新版

21世纪

高职高专系列教材

XML程序设计

◎龚小勇　瞿 中　王荣辉　编著

提供电子教案增值服务

机械工业出版社
CHINA MACHINE PRESS

21 世纪高职高专系列教材

XML 程序设计

龚小勇　瞿 中　王荣辉　编著

机 械 工 业 出 版 社

XML(eXtensible Markup Language)是指可扩展标记语言。本书全面、系统地介绍了XML的基本原理和各种实用技术。全书共7章,主要内容包括:XML简介、XML语法、XML模式、XML链接、格式化XML、访问XML、XML应用及前景等。

本书技术阐述与实训指导相结合,强调理论以够用为度,始终以介绍XML中已成熟的标准和应用技术为主。书中的应用实例来自工程实践领域,既可相对独立,又有一定的内在联系。

本书不仅适合作为高职高专院校计算机及相关专业的教材,也可以供XML技术爱好者参考。

图书在版编目(CIP)数据

XML程序设计/龚小勇等编著.—北京:机械工业出版社,2006.12
ISBN 978-7-111-20568-5

Ⅰ.X... Ⅱ.龚... Ⅲ.可扩充语言,XML—程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字(2006)第155103号

机械工业出版社(北京市百万庄大街22号 邮政编码 100037)

策 划:胡毓坚
责任编辑:车 忱
责任印制:洪汉军
三河市国英印务有限公司
2007年1月第1版·第1次印刷
184mm×260mm·16.25印张·398千字
0001-5000册
定价:23.00元

凡购本书,如有缺页,倒页,脱页,由本社发行部调换
销售服务热线电话:(010)68326294
购书热线电话:(010)88379639 88379641 88379643
编辑热线电话:(010)88379739
封面无防伪标均为盗版

21世纪高职高专计算机专业系列教材
编委会成员名单

出版说明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位，促进学生技能的培养，以及教材内容要紧密结合生产实际，并注意及时跟踪先进技术的发展等指导精神，机械工业出版社组织全国40余所院校的骨干教师，对在2001年出版的“面向21世纪高职高专系列教材”进行了修订，并将修订后的这套教材改名为“21世纪高职高专系列教材”。

在几年的教学实践中，本系列教材获得了较高的评价。因此，在修订过程中，各编委会保持了第1版教材“定位准确、注重能力、内容创新、结构合理和叙述通俗”的编写特色。同时，针对教育部提出的高等职业教育的学制将由三年逐步过渡为两年，以及强调以能力培养为主的精神，制定出了本次教材修订的原则：跟上我国信息产业飞速发展的节拍，适应信息行业相关岗位群对第一线技术应用型操作人员能力的要求，针对两年制兼顾三年制，理论以“必须、够用”为原则，增加实训的比重，并且制作了内容丰富而且实用的电子教案，实现了教材的立体化。

针对课程的不同性质，修订过程中采取了不同的处理办法。核心基础课的教材在保持扎实的理论基础的同时，增加实训和习题；实践性较强的课程强调理论与实训紧密结合；涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。此外，在修订过程中，还进行了将几门课程整合在一起的尝试。所有这些都充分地体现了修订版教材求真务实、循序渐进和勇于创新的精神。在修订现有教材的同时，为了顺应高职高专教学改革的不断深入，以及新技术新工艺的不断涌现和发展，机械工业出版社及教材编委会在对高职高专院校的专业设置和课程设置进行了深入的研究后，还准备出版一批适应社会发展的急需教材。

信息技术以前所未有的速度飞快地向前发展，信息技术已经成为经济发展的关键手段，作为与之相关的教材要抓住发展的机遇，找准自身的定位，形成鲜明的特色，夯实人才培养的基础。为此，担任本系列教材修订任务的教师，将努力把最新的教学实践经验融于教材的编写之中，并以可贵的探索精神推进本系列教材的更新。由于高职高专教育正在不断的发展中，加之我们的水平和经验有限，在教材的编审中难免出现问题和错误，恳请使用这套教材的师生提出宝贵的意见和建议，以利我们今后不断改进，为我国的高职高专教育事业作出积极的贡献。

