

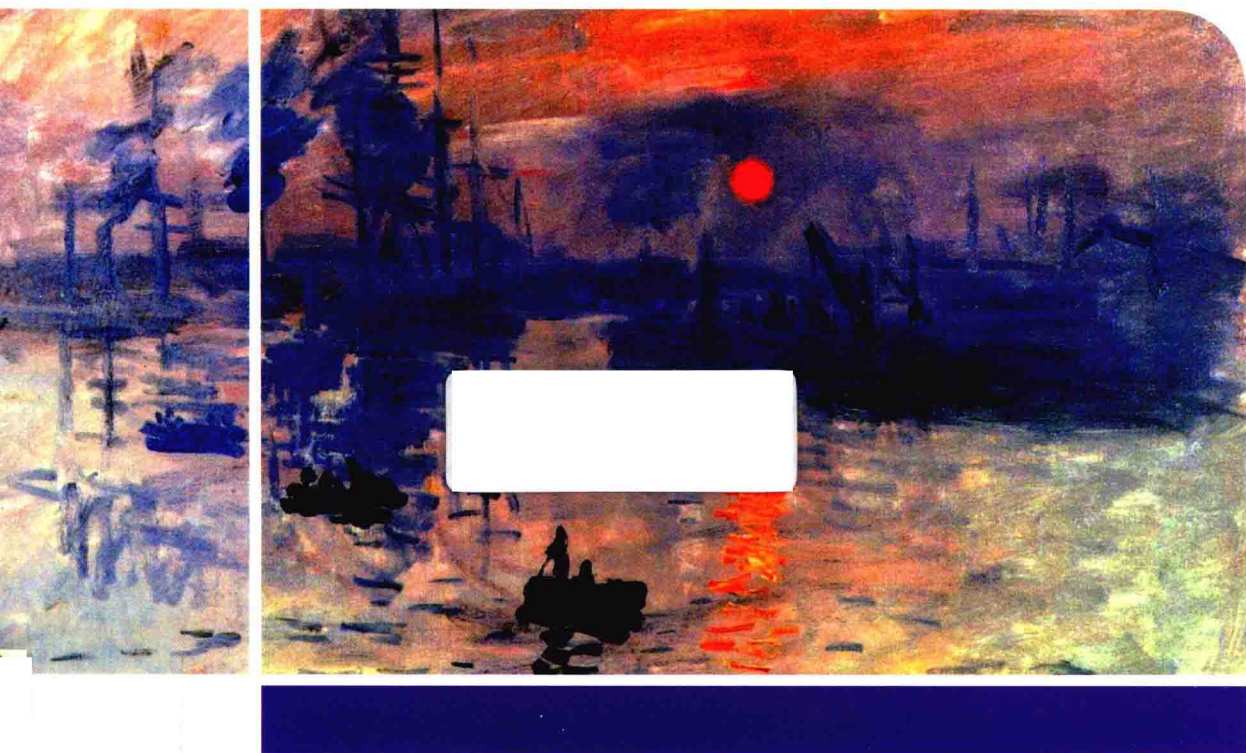


21世纪高等学校计算机
专业实用规划教材

XML 技术应用

(第二版)

◎ 贾素玲 王虹森 王 强 主编
王航飞 于 浩 编著



清华大学出版社



21世纪高等学校计算机
专业实用规划教材



XML 技术应用

(第二版)

◎ 贾素玲 王虹森 王 强 主编

王航飞 于 浩 编著



清华大学出版社

北京

内 容 简 介

本书从 XML 的基本概念开始,结合具体代码示例,由浅入深地介绍了 XML 基础应用,以及结合 XML 的相关技术的应用。

本书共分为 9 章。首先介绍了 XML 的基本概念和语法,接着介绍了用来约束 XML 文档的两种规范:文档类型定义和 XML Schema,然后介绍了如何利用文档对象模型来访问 XML 文档以及如何利用可扩展样式语言来转化 XML 文档,最后介绍了其他一些基于 XML 的应用技术,包括 XQuery、XLink、XPointer、XML Web Services。

