



Sams Teach Yourself
XML in 24 Hours

XML

- 详述XML的功能、句法及应用
- 使用XML来满足特定需求
- 在HTML文件中显示XML数据
- 定义XML模式和导入文档
- 通过HTML和XML操作数据库
- 使用XSL来定义XML文档的布局
- 用XSL执行数据库查询
- 在电子商务中使用XML
- 为在线商店建造订货机制
- 书中大量代码实例方便读者引用

速成教程

时光飞逝 时不我待

(美) Charles Ashbacher 著

启丁编译组译



机械工业出版社
China Machine Press

SAMS

XML速成教程

(美) Charles Ashbacher 著

启丁编译组 译



机械工业出版社
China Machine Press

本书从基础开始讲解,全面地介绍了XML的功能、句法及应用,是供XML的初学者和专业人员使用的一部优秀教程。

主要内容包括使用XML来满足你的单位的需求,在HTML文件中显示XML数据,定义XML模式和导入文档,通过HTML和XML操作数据库,使用XSL来定义你的XML文档的布局,用XSL执行数据库查询,在电子商务中使用XML,为在线商店建造订货机制等。该书的全部程序都是经过译者实际运行验证的,读者可以放心地学习和引用。

Charles Ashbacher: Sams Teach Yourself XML in 24 Hours.

Authorized translation from the English language edition published by Sams, an imprint of Macmillan Computer Publishing U.S.A.

Copyright © 2000 by Sams. All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2001 by China Machine Press.

本书中文简体字版由美国麦克米兰公司授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2000-2815

图书在版编目(CIP)数据

XML速成教程/(美)艾史拜克(Ashbacher, C.)著;启丁编译组译.-北京:机械工业出版社,2001.10

书名原文: Sams Teach Yourself XML in 24 Hours

ISBN 7-111-09281-3

I. X… II. ①艾… ②启… 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(2001)第055720号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 李 炎 张鸿斌

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2001年10月第1版第1次印刷

787mm × 1092mm 1/16 · 20印张

印数: 0 001-5 000册

定价: 33.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

JS651/02

译者序

本书从基础开始，按照一学时学习一课的原则讲解。全书共分24学时，由浅入深地全面介绍了XML的功能、句法及应用，是从头自学XML和培训XML专业人员的优秀教程。

读者可以跟随这本书从头学习。全书的主要内容包括：

- 如何使用XML来满足你的单位的需求
- 如何在HTML文件中显示XML数据
- 怎样定义XML模式和导入文档
- 如何使用XML和HTML操作数据库
- 如何使用XSL来定义你的XML文档的布局
- 如何使用XSL执行数据库查询
- 如何在电子商务中使用XML
- 怎样为在线商店建造订货机制

本书中所有的HTML程序及其调用的XML或CSS等程序，都是经过译者在Internet Explorer 5.0中运行并验证的。本书内容简明，要点都是通过程序实例来进行说明的，更便于读者掌握XML。在翻译过程中，译者对全书的程序代码中的笔误等进行了更正，并得到了原作者的同意和帮助，特此对原作者表示感谢。读者可以放心地学习和引用。

王启丁

2001年6月22日

前 言

当有人问我应该从哪儿开始学习编程时，我总是说：“学习HTML。”因为HTML软件是免费的，HTML语言也容易掌握，而且不用编译代码就能够及时看到结果。学习XML也基本上如此。HTML和XML有着共同的语言祖先。因此，只要你懂得HTML，就不难掌握XML。这本书假定你已经很熟悉HTML了。

在编写本书期间，第二大主要浏览器Netscape的前景尚不明朗。公众不清楚该公司被AOL收购后的发展方向，对Netscape浏览器的未来也存在很大的疑问。由于IE 5完全支持XML，而且占有一定的市场份额，所以我们决定在这本书中只使用IE 5。这也就避免了要不断地指出这两种浏览器在支持XML方面的差别的麻烦。最后，IE 5可以免费下载，因此，如果有人决定不使用它，那就不是经济方面的问题，而只是个人的问题了。