机械工业出版社

前　言

XML(eXtensible Markup Language,可扩展标记语言)是由W3C(万维网联盟)于1998年2月发布的一种标准。它是SGML的一个简化子集,它将SGML的丰富功能与HTML的易用性结合到Web的应用中,以一种开放的自我描述方式定义了数据结构,在描述数据内容的同时能突出对结构的描述,从而体现出数据之间的关系。这样组织的数据对于应用程序和用户都是友好的、可操作的。

本书全面、系统地介绍了XML的基本原理和各种实用技术,全书共分为7章。第1章简要介绍XML的历史起源、编辑浏览工具以及XML的优点。第2章概括地介绍了XML语法规则,并通过对实例的详尽分析,告诉读者如何建立和使用一个正确的XML文档。第3章主要从DTD和XML Schema两个角度向读者介绍了XML模式的使用,让读者能更明确地了解和掌握XML文档的整体结构。第4章介绍了XML链接,第5章介绍了格式化XML,通过这两章的学习,为读者今后进行更高层次的XML应用开发打下基础。第6章是XML高级编程的入门内容,介绍了通过数据岛、DOM和.NET等多种方法来访问XML数据和文档。第7章讨论了XML应用及前景,以进一步开阔读者视野,提高学习和掌握XML的积极性。

本书在编写过程中始终围绕着"理论够用、注重操作和应用"的原则。对于初学者来说,可以由浅入深地了解XML的语言规范和应用技术;对于已经有XML语言基础的读者来说,本书安排了大量的应用实例和练习,使读者可以从应用的角度来进一步提高XML的开发能力。本书可以作为高职高专学校的专用教材或作为XML技术爱好者的参考书。

本书第1、2章由重庆电子科技职业学院龚小勇编写,第3、4、5章由重庆电子科技职业学院王荣辉编写,第6、7章由重庆邮电大学瞿中编写,全书由龚小勇统稿。本书附代的软件包中包含了电子教案及所有章节的实例源代码和习题答案,可在http://www.cmpbook.com免费下载。

由于编者水平有限,编写时间仓促,书中难免存在缺点和错误,恳请广大读者批评指正。若有问题,可发电子邮件至jsjfw@mail.machineinfo.gov.cn。

作　者

目　录

第1章　XML简介

本章要点

- 标记语言的基本概念
- XML的优点及主要用途
- XML的编辑浏览工具

1.1　XML的产生

XML(eXtensible Markup Language)代表可扩展标记语言,是由World Wide Web Consortium(W3C)的XML工作组定义的。这个工作组是这样描述该语言的:“XML是SGML(Structured Generalized Markup Language)的子集,其目标是允许普通的SGML在Web上以目前HTML的方式被服务、接收和处理,XML被设计成易于实现,且可在SGML和HTML之间互相操作。”XML集SGML和HTML的优势于一身,具有易于编辑、便于管理、适于存档、容易查询等诸多优势,已经成为网络发展的又一个亮点。

1.1.1　标记语言

标记语言起源于传统印刷。印刷之前必须排版。在电子出版业出现之前,人们需要用手抄或者打字的方式先复制手稿。然后,再对副本以人工方式标记,并加上编辑说明,告诉印刷排版人员如何处理版面排放以及其他制作问题。排版人员根据文件上的标记和说明选择文本印刷格式,如字型、段落起始点、边界、对齐方式等,进行铅字排版,完成印刷前的制版工作。

采用计算机排版也有相同的处理程序。通过选择字体和设置字符间距、段落、换行符等,把“格式描述码”添加到文档中,告诉计算机关于文件的结构以及文件应显现什么样的外观。这个“格式描述码”是电子式的标注,即需要通过各种电子式的标注代码,将排版信息告知计算机,如大家所熟悉的Microsoft Word、写字板等文本编辑器都是借助标注代码来定义格式与外观的。

通俗地讲,标记语言就是一种用来给文本添加“标注代码”以指明文档中文本编排格式的语言。一般由定义文档格式的一些代码和控制标记组成。例如,Microsoft Word、写字板等文本编辑器都支持的文件存储格式RTF(Rich Text Format)实际上就是一种简单标记语言。

