



编程实例教程系列

XML 编程 实例教程

北京希望电子出版社 总策划
丘广华 张文敏 编 著

 科学出版社
www.sciencep.com



编程实例教程系列

XML 编程 实例教程

北京希望电子出版社 总策划
丘广华 张文敏 编 著

 科学出版社
www.sciencep.com

内 容 简 介

这是一本用实例介绍 XML 编程技术的专著。

本书共由 10 章内容构成。主要内容如下：XML 基础入门，文档类型定义 DTD，命名空间和属性的声明，层叠样式表 CSS，层叠样式表 CSS2，初识 XSL，XSL 高级应用，XML 结合 JavaScript 的应用，XML 存储结构以及整合 ASP 的应用，XML 的高级应用。

本书贯穿了“用实例说明问题”的指导思想，在各章的每一小节的最后都有一个完整的范例对相应知识点进行概括和整合，因此本书实用性和指导性较强，适合了解 XML 编程基本概念的初、中级软件开发人员参考使用。

需要本书或技术支持的读者，请与北京中关村 083 信箱（邮编 100080）发行部联系。电话：010-62528991，62524940，62521921，62521724，82610344，82675588（总机） 传真：010-62520573，E-mail：yanmc@bhp.com.cn

图书在版编目（CIP）数据

XML 编程实例教程/丘广华，张文敏 编著. —北京：
科学出版社，2004.4

ISBN 7-03-012780-3

I. X... II. ①丘...②张... III. 可扩充语言，XML
—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字（2004）第 003771 号

责任编辑：周艳

/ 责任校对：周凤明

责任印刷：双青

/ 封面设计：梁运丽

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2004 年 4 月第一版

开本：787×1092 1/16

2004 年 4 月第一次印刷

印张：27 1/2

印数：1—5 000

字数：634 296

定价：39.00 元

前 言

XML (Extensible Markup Language, 可扩展标记语言) 作为一种新兴的计算机技术, 已经在短短的几年间对互联网产生了深刻的影响。它使 Internet 应用程序的编写方式在基本结构的层面上发生了变革。一时间, XML 成为了计算机时尚技术中一个很抢眼的词汇, 各大软件公司纷纷将自己的相关产品向 XML 的大旗靠拢。

对于从未接触过 XML 的读者来说, 要理解 XML 技术中的许多概念是有一定困难的。相比在 Internet 上应用得更为广泛、也更为人们所熟知的标记语言—HTML, XML 的语法和规则都要严谨和复杂一些。这些问题也是许多读者在开始学习 XML 以及学习的进阶时期最常见的困难。而本书正是针对这些问题来组织相关知识点和实例进行讲解的。

本书主要包括了 4 个部分: XML 基础、样式表技术、XML 综合技术运用以及 XML 高级编程。首先, 本书从 XML 诞生的背景入手, 详细介绍了 XML 标准体系的作用和意义。并结合实例一步步阐明了 XML 的基本概念和语法基础。接着, 在样式表的章节中, 针对 CSS、XSL 两大样式表技术, 通过大量的实例向读者介绍了它们的基本语法以及实用技巧。在这 2 个部分的基础上, 本书进一步展开了 XML 综合技术的介绍。在这个知识单元中, 我们将 Internet 上最为实用的页面脚本编程和 ASP 技术整合到 XML 的应用中, 并结合 XML 的高级应用属性生成了一些功能较为强大的实用页面程序。这些应用实例是通过讲解过程中知识点的不断积累而逐步完善的, 从而方便读者更好地了解和掌握相应的技术要领。在最后部分, 是 XML 一些高端技术的应用, 这些章节中知识点的学习是建立在之前各章内容基础之上的, 所以采用了高密度、大信息量的方式进行介绍, 以便在有限的篇幅内尽可能多的将之前学习的相对分散的 XML 知识点, 与互联网中运作实用技术紧密联系在一起。

对具体的章节而言, 本书的 10 章内容都有着独立的主题。而在学习过程中, 这些主题又是逐步深入的。XML 入门部分和样式表语法为综合技术运用奠定了基础, 而这些基本语法和技术特点也正是在综合性范例中得到了充分的体现。

在讲解的方式上, 本书始终贯穿了“用实例说明问题”的编写思想, 努力追求将每一个知识点的理论内容都通过相应的例子来进行形象的说明。因此, 本书的各章节中在介绍语法及使用技巧的同时, 都匹配了简单的例子; 而且在每一小节的最后, 都有一个完整的范例对小节的知识点进行概括和整合, 以加深读者对这些知识的理解。

如果读者想要在 Internet 上访问到更多 XML 的信息以补充从本书中学到的内容, 可以参考 W3C 在 <http://www.w3.org> 中专门为网页制作人员提供的各种信息和服务。另外, Microsoft Developer Network (微软开发商网络, MSDN) 在 <http://msdn.microsoft.com/xml/default.asp> 中也提供了 Microsoft 产品所支持的、有关 XML 的大量信息。本书是专门为 Microsoft Internet Explorer 5.0~6.0 编写的。读者可以从 <http://www.microsoft.com/windows/ie> 下载最新的 Internet Explorer 6.0。在整本书中, 当某些信息属性只适用于某个 Internet Explorer 版本时, 也会在相应的位置及时指出。