本书可作为高等院校计算机相关专业的参考书,也可供相关专业技术人员和教育工作者查阅使用。

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 技术应用/贾素玲,王虹森,王强主编. —2 版. —北京:清华大学出版社,2017
(21 世纪高等学校计算机专业实用规划教材)
ISBN 978-7-302-45617-9

I. ①X… II. ①贾… ②王… ③王… III. ①可扩展语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 283925 号

责任编辑:黄 芝 王冰飞

封面设计:刘 键

责任校对:焦丽丽

责任印制:杨 艳

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>, 010-62795954

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:16.75 字 数:402 千字

版 次:2007 年 7 月第 1 版 2017 年 2 月第 2 版 印 次:2017 年 2 月第 1 次印刷

印 数:1~2000

定 价:35.00 元

产品编号:069914-01

前 言

毫无疑问,21 世纪是 Web 的时代。随着 Internet 的飞速发展,各种新技术应运而生。标记语言特别是 HTML 的出现改变了计算机的发展方向,以文档对象为基础、以浏览器为载体、内容更丰富更具实时性的图形界面取代了单调的命令行界面,没有任何网络基础的普通用户也可以通过鼠标点击而轻松地阅读新闻和收发邮件。然而,HTML 本身却存在着很多缺陷。首先,HTML 是一种样式语言,目前在 Internet 中扮演的只是数据表示的角色,随着信息量的增多,HTML 变得越来越难以胜任。其次,HTML 对浏览器的过度依赖也形成了 HTML 标准的严重不统一,从而导致许多信息表示只能由某种特定的浏览器来解释,HTML 的这些不足使得人们重新思考 HTML 在 Internet 上的角色,并开始研究一门新的语言来弥补 HTML 的缺陷,XML 的产生正是这种思考的最终结果。

XML 是一种专门为 Internet 所设计的标记语言,它的重点是管理数据本身,数据的表示形式交给其他的技术来解决。这种明确的分工带来了更高效的程序设计、更快速的搜索引擎、更统一的数据表示方式及更方便的数据交流。

XML 是一种定义语言的语言,它克服了 HTML 的缺点,给了程序员更自由的空间。近几年,XML 在除数据表示以外的其他方面也得到越来越广泛的应用。从数据存储到数据交换再到系统整合,XML 都发挥着强大的作用。基于 XML 的新技术也如雨后春笋般层出不穷,Web Services、AJAX 等技术相继形成,并逐渐发展成熟。W3C(万维网联盟)也不断为 XML 制定新的标准,并对其进行完善。

因此,作为计算机相关专业的学生,了解并掌握 XML 是十分必要的。本书面向 XML 的初学者,避开 XML 的高级应用,从最简单、最基础的内容入手,使读者对基础知识牢牢掌握,这样再学习和研究高级应用就会容易得多。

全书共分为 9 章。第 1 章从标记语言入手,引出 XML 的概念和相关技术,并介绍如何创建一个最简单的 XML 文档。第 2 章重点介绍 XML 语法,并学习如何创建格式良好的 XML 文档。第 3 章介绍文档类型定义(DTD),DTD 可以详细说明允许或者不允许哪些元素出现在文档中,以及这些元素的确切内容和结构。在第 3 章中,读者将学习如何定义元素、实体和属性,如何将 DTD 附加到文档上,如何使用 DTD 验证文档的正确性以及如何编写自己的 DTD 来解决遇到的问题。第 4 章介绍命名空间和 XML Schema,命名空间可以解决名称冲突的问题,而 Schema 则弥补了 DTD 的一些不足。第 5 章介绍文档对象模型(DOM),DOM 提供了一系列接口,使应用程序可以方便地访问 XML 文档。第 6 章为可扩展的样式语言(XSL),XSL 是比 CSS 更复杂、更强大的样式语言,XSL 转换是 XSL 的一种,它可以将 XML 树状结构转化为其他形式的树状结构。第 7 章介绍 XQuery 查询语言,能够方便地从 XML 文档中提取需要的信息。第 8 章介绍 XLink 和 XPointer,XLink 提供多方