除了几处例外，本书只把JavaScript语言用于提供指令编码的动态功能。经过认真思考之后，我们决定不花时间来介绍或复习JavaScript。然而，不介绍它并不会像想像的那么严重。本书中的程序都是使用非常标准和基本的程序构架，因而，只要熟悉任何程序语言就足够了。如果你是编程方面的新手，并且仍然希望学习XML的话，可以找一些好的JavaScript参考资料开始阅读。有了这本书和JavaScript手册的帮助，任何决心取得成功的读者都可以达到目的。

为什么学习XML很重要？这个问题有三个答案：World（世界）、Wide（广泛）和Web。虽然Web以异常快的速度在增长，并且有关它的新用途的想法也在以超过细菌繁殖的速度增长，但是它仍然处于婴儿阶段。自然，任何用于在Web上进行操作的工具都将要跟上这个奔腾的潮流。在这大潮中还有另外的很强大的趋势，即电子商务或e商务。关于电子商务普及的许多预测都过分的乐观，因为实践证明，在Web上实现电子商务比最初想像的要困难得多。

虽然有这些问题，但是毫无疑问，电子商务已经变成世界经济的一个重要部分。在电子商务背后的驱动力量是其容易捕获、存储、修改和解释Web环境中的数据的能力。用XML就能够完成这个任务。

如何使用本书

虽然本书中共介绍了24课，每课学习一小时。但是，我们也对其中的小节进行了协调，以便精细地调整每学时的长度。当我进入一些公共场所，例如，棒球比赛、公共汽车、献血中心时，常常看见人们手里拿着计算机方面的材料。通常我会询问他们能够抽出多少时间来学习，大多数的回答并不是一小时，而是15~30分钟。因此，为了使该书更适合现代生活中快节奏的现实，已经将各学时分为若干小节，其中没有一个小节是需要花费30分钟以上的。而且，大多数的小节都可以在15分钟内学习完。

在程序清单中，为了便于引用而添加了行号。这些行号不是HTML或JavaScript的一部分，当你输入程序代码时请省去这些编号。

要想从本书中获得最大的收益，你应该阅读每一学时的每一部分，阅读并弄懂每一个问题，参加全部小测验并完成全部练习题。为了便于读者学习，例子中的所有源程序都可以在本书的协作Web站点www.sampublishing.com上找到（原书书号为ISBN 0-672-31950-0）。一旦你学习完这本书，就能掌握一系列技巧。这些技巧将会帮助你准备好使用XML处理Web上的数据，从而进入那激动人心的和动态的世界。



注释代表了与周围讨论的主题有关的一些有趣的信息。

目 录

译者序
前言

第一部分 基 础

第1学时 XML入门	1
1.1 标记语言的历史	1
1.2 格式正确的XML文档	3
1.3 XML作为元语言	4
1.4 你的第一个XML程序	4
1.5 XML的自说明方面	7
1.6 XML的影响	8
1.7 课时小结	9
1.8 专家答疑	9
1.9 课外作业	9
1.9.1 思考题	9
1.9.2 解答	9
1.9.3 练习题	9
第2学时 使用HTML文件显示XML 文件中的数据	11
2.1 动态更新HTML标签的内容	11
2.2 使用HTML文件显示XML文件中的数据	12
2.2.1 介绍ActiveX控件	13
2.2.2 显示XML数据的编程例子	13
2.2.3 把组件添加到XML文件中	16
2.3 从多个XML文件导入数据	17
2.4 课时小结	19
2.5 专家答疑	19
2.6 课外作业	20
2.6.1 思考题	20
2.6.2 解答	20
2.6.3 练习题	21
第3学时 对象和XML	22

3.1 对象和XML之间的关系	22
3.2 创建HTML文件的层次树	24
3.3 节点和节点列表对象	25
3.4 创建XML文件的通用数据提取器	26
3.5 错误处理	28
3.6 课时小结	30
3.7 专家答疑	30
3.8 课外作业	30
3.8.1 思考题	30
3.8.2 解答	31
3.8.3 练习题	31
第4学时 XML数据岛	32
4.1 XML数据岛的基础	32
4.1.1 使用XML数据岛中的联机代码	32
4.1.2 把文件导入XML数据岛	35
4.1.3 把多个文件导入XML数据岛	36
4.2 使用XML数据岛来更新节点的内容	40
4.3 使用XML数据岛来读取存储消息的文件	44
4.4 课时小结	47
4.5 专家答疑	48
4.6 课外作业	48
4.6.1 思考题	48
4.6.2 解答	48
4.6.3 练习题	48
第5学时 文档类型定义	50
5.1 预定义实体	50
5.2 可解析实体和不可解析实体	52
5.3 文档类型定义	53
5.3.1 把可解析实体放置在DTD中	54
5.3.2 把不可解析实体放置在DTD中	55
5.3.3 确认DTD	56
5.3.4 限定DTD中的数据组件	57