本书由丘广华、张文敏和农超组织编写, 其他参加本书的编写、录排、校对工作的人员还有: 罗贤锋、葛丽、刘卫宏、刘向军、田野、刘聪、田军、张丽、陈曙辉、邓波、王

东霞、李志、张巧莉、赵军锁、姜南、龚志翔、汪建华、郭祥雷、赵文辉、黄晓润、车自友、陈培等。龚超同志对全稿进行了严格细致的复审。本书的出版是集体劳动的结晶。还要特别感谢希望电子出版社第三编辑室的老师们，他们为本书的出版付出了很多心血！

由于时间仓促，且作者的经验和水平有限，文中难免有不妥之处，我们殷切地期望你能给我们提出中肯的意见，以便于提高水平，把更好的图书呈现给大家！（联系方式：admin_mail@163.com）。

作者

目 录

第 1 章 XML 基础入门.....	1	2.2.3 元素内容声明.....	49
1.1 什么是 XML.....	1	2.2.4 元素的一些规则.....	50
1.1.1 标记语言.....	1	2.2.5 实例 2-3——推荐丛书的例子	52
1.1.2 SGML.....	2	2.3 实体	54
1.1.3 HTML.....	2	2.3.1 通用内部解析实体.....	55
1.1.4 XML.....	5	2.3.2 参数内部解析实体.....	57
1.1.5 XML 与 SGML、HTML 的关系.....	6	2.3.3 实例 2-4——网上书店的例子	58
1.1.6 实例 1-1——分别用 XML 和 HTML 实现的人事档案.....	9	2.4 创建外部实体	61
1.2 XML 的结构和语法.....	16	2.4.1 通用外部解析实体.....	61
1.2.1 XML 文档结构.....	16	2.4.2 通用外部未解析实体.....	62
1.2.2 实例 1-2——交易清单的例子.....	20	2.4.3 参数外部解析实体.....	64
1.2.3 XML 标记语法.....	23	2.4.4 实例 2-5——网上商城的例子	64
1.2.4 XML 文档语法.....	25	2.5 包含及忽略 DTD 片断	68
1.2.5 实例 1-3——嵌套 HTML 源代码的 XML 文档.....	31	2.5.1 使用注释.....	68
1.2.6 建立正规有效的 XML 文档.....	34	2.5.2 使用 INCLUDE 和 IGNORE.....	71
1.2.7 CSS 与 XSL.....	35	2.5.3 实例 2-6——管理球员资料	72
1.3 XML 的应用.....	36	2.6 本章总结	76
1.4 XML 应用工具.....	37	第 3 章 命名空间和属性的声明	77
1.4.1 XML 编辑器.....	37	3.1 命名空间	77
1.4.2 XML 浏览器.....	38	3.1.1 什么需要命名空间.....	77
1.5 本章总结.....	38	3.1.2 创建命名空间.....	78
第 2 章 文档类型定义 DTD.....	40	3.1.3 缺省名称空间.....	79
2.1 文档类型定义.....	40	3.1.4 实例 3-1——职工资料表的例子	80
2.1.1 定义 DTD.....	41	3.2 属性的声明	82
2.1.2 定义外部 DTD.....	43	3.2.1 在 DTD 中声明属性.....	82
2.1.3 公用 DTD.....	44	3.2.2 属性类型.....	83
2.1.4 实例 2-1——用内部 DTD 实现医疗病历.....	45	3.2.3 实例 3-2——宠物交易	87
2.1.5 实例 2-2——用外部 DTD 实现扩展的医疗病历.....	46	3.3 属性的高级应用	90
2.2 元素.....	48	3.3.1 属性的缺省值.....	90
2.2.1 元素简介.....	48	3.3.2 2 个特殊属性.....	92
2.2.2 元素声明.....	49	3.3.3 实例 3-3——NBA 球队资料	93
		3.4 本章总结	96
		第 4 章 层叠样式表 CSS.....	98
		4.1 层叠样式表 CSS	98
		4.1.1 CSS 的概念.....	98