下面通过一个示例来认识RTF标记语言。

【例1-1】　查看一个简单RTF文档的标记。

1)打开写字板,输入“We are learning markup language!”,选择字体“Times New Roman”,字号20,如图1-1所示。

2)单击“保存”按钮,在“保存为”对话框中,选择保存在“例题”文件夹,输入文件名“1.1”,选择保存类型为RTF,如图1-2所示,然后点击“保存”按钮。

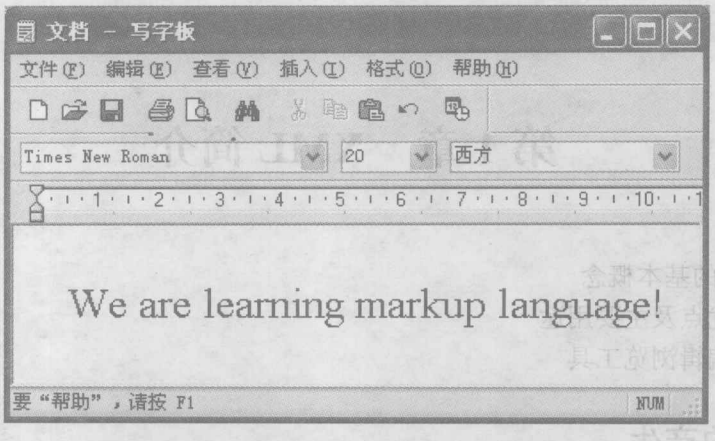

图 1-1　在写字板中输入一段文字

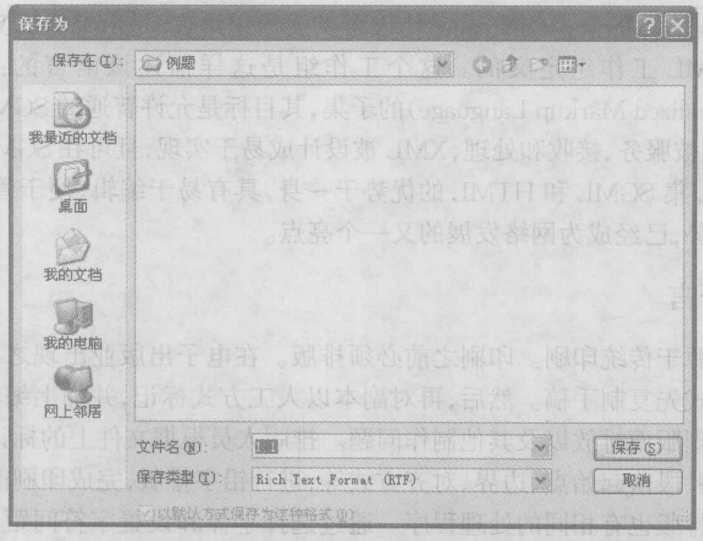

图 1-2　保存 RTF 文档

3）在记事本中打开刚刚建立的“1.1.RTF”文件，看到的是加上标记后的样子。

可以看到在图 1-3 中显示的内容除了刚才输入的“We are learning markup language!”外，还有其他字符，这些字符就是 RTF 标记。这些标记告诉支持 RTF 格式的文本编辑器如何显示文件内容。例如，标记“ \fcharset0 Times New Roman”指定了显示字体“Times New Roman”。因为记事本不能解读 RTF 标记，所以只能将文件的所有代码显现出来。