5.4 把DTD分离成外部文件	59	8.6 课外作业	106
5.5 xmlDOM对象的DTD相关属性	61	8.6.1 思考题	106
5.6 专家答疑	64	8.6.2 解答	106
5.7 课外作业	64	8.6.3 练习题	107
5.7.1 思考题	64	第9学时 数据库和XML	108
5.7.2 解答	64	9.1 你的基本数据记录的定義	108
5.7.3 练习题	65	9.2 读取Catalog数据库中的<BOOK>节点	114
第二部分 超越基础		9.3 使用按钮单击来移动通过 <Catalog>数据库	117
第6学时 属性列表和编写正确的DTD	67	9.4 课时小结	127
6.1 属性列表	67	9.5 专家答疑	127
6.2 外部实体	73	9.6 课外作业	127
6.3 处理指令	75	9.6.1 思考题	127
6.4 课时小结	75	9.6.2 解答	128
6.5 专家答疑	75	9.6.3 练习题	128
6.6 课外作业	75	第10学时 使用XML数据源对象	129
6.6.1 思考题	75	10.1 数据源对象	129
6.6.2 解答	76	10.2 读取有多层节点的数据	134
6.6.3 练习题	76	10.3 课时小结	141
第7学时 XML中的数据分类	77	10.4 专家答疑	141
7.1 XML中的数据类型介绍	77	10.5 课外作业	141
7.2 修改节点的数据类型	83	10.5.1 思考题	141
7.3 错误处理	86	10.5.2 解答	142
7.4 从错误中恢复	89	10.5.3 练习题	142
7.5 课时小结	90	第三部分 可扩展的样式表语言： 创建样式表和访问数据库的方法	
7.6 专家答疑	90	第11学时 介绍可扩展样式表语言 (XSL)	143
7.7 课外作业	90	11.1 级联样式表	143
7.7.1 思考题	90	11.2 XML名称空间	146
7.7.2 解答	91	11.3 XSL名称空间	148
7.7.3 练习题	91	11.4 在Web开发中XSL模板和样式的作用	152
第8学时 逐步读取节点树并添加 和替换节点	92	11.5 课时小结	152
8.1 逐步读取节点树	92	11.6 专家答疑	152
8.2 添加节点到树中	96	11.7 课外作业	152
8.3 替换树中的节点	103	11.7.1 思考题	152
8.4 课时小结	105	11.7.2 解答	153
8.5 专家答疑	106		