4.1.2 CSS 的基本语法格式.....	99	5.3.1 列表符号的设定.....	173
4.1.3 实例 4-1——CSS 带来的 崭新歌词界面.....	104	5.3.2 实例 5-3——家常菜谱 页面的制作.....	177
4.2 设置字体.....	107	5.4 浮动框和滚动条.....	181
4.2.1 font 属性集.....	108	5.4.1 元素单元的布局.....	181
4.2.2 font 属性集的子属性.....	109	5.4.2 对元素设置滚动条属性.....	184
4.2.3 display 属性.....	113	5.4.3 实例 5-4——影视人物介绍.....	184
4.2.4 实例 4-2——论文格式的显示.....	113	5.5 完整的 CSS 样式表运用.....	188
4.3 设置色彩和背景.....	116	5.5.1 CSS 样式表属性设置.....	188
4.3.1 定义前景色.....	116	5.5.2 不同属性设置的配合使用.....	188
4.3.2 定义背景色.....	118	5.5.3 实例 5-5——综合网站页面设计.....	189
4.3.3 实例 4-3——主题网站页面的 制作.....	122	5.6 本章总结.....	197
4.4 设置边界.....	125	第 6 章 初识 XSL.....	199
4.4.1 margin 属性组.....	126	6.1 使用 XSL 显示 XML 文件.....	199
4.4.2 border 属性组.....	128	6.1.1 使用 XSL 转换 XML 的 基本步骤.....	200
4.4.3 实例 4-4——在线心理 测试的表单样式.....	133	6.1.2 XSL 文件的基本元素.....	201
4.5 元素填充区设定和段落排版.....	137	6.1.3 实例 6-1——仙乐飘飘.....	203
4.5.1 元素边界填充设定.....	137	6.2 处理多个选择.....	207
4.5.2 文本效果设定.....	139	6.2.1 用 for-each 处理循环.....	207
4.5.3 单词、字符间距设定.....	142	6.2.2 应用模板.....	209
4.5.4 实例 4-5——一篇中英文 对照散文的段落排版.....	142	6.2.3 实例 6-2——会议记录.....	210
4.6 元素的选择.....	146	6.3 为 match 属性指定匹配模式.....	215
4.6.1 使用 ID 选择符.....	146	6.3.1 匹配模式.....	215
4.6.2 实例 4-6——列车时刻表的制作.....	149	6.3.2 实例 6-3——环宇影视.....	219
4.7 本章总结.....	154	6.4 使用 XPath 表达式.....	225
第 5 章 层叠样式表 CSS2.....	155	6.4.1 XPath 节点的种类.....	225
5.1 CSS2 的新增功能.....	155	6.4.2 XPath 轴线.....	226
5.1.1 CSS2 中的新特点.....	155	6.4.3 基本的 XPath 表达式.....	226
5.1.2 使用 CSS2 的新增属性.....	156	6.4.4 基本的 XPath 表达式.....	227
5.1.3 实例 5-1——用户登录 界面的制作.....	160	6.4.5 实例 6-4——学生选课情况表.....	228
5.2 元素空间定位.....	164	6.5 使用 XPath 谓词断言.....	232
5.2.1 相对定位和绝对定位.....	164	6.5.1 节点集类型.....	232
5.2.2 设定元素大小.....	166	6.5.2 字符串类型.....	233
5.2.3 实例 5-2——十二星座图的制作.....	169	6.5.3 数值类型.....	234
5.3 元素列表属性.....	173	6.5.4 布尔类型.....	234
		6.5.5 实例 6-5——学生成绩情况表.....	235
		6.6 本章总结.....	240
		第 7 章 XSL 高级应用.....	242

7.1 转换为 HTML	242	8.3.2 transformNode 属性.....	308
7.1.1 DOMDocument 对象.....	242	8.3.3 实例 8-3——控制通讯录 显示方式.....	309
7.1.2 基本步骤.....	244	8.4 XSL 的循环技术.....	315
7.1.3 实例 7-1——CD 大卖场.....	245	8.4.1 使用脚本语言的循环过程 处理信息.....	316
7.2 条件处理.....	250	8.4.2 实例 8-4——可控循环 列表的制作.....	318
7.2.1 使用 xsl:if.....	250	8.5 结合数据源属性的 XML.....	324
7.2.2 使用 xsl:choose.....	251	8.5.1 SPAN 元素对 XML 数据岛的支持.....	324
7.2.3 实例 7-2——股票行情分析.....	253	8.5.2 TransformNode 对象	325
7.3 对输出元素排序.....	258	8.5.3 实例 8-5——有翻页 功能的通讯录.....	326
7.3.1 使用 order-by 属性	258	8.6 检索 XML 节点信息.....	332
7.3.2 使用 xsl:sort 元素.....	260	8.6.1 使用 XSL 搜索元素	332
7.3.3 实例 7-3——为学生资料排序.....	261	8.6.2 使用 XML 数据岛配合 XSL 搜索元素	335
7.4 决定输出内容.....	266	8.6.3 实例 8-6——搜索通讯录	335
7.4.1 使用 xsl:comment.....	266	8.7 本章总结	341
7.4.2 使用 xsl:text.....	268	第 9 章 XML 存储结构以及整合 ASP 的应用	342
7.4.3 使用 xsl:element	269	9.1 处理交互式信息	342
7.4.4 使用 xsl:attribute.....	270	实例 9-1——心理测试程序	342
7.4.5 实例 7-4——变换客户资料.....	271	9.2 脚本语言与 XML 的存储结构.....	353
7.5 使用 eval 元素.....	276	实例 9-2——随机测试的生成	353
7.5.1 在 XSL 中使用 Script.....	276	9.3 使用服务器端语言	362
7.5.2 使用 XSL 运行时方法	279	9.3.1 服务器端语言简介	362
7.5.3 实例 7-5——消费清单.....	280	9.3.2 ASP 语言简介	363
7.6 联合样式表.....	285	9.3.3 ASP 内建对象	363
7.6.1 使用 xsl:import	285	9.3.4 使用 ASP 创建 XML DOM 对象.....	364
7.6.2 使用 xsl:include	287	9.3.5 实例 9-3——用 ASP 显示 XML 文件.....	365
7.6.3 实例 7-6——文章段落的显示.....	288	9.4 修改存储节点	369
7.7 本章总结.....	292	9.4.1 使用 Response 对象	369
第 8 章 XML 结合 JavaScript 的应用	294	9.4.2 使用 Request 对象.....	370
8.1 XML 综合技术的运用.....	294	9.4.3 修改 DOM 节点	371
8.1.1 页面制作中 XML 技术的 综合运用.....	294	9.4.4 实例 9-4——修改图书信息	372
8.1.2 XML 实用页面编程的结构.....	295	9.5 添加和删除节点	377
8.1.3 实例 8-1——通讯录的制作.....	296		
8.2 结合脚本语言的 XML 运用.....	301		
8.2.1 脚本语言	302		
8.2.2 JavaScript	302		
8.2.3 实例 8-2——通讯录的用户登录.....	303		
8.3 XML 数据岛的应用.....	308		
8.3.1 将 XML 数据链入 HTML.....	308		

