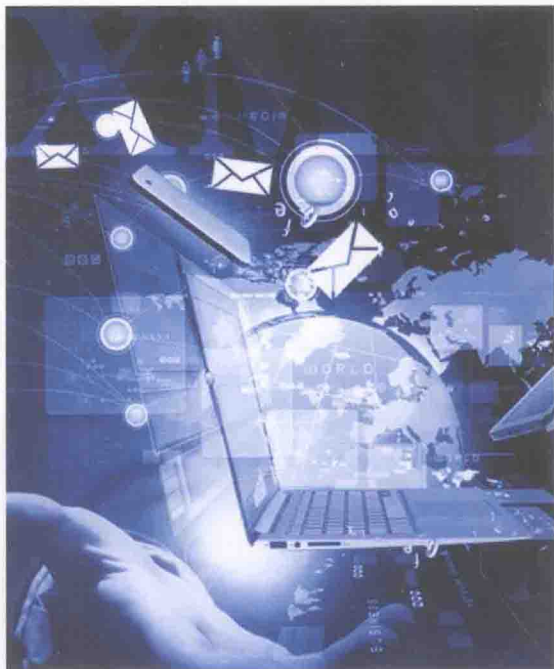


XML基础教程

- ◆ XML基础知识
- ◆ 格式良好的XML文档
- ◆ 有效的XML文档
- ◆ XML文档的显示技术
- ◆ XML解析器
- ◆ XML与数据库
- ◆ XML Spy 2013编辑软件的使用法



胡 静 常 瑞 张 青 郭纯一 编著



清华大学出版社

高等学校计算机应用规划教材

XML 基础教程

胡静 常瑞 张青 郭纯一 编著

清华大学出版社

北 京

内 容 简 介

本书从初学者角度出发,以通俗易懂的语言,详实丰富的实例,介绍了与 XML 有关的各种主要技术。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、级联样式表、可扩展样式表、解析器和数据库的集成等知识,最后还通过一个综合案例演示了 XML 在实际项目开发中的应用。

本教程注重基础、讲究实用、力求由浅入深,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软件工程、计算机科学与技术等相关专业的研究生参考用书,也可作为相关专业的高年级本科教材,还可作为初学者学习 XML、Android 移动应用开发、JavaEE 开发的培训教材。

本书的电子教案、习题答案和实例源文件可以到 <http://www.tupwk.com.cn/downpage/index.asp> 网站下载。

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。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

XML 基础教程/胡静 等编著. —北京:清华大学出版社,2015

(高等学校计算机应用规划教材)

ISBN 978-7-302-39620-8

I. ①X… II. ①胡… III. ①可扩展语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 049229 号

责任编辑:胡辰浩 袁建华

装帧设计:孔祥峰

责任校对:曹 阳

责任印制:何 芊

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>, 010-62794504

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:16.25

字 数:375 千字

版 次:2015 年 4 月第 1 版

印 次:2015 年 4 月第 1 次印刷

印 数:1~4000

定 价:35.00 元

产品编号:054095-01

前 言

物竞天择，适者生存——在以计算机与互联网技术为代表的 IT 时代，各种各样的新技术如雨后春笋般涌现，然而真正能够历经磨炼生存下来的却寥寥无几。毫无疑问，XML 便是其中的佼佼者。XML 是 SGML 的一个子集，保留了灵活性，去掉了复杂性。很快 XML 便获得了巨大的成功，XML 标准开始突飞猛进地发展，大批的软件开发商争先恐后地采纳这个标准，这一切令人叹为观止。如今 XML 在 IT 领域已经拥有不可动摇的地位，很难想象有一个重要的应用程序不使用 XML 来保存它的配置文件或数据文件。

XML 是由 W3C 定义的一种语言，是表示结构化数据的行业标准。XML 在电子商务、移动应用开发、Web Service、云计算等技术和领域中起着非常重要的作用。这些名人曾这样评论 XML。

- 微软总裁比尔·盖茨：XML 将为每一种流行的编程语言带来一个语言革命，其影响力甚至超过 HTML 为演示世界带来的影响。
- 微软 CEO 史蒂夫·鲍尔默：XML 的出现，对于信息技术的影响不亚于 GUI 和浏览器。
- IBM 资深专家 Goldfarb：我为 XML 感到骄傲，WWW 正在变成以 XML 为基础。XML 是未来的发展趋势，无论是网页设计师还是网络程序员，都应该及时学习和了解，等待只会让你失去机会。