向的超文本链接,XPointer 不仅可以链接到特定文档,还可以链接到特定文档的特定部分。

第 9 章介绍 XML Web Services 的思想和框架,并引出 WSDL 与 SOAP。

第 2~7 章为本书的重点,应该着重掌握。

由于时间仓促,本书不足之处恳请读者批评指正。

编 者
2017 年 1 月

目 录

第 1 章 XML 概述	1
1.1 XML 的概念	1
1.1.1 标记语言	1
1.1.2 XML 的起源	2
1.2 XML 的特性和优点	3
1.2.1 XML 的特性	3
1.2.2 XML 的优点	5
1.3 XML 的处理过程	5
1.4 XML 的设计目标	6
1.5 XML 的相关技术	7
1.6 创建并显示简单的 XML 文档	10
1.6.1 Hello XML World	10
1.6.2 显示 XML 文档	11
本章小结	13
思考题	13
第 2 章 XML 语法	14
2.1 XML 文档结构	14
2.1.1 序言区	14
2.1.2 主体区	15
2.1.3 尾声区	16
2.2 元素和标记	16
2.2.1 元素的命名	16
2.2.2 标记	17
2.2.3 空元素	18
2.2.4 元素的嵌套	18
2.3 属性	19
2.3.1 属性的命名	19
2.3.2 属性的值	20

2.4 引用.....	21
2.4.1 实体引用	21
2.4.2 字符引用	22
2.5 处理指令.....	22
2.6 注释.....	23
2.7 CDATA 节	24
2.8 创建格式良好的 XML 文档	24
2.8.1 XML 文档的声明	24
2.8.2 元素	25
2.8.3 属性	27
2.8.4 其他	28
2.8.5 文档全文	29
本章小结	35
思考题	36
第 3 章 文档类型定义	37
3.1 DTD 的概念	37
3.2 DTD 的调用	38
3.2.1 内部 DTD 的声明	39
3.2.2 外部 DTD 的声明	40
3.2.3 内部 DTD 和外部 DTD 的联合使用	42
3.3 DTD 的结构	42
3.4 元素的声明.....	43
3.4.1 #PCDATA	43
3.4.2 空元素	45
3.4.3 子元素	46
3.4.4 混合型	52
3.4.5 ANY	53
3.5 实体的声明.....	53
3.5.1 内部一般实体	53
3.5.2 外部一般实体	57
3.5.3 内部参数实体	59
3.5.4 外部参数实体	62
3.6 属性的声明.....	63
3.6.1 设置属性的类型	64
3.6.2 属性的默认设置	72
本章小结	75
思考题	76

第 4 章 命名空间和 XML Schema	77
4.1 命名空间的概念	77
4.2 命名空间的语法	79
4.2.1 定义命名空间	79
4.2.2 多个命名空间	79
4.2.3 属性	81
4.2.4 默认命名空间	82
4.3 命名空间的有效性	83
4.4 XML Schema 与 DTD	84
4.5 XML Schema 有效性检验	86
4.6 XML Schema 语法	87
4.6.1 简单类型	88
4.6.2 复杂类型	90
4.6.3 xsd:any 元素和 xsd:anyAttribute 元素	95
4.6.4 数据类型	97
4.6.5 文档举例	99
本章小结	100
思考题	101
第 5 章 文档对象模型	103
5.1 DOM 概述	103
5.1.1 DOM 的概念	103
5.1.2 DOM 的结构	104
5.1.3 DOM 的工作方式	105
5.1.4 DOM 的规范	105
5.1.5 正确理解 DOM	106
5.2 DOM API	106
5.2.1 DOM API 的概念	106
5.2.2 DOM 接口继承树	107
5.2.3 DOM API 核心接口	108
5.3 利用 DOM API 处理结构化文档	126
5.3.1 遍历 XML 文档	126
5.3.2 处理结点	127
5.4 DOM 的简单应用	129
5.4.1 Java XML 解析器	129
5.4.2 遍历文档	130
5.4.3 添加结点	133
5.4.4 删除结点	142