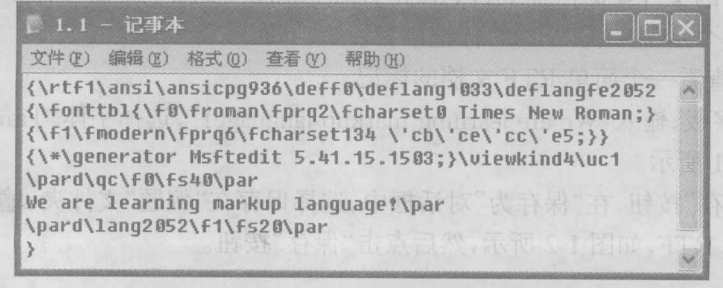

图 1-3　用记事本打开“1.1.RTF”文件

下面用 Microsoft Word 文本编辑器打开“1.1.RTF”文件，看看显示结果如何。

图 1-4 给出了用 Word 打开“1.1.RDF”文件的显示结果，与图 1-1 中写字板编辑的结果相同。这是因为 Microsoft Word 能解读其中的 RTF 标记，所以，它按 RTF 标记指定的字体“Times New Roman”和字号 20 显示文字“We are learning markup language!”。

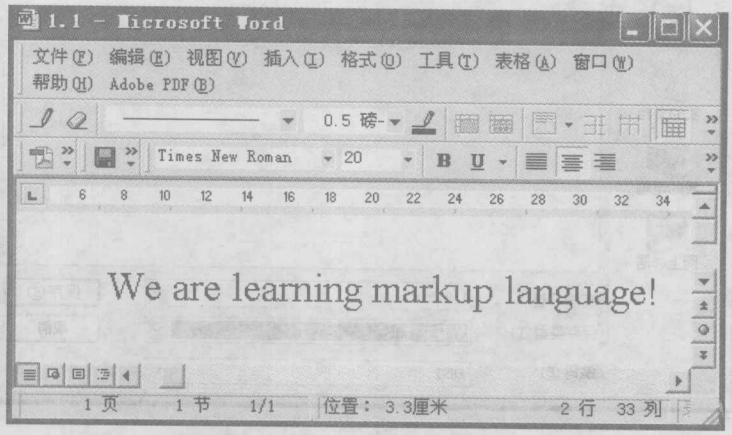

图 1-4 在 Microsoft Word 中打开“1.1.RTF”文件

下面来看另一种常用的标记语言 HTML(Hyper Text Markup Language)。HTML 也定义了一系列标记，每个标记表明了一定的显示格式，主要用于编写各种各样的网页，网页中可包含标题、文本、表格、列表、图片、超链接等内容。HTML 文档(即同时包含了纯文本和关于文本显示格式的 HTML 标记的文档)由一个 HTML 处理工具，例如一个浏览器，进行读取，然后再根据上述标记规则来加以显示。尽管 HTML 文档具有如此强大的功能，但是它却是一种纯文本格式的文档。可以用记事本等文本编辑工具来手工编写，也可以用 Dreamweaver 和 FrontPage 等可视化网页编辑工具自动生成 HTML 文档。

【例 1-2】 查看 HTML 的标记格式及显示效果。

1) 在记事本中，输入一段 HTML 文档，如图 1-5 所示，其中，用尖括号括起来的是标记。

图 1-5 用记事本编辑 HTML 文档

2) 选择“文件”菜单中的“保存”。在“另存为”对话框中，选择保存在“例题”文件夹，输入文件名“1.2.htm”，然后选择保存类型为“所有文件”，如图 1-6 所示。