11.7.3 练习题	153	14.4 课时小结	185
第12学时 使用XSL模板从XML		14.5 专家答疑	185
文件中读取数据	154	14.6 课外作业	185
12.1 把XSL模板从HTML文件中分离	154	14.6.1 思考题	185
12.2 编程例子	154	14.6.2 解答	186
12.2.1 XML文件	154	14.6.3 练习题	186
12.2.2 XSL模板	156	第15学时 使用条件XSL语句和嵌入脚本	
12.2.3 HTML文件	157	代码来执行数据库查询	187
12.3 改变数据读取的顺序	160	15.1 XSL元素	187
12.4 添加模板来读取不在<Author>或		15.2 xsl:if元素	188
<Publisher>节点中的数据	162	15.2.1 读取Catalog数据库并提取具有	
12.5 使用单独的文件的优点	164	BestSeller属性的Book记录的	
12.6 课时小结	164	xsl:if例子	188
12.7 专家答疑	164	15.2.2 只选择BestSeller属性有特殊值的	
12.8 课外作业	164	记录的xsl:if例子	190
12.8.1 思考题	164	15.2.3 在XSL文件中使用多个xsl:if元素	191
12.8.2 解答	165	15.3 xsl:choose、xsl:when和xsl:otherwise	
12.8.3 练习题	165	元素	192
第13学时 使用多个XSL模板从多个XML		15.3.1 xsl:choose和xsl:when例子	192
文件中读取数据	166	15.3.2 xsl:otherwise例子	196
13.1 操作两个数据库	166	15.4 xsl:eval元素	197
13.2 操作由XSL模板创建的节点	171	15.4.1 xsl:eval元素的例子	197
13.3 交叉引用两个数据库	172	15.4.2 使用JavaScript函数的xsl:eval例子	199
13.4 课时小结	175	15.5 课时小结	201
13.5 专家答疑	175	15.6 专家答疑	201
13.6 课外作业	176	15.7 课外作业	201
13.6.1 思考题	176	15.7.1 思考题	201
13.6.2 解答	176	15.7.2 解答	202
13.6.3 练习题	177	15.7.3 练习题	202
第14学时 使用XSL模板执行基本的		第16学时 使用XSL选择和布尔运算符	
数据库查询	178	来执行数据库查询	203
14.1 使用XSL模板中的选项来创建		16.1 xsl:comment元素	203
可选择的数据提取	178	16.2 XSL选择运算符	205
14.2 排序数据	178	16.3 ALL运算符	210
14.3 筛选数据	181	16.4 XSL布尔运算符	212
14.3.1 使用比较运算符进行筛选	181	16.5 课时小结	214
14.3.2 根据XML属性进行筛选	182	16.6 专家答疑	214

16.7 课外作业	215	19.4 属性类型声明	244
16.7.1 思考题	215	19.5 IE中支持的属性类型	246
16.7.2 解答	215	19.6 课时小结	246
16.7.3 练习题	215	19.7 专家答疑	246
第17学时 使用XSL比较和集合运算符以及 XSL方法来执行数据库查询	217	19.8 课外作业	247
17.1 相等运算符	217	19.8.1 思考题	247
17.2 比较运算符	218	19.8.2 解答	247
17.3 all和any运算符	220	19.8.3 练习题	247
17.4 使用value()函数访问节点的内容	221	第20学时 创建在线书店	249
17.5 selectSingleNode()方法	221	20.1 <Catalog>数据库的结构	249
17.6 end()方法	222	20.2 附加信息的数据库	253
17.7 index()方法	222	20.3 购买项目的数据格式	254
17.8 课时小结	223	20.4 书店的开始网页	255
17.9 专家答疑	223	20.5 课时小结	257
17.10 课外作业	224	20.6 专家答疑	257
17.10.1 思考题	224	20.7 课外作业	257
17.10.2 解答	224	20.7.1 思考题	257
17.10.3 练习题	224	20.7.2 解答	257
第四部分 电子商务和其他高级论题中的XML		20.7.3 练习题	258
第18学时 文档对象模型	225	第21学时 访问在线书店的数据库	259
18.1 数据提取和节点集合方法	225	21.1 构造用于搜索数据库的网页	259
18.2 文档对象模型	226	21.2 搜索在线书店数据库	260
18.3 文档对象方法	230	21.3 显示已购买的书籍的列表	269
18.4 节点对象的属性和方法	232	21.4 课时小结	272
18.5 nodeList对象的属性和方法	233	21.5 专家答疑	272
18.6 课时小结	233	21.6 课外作业	272
18.7 专家答疑	234	21.6.1 思考题	272
18.8 课外作业	234	21.6.2 解答	272
18.8.1 思考题	234	21.6.3 练习题	273
18.8.2 解答	234	第22学时 处理和显示在在线书店中的 购买	274
18.8.3 练习题	235	22.1 World Wide Web的客户-服务器模型	274
第19学时 XML模式语言	236	22.2 Querystring环境变量	274
19.1 模式基础	236	22.3 Microsoft Personal Web Server	279
19.2 ElementType声明	237	22.4 在服务器上创建文本文件	279
19.3 赋值给minOccurs和maxOccurs	242	22.5 课时小结	281
		22.6 专家答疑	281