9.5.1 使用表单接受用户信息.....	377	10.3 在服务器上操作 XML 文件.....	398
9.5.2 实例 9-5——完整的图书 信息管理系统.....	378	实例 10-3——在服务器上操作 XML 文件.....	399
9.6 本章总结.....	384	10.4 XML 网络留言板.....	403
第 10 章 XML 的高级应用.....	385	实例 10-4——XML 网络留言板.....	403
10.1 把数据库记录转换为 XML.....	385	10.5 投票调查系统.....	411
10.1.1 XML 的优势.....	385	实例 10-5——投票调查系统.....	411
10.1.2 转换注意事项.....	386	10.6 XML 网络聊天室.....	417
10.1.3 转换的 2 种基本方法.....	388	实例 10-6——XML 网络聊天室.....	417
10.1.4 实例 10-1——把数据库记录 转换为 XML.....	389	10.7 本章总结.....	425
10.2 实现上下文菜单.....	392	参考文献.....	427
实例 10-2——实现上下文菜单.....	392		

第 1 章 XML 基础入门

主要内容:

- XML 的概念。
- XML 语言的结构和语法。
- XML 的应用及处理工具。
- 实例 1-1——分别用 XML 和 HTML 实现的人事档案。
- 实例 1-2——交易清单的例子。
- 实例 1-3——嵌套 HTML 源代码的 XML 文档。

XML (Extensible Markup Language, 扩展标记语言) 作为一种专门在互联网上传递信息语言, 已经被广泛认为是继 Java 之后 Internet 上最激动人心的新兴技术。欢迎走进 XML 的世界。

XML 是由 W3C (World Wide Web Consortium, 万维网联盟) 定义的一种语言。相比当前更为大家所熟知的另一种在互联网上使用的标记语言 HTML (Hypertext Markup Language, 超文本标记语言), XML 极大地强化了保存信息和处理信息的能力。同时, 由于 XML 语法的灵活性和真正的“扩展”性, 使其能适应各种不同的数据类型: 从一条简单的字符串到一个庞大的数据库, 都是 XML 强大的信息描述能力可以一展身手的地方。

现在, 就让我们一步步熟悉 XML。

1.1 什么是 XML

语言的出现是人类文明进程的一个重要的里程碑, 而人们之所以可以用语言来进行信息的交流, 正是由于组成语言的字符是一种具有特定含义的标记。在这种标记具有了人们预定的、普遍认可的含意之后, 语言就成为了人们沟通交流的重要方式。随着语言的发展, 特别是有了可以记录的文字之后, 为了顺应各种不同的需求, 人们又发明了各式各样的书写格式和记录规则。到了今天, 计算机和信息技术的突飞猛进又让我们已经熟知的“语言”有了质的飞跃, 人们开始渐渐习惯了信息时代的语言。虽然经历了如此多的变化, 但语言的本质没有变, 依然是由人们事先预定的、由特定含义的标记组成的交流信息的方式。

1.1.1 标记语言

XML 是一种标记语言, 所以先介绍一下什么是标记语言。

在标记语言 (Markup Language) 中, 标记是用于指明文档中文本格式排列的代码系统, 所以标记语言就是一种描述文档内容如何显示的语言。换句话说, 标记是一种传输元数据 (即关于数据集本身的信息) 的方法, 而标记语言就是使用文字串或标记来界定和描述这

些数据的语言。XML 和 HTML 都是一种标记语言，而两者都是由同一种语言—SGML（Standard Generalized Markup Language，标准通用标记语言）发展而来的。

1.1.2 SGML

上世纪 60 年代末，IBM 公司的 Charles Goldfarb Ed Mosher 和 Ray Lorie 为了解决由不同专用格式创建的法律文件在公司不同部门间相互移植的问题，非常有建设性地提出了在不同操作系统上传递文档的 3 个要求：