3) 双击“例题”文件夹中的“1.2.htm”文件，在 IE 浏览器中显示的结果如图 1-7 所示。

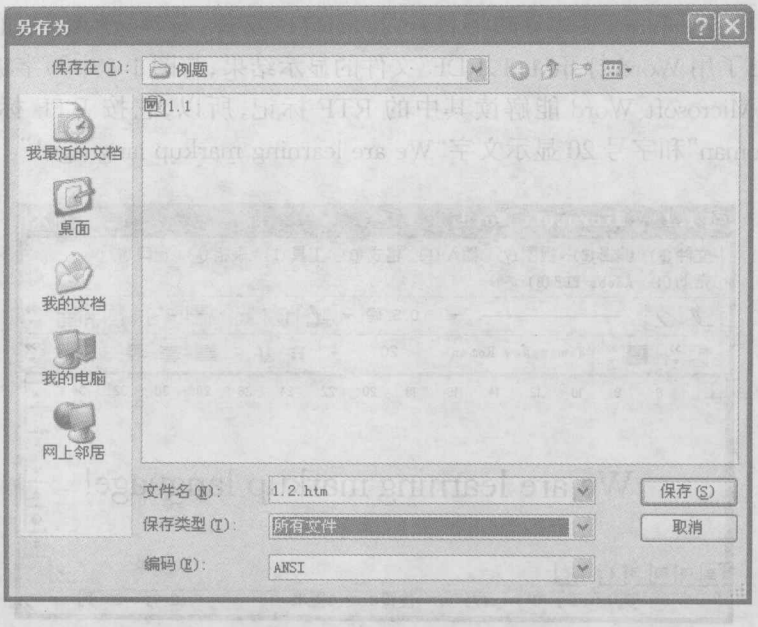

图 1-6　保存 HTML 文档

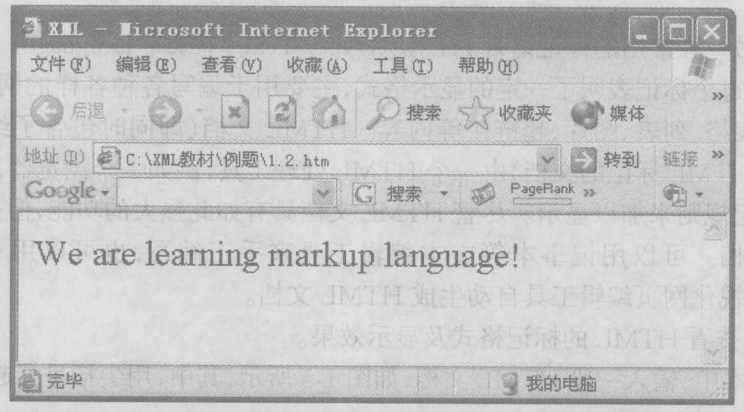

图 1-7　在 IE 中显示“1.2.htm”文件

IE 浏览器用字体“Times New Roman”显示“We are learning markup language!”。这是因为已经在“1.2.htm”文件中设置了＜FONT face＝″Times New Roman″ size＝″5″＞的标记，这个标记告诉 IE 浏览器要用“Times New Roman”字体显示标记后面的“We are learning markup language!”。

通过上面两个例子，读者对标记语言及其作用已经有了一定的了解。那么，作为一种新的标记语言，XML 是如何产生的？它与 RTF、HTML 有何不同呢？下面的内容将回答这些问题。

1.1.2　XML 的来源

XML 有两个先驱——SGML 和 HTML，这两个语言都是非常成功的标记语言，但是它们都在某些方面存在着与生俱来的缺陷。XML 正是为了解决它们的不足而诞生的。

SGML 从 20 世纪 80 年代初开始使用，是所有标记语言的母语言，它为语法标记提供了异

常强大的工具,同时具有极好的扩展性,因此在分类和索引数据中非常有用。目前,SGML 多用于科技文献和政府办公文件中。

但是,SGML 非常复杂,其复杂程度对于网络上的日常应用简直不可思议。还有最关键的一点是,几个主要的浏览器厂商都明确拒绝支持 SGML,这无疑是 SGML 在网上传播遇到的最大障碍。

相反,HTML 免费、简单,而且它获得了广泛的支持。HTML 最初于 1990 年由 CERN 设计,它是一个非常简单的 SGML 语言,可以方便普通人的使用。而正如设计之初所构想的那样,HTML 目前在世界范围内得到了广泛的应用。

但 HTML 的功能和扩展性有限。1996 年人们开始致力于设计一种新的标记语言,它既具有 SGML 的强大功能和可扩展性,同时又具有 HTML 的简单性。万维网联盟 W3C 决定专门成立一个 SGML 专家小组来从事此项工作,由 Sun 公司大名鼎鼎的 Jon Bosak 担任小组的指挥。

事实上,Bosak 和他领导的专家小组对 SGML 所做的贡献就像 Java 研究组对 C++ 做出的贡献一样。SGML 中所有非核心的、未被使用的和含义模糊的部分都被删除,剩下的就成为短小精干的标记语言——XML。对于 XML 的描述只有 26 页,而当初 SGML 的描述却长达 500 页之多。值得一提的是,对于 XML 的描述尽管篇幅只是 SGML 的二十分之一,但是 SGML 中所有的精华都被保留了下来。