学习和掌握 XML 的理由如下。

- XML 是一门年轻的技术。
- XML 是最前沿的技术。
- XML 是应用广泛的技术，其发展前景无可限量。
- XML 是一门综合性很强的技术。

XML 越来越热，关于 XML 的基础教程也随处可见，可是一大堆的概念和术语往往让人望而生畏。有些图书起点太高，初学者难以理解基本概念，一开始学习就困难重重，容易产生厌倦心理而放弃学习；有的图书又过于简单，读者学完之后还是不会做实际的事情，不能达到一定的高度。

概括起来，本书具有以下主要特点。

- 注重基础，讲究实用，力求从入门到精通。
- 充分体现案例教学。本书以易学易用为重点，例子实用、知识丰富、步骤详细、学习效率 high，特别适合入门者。
- 配有源代码，加速学习。本书的所有示例均在 XML Spy 2013 开发环境下调试通过，读者可直接下载所有例子的源程序，并通过教材中介绍的步骤学习要点。

本书在讲述 XML 基本概念的基础上，系统地介绍了 XML 技术中业已成熟的标准和

应用技术，并给出了基于 XML 的应用实例。全书共分 10 章，各章的主要内容如下：

第 1 章是 XML 简介，讲述标记语言的发展、HTML 的局限性、XML 的实现机制、XML 的优势与特点，并给出了 XML 文档范例。这一章还用不少的篇幅介绍了 XML 技术的应用领域与应用前景，以及与 XML 相关的各种技术。

第 2 章讲解 XML 的语法，包括 XML 文档的构成、XML 文档的声明与注释、XML 元素的组成与命名、XML 元素属性的定义规则、特殊的 CDATA 文本段、XML 命名空间的概念与应用等。XML 的语法并不复杂，但必须遵守，只有符合这些语法规则的 XML 文档才是一个格式良好的 XML 文档。

第 3 章讲解文档类型定义 DTD。介绍了 DTD 的基本结构，重点阐述如何使用 DTD 为 XML 文档建立语义约束，包括如何在 DTD 中定义元素及元素类型，分析了 DTD 所支持的各种属性类型，说明了如何在 DTD 中定义各种实体，指出了 DTD 的局限性及现状。

第 4 章讲解描述和约束 XML 文档的语言——XML Schema。对比 DTD 中存在的缺陷引出了 Schema，以一个 Schema 文档为例，介绍 Schema 的基本结构，详细分析 Schema 中的简单类型和复杂类型，以及如何进行数据类型的定义、元素的定义和属性的定义，分析了 Schema 命名空间的作用，说明了验证 XML 文档有效性的两种方法。

第 5 章介绍如何使用 CSS(层叠样式表)来格式化输出 XML 文档的内容。XML 文档本身只包含数据而不包含这些数据的显示格式信息，然而利用简单的 CSS 技术就能实现将 XML 文档中的数据以设计者所设定的各种格式在浏览器中显示出来。

第 6 章讲解 XSL(可扩展样式表)技术，利用该技术不仅能够把 XML 文档转换为 HTML 文档，实现在浏览器中的格式化显示，而且还可以将 XML 文档转换为其他各种基于文本的文档，以实现跨平台的数据共享和交换。

第 7 章详细展示 XML 文档的解析过程，包括 DOM 树模型、DOM 的结构、DOM 基本接口、DOM 的节点访问和 DOM 对 XML 文档的相关操作等内容。DOM 解析器的主要功能是检查 XML 文件是否有结构上的错误，剥离 XML 文件中的标记，读出正确内容，并交给下一步应用程序处理。

第 8 章介绍一种高效的解析器——SAX 解析器，包括 SAX 的优缺点、工作机制、事件处理器、SAX 事件、常用接口、回调方法、SAX 错误信息和 SAX 对 XML 文档的相关操作。在这一章中还比较了 SAX 与 DOM 两种截然不同的解析方式，并给出了将两者结合应用的具体实例。

第 9 章介绍了 XML 与关系数据及关系数据库的集成，阐述了数据库技术的发展、XML 的数据交换及存取机制、在数据库技术中引入 XML 的原因以及二者的结合对数据交换的影响，并全面介绍了 .NET 平台下 XML 与关系数据库系统(以 SQL Server 2005 为例)互换数据所采用的各种技术，以及 SQL Server 2005 对 XML 的支持。

第 10 章通过一个综合性的实例，系统地介绍了 DOM、SAX、CSS 等多种 XML 技术的应用，演示了在 .NET 平台下利用 XML 进行实际项目开发的完整过程。