- 文件处理程序支持同一个公共文档格式。
- 这个公共文档格式在特定的领域是专用的。
- 为了达到高度可靠性，文档格式必须遵守一些特殊规则。

由此，这种文档格式编排系统成为了第 1 种现代标记语言 GML（Generalized Markup Language, 通用标记语言）。在之后的几年里，在 Goldfarb 领导的一个开发组织的不懈努力下，终于推出了 SGML（Standard Generalized Markup Language，标准通用标记语言）。1986 年，国际标准化组织 ISO 批准采用 SGML 作为一种国际性的数据存储和交换的标准。

SGML 是一种非常强大的标记语言，它已经在很多大型制造公司、信息技术发布者及政府部门广泛采用。出版商也使用它制作各类文档，如书刊、报告、手册等。然后，这些 SGML 文档被转换为合适的格式，接着交给排版者和印刷者处理。在生产制造领域，SGML 还被用来规范技术指标。

SGML 让使用者可以在同一种信息资产上套用不同的处理方式，从而实现“一次建设，多次使用”的目的。举例来说，旅行社一些日程安排信息是共享的，所以设计者可以使用相同的 SGML 来源以建立组织者和游客的日程表。而在排版方面，建立游客日程表时只须打印一些简单的活动安排和时间表，而建立导游的日程表则再加上一些联系方式及资费代码等信息。SGML 文件是由标记和内容组成的简单 ASCII 数据串流，分析器会读取该文件，再由识别标记来决定这些信息的结构和内容。正是因为这种简单而标准的组成格式，SGML 文件具有高可移植性并且可以执行于任何有分析器的平台上。

然而，同样因为 SGML 是 ISO 的标准，因此它具有稳定到难以修改的特性。建立每一个 ISO 标准的委员会每隔 5 年才复检和维护该标准一次，并决定该标准是否仍然需要。如果答案是肯定的，再检查该标准是否应该更新。但是在商业信息管理的变更极为快速的今天，5 个月之内更新一次恐怕都无法满足实际的需求。此外，SGML 的“全能”也必然使它成为了一种复杂的不易学习和理解的标记语言。它的复杂性及其实现所需要的大量资金也注定了大多数商业用户和个人用户无法享受此项技术所带来的益处。可以说，SGML 赶不上 Web 的速度，它是在慢速、昂贵计算机的旧时代中诞生的。

因此，出于应用的需要，SGML 衍生出了 2 种真正实用的标记语言：HTML 和 XML。

1.1.3 HTML

1989 年，一位英国的研究人员 Tim Berners-Lee 在位于瑞士日内瓦的欧洲离子物理实验室，发现处理该研究中心大量研究资料时，资料之间的不可移植和不兼容是一件非常让人头痛的问题。于是，他构想了这样一个信息世界：在其中存取数据都采用一种统一的简单方式进行，传递信息的双方不考虑使用什么终端和程序，包含任何地点的任何计算机可以

使用一个简单的和常用的程序存取数据。他意识到传统的线性思考不可能达到这种跨越系统交流信息的要求，于是萌生了超级链接的概念。结合已有的 SGML 语言基础，HTML（Hypertext Markup Language，超文本标记语言）作为一个描述 Web 信息这种广泛使用文档类型的 SGML 应用程序而诞生了。

HTML 是建立 Web 网页的标记语言，和 SGML 一样，也是全球通用的标记语言。有人估算过，在今天的 Web 上 HTML 网页个数的数量级大约是 10 的 10 次方。它以简单明了的表达式、灵活的可移植性充斥着互联网的每个角落。一个初学者可以通过一本简单的教材在短时间内学会 HTML，而且建立和检查 HTML 代码的工具都很容易找到。HTML 使用二维空间也就是平面的结构格式化和处理文字，这些结构会呈现在 Web 浏览器中。浏览器供应商普遍在他们的产品中内建 HTML 模型，从而提供了丰富的程序化接口以延伸 HTML 的功能和表现方式。Internet 的使用者只要在浏览器网页中的超链接上按一下，就可以从数千公里以外的 Web 服务器上下载一个网页，浏览异地网络资源。

1. 一个 HTML 的示例

HTML 提供了固定的预定义元素集，可以使用它来标记一个典型、通用的 Web 页的各个组成部分。在最新版本 4.01 中，标记符号的个数已经从最初 1.0 版的 12 个拓展到了将近 100 个，浏览器检查 Web 页面中的这些 HTML 标记，并使用相应的格式将页面显示出来。

例如程序代码 1-1 所示的 HTML 页面。

程序代码 1-1

```
<HTML>
<HEAD>
<META http-equiv=Content-Type content="text/html; charset=gb2312">
<TITLE>Home Page</TITLE>
</HEAD>
<BODY>
<H1 align=center><IMG SRC="Butterfly.gif">欢迎来到爱飞翔的家</H1>
<H2 align=center><EM>Welcome to my home!</EM></H2>
<P><SPAN style='font-family:楷体_GB2312'>你想看看什么呢？请选择：</SPAN></P>
<UL>
<LI><A HREF="Story.htm"><B>爱飞翔的故事</B></A></LI>
<LI><A HREF="butterflyhome.htm"><B>蝴蝶之乡</B></A></LI>
<LI><A HREF="note.htm"><B>一起飞翔的朋友们</B></A></LI>
</UL>
<P><SPAN style='font-family:楷体_GB2312'>下面几个地方也有我的伙伴，
想去他们那边看看么？</SPAN></P>
<UL>
<LI>
<A HREF="http://www.hudieworld.com/index.asp">蝴蝶世界</A>
</LI>
<LI>
<A HREF="http://butterfly.scib.ac.cn/">青青蝴蝶</A>
```

```
</LI>  
</UL>  
</BODY>  
</HTML>
```