22.7 课外作业	281	23.7 专家答疑	293
22.7.1 思考题	281	23.8 课外作业	294
22.7.2 解答	282	23.8.1 思考题	294
22.7.3 练习题	282	23.8.2 解答	294
第23学时 XML中的链接: XLink		23.8.3 练习题	294
和XPointer	283	第24学时 附加的标记语言	296
23.1 HTML中的链接	283	24.1 向量标记语言	296
23.2 XLink结构	286	24.1.1 位图图像与向量图形的对比	297
23.3 XLink的全局属性	286	24.1.2 VML的例子	297
23.3.1 type	287	24.2 数学标记语言	299
23.3.2 actuate	287	24.3 虚拟真实建模语言	301
23.3.3 show	287	24.4 房地产列表标记语言	301
23.3.4 href	287	24.5 人力资源管理标记语言	305
23.3.5 role	287	24.6 VoxML	305
23.3.6 title	287	24.7 课时小结	307
23.3.7 from和to	288	24.8 专家答疑	307
23.3.8 XLink全局属性的例子	288	24.9 课外作业	308
23.4 为XLink对象编写DTD	289	24.9.1 思考题	308
23.5 XPointer	290	24.9.2 解答	308
23.6 课时小结	293	24.9.3 练习题	308

第一部分 基础

第1学时 XML入门

在万维网（World Wide Web）短暂的历史中，有一个长长的所谓“下一代的伟大事物”的列表，这些事物的实际效果往往小于天花乱坠的广告宣传。扩展标记语言（Extensible Markup Language, XML）显然是这些“下一代的伟大事物”之一。但是，要预测什么时候这种广告宣传才会结束尚为时过早。然而，毫无疑问，Web商务将会继续，并且XML在促进系统之间的数据交换方面是很有价值的工具。XML是基于HTML所采用的同一个标记语言原理，从而会使读者容易学习。由于这些原因，XML的寿命至少应该与Web的寿命一样长。

在这一学时中，你将要学习下列内容：

- 标记语言的简要历史，以及为什么创建XML在制作Web网页方面补充HTML。
- XML作为元语言的作用。
- XML数据文件的基本组织。
- XML的自说明特性。
- 在Web上传输数据方面XML的影响。

1.1 标记语言的历史

标记语言早于Web许多年出现，其起源可以追溯到商业出版初创阶段。在编辑们准备出版手稿时，常常用符号标记出文本的各个章节。例如，需要用斜体开始的节就会用字母I来标记，要指明斜体应该在何处结束时，该节的结尾也要标记。当然，为了描述得更清楚一些，结束符号应该与起始符号有所不同。

用于指定文档结构的第一个正式的标记语言是IBM公司在20世纪60年代创建的。它就是众所周知的通用标记语言（Generalized Markup Language, GML），最初它被用于标准化IBM的内部文档。不久，它就被扩展为标准通用标记语言（Standard Generalized Markup Language, SGML）。SGML成为了被许多不同工业采纳的信息表示标准。1986年，国际标准化组织（International Organization for Standardization, ISO）把SGML确定为正式的标准。要查询有关国际标准化组织的更多信息，要访问SGML标准的文档，请进入到它的Web站点 <http://www.iso.ch/welcome.html>。

尽管你可能还从来没有用SGML创建过文档，可能还不清楚作为SGML的子集，HTML是怎样产生的。首先，你需要了解当打开浏览器并进入到Web地址，如<http://www.ashbacher.com>时，就进行了下列的三步处理：

- 1) 浏览器与Internet上的目录服务计算机联系，以获得请求的Web网页的Internet地址。该网

页是用HTML编写的文件。

2) 如果找到了该地址, 浏览器就会与Web站点联系并请求获得该文件。

3) 如果请求的文件可用, 站点便会通过Internet将它发送到浏览器, 浏览器就会翻译文件并显示文件内容。

上述的全部操作几乎已经成为常规, 并且大部分是透明的。然而, 在十九世纪八十年代末以前还没有这个过程。创建这个过程的灵感迸发于欧洲粒子物理实验室 (CERN), 即在瑞典的高能物理研究中心。