本书从 XML 的基础知识讲起，语言通俗易懂，并配有很多实例和插图，使读者对每一章所讲述的内容都能有深刻的理解并加以巩固，十分适合初学者和有一定 XML 基础的

人员使用。

本书由胡静、常瑞、张青、郭纯一编写并由胡静统稿。此外，参加本书编写的人员还有耿超、李俊艳、向春阳、王亚敏、丁雷道、张亚楠、陶永才、史晓东、李冬芳、谢琦、高宇飞、吴保东、张丹丹、韩颖、王战红、姚瑶、段赵磊等人。同时，对清华大学出版社表示感谢。

本书的电子教案、习题答案和实例源文件可以到 <http://www.tupwk.com.cn/downpage/index.asp> 网站下载。

由于时间较紧，书中难免有错误与不足之处，恳请专家和广大读者批评指正。在编写本书的过程中参考了相关文献，在此向这些文献的作者深表感谢。我们的电话是 010-62796045，信箱是 huchenhao@263.net。

编 者

2014 年 12 月

目 录

第 1 章 XML 简介	1
1.1 XML 的产生	1
1.1.1 SGML 的诞生	1
1.1.2 什么是 XML	2
1.1.3 XML 和 HTML 的不同	4
1.2 XML 的现状及其发展	6
1.2.1 XML 应用领域	6
1.2.2 XML 发展前景	7
1.3 XML 相关技术	10
1.4 XML 编辑工具	15
1.4.1 普通文本编辑工具	15
1.4.2 本书的开发环境	16
1.4.3 XML Spy 简介	16
1.4.4 使用 XML Spy 编辑 XML 文档	17
1.4.5 XML Spy 的视图格式	20
1.5 本章小结	21
1.6 思考和练习	21
第 2 章 格式良好的 XML 文档	22
2.1 XML 文档的分类	22
2.1.1 格式不良的 XML 文档	23
2.1.2 格式良好的 XML 文档	23
2.2 XML 文档的整体结构	24
2.3 XML 声明	26
2.3.1 XML 声明中的 version 属性	26
2.3.2 XML 声明中的 encoding 属性	26
2.3.3 XML 声明中的 standalone 属性	27

2.4 XML 文档的处理指令和 注释	27
2.4.1 处理指令	27
2.4.2 注释	28
2.5 XML 元素的基本规则	29
2.5.1 XML 元素的命名规则	29
2.5.2 根元素	29
2.5.3 元素的构成	30
2.5.4 元素的嵌套	31
2.5.5 元素的属性	33
2.6 实体引用和 CDATA 段	34
2.6.1 实体引用	35
2.6.2 CDATA 段	36
2.7 命名空间	37
2.7.1 有前缀和无前缀命名空间	38
2.7.2 在标记中声明命名空间	39
2.7.3 命名空间的作用域	40
2.8 本章小结	40
2.9 思考和练习	41
第 3 章 有效的 XML 文档——DTD	42
3.1 DTD 概述	42
3.2 DTD 的基本结构	43
3.2.1 内部 DTD	43
3.2.2 外部 DTD	44
3.2.3 DTD 的基本结构	45
3.3 DTD 元素定义	45
3.3.1 元素定义	45
3.3.2 元素类型	45
3.4 DTD 属性说明	48
3.4.1 声明属性的语法	48
3.4.2 属性的默认值	49