5.4.5 替换结点	148
本章小结	150
思考题	151

第6章 可扩展的样式语言 152

6.1 XSL 的概念	152
6.2 XSLT 概述	152
6.2.1 转换语言	152
6.2.2 树状结构	153
6.2.3 XSLT 文档	154
6.3 XSLT 模板	155
6.3.1 xsl:stylesheet 元素、xsl:template 元素和 match 属性	155
6.3.2 xsl:apply-templates 元素、xsl:value-of 元素和 select 属性	156
6.3.3 xsl:for-each 元素	158
6.4 XSLT 匹配结点的模式	159
6.4.1 匹配根结点、子结点及其他后代结点	159
6.4.2 匹配元素名称	160
6.4.3 通配符	160
6.4.4 使用 ID 匹配单个元素	161
6.4.5 使用@匹配属性结点	161
6.4.6 使用 comments()匹配注释结点	162
6.4.7 使用 processing-instruction()匹配处理指令结点	163
6.4.8 使用 text()匹配文本结点	163
6.4.9 使用或操作符	164
6.4.10 使用[]进行测试	164
6.5 XSLT 输出	166
6.5.1 属性值模板	166
6.5.2 xsl:element 元素	166
6.5.3 xsl:attribute 元素和 xsl:attribute-set 元素	167
6.5.4 xsl:processing-instruction 元素	168
6.5.5 xsl:comment 元素	169
6.5.6 xsl:text 元素	169
6.5.7 xsl:copy 元素	170
6.5.8 xsl:number 元素	170
6.5.9 xsl:sort 元素	172
6.5.10 xsl:variable 元素	173
6.5.11 xsl:if 元素和 xsl:choose 元素	173
6.5.12 xsl:call-template 元素	175
6.5.13 xsl:output 元素	177

6.6	XSLT 合并样式表	177
6.7	XPath 简介	178
6.7.1	XPath 表达式	178
6.7.2	XPath 结点轴	179
6.7.3	XPath 表达式类型	180
6.7.4	XPath 路径定位举例	184
6.8	XSL-FO 简介	185
6.8.1	XSL-FO 区域	185
6.8.2	XSL-FO 主要元素	186
	本章小结	189
	思考题	191
第 7 章	XQuery 查询语言	193
7.1	XQuery 概述	193
7.1.1	XQuery 与 XPath、XSLT 的关系	193
7.1.2	XQuery 示例	193
7.2	XQuery 常用表达式	194
7.2.1	基本表达式	194
7.2.2	算术表达式	194
7.2.3	比较表达式	195
7.2.4	逻辑表达式	196
7.2.5	序列表达式	196
7.2.6	条件表达式	197
7.2.7	量词表达式	197
7.2.8	FLWOR 表达式	198
7.3	XQuery 其他语法	200
7.3.1	命名空间声明	200
7.3.2	变量声明	201
7.3.3	函数声明	201
7.4	XQuery 更新功能	202
7.4.1	插入表达式	202
7.4.2	删除表达式	203
7.4.3	替换表达式	203
7.4.4	更名表达式	204
7.4.5	转换表达式	204
	本章小结	205
	思考题	206

第 8 章 在 XML 中使用链接	207
8.1 XML 中的链接	207
8.2 XLink 概述	207
8.2.1 XLink 的概念	207
8.2.2 XLink 的设计原则	208
8.3 链接元素	209
8.3.1 简单链接	209
8.3.2 链接行为	210
8.3.3 扩展链接	211
8.3.4 弧元素	212
8.3.5 行外链接	214
8.4 XPointer 概述	215
8.5 使用 XPointer 访问信息	217
本章小结	219
思考题	220
第 9 章 XML Web Services	221
9.1 Web Services 概述	221
9.1.1 Web Services 发展历程	221
9.1.2 Web Services 的特点	222
9.1.3 Web Services 体系结构	223
9.1.4 Web Services 协议	225
9.2 WSDL	225
9.2.1 WSDL 结构	226
9.2.2 WSDL 主要元素	227
9.2.3 WSDL 示例	228
9.2.4 WSDL 绑定类型	229
9.3 SOAP	230
9.3.1 SOAP 简介	230
9.3.2 SOAP 设计目标	231
9.3.3 SOAP 示例	231
9.3.4 SOAP 消息交换模式	232
9.3.5 SOAP 消息	233
9.3.6 SOAP 编码规则	234
9.3.7 HTTP 上的 SOAP	234
9.3.8 在 RPC 中使用 SOAP	235