在该实验室的科学家们习惯于交换包含参考文献的在线论文。要查看参考材料, 必须进行搜索, 或者至少要人工地打开另外的文档。除此之外, 由于编写在计算机上显示的复杂公式当时还有些问题, 软件常常不能按希望的方式显示在线论文。为解决这些问题, 欧洲粒子物理实验室的两位研究员Tim Berners-Lee和Anders Berglund敏感地观察到了对电子文档交换系统的需求。

研究人员的设想是创建符号或标签的标记语言, 并把该语言嵌入到文档中, 以告知查看它的程序 (浏览器) 怎样显示其文本。随后就创建了由小于符号 (<) 和大于符号 (>) 限制的标签, 其完整的标签集就变成了众所周知的超文本标记语言 (Hypertext Markup Language, HTML)。浏览器检查并翻译HTML标签, 并把HTML标签作为模板使用以确定怎样显示文件的内容。这个简单而功能强大的解决方案工作得如此成功, 以至于现在人们都把Berners-Lee看作是万维网的发明人。

随着Web的继续增长, HTML也继续以极快的速度在改变。World Wide Web联盟 (W3C) 是制定和发布SGML和HTML标准的组织。它目前是由Berners-Lee领导。要获得有关HTML的历史和最近的标准文档方面的更多信息, 请访问网址为<http://www.w3c.org>的Web站点。

HTML与XML之间的关系

程序清单1-1展示了HTML文件的基本构架。

程序清单1-1 HTML文件的基本构架

```
1: <HTML>
2: <HEAD>
3: <TITLE> </TITLE>
4: </HEAD>
5: <BODY>
6:
7: </BODY>
8: </HTML>
```

关于HTML文件要记住的主要一点是: 标签把文件分成几个部分, 每一部分都有独特的目的。正如你将看到的那样, XML文件也有这个特性。

事实证明, HTML的成功超出了任何人的期望。现在人们往往认为Berners-Lee会因为Internet而变得很富有, 但实际上并非如此。因为HTML几乎是专门为显示静态文本而设计的, 对于许多复杂的操作, HTML所能够表达的内容受限。

XML是使用最广泛的HTML扩展, 它弥补了HTML的这个缺陷 (第24学时将要讨论HTML的

其他一些扩展)。XML本身也是标记语言，它使用与HTML相同的标签符号。用XML可以定义自己的标签。由于可自定义标签就使XML比HTML具有更多的灵活性。在<http://www.w3c.org>站点上有当前的XML标准。

1.2 格式正确的XML文档

你可以使用HTML做许多事情。但是，在某些方面它的规则不够严格，从而会强制浏览器填充一些错误信息。而XML就比较严格，它强制严格遵守下列规则：

- 每个起始（Opening）标签必须有对应的结束（Closing）标签。
- 嵌套的标签对不能够用另外的标签覆盖。
- 标签的名称是区分大小写的。

HTML要求许多标签要成对（例如，每个起始标签必须与一个结束标签对应）出现。然而，成对的要求对于某些标签是可选择的，并且有的只需要起始标签。最常见的例子是段落标签（<p>）。例如，浏览器能够接受下列HTML代码，这些代码会创建三个段落：

```
<p>The first paragraph is here.  
<p>The second paragraph is here.  
<p>The third paragraph is here.
```

当浏览器分析这段代码时，下一个段落标签的出现是终止前一个段落的暗示信号。这个暗示信号带来的问题是由浏览器决定结束标签应该在什么地方，而不总是由作者来决定结束标签应该在什么地方。

XML则通过严格的规则来减少这种随意性，即规定每个起始标签都必须有一个对应的结束标签。如果要使文档格式正确，这是必须遵守的规则之一。例如，前面例子相应的格式正确的代码如下所示：

```
<p>The first paragraph is here.</p>  
<p>The second paragraph is here.</p>  
<p>The third paragraph is here.</p>
```

HTML包含了执行操作的一些附加标签，而不只是标记要修改的文本段。断行标签（
）就是一个最好的例子。为了满足XML的标签成对的要求，起始标签和结束标签是联合使用的。在这种类型的标签中，正斜杠（/）被放置在紧靠大于符号之前的标签内。这种标签叫做空标签（empty tag）。例如，与HTML断行标签等效的XML标签是
。