3.4.3 属性的类型	50	5.2.3 CSS 的创建与应用	85
3.5 DTD 实体声明	54	5.3 CSS 基本语法	87
3.5.1 实体的概念和分类	54	5.3.1 定义样式	87
3.5.2 通用实体	55	5.3.2 对 XML 文档有效的 CSS 选择符	88
3.5.3 参数实体	56	5.4 XML 与 CSS 结合的方式	89
3.6 DTD 现状和 Schema 的优势	57	5.4.1 调用外部样式表文件	89
3.6.1 DTD 现状	57	5.4.2 在 XML 文档内部定义 样式	90
3.6.2 Schema 的优势	58	5.4.3 混合方法指定样式	91
3.7 本章小结	59	5.4.4 使用多个样式文件	91
3.8 思考和练习	59	5.5 CSS 属性	92
第 4 章 有效的 XML 文档 ——Schema	60	5.5.1 字体属性	93
4.1 Schema 概述	60	5.5.2 文本属性	93
4.2 XML Schema 的基本结构	61	5.5.3 颜色和背景属性	94
4.2.1 XML Schema 文档示例	61	5.5.4 设置文本的显示方式	95
4.2.2 XML Schema 的主要组件	63	5.6 CSS 的显示规则	96
4.3 XML Schema 中的数据类型	67	5.7 本章小结	97
4.3.1 简单类型	67	5.8 思考和练习	98
4.3.2 复杂类型	72	第 6 章 使用 XSL 显示 XML 文档	100
4.4 XML Schema 的命名空间	74	6.1 XSL 概述	100
4.4.1 名称重复	74	6.1.1 CSS 的局限性及 XSL 的特点	100
4.4.2 命名空间	74	6.1.2 XSL 的构成	101
4.4.3 使用命名空间	75	6.1.3 XSL 转换入门	102
4.5 XML 有效性的验证	76	6.2 XSL 文档结构	103
4.5.1 使用开发工具验证	76	6.2.1 创建一个 XSL 实例	103
4.5.2 编程验证	77	6.2.2 XSL 入门	106
4.6 本章小结	80	6.3 XSL 模板	107
4.7 思考和练习	80	6.3.1 使用 template 元素定义 模板	107
第 5 章 使用 CSS 显示 XML 文档	83	6.3.2 使用 apply-templates 元素 处理子节点	108
5.1 样式表概述	83	6.3.3 XSL 的默认模板规则	112
5.1.1 显示 XML 的两种 常用样式表	83	6.3.4 使用命名模板	113
5.1.2 样式表的优势	84	6.4 XSLT 的元素	113
5.2 CSS 简介	85		
5.2.1 CSS 基本概念	85		
5.2.2 CSS 的历史	85		

6.4.1 使用 <code>xsl:value-of</code> 获得 节点值.....	113	8.2 SAX 的特点.....	149
6.4.2 使用 <code>xsl:for-each</code> 处理 多个元素.....	115	8.3 SAX 工作机制.....	150
6.4.3 使用 <code>xsl:sort</code> 对输出 元素排序.....	118	8.3.1 事件处理器.....	150
6.4.4 用于选择的元素 <code>xsl:if</code> 和 <code>xsl:choose</code>	119	8.3.2 SAX 事件.....	151
6.5 XSL 的模式语言.....	122	8.3.3 SAX 常用接口.....	152
6.5.1 相对路径和绝对路径.....	122	8.3.4 SAX 回调方法.....	154
6.5.2 匹配节点的模式.....	122	8.4 使用 SAX 解析 XML.....	155
6.6 使用 XMLSpy 管理 XSL 操作.....	127	8.4.1 SAX 解析 XML 文档.....	155
6.7 本章小结.....	129	8.4.2 处理空白.....	156
6.8 思考和练习.....	129	8.4.3 实体.....	156
第 7 章 XML 解析器——DOM.....	132	8.5 SAX 错误信息.....	156
7.1 DOM 概述.....	132	8.6 SAX 与 DOM.....	157
7.2 DOM 的结构.....	133	8.7 本章小结.....	160
7.3 节点类型.....	135	8.8 思考和练习.....	160
7.4 DOM 基本接口.....	136	第 9 章 XML 与数据库.....	161
7.4.1 Node 接口.....	137	9.1 XML 与数据库技术的发展.....	161
7.4.2 Document 接口.....	137	9.1.1 数据库技术的发展.....	162
7.4.3 NodeList 接口.....	139	9.1.2 XML 与数据库技术的 结合.....	163
7.4.4 NamedNodeMap 接口.....	139	9.1.3 XML 在数据库中的 应用模式.....	163
7.4.5 Element 接口.....	139	9.2 XML 的数据交换与 存储机制.....	164
7.4.6 Text 接口.....	141	9.2.1 XML 的数据交换机制.....	164
7.5 DOM 的使用.....	141	9.2.2 XML 的数据交换类型.....	165
7.5.1 修改 XML 文档.....	141	9.2.3 XML 的数据存取机制.....	166
7.5.2 生成 XML 文档.....	143	9.2.4 XML 数据交换技术的 工程应用.....	167
7.5.3 处理空白.....	145	9.3 XML 与数据库的数据 交换技术.....	168
7.5.4 验证格式良好与有效性.....	146	9.3.1 ADO.NET 简介.....	168
7.6 浏览器对 DOM 的支持.....	146	9.3.2 .NET 中的 XML 特性.....	170
7.7 本章小结.....	146	9.3.3 从数据库到 XML 文档.....	171
7.8 思考和练习.....	147	9.3.4 从 XML 文档到数据库.....	179
第 8 章 XML 解析器——SAX.....	148	9.4 SQL Server 2005 对 XML 的支持.....	182
8.1 SAX 简介.....	148		