9.4 UDDI	235
本章小结.....	237
思考题.....	237
附录 A XMLSpy 简介	238
附录 B XML 的 Java API	245
参考文献.....	251

第 1 章

XML 概述

本章将介绍 XML 的概念、XML 的特性及 XML 的相关技术等内容,同时帮助读者理解与 XML 相关技术的关键部分是如何协同工作的。希望通过本章的学习,能够使读者对 XML 有一个大概的了解。

1.1 XML 的概念

在描述什么是可扩展标记语言(eXtensible Markup Language,XML)以及 XML 的起源之前,首先介绍什么是标记语言(Markup Language)。

1.1.1 标记语言

标记语言就是使用某种“记号”来表示某种特殊信息的语言,它是一套标记符号和相关语法的集合。它通过其特定的标记符号和语法传达某种标记信息。标记语言由一些格式标记(code)或者控制标记(tag)组成,这些标记决定了信息显示的格式或数据的意义。但是,标记语言本身不能单独存在,它只能用于装饰它所包括的文本内容。如以下这一段代码:

```
<bold>FootballTeams</bold>
```

<bold></bold>就是所使用的标记,FootballTeams 就是标记所修饰的文本内容。标记与文本内容相结合,对文本内容的显示格式进行了控制。在这里就是使得文本 FootballTeams 以加粗的形式显示。同样,再看下一段代码:

```
<Country>Germany</Country>
```

<Country></Country>也是一种标记,它用来说明它所包括的文本信息的意义。

目前大多使用两种标记语言:专用的标记语言与通用的标记语言。专用的标记语言针对特定应用程序与设备,产生专用的编码,这些标记语言是为了满足特定的需求而建立的。通用标记语言则描述文件中文字的意义与结构,但并不定义如何使用文字,这类标记语言并不是为了满足任何特定应用程序而设计,而是为了让许多不同的、一般性的程序使用。通用标记语言所描述的文件比专用标记语言所描述的文件拥有更好的移植性。

1. 专用标记语言

专用标记语言(Specific Markup Language)用来表示某种特殊的信息,通常被某一种或

者几种应用软件所支持,这是基于特殊用途的需要而发展起来的。超文本标记语言(HTML)就是一种专用标记语言,它是为了应用在 WWW 上面的网页而专门设计的,其重点在于信息的显示。

其他的应用软件包括 Word、写字板等也是通过专用标记语言中的标记信息来处理文本中所有特殊格式的,如文字的大小、字体、样式和颜色等。

2. 通用标记语言

通用标记语言(Generalized Markup Languages)的设计并不针对某一特殊的应用软件,或者是某一特殊的用途。通常而言,它只用于描述数据的内容和结构,是一般化的语言。通俗地讲,这种语言就是一种元标记语言,是用于定义语言的语言。这种语言使得许多应用软件都能够识别它,从而提供了一种在不同的应用软件之间相互交流数据的平台。这种语言的特点是,它具有经国际标准化组织通过并被全世界普遍接受的标准。标准通用标记语言(Standard Generalized Markup Language, SGML)就是一种通用标记语言,使用该标记语言,不仅便于计算机处理,而且也便于不同的应用软件之间进行数据共享。XML 正式 SGML 的一个子集,它是从 SGML 发展而来,也是通用标记语言的一种。

1.1.2 XML 的起源

1996 年,万维网联盟(W3C)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。这种语言就是 XML,它继承了 SGML 的规范,而且实际上就是后者的一个子集。

SGML 已经提供了一种可以无限扩展的语言,它允许任何人能够根据自己的需要加以扩充。XML 之所以要较 SGML 更为简化,很大程度上是出于易用性的考虑。人们对标记的读写过程应该使用现有的、简便的、通用的工具,同时,也应当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 XML 的解释器则简单得多。此外,XML 使得现有的 Internet 协议与应用软件之间沟通更加方便,从而简化了数据处理和传输。作为 SGML 的子集,XML 还保持了对现有面向 SGML 的系统的向下兼容性。这样,用 XML 标记过的数据仍然可以在这些系统中使用,为基于 SGML 的行业节省了大笔改造费用,同时与 Web 的结合也使得它们更易于访问。