这段代码在 IE 中运行, 显示结果如图 1-1 所示。

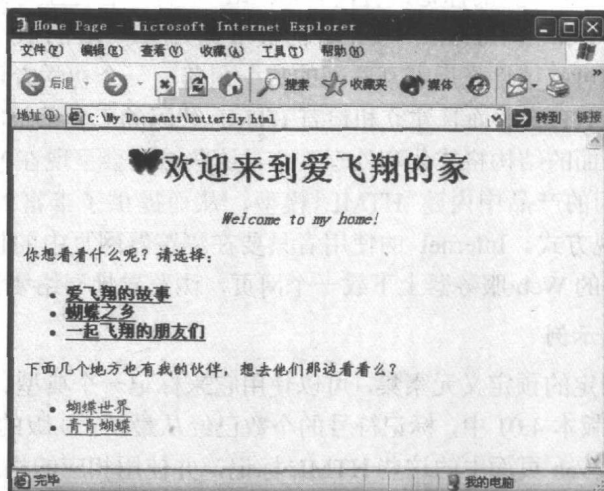


图 1-1 一个 HTML 页面

在 HTML 中, 每一个元素都以起始标签开头: 起始标签是前面带有左尖括号“<”, 后面带有右尖括号“>”, 并且包含元素名称和其他信息的文本块。大部分元素都以结束标签结尾, 该标签很像它所对应的起始标签, 不过其元素名称在斜杠 (/) 后面。元素内容是位于起始标签和结束标签之间的文本 (如果有的话)。元素可以嵌套。

2. HTML 的特点

当然, 像前文提到的那样, 最新版本的 HTML 有 100 多个标准元素, 这里所列举的只是其中最基本也是最常用的一些。虽然在很多“所见即所得”的网页制作软件中, 简单的页面制作已经可以将代码的编写完全放到图形操作界面中, 用户只需操作工具菜单就能完成编写。但要了解网页的构成, 编写真正符合实际需求的页面, 明白这些标记的用法是必需的。

作为使用者, 记住全部的元素是没有必要的。只要熟悉了这些元素的使用方法, 了解它们的特定用途, 在制作 HTML 页面时查一下相关资料就可以轻松应用了。显示 HTML 页的浏览器可以识别这些标准元素中的每一个元素, 并且知道怎样格式化和显示它们。比如, 浏览器通常用最大的字体显示 H1 标题, H2 标题用小一号的字体, 元素 P 则使用更小的字体表示。它在无顺序编号的列表中把一个 LI 元素显示成一个带有项目符号的有缩进的段落。此外, 它把元素 A 转换成带有下划线的超链接, 在用户单击时就可以链接到指定的地方或页面。

HTML 的简单易用也注定了它天生的一些致命缺陷。尽管不断更新版本扩张了基本元素的个数, 面对各种各样的信息类型和用户自定义对象, HTML 仍然无法表达。同时, HTML 无法执行某种程度的功能: 如精确的空格键控制、定位排列等等一系列网页制作者需要的

功能。为了解决 HTML 格式化的限制问题,又出现了 CSS (Cascading style sheets, 串接样式表)。HTML 的格式设定得到了一些改进,但是这些功能是外挂于 HTML 的基本功能之上,而且每一种浏览器都可能支持不同的样式属性,使得页面在设计者和使用不同浏览器的访问者不一定得到统一的样式。为了照顾广泛的浏览用户,则可能要牺牲一小部分用户的利益。

另外,HTML 是由非常不严谨的建议标准所管理,其中只有一点点的规则是用于处理网页数据,而且大部份的浏览器也不一定可以正确的执行这些处理网页数据的语法。比方说,段落元素<P>的结束卷标</P>是选择性的,这暗示不一定要使用</P>这个结束标签。也许对于有的程序设计员来说,编写页面代码时,不使用结束卷标带来了一些方便。但是如果某个元素真的需要结束卷标时,对代码进行排错就非常的困难。

没有严格的标准导致了在代码编写过程中,程序设计师会继续撰写不好的程序代码,因为撰写的好坏与否并没有差别,因此坏的程序代码招致更坏的程序代码。再加上可用元素的数目十分有限,所以 HTML 的程序设计师会以某些技巧来使用卷标,但是这些卷标对描述数据来说却不是必要的。

虽然 HTML 的普及在很大程度上推进了 Web 技术的发展,但先天不足的缺陷注定了它无法运用于更广泛的领域。

1.1.4 XML

1. XML 的诞生

为了让 SGML 在 Web 上有更广泛的应用,1996 年 7 月,万维网协会开始设计一种可扩展的标记语言,使其能够将 SGML 的灵活性和强大功能与已经被广泛采用的 HTML 结合起来。1998 年 1 月 10 日,SGML 以全新的面目出现了,并被命名为 XML(Extensible Markup Language, 扩展标记语)。当时发布的是 XML1.0 第一版,同时公布的还有 XML 计划的以下 3 个组成部分:

- XML 的句法。
- XLL 可扩展链接语言 XML 的语义链接。
- XSL 可扩展类型语言 XML 的表现。

2. 一个 XML 的示例

作为 SGML 的简化版本,XML 具有了更广泛的易用性。简单明了的语法结构使初学者比较容易就可以掌握 XML。人们对标记的读写过程应该使用现有的、最简便通用的工具,同时,也要简化计算机对文档和数据交换的处理。由于有太多的可选功能,SGML 变得过于复杂,以至于很难编写出针对这种语言的普通解释器,而相比之下 XML 的解释器则简单得多。此外,作为一个不错的 SGML 子集,XML 还保持了对现有的面向 SGML 的系统的向下兼容性,这样,用 XML 标记过的数据就仍然可以在这些系统中使用,为基于 SGML 的行业节省了大笔的改造费用,同时,与 Web 的结合也使得它们更便于被访问。

在数据表达方式上,XML 是一种界定文本数据的简便而标准的方法,由此它曾经被人称作“Web 上的 ASCII 码”。好比你可以根据个人的喜好选择编程语言来创建任何一种数据结构,然后同其他人在不同的计算平台上使用不同的语言来共享一样。XML 的标记可以用

来描述你要表达的概念，而属性则用来控制它们的结构。所以，你可以定义出能够与别人共享的语法。

程序代码 1-2 就是一个 XML 文档的示例。

程序代码 1-2

```
<?xml version="1.0"?>
<personlist>
  <employee>
    <name>张雨</name>
    <sex>男</sex>
    <age>34</age>
    <department>市场部</department>
    <headship>部门经理</headship>
  </employee>
  <employee>
    <name>叶佩</name>
    <sex>女</sex>
    <age>27</age>
    <department>财务部</department>
    <headship>职员</headship>
  </employee>
</personlist>
```

图 1-2 就是在浏览器中打开该 XML 文件显示的结果。

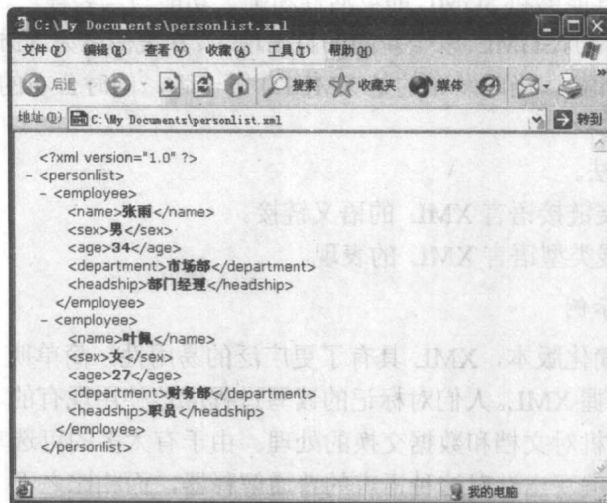


图 1-2 在 IE 浏览器中看到一个 XML 文档

1.1.5 XML 与 SGML、HTML 的关系

1. XML 与 SGML 的关系

SGML 是所有标记语言的母语言。从前面的介绍中我们已经了解到 HTML 和 XML 都派生自 SGML。因此，这几种语言都有一些共同点，如相似的语法和标记符的使用。但是

XML 从根本上来讲就是 SGML 的一个子集, 而 HTML 是 SGML 定义的一种应用。和 SGML 一样, XML 也可以定义新的应用。例如 RDF(Resource Description Format, 资源描述格式) 和 CDF(Channel Description Format, 频道描述格式) 都是使用 XML 定义的应用。

XML 和 SGML 是兼容的, XML 文档可以通过任何 SGML 制作或浏览工具阅读。但是 XML 没有 SGML 那些过于规范以至于复杂的要求, 针对于有限带宽的网络, XML 的设计更适用于 Internet。正如 XML 规范的缔造者之一 Tim Bray 所说的那样: “XML 的设计出发点是取 SGML 的优点去除复杂的部分, 使其保持轻巧而可以在 Web 上工作”。

不同于 HTML 和 XML, SGML 可能永远不会在 Internet 上被广泛接受, 因为它从来没有为某个网络协议的需求而设计或优化过。然而, 作为标记语言的基础, 强大的功能和规范的结构使得 SGML 在高端复杂结构的发布应用中将继续适用。

图 1-3 就表达了 3 种语言的关系。

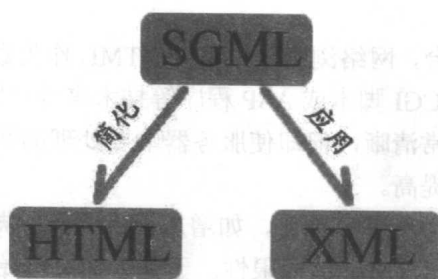


图 1-3 3 种标记语言之间的关系

2. XML 与 HTML 的比较

在具体的应用中, XML 和 HTML 可以说是处在同一个层面上。XML 的开发也是针对了 HTML 的不足之处, 而 HTML 的简单易用也很早就被人们普遍接受。习惯 HTML 而很少涉及其缺陷的人也许依然会问: 有了 HTML 为什么还要用 XML? 而只是了解了 XML 种种好处的人也可能会有这样的问题: XML 可以取代 HTML 吗? 单纯介绍 2 种语言的产生和特点并不足以解答问题, 接下来就进一步讨论一下这个话题。

(1) 有了 HTML 为什么还要用 XML?