9.4.1 SQL Server 2005 对 XML 的支持	182
9.4.2 XML 数据类型	183
9.4.3 XML 类型的方法	184
9.4.4 发布 XML 数据	185
9.4.5 在表中插入 XML 数据	188
9.5 本章小结	190
9.6 思考和练习	191
第 10 章 基于 XML 的论坛开发	192
10.1 系统功能分析	192
10.1.1 论坛功能	192
10.1.2 系统模块	193
10.2 论坛系统 XML 文件的设计	193
10.2.1 users.xml	194
10.2.2 section.xml	195
10.2.3 topic.xml	197
10.2.4 reply.xml	198
10.3 访问 XML 数据的公共类	198
10.3.1 系统配置	199
10.3.2 两个基本公共类	199
10.3.3 用户信息访问类	200
10.3.4 版块信息访问类	204
10.3.5 帖子信息访问类	209
10.3.6 回复信息访问类	215
10.4 帖子相关模块的设计与实现	217
10.4.1 帖子的浏览	217
10.4.2 特定帖子回复的浏览	223
10.4.3 已登录用户发表新帖	225
10.4.4 已登录用户回复旧帖	226
10.5 用户信息模块的设计与实现	227
10.5.1 用户注册	227
10.5.2 会员登录	230
10.5.3 会员注册信息查询与修改	231
10.5.4 会员发帖或回复信息查询与管理	234
10.6 管理模块的设计与实现	237
10.6.1 管理员登录	237
10.6.2 版块管理	237
10.6.3 帖子管理	242
10.6.4 其他管理	244
10.7 本章小结	246
10.8 思考和练习	246
参考文献	247

第1章 XML简介

在互联网的发展历史上，有两种非常核心的技术，这就是 Java 和 XML。Java 提供了程序代码的平台无关性，而 XML 则保证了数据的平台无关性，被誉为因特网上的世界语，已成为 Web 应用中数据表示和数据交换的标准。但是，人们对 XML 的认识远远没有对 HTML 的认识彻底和清晰。那么，究竟什么是 XML？XML 和 HTML 有什么不同，它们的本质区别是什么？同时，由于 XML 的优越性及 XML 的不断发展壮大，XML 下的标准和规范不断变化，了解这些标准的来龙去脉以及它们之间的关系，对于掌握 XML 是至关重要的。

本章首先介绍标记语言的发展历史，在与有关标记语言比较的基础上，引出 XML 语言，然后对 XML 语言的特点、作用以及与之相关的技术进行简要的介绍。通过本章的学习，读者将会了解到 XML 技术的具体含义及其广阔的应用前景。

本章的学习目标：

- 掌握 XML 的特点
- 理解 XML 与 HTML 的不同
- 了解 XML 的应用领域
- 掌握 XML 的技术规范
- 熟悉 XML 文档的编辑软件

1.1 XML 的产生

XML 的全称是 Extensible Markup Language，意思是可扩展的标记语言，它是 SGML 的一个子集，现在广为使用的 HTML 也是 SGML 家族中的一员。

HTML、XML 以及 SGML 都属于标记语言，标记语言不是像 Java、C 一样的编程语言，它本身并无任何“动作行为”。标记语言只是用一系列约定好的标记来对电子文档进行标记，从而为电子文档额外增加语义、结构和格式等方面的信息。标记语言比编程语言简单得多，只需按规定为文本文件添加一些特殊标记即可，这些特殊标记用于传递更多额外的信息。

1.1.1 SGML 的诞生

在 20 世纪 60 年代，IBM 的研究人员提出在各文档之间共享一些相似的属性，如字体大小和版面。IBM 设计了一种文档系统，通过在文档中添加标记，来标识文档中的各种元

素, IBM 把这种标记语言称作通用标记语言(Generalized Markup Language), 即 GML。

由于在当时的信息交换过程中, 经常会发生数据格式不同的问题, 随着网络技术的不断发展, 这一问题日益严重, 制约了人们的信息交流。经过若干年的发展, 1984 年国际标准化组织(ISO)开始对此提案进行讨论, 并于 1986 年整理为 SGML(Standard Generalized Markup Language), 即标准通用标记语言。SGML 是一种定义电子文档结构和描述其内容的国际标准语言, 是所有电子文档标记语言的起源, 早在 Web 发明之前就已存在。SGML 具有良好的扩展性和可移植性, 在任何一种环境下都可以正常使用。但 SGML 强大功能的背后是它的复杂度太高, 不适合网络的日常应用。另外, SGML 价格昂贵, 开发成本高。更为重要的是, 它不被主流浏览器厂商所支持。这些原因均使得 SGML 的推广受到了阻碍。标记语言的发展历史如图 1-1 所示。

