HTML 是一种预定义标记语言，它定义了一套固定的标记，用来描述一定数目的元素。它只识别诸如<html>、<p>等已经定义的标记，对于用户自定义的标记是不识别的。

XML 允许使用者创建和使用他们自己的标记，而不是 HTML 的有限词汇表。这使得 XML 具有无限的可扩展性。

4. XML 是自描述语言

XML 文档通常包含一个文档类型定义 (Document Type Definition, DTD)，DTD 将在本书的第 3 章加以介绍。DTD 就是一组标记符的语法规则，表明 XML 文档是怎样组织的。例如，DTD 可以表示一个<学生>必须有一个子标记<姓名>，可以有或没有子标记<爱好>等。XML 文档由 DTD 和 XML 文本组成，XML 文档是自描述的。

5. XML 是独立于平台的

XML 是纯 ASCII 文本，是一种通用的数据格式，可以在不同的计算机平台和不同的计算机程序间方便、平稳地交换数据。XML 表示的信息独立于平台，这里的平台既可以理解为不同的应用程序，也可以理解为不同的操作系统，或不同的硬件环境。XML 描述了一种规范，利用它 Microsoft 的 Word 文档可以和 Adobe 的 Acrobat 交换信息，可以和数据库交换信息。

另外，ASCII 的抗破坏能力是很强的。而对于压缩数据和 Java 对象来说，只要破坏一个数据，文件就不可阅读。

6. XML 不进行任何操作

虽然 XML 标记可用于描述订单之类的数据结构，但它不包含可用于发送或处理该订单以及确保按该订单交货的任何代码。设计人员必须另外编写代码来实现对 XML 数据的这些操作，即要为 XML 文档开发相应的应用程序来处理 XML 的数据。第 7 章将介绍 XML 应用程序开发。

7. XML 具有良好的保值性

随着软件的更新和升级，大量的数据文件需要配合转换，这将导致低效率甚至文件缺损。采用 XML 则不存在这些问题，用户可以方便地将 XML 文档转换成其他的格式。XML 具有良好的保值性，XML 可以解决如何长久保存 21 世纪大量电子化文档的问题。

综上所述，XML 是一种描述语义和数据结构的语言，可以保存在任何可以存储文本的文档中，XML 具有自描述性、保值性、内容和显示相分离、可扩展、独立于平台等特点。

1.2 XML 的用途

XML 是被设计用来存储数据、携带数据和交换数据的，它不是为了显示数据而设计的。

XML 的主要用途包括以下几个方面。

(1) 从 HTML 中分离数据。通过 XML, 用户可以在 HTML 文件之外存储数据。在不使用 XML 时, 数据必须存储在 HTML 文件之内; 使用了 XML, 数据就可以存放在分离的 XML 文档中。这种方法可以让用户精力集中到使用 HTML 做好数据的显示和布局上, 并确保数据改动时不会导致 HTML 文件也需要改动, 可以方便维护页面。XML 数据同样可以以“数据岛”的形式存储在 HTML 页面中。用户仍然可以集中精力到使用 HTML 格式化和显示数据上去。

(2) 交换数据。通过 XML, 可以在不兼容的系统之间交换数据。在现实生活中, 计算机系统和数据库系统所存储的数据有多种形式, 对于开发者来说, 最耗费时间的就是在遍布网络的系统之间交换数据。把数据转换为 XML 格式存储将大大减少交换数据时的复杂性, 还可以使得这些数据能被不同的程序读取。

(3) B2B 电子商务。使用 XML, 可以在网络中交换金融信息。XML 正在成为遍布网络的商业系统之间交换金融信息所使用的主要语言。许多与 B2B 有关的完全基于 XML 的应用程序正在开发中。

(4) 共享数据。既然 XML 数据是以纯文本格式存储的, 那么 XML 提供了一种与软件和硬件无关的共享数据方法。这样创建一个能够被不同的应用程序读取的数据文件就变得简单了。同样, 升级操作系统、升级服务器、升级应用程序、更新浏览器也变得容易多了。

(5) 存储数据。大量的数据可以存储到 XML 文件中或者数据库中。应用程序可以读写和存储数据, 一般的程序可以显示数据。

(6) 充分利用数据。既然 XML 是与软件、硬件和应用程序无关的, 所以可以使数据被更多的用户、更多的设备所利用, 而不仅仅是基于 HTML 标准的浏览器。其他客户端和应用程序可以把 XML 文档作为数据源来处理, 就像它们对待数据库一样, 数据可以被各种各样的“阅读器”处理, 这对残疾人来说是很方便的。

(7) 创建新的语言。XML 定义了一套元句法, 遵循这套元句法, 任何人都可以创建与特定领域有关的标记语言。XML 已创建的标记语言有:

① MusicML (Music Markup Language, 音乐置标语言) 用于描述音乐符号的置标语言。

② MathML (Mathematical Markup Language, 数学置标语言) 用于描述数学公式, 可以将数学公式精确地显示在浏览器上。

③ CML (Chemical Markup Language, 化学标记语言) 用于描述化学的置标语言, CML 可描述分子与晶体结构、化合物的光谱结构等。

④ SVG (Scalable Vector Graphics, 可伸缩矢量图形) 用于显示矢量图形的置标语言。

⑤ WML (Wireless Markup Language, 无线置标语言) 用于手机上网的置标语言。

⑥ SMIL (Synchronized Multimedia Integration Language, 同步多媒体集成语言) 用于体现多媒体效果的置标语言。

浏览器不必事先了解多种不同的标记语言使用的每个标记。关于如何显示这些标记的内容的详细指令是附加在文档上的另外的样式单提供的。用户可以创建自己需要的标记, 当需要时, 由样式单告诉浏览器如何显示这些标记。

XML 作为数据描述语言是非常优秀的, 但并不是所有的电子数据转换成 XML 都是最合适的。例如, XML 的文本表现手法、标记的符号化等会导致 XML 数据比二进制表现方法的数据量增加, 尤其当数据量很大时, 会成为很大的问题。也就是说, XML 的引入要根据具体

需求, 比较其优缺点, 在充分发挥 XML 优点的领域进行。

尽管 XML 具备了作为通用数据描述语言的优秀特征, 但 XML 不是编程语言, 说到底它是一种数据描述的技术。因此, XML 文件的显示、文件结构的变更、应用程序的操作等 XML 相关技术也是非常重要的。

1.3 XML 的技术规范

XML 相关规范可分为以下 3 部分来理解。

- XML 基本规范 (XML1.0、XML1.1)。
- XML 关联技术规范 (DTD、XML 命名空间、XML Schema、XPath、XSL、XSLT、XLink、XPointer、XQuery、DOM、SAX 等)。
- XML 应用规范 (XHTML、SVG、SMIL、MathML、GML、RDDL、XBL、XBRL、RSS 等)。

由于篇幅有限, 本教材仅介绍 XML 的基本规范和 XML 的关联技术规范。

1.3.1 XML 基本规范

W3C 于 1998 年 2 月批准了 XML 的 1.0 版本, 成为一个国际标准。其设计目标包括:

- XML 可以直接应用于 Internet;
- XML 应该支持各种不同的应用;
- XML 应与 SGML 兼容;
- 处理 XML 文档的应用程序应该容易编写;
- 可选特性应减到最小, 最好没有, 最好无二义性;
- XML 文档应便于用户阅读、理解;
- XML 的设计过程应简单、迅速;
- 设计的 XML 应该正式、简洁;
- XML 文档应易于创建;
- 标记的简洁性并不重要。

2004 年 2 月 4 日 XML 1.1 推荐标准正式发布, XML 1.1 主要是为了适应新版 Unicode 而做的一些调整, 没有重大的修改。由于 XML 1.0 所依赖的字符规范 Unicode 标准已经从 2.0 版升级到了 3.1 版甚至更高, 许多在 Unicode 2.0 中没有的字符可能已经被使用于 XML 1.0 的字符数据 (character data) 中了。然而, XML 1.0 在关于什么是结构良好 (well-formed) 的 XML 方面仍保持不变, 使新字符不能出现在 XML 名称中 (如元素类型名、属性名、枚举属性值、PI 目标等)。而 XML 1.1 对名称的规定是: 任何不被禁止的 (由于某个特定原因), 就是允许的, XML 1.1 几乎允许所有字符 (包括那些尚未被指定的) 在名称中出现。

1.3.2 XML 关联技术规范

XML 的关联技术规范较多, 主要介绍如下几个。

1. DTD (Document Type Definition, 文档类型定义)

DTD 可定义合法的 XML 文档构建模型。它使用一系列合法的元素来定义文档的结构, 它可用来验证一个 XML 文档的有效性。DTD 是一系列关于元素类型 (Element Type)、属性 (Attribute)、实体 (Entities) 和注记 (Notations) 的定义。DTD 是从 SGML 继承下来的, 目前大多数的面向 XML 应用, 都对 DTD 做了很好的支持, DTD 的工具也相对较为成熟。

2. XML 命名空间 (XML Namespaces)

XML 命名空间提供了一种避免元素命名冲突的方法。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的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。

命名空间表示了名称的来源和使用范围。通过命名空间可以区分来自不同的 XML 应用但具有相同名称的标识, 还可以把来自某种 XML 应用的相关元素和属性集合在一起, 以方便软件识别和处理。

3. XML Schema

2001 年 5 月 2 日, XML Schema 成为了 W3C 的正式推荐标准。XML Schema 的功能类似 DTD, 也可以用来约束 XML 文档结构, 验证 XML 文档的有效性。但 XML Schema 的功能更强, 不仅可以约束 XML 文档的结构, 而且可以约束 XML 文档的内容及其语义; XML Schema 提供了丰富的内建数据类型, 可以根据需要自定义数据类型; 并且 XML Schema 使用的是 XML 语法, 可以用 XML 解析器来解析 XML Schema。