这以后,XML 不断发展演化,并且从化学标记语言(Chemistry Markup Language,CML)和数学标记语言(Mathematical Markup Language,MathML)中汲取了大量的经验。1997 年春天,可扩展链接语言(eXtensible Link Language,XLL)草案已被拟定,到了 1997 年夏天,微软也开始了关于频道描述格式(Channel Definition Format,CDF)的定义工作,这应该算是 XML 的第一个真正的应用。

最后,W3C 于 1998 年 2 月批准了 XML 的 1.0 版本,一种崭新而大有前途的标记语言诞生了。

1.1.3 XML 概述

XML 与 RTF、HTML 的不同在于"X"(eXtensible),即可扩展性。XML 不像 RTF、HTML 那样,提供了一组事先定义好的标记,而是提供了一个定义标记的标准,利用这个标准,用户可以根据实际需要定义自己的标记。因此,可以使用 XML 描述任意类型的数据。例如,假设要描述一个网上音乐店客户的编号、客户地址、姓名、订单等信息,就必须创建用于每项数据的标记。新创建的标记可在文档类型定义(Document Type Definition,DTD)中加以描述。下面介绍一个非常简单的例子。

【例 1-3】 用 XML 文档描述网上音乐店的客户信息。

```
<? xml version="1.0" encoding="gb2312"? >
<! DOCTYPE 客户名单 SYSTEM "C: \ XML 教材 \ 例题 \ 1.3.dtd" >
<客户名单>
    <客户>
        <编号>KH-0165</编号>
        <客户地址>重庆</客户地址>
        <姓名>任建兴</姓名>
```

```
        <订单>DD-345</订单>
    </客户>
    <客户>
        <编号>KH-0166</编号>
        <客户地址>北京</客户地址>
        <姓名>马辛</姓名>
        <订单>DD-346</订单>
    </客户>
</客户名单>
```

这一段代码是一个非常简单的 XML 文档，看上去它和 HTML 非常相像，但细心的人会发现这里的标记所代表的不再是显示格式，而是对于客户信息的语义解释。不过，仅仅给数据设置标记还不够，为了让计算机读懂这些数据，XML 还必须指定标记的语法。

换句话说，如果想让计算机应用程序读懂并能处理这段数据，它还必须知道什么是一个有效的标记，如何处理一个有效的标记。具体地说，浏览器如何知道标记“<编号>”是什么含义？它究竟是不是一个合法的标记？因此，XML 必须能够告诉应用程序它所采用的标记的语法，以便应用程序能够对其进行正确的处理。

在 XML 中，标记的语法是通过文档类型定义 DTD 或模式 Schema 来描述的。也就是说，通过 DTD 或 Schema 来描述什么是有效的标记，从而进一步定义 XML 的结构。后续章节中将详细讨论 DTD 和 Schema 的定义方法。这里先了解一下例 1-3 用到的 DTD 文件“1.3.dtd”中的内容。

```
<? xml version="1.0" encoding="UTF-8"? >
<! ELEMENT 客户名单 (客户+)>
<! ELEMENT 客户 (编号, 客户地址, 姓名, 订单)>
<! ELEMENT 编号 (#PCDATA)>
<! ELEMENT 客户地址 (#PCDATA)>
<! ELEMENT 姓名 (#PCDATA)>
<! ELEMENT 订单 (#PCDATA)>
```

至于 XML 文档的外观显示，可通过搭配样式单(或称为样式表)来描述，如层叠样式单 CSS(Cascading Style Sheets)和可扩展样式单语言 XSL(eXtensible Style sheet Language)。通过指定和 XML 相关联的样式单文件，为 XML 文档定义显示格式。样式单的使用将在后续章节中作具体描述。