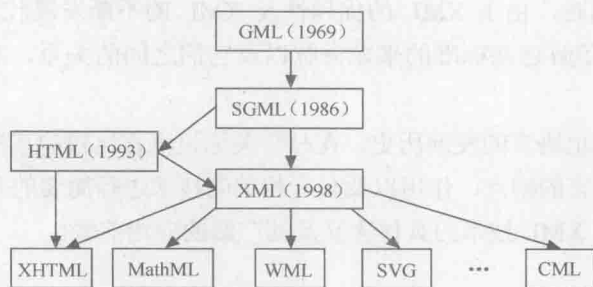


图 1-1 标记语言的发展历史

1.1.2 什么是 XML

超文本标记语言 HTML(Hypertext Markup Language)是目前网络上应用最广泛的语言, 也是构成网页文档的主要语言。HTML 里有很多标记, 都是在 HTML 4.0 里规范和定义的, 而 XML 允许用户自己创建这样的标记, 所以叫做可扩展性。XML 文件是由标记以及它所包含的内容构成的文本文件, 这些标记可自定义, 其目的是使得 XML 文件能够很好地体现数据的结构和含义。W3C 推出 XML 的主要目的是使得 Internet 网络上的数据相互交流更方便, 让文件的内容更加显而易见。

XML 同 HTML 一样, 都来自 SGML。SGML 十分庞大, 既不容易学习, 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——欧洲粒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, 如 HTML 4.0 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。首先, HTML 可以指定一个文档的内容和格式, 但不能指定文档的结构。也就是说, HTML 是面向表示的标记语言, 用来告诉浏览器如何在网站上显示信息, 而非面向结构。其次, HTML 只能应用于信息的显示, 它可以使文本加粗, 以斜体或下划线形式显示, 但它几乎没有语义结构。

HTML 显示数据是按照布局而非按照语义。随着网络应用的飞速发展, 各行各业对各种信息有着不同的需求, 这些不同类型的信息未必都是以网页的形式显示出来。例如, 当通过搜索引擎进行数据搜索时, 按照语义而不是按照布局来显示数据显然会具有很多优点。再次, HTML 可扩展性差, HTML 中标记的名称是固定不变的, 因而其提供的功能与使用的属性也是固定的。因此, HTML 不允许网页设计者自行创造标记。例如, HTML 文档包括了格式化、结构和语义的标记。就是 HTML 中的一种格式化标记, 它使其中的内容变为粗体; <TR>也是 HTML 中的一种结构标记, 指明内容是表格中的一行。也就是说, HTML 不是一种元语言, 不能创建某一特定领域的标记集。虽然作为一般的应用, HTML 已经够用了, 但科学家无法用 HTML 书写数学公式、化学方程式以及分子晶体结构, 这样使它的发展受到了极大的限制。

总而言之, HTML 的缺点使其交互性差, 语义模糊。随着互联网应用发展的需求, HTML 越来越难以满足网络数据交互和业务集成的需求。

有人建议直接使用 SGML 作为 Web 语言, 这固然能解决 HTML 遇到的困难。但是 SGML 太庞大了, 用户学习、使用不方便尚且不说, 要全面使用 SGML 的浏览器就非常困难, 于是自然想到只使用 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——XML 应运而生了。

XML 是 SGML 的一个精简的子集, 其复杂程度大约只有 SGML 的 20%, 但却有 SGML 80% 的功能, 因此它一经推出即受到用户的欢迎。XML 保留了 SGML 的可扩展功能, 这使 XML 从根本上有别于 HTML。XML 是一种元标记语言, 要比 HTML 强大得多, 它不再是固定的标记, 而是要用户根据描述数据的需要自己定义各种标记。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, 在 HTML 中, 一首歌可能是用定义标题标记<dt>、定义数据标记<dd>、无序列表标记和列表项标记来描述的。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>春暖花开
<ul>
  <li>词: 梁芒
  <li>曲: 洪兵
</ul>
```

而在 XML 中, 同样的数据可能标记如下。