在要求标签遵守正确的嵌套规则方面，HTML也比较松散。例如，下面两行HTML代码就包含没有正确嵌套的标签：

```
<STRIKE><BOLD>This is our first example.</STRIKE></BOLD>  
<SUP><SMALL>This is our second example.</SUP></SMALL>
```

上面的两行代码在XML中都是不合法的。XML的正确格式代码必须是结束标签必须与上一个起始标签相匹配。例如，下面两行代码是与前面例子等效的格式正确的代码：

```
<STRIKE><BOLD>This is our first example.</BOLD></STRIKE>  
<SUP><SMALL>This is our second exemple.</SMALL></SUP>
```



由于手持设备的内存比较小，把这些规则用于手持设备时还会带来额外的好处。HTML的松散规则使得为这些设备编写浏览器的程序困难得多，因为用于检查嵌套不当和丢失结束标签所需要的全部额外代码会使执行文件增大很多。如果能够避免HTML的这个问题，这个代码就能够更适合于较小的设备。

1.3 XML作为元语言

元语言（metalanguage）是用来说明其他语言的语言。例如，“名词”、“动词”、“副词”和“形容词”等词就属于元语言。这些词描述了大多数人类语言的句子的结构。因此，它们是用于描述语言的语言的一部分。元语言的优点是它能够用来定义多个实际的语言。

首先，XML是用来定义其他语言的元语言。要明白这一点，你就需要接受“语言”这个词的普通定义。为了本书的需要，我们把数据库中的任何数据格式都认为是语言。因为XML能够用于定义数据的格式，所以说它是一种元语言。

作为一种元语言，你可以使用XML定义自己的标签。这就提供了胜过基本的HTML的突出的优点。例如，在使用如<Message>之类的标签包围一个XML条目时，在标签之间的数据的含义是很清楚的。把标签编写成与你的数据的特殊格式相匹配，就能使程序代码容易读懂和编写。常常选择能够表达数据所代表的含义的名称来做标签名称。对于标签名称的惟一限制也就是其他编程语言所常用的限制，包括：

- 不能使用关键字。
- 在标签名称中不能有空格符、标点符号和其他分隔符（诸如括号、方括号和大括号）。
- 标签名称中可以使用下划线（_）和数字。但是，标签名称的第一个字符不能够用数字，而下划线可以作为标签名称的第一个字符。

当在标签名称中要使用多个单词时，可以直接借用广泛用于编程中的策略，即用多个单词组合成名称，并将其中每个单词的首字母大写。例如，如果要定义一个标签来代表客户在银行账户中钱的数量，可以把标签名称取为MoneyInAccount。这个标签既符合命名规则又保留了标签的含义。

1.4 你的第一个XML程序

第一个程序的例子是由两个文件组成的。一个是XML文件，另外一个HTML文件。其中的XML文件如程序清单1-2所示。在第2学时“使用HTML文件显示XML文件中的数据”中，你将会看到其中的HTML文件。这个例子以及贯穿本书的许多其他例子，都使用电子邮件的基本结构作为模型。

程序清单1-2 简单的XML数据文件

```
1: <?xml version="1.0"?>
2: <!-- This is our first XML file -->
3: <MESSAGE>
4: <TO>STUDENT</TO>
5: <FROM>AUTHOR</FROM>
6: <SUBJECT>Introduction to XML</SUBJECT>
```

```

7: <BODY>Welcome to XML!</BODY>
8: </MESSAGE>

```

可以使用IE 5.0来验证格式正确的XML文件的属性。为此，只需要在IE中单击“文件”菜单，选择其中的“打开”命令，选取要打开的XML文件即可。如果该文件格式正确，其结果将类似于图1-1所示。

下面让我们详细地研究一下这个程序。该程序的第1行是文件的说明，包括XML规范的版本号。虽然这一行不是非要不可，但是在XML文件中应该总是包括这一行为好。这一行的句法很特殊。它的所有字母都必须小写，必须使用问号，而且版本号必须在双引号中。