从上述例题中可以看出，与 HTML 不同，XML 将结构、内容和显示格式分离，可通过编写多个样式单，将同一个 XML 源文档用不同的方式呈现出来。

1.1.4 XML 的设计目标

下面是 W3C Web 站点(http://www.w3.org/TR/REC-xml)上的 XML 正式规范中阐述的 XML 的 10 个设计目标：

(1) XML 应该可以直接用于 Internet。XML 的主要设计目标是在 Web 上保存并传递信息。

(2) XML 应该支持各种应用程序。尽管 XML 的主要目的是通过服务器和浏览器程序在 Web 上传递信息,但是它还可以被其他类型的程序使用。例如,XML 已被用于在不同程序之间交换信息,用来发布和更新软件等。

(3) XML 应该与 SGML 兼容。XML 是 SGML 的专用子集,这种特性的一个好处是 SGML 软件工具可以很容易地适用于 XML。

(4) 编写处理 XML 文档的应用程序应该很简单。如果希望 XML 有实用性,那么它必须很容易编写出能够处理 XML 文档的浏览器和其他程序。实际上,从 SGML 中派生出 XML 子集的主要原因是,编写处理 SGML 文档的程序很笨拙。下列设计目标主要是为了支持这个基本目标而服务的。

(5) XML 中可选特性的数目应该尽可能地少,理想情况是零。这使得编写处理 XML 文档的程序更容易。SGML 中有大量冗余的可选特性是它为什么被认为对于定义 Web 文档来说不实用的主要原因。

(6) XML 文档应该便于人阅读而且相当清晰。这种便于人阅读的特性使 XML 区别于大部分被数据库和字处理文档所使用的专用格式。人们可以很容易地阅读 XML 文档,因为它是用纯文本编写的,而且具有类似树型的逻辑结构。可以通过为文档元素、属性和实体选择有意义的名字,并且增加有用的注释来增强 XML 的可读性。

(7) XML 设计应该很快地准备好。当然,只有当程序员和用户团体都采纳 XML 时,XML 才是一种可行的标准。因此,在这个团体开始采纳另一个标准之前,这个标准还需要完善,软件公司应该以很快的速度生成该标准。

(8) XML 的设计应该正式而且简洁。XML 规范是用一种定义计算机语言的正式语言编写的,这种正式语言解决了二义性的问题,使得它更容易编写 XML 文档,尤其是 XML 处理软件,这就进一步地鼓励了人们采纳 XML。

(9) XML 文档应该易于创建。要让 XML 成为一种适用于 Web 文档的实用标记语言,不仅要求 XML 处理程序必须很容易编写,而且要求 XML 文档本身必须很容易创建。

(10) XML 标记的简洁是最不重要的。为了满足前面目标 6 的要求(XML 文档应该便于人阅读且相当清晰),XML 标记不应过于简洁,以致于含义模糊。

1.2 使用 XML 的原因

SGML 虽然功能强大,但太复杂,无法有效地在网上传递信息。由于有太多的可选功能与其他特性,令编写在网页浏览器中处理与显示 SGML 信息的软件变得非常困难。

HTML 虽然源于 SGML,由于种种原因,HTML 越来越侧重于信息的表示,标签中原本就很微弱的信息描述含义也被削弱了。它难以满足网络进一步发展的需要。

1.2.1 HTML 的缺点和不足

HTML 是最早应用于网络信息传输的标记语言,也是近几年互联网上最普及的一种网页制作通用语言。它侧重于主页表现形式的描述,大大丰富了主页的视觉和听觉效果,为推动信息和知识的网上交流发挥了不可取代的作用。下面对 HTML 作一个简单介绍。