```
<song>金曲 TOP1
<title>春暖花开</title>
<composer>洪兵</composer>
  <lyricist>梁芒</lyricist>
</song>
```

在这段代码中没有使用通用的标记如<dt>、等, 而是使用了更有意义的标记, 如

<song>、<title>、<composer>等。这种用法使源代码易于阅读，使人能够看出代码的意义。

XML 具有以下特点。

- XML 描述的是结构和语义，而不是格式化。
- XML 将数据内容和显示格式相分离。
- XML 是元标记语言。XML 的标记不是预先定义好的，而是自定义的。
- XML 是自描述语言。XML 使用 DTD 或者 Schema 后就是自描述的语言。XML 文档通常包含一个文档类型声明，因而 XML 文档是自描述的；不仅人能读懂 XML 文档，计算机也能处理。
- XML 是独立于平台的。
- XML 不进行任何操作。
- XML 具有良好的保值性。XML 良好的保值性和自描述性使它成为保存历史档案，如政府文件、公文、科学研究报告等的最佳选择。

XML 标准的发展没有 HTML 那样迅速，直到 1998 年 2 月，W3C 才发布了 XML 1.0 推荐标准，又于 2004 年 2 月，发布了 XML 1.1 的推荐标准，这是最新的 XML 版本，不过目前大多数的应用遵循的还是 W3C 于 2000 年 10 月 6 日发布的 XML 1.0 标准。

1.1.3 XML 和 HTML 的不同

从前面的介绍中，可以感觉出 HTML 和 XML 的明显不同。HTML 标记用途很简单，也很明确，就是使用 HTML 标记创建的文档可以用浏览器显示相似的内容，并显示美观的网页编排。

而 XML 则属于一种文档格式的革命，它能让用户自行定义文档结构，给予文档一种全新的生命，让计算机能够读懂文档。XML 的设计目的是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，从而提高处理数据的效率和灵活性。

下面对二者之间的差异进行比较。

(1) XML 和 HTML 都来自于 SGML，它们都含有标记，有着相似的语法，区别在于：HTML 不具有扩展性，它用固有的标记来描述、显示网页内容。例如，<H1>是第一级标题标记，有固定的尺寸——20 磅的 Helvetica 字体的粗体。如果 HTML 语言没有定义用户所需的标记，用户就没有办法了。这时只能等待 HTML 的下一个版本，希望在新版本中能包括所需的标记。相对的，XML 是元标记语言，可用于定义新的标记语言。如果将 HTML 看成是在织毛衣，那么 XML 就是关于如何织毛衣的指导书。学会 XML，用户不仅可以织毛衣，还可以织袜子、手套等。

(2) HTML 的核心不是为了体现数据的含义，而是为了体现数据的显示格式。HTML 网页将数据和显示混在一起，而 XML 则将数据和显示分开来。XML 的核心是描述数据的组织结构，让 XML 可以作为数据交换的标准格式。由于 XML 文档本身不受表现形式的羁绊，只要对 XML 文档作适当的转换，就可以将其变成不同的形式，如网页、PDF 文档和 Word 文档等，达到“一次编写，多处使用”的目的，提高了内容的可重用性。

(3) 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”。

下面这些语句在 HTML 中随处可见。

```
<b><i>sample</b></i>1-  
<td>sample</TD>  
<font color=red>samplar</font>
```

而在 XML 文档中，上述几种语句的语法都是错误的。XML 严格要求嵌套、配对和遵循 DTD 的树形结构。

XML 和 HTML 的更多不同在表 1-1 中进行了详细对比。

表 1-1 XML 和 HTML 的对比

比较内容	HTML	XML
是否预置标签	预置大量标签	自定义标签
可扩展性	不具有可扩展性	是元标记语言，可用于定义新的标记语言，具有很好的可扩展性
侧重点	侧重于如何表现信息	侧重于传输和存储数据，核心是数据本身
语法要求	松散、不严格	严格要求嵌套、配对，并遵守 DTD 或 Schema 定义的语义约束
可读性及可维护性	难以阅读和维护	结构清晰，便于阅读和维护
数据和显示的关系	数据与显示混为一体，难以分离	数据与显示分离
与数据库的关系	与数据库没有关系	与关系型数据库的数据表对应，可进行转换
是否区分大小写	大部分浏览器不区分大小写	严格区分大小写
编辑工具	文本编辑工具，大量所见即所得的编辑器(如 Dreamweaver)	文本编辑工具，大量 XML 编辑器(如 XMLSpy)
处理工具	任何浏览器都可	需要专门的程序进行处理

下面再通过具体的实例将 HTML 和 XML 进行对比，【例 1-1】中的 example1_1.html 是一个简单的 HTML 文件。

【例 1-1】 example 1_1.html 文件的源代码如下。