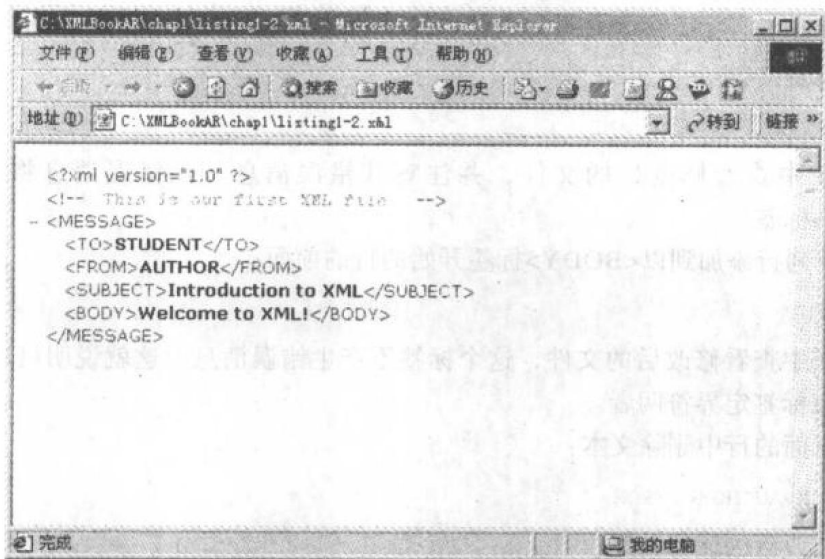
XML的注释句法与HTML中的注释句法相同。第2行是一行注释。



与HTML中的情况相同，注释行的长度可以超过一行。

图1-1

在IE中查看时，你的第一个XML文件的外观



该程序的第3行至第8行中包含了用户定义的标签，这些标签划定了数据元素列表的界限。<MESSAGE>标签对定义了文件的数据类型，并且是层次结构的根，如图1-2所示。



在XML中，结构的定义是通常所说的数据类型（data type）。例如，程序清单1-2定义了MESSAGE数据类型是由TO、FROM、SUBJECT和BODY等元素组成的。对应于数据类型的实际代码通常叫做该类型的对象（object）。在程序清单1-2中，第3行至第8行创建了MESSAGE对象。



只要起始标签和结束标签一致，标签名称就不必都用大写字母。例如，下面的标签对格式都是可以接受的：

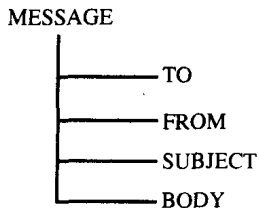
```

<Message>...</Message>
<message>...</message>
<MESSAGE>...</MESSAGE>

```

图1-2

用层次树形式表示的
第一个XML文件的
代码



从第4行到第7行中的每一行都定义了MESSAGE对象的一个组件。如前所述，每个组件都必须以适当的结束标签终止。

为了让你亲手积累XML句法的经验，这一节的其余部分说明应该怎样对上述XML文件进行修改。一定要确保每次修改之后，能重新使文件回复到原来的格式。

请把下面一行

```
<BODY>Welcome to XML!</BODY>
```

修改为：

```
<BODY>Welcome to XML!
```

在IE中查看修改后的文件，并注意其错误消息。该错误消息将会告知你没有结束的<BODY>标签。

把下列行添加到以<BODY>标签开始的行的前面：

```
<CC/>
```

在IE中查看修改后的文件，这个标签不产生错误消息。这就说明可以用单个标签来代表起始和结束标签定界符两者。

从下面的行中删除文本：

```
<FROM>AUTHOR</FROM>
```

使修改后的该行成为：

```
<FROM></FROM>
```

在IE中查看修改后的文件，该标签将被压缩成下面的格式：

```
<FROM/>
```

这项操作说明在标签对中不一定必须有信息。

请把下列行添加到代码中以<BODY>标签开始的行的前面：

```
<POSTSCRIPT><NOTE>Enjoy yourself!</NOTE></POSTSCRIPT></NOTE>
```

在IE中查看修改后的文件，并注意其错误消息。它会提示你结束标签没有按正确顺序放置。

把下列行添加到文件中：

```
<POSTSCRIPT><NOTE>Enjoy yourself!</NOTE></POSTSCRIPT>
```

在IE中查看修改后的文件，并注意其外观。在这种情况下，特别要注意数据元素的缩排和<MESSAGE>及<POSTSCRIPT>标签前面的红色连字符的外观。其范例如图1-3所示。