XML 是一种界定文本数据的简便而标准的方法,它曾经被称为“Web 上的 ASCII 码”。就好像可以使用自己喜爱的编程语言来创建任何一种数据结构,然后同其他人在其他计算平台上使用的其他语言来共享一样。XML 的标记用来说明用户所描述的概念,而属性则用来控制它们的结构。所以,用户可以定义自己所设计出的语法并同其他人共享。

为了使用 XML 文档,W3C 为 XML 标准化了一套应用程序编程接口(Application Programming Interface, API),这样就可以轻松地编制读写 XML 的程序;同时,开发者团体还设计了一套特殊的,免费赠送的,基于事件的替代 API。此外,XML 在设计时就准备支持非欧洲语言和进行国际化。同 HTML 4.01 一样,XML 基于在 ISO/IEC 10646 字符集标准(等同于现在著名的 Unicode 标准)中定义的通用字符集(Universal Character Set, UCS)。可以说,所有促使 HTML 得以流行的特性都出现在了 XML 中。

但是,XML 并不单单是 HTML 的替代品,它与 HTML 有着本质的不同。接下来介绍

XML 的优点,从中可以了解到 XML 的特性以及它与 HTML 相比较有什么优势。

1.2 XML 的特性和优点

XML 之所以能够在世界上这么流行,自然有它自己的优越性,尤其是与 HTML 相比较,它所特有的性质克服了 HTML 的一些固有缺陷。首先,HTML 是一种样式语言,它目前在 Internet 中扮演的只是数据表示的角色,随着信息量的增多,HTML 变得越来越难以胜任。其次,HTML 对浏览器的过渡依赖性也形成了 HTML 标准的严重不统一,从而导致许多信息表示只能由某种特定的浏览器来解释。下面首先介绍 XML 的特性。

1.2.1 XML 的特性

与 HTML 等其他专用标记语言相比较,XML 作为元标记语言主要包含以下特性。

1. XML 的核心是数据

与 HTML 重视文档的格式不同,在 XML 中数据与样式分离,这种分离使得文档的数据从样式中彻底独立出来。XML 的用户可以随心所欲地设计自己的数据内容,而无须考虑这些内容如何显示。同时样式同数据分离后,开发者就可以根据不同的应用设计不同的样式。例如,要在浏览器上显示 XML 文档,就可以为文档套用专门的 CSS 样式表,使得文档输出有类似于 HTML 的效果;如果文档要输出到文本编辑软件 Word 中进行打印,就可以套用 Word 的样式表,使得输出具有 Word 的效果。这种数据和样式分离的设计,不仅提高了 XML 文档的利用率,而且提高了 XML 的数据容量和质量,还大大方便了对于数据的查询和提取。

众所周知,在 HTML 中,大部分的标记是用来控制文档的布局和外观的,如下面这一段 HTML 代码。

```
<h1>FootballTeams</h1>
<table border = "1" cellpadding = "1">
  <tr>
    <td>Name</td>
    <td>Country</td>
  </tr>
  <tr>
    <td>FC Bayern Munchen</td>
    <td>Country</td>
  </tr>
  <tr>
    <td>FC Barcelona</td>
    <td>Spain</td>
  </tr>
</table>
```

这段代码在 Web 浏览器 Firefox 中显示的效果如图 1-1 所示。

可以看到,上面这么长的一段代码,实际展现在用户面前的有用数据相当少。因此,作



图 1-1 以 HTML 格式呈现的足球队资料

为存储数据的媒介,HTML 并不是一个好的选择,同样的数据使用 XML 来描述和存储会简练方便很多,如下所示。

```
<?xml version = "1.0" encoding = "UTF - 8"?>
<FootballTeams>
  <Team>
    <Name>FC Bayern Munchen</Name>
    <Country>Germany</Country>
  </Team>
  <Team>
    <Name>FC Barcelona</Name>
    <Country>Spain</Country>
  </Team>
</FootballTeams>
```

2. XML 数据的自我描述性

由于 XML 是一种元标记语言,因此它与 HTML 不同,不需要使用专门的标记符号进行元素和数据的描述,用户可以自定义标记的名称,从而使得 XML 的数据具有了自我描述性,如上例中的“< Name> FC Bayern Munchen</Name>”。