```
<html>  
<head>  
  <title>订单信息</title>  
</head>  
<body>  
  <h1>订单号: 1001</h1>  
  <h2>商品名称: 运动服</h2>  
  <h2>单价: 200 元</h2>  
  <h2>数量: 15 双</h2>  
</body>  
</html>
```

上面的标记,如<html>、<head>、<body>、<h1>等都是固定的,而在创建一个 XML 文档时,则可以由用户自己定义各种标记并以任何名字为它们命名。与之对应的 XML 文件 example1_1.xml 如下。

```
<?xml version="1.0" encoding="gb2312"?>
<订单>
  <订单号>1001</订单号>
  <商品名称>运动服</商品名称>
  <单价>200</单价>
  <数量>15</数量>
</订单>
```

从 example1_1.xml 中可以很清楚地看出数据的组织结构,所以 XML 文档其实什么都不做,它只是用 XML 标记存储信息的文件。

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。事实上,它们是基于两个不同的目标而开发的。HTML 着重于如何描述将文件显示在浏览器中,XML 和 SGML 相近,着重于如何描述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。

1.2 XML 的现状及其发展

XML 具有许多优良的特性,并且使用方便,因此受到了越来越多的欢迎。目前,许多大公司和开发人员已经开始使用 XML,包括 B2B 在内的很多优秀应用都已经证实了 XML 将会改变今后创建应用程序的方式。当然,XML 的意义远非如此,其潜在的影响是无穷的。

1.2.1 XML 应用领域

XML 在实际使用的过程中发挥着巨大的作用。目前,越来越多的行业开始采用 XML 来实现特定的功能。

XML 的用途主要包括以下几个方面。

1. 从 HTML 中分离数据

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。HTML 只要做好数据的显示和布局,这样数据改动时不会导致 HTML 文件也需要改动。

2. 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,并且还可以使得这些数据能被不同的程序读取,如图 1-2 所示。

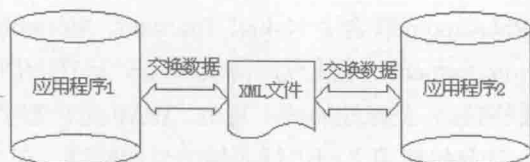


图 1-2 XML 实现不同应用程序之间的数据交互

3. 存储和共享数据

XML 提供了一种与软件和硬件无关的存储和共享数据的方法,大量的数据可以存储到 XML 文件或者数据库中。应用程序可以读写和存储数据,一般的程序可以显示数据。

4. 充分利用数据

XML 是与软、硬件和应用程序无关的,所以可以使数据可以被更多的用户、更多的设备所利用,而不仅仅是基于 HTML 标准的浏览器。别的客户端和应用程序可以把 XML 文档作为数据源来处理,就像他们对待数据库一样,设计者的数据可以被各种各样的“阅读器”处理。

5. 创建新的语言

利用 XML 可以设计与特定领域有关的标记语言,如 MusicML、MathML、CML、SVG、WML、SMIL 等。XML 允许各种不同的专业(如音乐、化学、数学等)开发与自己的特定领域有关的标记语言,这就使得该领域的人们可以交换笔记、数据和信息。

XML在数学领域中的应用称为数学标记语言(Mathematical Markup Language, 简称 MathML), MathML适合描述数学方程式,利用它数学家们第一次可以把数学公式精确地显示在浏览器上。化学标记语言CML(Cheical Markup Language)可能是第一个XML应用,可以描述分子等信息。

1.2.2 XML 发展前景

XML 自从 1998 年 2 月成为推荐标准后,许多厂商加强了对它的支持力度,包括 Microsoft、IBM、Oracle、Sun 等,它们都推出了支持 XML 的产品或改造原有的产品以支持 XML, W3C 也一直在致力于完善 XML 的标准体系。作为互联网的新技术,XML 的应用非常广泛,可以说 XML 已经渗透到了互联网的各个角落。

XML 的开放性、严谨性、灵活性和结构性备受网络开发者的青睐。Web 的飞速发展给予了 XML 充分展示自我的空间,它提供给使用者更为强大的功能,带给程序员更为便利的开发环境。在以下领域,XML 将一展风采。

1. 移动通信领域

随着移动电话与互联网结合,无线上网的趋势正在形成,有人预言,随着无线带宽的增加和无线上网技术的迅速发展,.move 将代替.com 成为新的潮流。为了满足人们随时随