HTML 推动了 Web 技术的发展, 从今天它在网页技术中占的比重就可以明确这一点。我们当然不能否认 HTML 在网络信息发展中的重要地位, 但 HTML 终归只是一个标记的集合, 作为一种表达的技术, HTML 并不一定能揭示自身标记中的含义。举一个最简单的例子: `<h2>Kale</h2>` 这句话在网络浏览器中有特定的表现, 但是 HTML 却并没有告诉我们它到底是什么。Kale 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: 是一个电脑公司、一种动物、还是一个人的名字? HTML 并没有告诉我们 Kale 的具体内容。

今天网络的成长速度远远超过了人们的想象, 电子商务、电子图书、远程教育等全新领域异军突起迅猛发展, 并逐渐成为互联网世界必不可少且非常重要的组成部分, 随之而来的是 Web 文件的复杂化、多样化和智能化。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的不同需求而以不同的效果和形式表达再现出来。人们才开始学

着使用 HTML 来表达自己的超链接式的想法时,就发现这种语言已经远远不能满足网络庞大的潜在应用能力的需要。HTML 标记集合是固定的,用户不能新增自己需要的有意义而又可以让其他人使用的标记。

举一个简单的例子来说,假设我们要在网上拍卖一部二手电脑,而在一个 Web 文件中包含了有关这部电脑的全部信息:包括基本硬件配置、购买时间、新旧程度、附件外设、物主信息、售价、支付地点及方式等等。这样的信息发布之后,和大量类似的出售信息堆在一起。而买主面对这些信息也会根据个人的喜好来进行选择:有些人关心硬件设施,有些人关心新旧程度,而有些人关心的也许是在自己的居住地附件弄到一部便宜的能用的机器,更有甚者可能只想了解一些当前二手电脑的行情而已。这样一来这个 Web 文件必须具有极高的复杂性与灵活性以应付如此复杂多变的具体应用。HTML 所欠缺的恰恰就是这一点,它的简单与普遍性原则虽然曾使 Web 的推广受益匪浅,但在某些方面,它反过来也成为了 Web 发展的障碍。

此外,作为一个应用平台,网络浏览器如果以 HTML 作为数据标准,那么网络上的应用程序都要依赖服务器上的 CGI 脚本或 ASP 程序等技术来处理网页上的数据。这样的做法虽然使得数据处理的结构非常清晰,但却使服务器端要处理的数据量过大,从而降低了网络速度,制约了网络效率的提高。

尽管人们已付出很多努力改进 HTML,如增加表格框架、脚本语言,扩充标准元素等等。但均未能从根本上改变 HTML 的局限性。于是人们又开始将精力放在重构一种新的 Web 标准上,并随之相继诞生了许多新的技术。但由于种种原因,这些技术大部分都是昙花一现,没有生存下来。而这其中,惟有 XML 不仅在使用中得到各方肯定,更在短短几年间迅速崛起并得到计算机界几个巨头 Microsoft、IBM、HP 等大公司的全力支持。XML 一可扩展标识语带领着一场不可阻止的 Web 变革正在席卷全球。

由此,第 1 个问题解决了:XML 有效克服了 HTML 的缺陷,在 Web 上有着很旺盛的生命力。

(2) XML 能够取代 HTML 吗?

XML 能够取代 HTML 吗?就目前来说,这个问题的答案是否定的。HTML 仍然是用来告诉浏览器如何在网站上显示信息的主要语言。两者根本谈不上替代的关系。XML 中并没有任何与可视化表现形式有关的内容。

HTML 兼顾了数据及其表达方式(虽然两方面做得似乎都不理想),而 XML 只关心数据本身。XML 的研究者建立了一套完善的体系来处理数据和表达方式之间的关系。与依靠扩充代码部分的传统数据表示方法不同,XML 样式的表现技术是由数据驱动的,这使得它几乎可以表达任意复杂的数据形式。但是,不论采用什么技术,XML 的样式化都是通过另一个被称作样式单的文档来实现的。在这个文档中,设计者会格式化样式和决定何时应用样式的规则。同时,这个样式单又可以用于其他多个文档中,以产生类似的效果。XML 中的数据会依据样式和规则来显示,但脱离了这些样式单数据格式就无法表达。而这个问题在 HTML 中是不存在的。

使用浏览器查看一个 XML 文件时通常可以采取 3 种方式,第 1 种是直接开启一个附样式表的 XML 文件(请注意,是附带了样式表的 XML 文件)。要利用样式表,就需要利用 HTML 来告诉浏览器如何安排 XML 数据。另外 2 种显示 XML 文件的方法:数据链路