用户可以很清楚地明白这个元素的用处就是用来描述球队的名称,而元素标记中的内容就是具体的球队名称。由于 HTML 必须使用专门的标记,特别是大量的用于控制显示格式的标记来进行文档设计,因此无法直接理解每一个标记中所包含内容的含义,这给网上自动化数据处理等造成了很大的麻烦。

XML 这种数据的自我描述性使得 XML 可以在任意平台下使用,也可以在任意时刻使用,它为网上数据的自动化处理提供了解决途径。

3. XML 支持 Unicode 字符集

在以往的 Web 程序中,从程序的设计到元素的命名,基本上所有的标记和语言都是英文。XML 支持 Unicode 所有的字符集,它允许使用双字节的字符来定义标记和编写程序,因此对于中国的程序员而言,可以很方便地使用汉字来命名 XML 文档中的元素和属性,从而使文档更具可读性。

1.2.2 XML 的优点

基于上述特性,概括出 XML 的三大优点。

(1) XML 可以实现异构数据之间的数据转换。由于 XML 是一种元标记语言,它可以应用于任意平台之上。因此它具有了同编程语言 Java 等一样的跨平台特性,这种特性为异构数据的交互提供了一种数据交换的标准。它是一种公共的交互平台,一种数据源只要能够将它的数据转换成 XML 格式就能够被另外一种数据源有效地识别,这为数据的迁移以及整合提供了很好的工具。

(2) XML 具有较好的保值性。XML 的保值性来自它的先驱之一,即 SGML 语言。SGML 是一套有着十几年历史的国际标准,它最初设计的一大目标就是要为文件提供 50 年以上的寿命。文件的寿命是一个非常重要的问题,现在人们可以通过流传至今的大量历史文献来了解我国悠久而辉煌的历史,如果没有这些文字资料,读者恐怕对“唐宋元明清”没有一点概念,同样下一代也要通过这样的文字资料来了解历史。现在大部分资料都已经电子化了,如果若干年以后,后人面对一大堆用 Word 写的文档,而没有专用的软件工具打开这些文档,那么这些文档包含的信息就有可能丢失。这就是 SGML 和 XML 的设计初衷。它们不但能够长期作为一个通用的标准,而且很容易向其他格式的文件转化。

(3) XML 遵循严格的语法要求。虽然 XML 中可以自由定义标记,但是与 HTML 相比,XML 的书写格式更加严格。HTML 的语法要求并不严格,浏览器可以显示有文法错误的 HTML 文件。但 XML 不同,它不但要求标记配对、嵌套,而且还要求严格遵守 DTD(文档类型定义)的规定。

读者可能会认为编写 XML 文件时要严格遵循语法要求是一件非常麻烦的事情。实际上一个具有良好语法结构的网页文件可以提供较好的可读性和可维护性,从长远来看是大有益处的。这还大大减轻了浏览器开发人员的负担,提高了浏览器的效率。

1.3 XML 的处理过程

基本上可以说 XML 是一种文档格式,它是一系列关于 XML 文档看起来是什么样的规则集合。与 XML 标准的符合程度有两个级别:一个是文档的结构完整性;另一个是文档的正确性。在本书中将介绍如何编写结构完整以及格式良好的文档。

XML 除了像 HTML 一样可以用于 Internet 以及 Web 上页面文件格式外,还可以用于不同程序之间的数据交换。同其他所有的数据格式一样,XML 文档需要具有一定的内容之后才能被正式应用,因此对于数据看起来应该是什么样子的,仅仅一个 XML 文档的规范是不够的,用户需要了解如何编辑 XML 文档,处理程序是如何读取 XML 文档的数据,并将这些数据传输给应用程序,以及应用程序是如何处理这些数据的。

要了解 XML 的处理过程,首先必须了解编辑器、语法分析和处理程序、用户端应用程序。

1. 编辑器

XML 文档在大部分情况下是通过编辑器创建的。编辑器可以是基本的文本编辑器,如 Notepad(记事本),也可以是一些所见即所得的编辑器,如 XMLSpy、Adobe FrameMaker