HTML 提供了固定的预定义元素集,可以使用它来标记一个 Web 页的各个组成部分。

预定义的元素有:标题、图片、列表、表格、图像、链接等。

【例 1-4】 用 HTML 创建个人主页。

```
<HTML>
<HEAD><TITLE>Home Page</TITLE></HEAD>
<BODY>
  <H1><IMG SRC="MainLogo.gif">Michael Young's Home Page</H1>
  <P><EM>Welcome to my Web site! </EM></P>
  <H2>Web Site Contents</H2>
  <P>Please choose one of the following topics:</P>
  <UL>
    <LI><A HREF="Writing.htm"><B>Writing</B></A></LI>
    <LI><A HREF="Family.htm"><B>Family</B></A></LI>
    <LI><A HREF="Photos.htm"><B>Photo Gallery</B></A></LI>
  </UL>
    <H2>Other Interesting Web Sites</H2>
    <P>Click one of the following to explore another Web site:</P>
  <UL>
    <LI><A HREF="http://www.yahoo.com/">Yahoo Search Engine</A></LI>
    <LI><A HREF="http://www.amazon.com/">Amazon Bookstore</A></LI>
    <LI><A HREF="http://mspress.microsoft.com/">Microsoft Press</A></LI>
  </UL>
</BODY>
</HTML>
```

如图 1-8 所示,Microsoft Internet Explorer 显示了该页面:

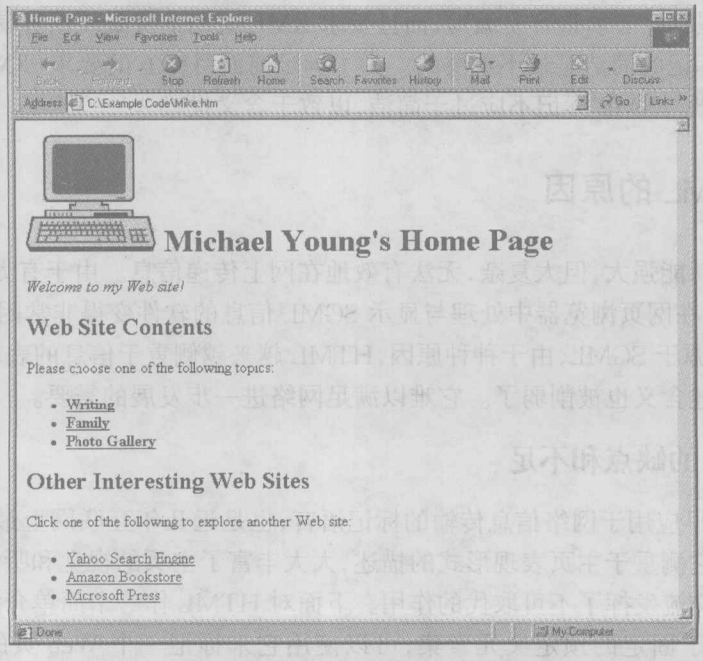

图 1-8 用 HTML 创建个人主页

每一个元素都以起始标签(start-tag)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(end-tag)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(content)是位于起始标签和结束标签之间的文本。图 1-9 显示了一个 HTML 元素的组成部分。

图 1-9 HTML 元素的组成

HTML 的一些元素如表 1-1 所示。

表 1-1 HTML 元素举例

HTML 元素	所标记的页面组成部分
HTML	整个页面
HEAD	标题信息
TITLE	页面的标题,它显示在浏览器的标题栏
BODY	浏览器显示的文本主题
H1	顶级的标题
H2	第二级的标题
P	文本段落
UL	项目符号列表(未排序的列表)
LI	列表中的一个项目(列表项)
IMG	图像
A	链接到另一个位置或页面(一个锚元素)
EM	一块斜体字(强调的)文本
B	一块粗体字文本

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。例如,浏览器通常用最大的字体显示 H1 标题,H2 标题用一个较小的字体,元素 P 表示更小的字体。它在未排序的列表中把一个 LI 元素显示成一个带有项目符号的、缩进的段落。此外,它把元素 A 转换成带有下划线的链接,用户可以单击它访问一个不同的地方或页面。

尽管自从第一个 HTML 版本后,预定义的 HTML 元素已经得到了极大的扩展,但是 HTML 仍然无法适用于已定义的众多文档类型。HTML 有以下主要缺点:

(1) HTML 的标记是固定的,用户不能自行新增。如果 HTML 中没有所需的标记,用户就没有办法了,这时只好等待它的下一个版本,希望在新版本中能够包括所需的标记。

(2) HTML 标记只表达如何显示信息,缺乏对内容含义的表达能力。除少数标记,如<P>、<Title>外,几乎全部都是用来表示网页的布局和显示外观的,它并不能提示 HTML 文档中