XML DTD 已不能满足那些对数据交换或者描述能力要求较高的应用, 以 XML Schema 来代替 XML DTD 已经成为一种必然趋势。

4. XPath (XML Path Language, XML 路径语言)

Xpath 1.0 于 1999 年 11 月 16 日成为 W3C 推荐标准, Xpath 2.0 于 2007 年 1 月 23 日被确立为 W3C 推荐标准。XPath 提供了在一个 XML 文档中定位和链接信息的通用语法和语义, 是在 XML 文档中查找信息的语言。XPath 可用来在 XML 文档中对元素和属性进行遍历。通过它可以方便地标识到 XML 文档中的节点。XPath 是 W3C XSLT 标准的主要元素, 并且 XQuery 和 XPointer 同时被构建于 XPath 表达式之上。因此, 对 XPath 的理解是很多高级 XML 应用的基础。

5. XSL (eXtensible Stylesheet Language, 可扩展样式语言)

XSL 1.0 于 2001 年 10 月 15 日成为 W3C 推荐标准, XSL 1.1 在 2006 年 12 月 5 日被确立为 W3C 推荐标准。XML 将内容和显示格式分开, 可以让用户按照各自希望的格式显示同一 XML 文档的数据内容。为 XML 文档提供格式控制信息的是样式表, 而适用于 XML 文档的样式表语言有 XSL 和 CSS 语言。XSL 是专门为 XML 设计的样式表语言, 它采用的是 XML 语法, 它的优势在于可以用于转换, 包括可以把 XML 文档转换为 HTML 格式 (CSS 是用于分离 HTML 文档中的数据和格式), 这是目前 XSL 的主要用途。而且同一个样式表可以用于多个具有相似源树结构的文档。输出的媒介不仅限于浏览器, 还可以是打印机、语音

设备或其他媒体。XSL 实际上由两个基本点不同的标准构成：转化语言 XSLT (XSL Transformations, 可扩展样式语言转换) 和格式化对象 (Format Object)。

6. XSLT (XSL Transformations, 可扩展样式语言转换)

XSLT 1.0 于 1999 年 11 月 16 日成为 W3C 推荐标准, 目前最新的是在 2007 年 1 月 23 日发布的 W3C 推荐标准 XSLT 2.0。XSLT 是为显示和打印 XML, 而从 XSL 规范独立出来的规范。XSLT 的基本原理是模式和模板匹配, 是根据 XSL 样式表将一个 XML 文档转换到另一个 XML 文档的语言。XSLT 本身也是用 XML 文档表示的, 它描述将源树转换到结果树的转换规则。XSLT 表达的转换也叫样式表, 样式表包含一系列模板规则, 一个模板规则包含一个模式和一个模板。另外, XSLT 也可以用于排版以外的各种用途。

7. XLink (XML Linking Language, XML 链接语言)

在 2001 年 6 月 27 日, XLink 被确立为 W3C 推荐标准。XLink 是用于在 XML 文档中创建超级链接的语言, 类似于 HTML 链接, 但功能更为强大。XML 文档中的任何元素均可成为 XLink。XLink 支持简易链接, 也支持可将多重资源链接在一起的扩展链接; 通过 XLink, 链接可在被链接文件外进行定义。

8. XPointer (XML Pointer Language, XML 指针语言)

XPointer 于 2003 年 3 月 25 日成为 W3C 推荐标准。XPointer 使超级链接可以指向 XML 文档中更多具体的部分 (片断), XPointer 使用 XPath 表达式在 XML 文档中进行定位。XPointer 可以链接到特定位置, 甚至可以链接 XML 文档中的某个位置范围。

9. XQuery (XML Query Language, XML 查询语言)

XQuery 1.0 在 2007 年 1 月 23 日被确立为 W3C 推荐标准。XQuery 与多种 W3C 标准相兼容, 如 XML、Namespaces、XSLT、XPath 以及 XML Schema。XQuery 是用于 XML 数据查询的语言。XQuery 对 XML 的作用类似 SQL 对数据库的作用, XQuery 被构建在 XPath 表达式之上, XQuery 被所有主要的数据库引擎 (IBM、Oracle、Microsoft 等) 支持。

10. XML DOM (XML Document Object Model, XML 文档对象模型)

DOM Level1 是 1998 年 10 月 1 日成为 W3C 的推荐标准, 目前最新的是 2004 年 4 月 7 日发布的 DOM Level3。DOM 定义了访问诸如 XML 和 XHTML 文档的标准, 是一个使程序和脚本有能力动态地访问和更新文档的内容、结构以及样式的平台和语言中立的接口。W3C DOM 被分为 3 个不同的部分:

- 核心 DOM: 用于任何结构化文档的标准模型;
- XML DOM: 用于 XML 文档的标准模型;
- HTML DOM: 用于 HTML 文档的标准模型。

XML DOM 是用于 XML 的标准对象模型和用于 XML 的标准编程接口, 它是中立于平台和语言的。XML DOM 定义了所有 XML 元素的对象和属性, 以及访问它们的方法 (接口)。