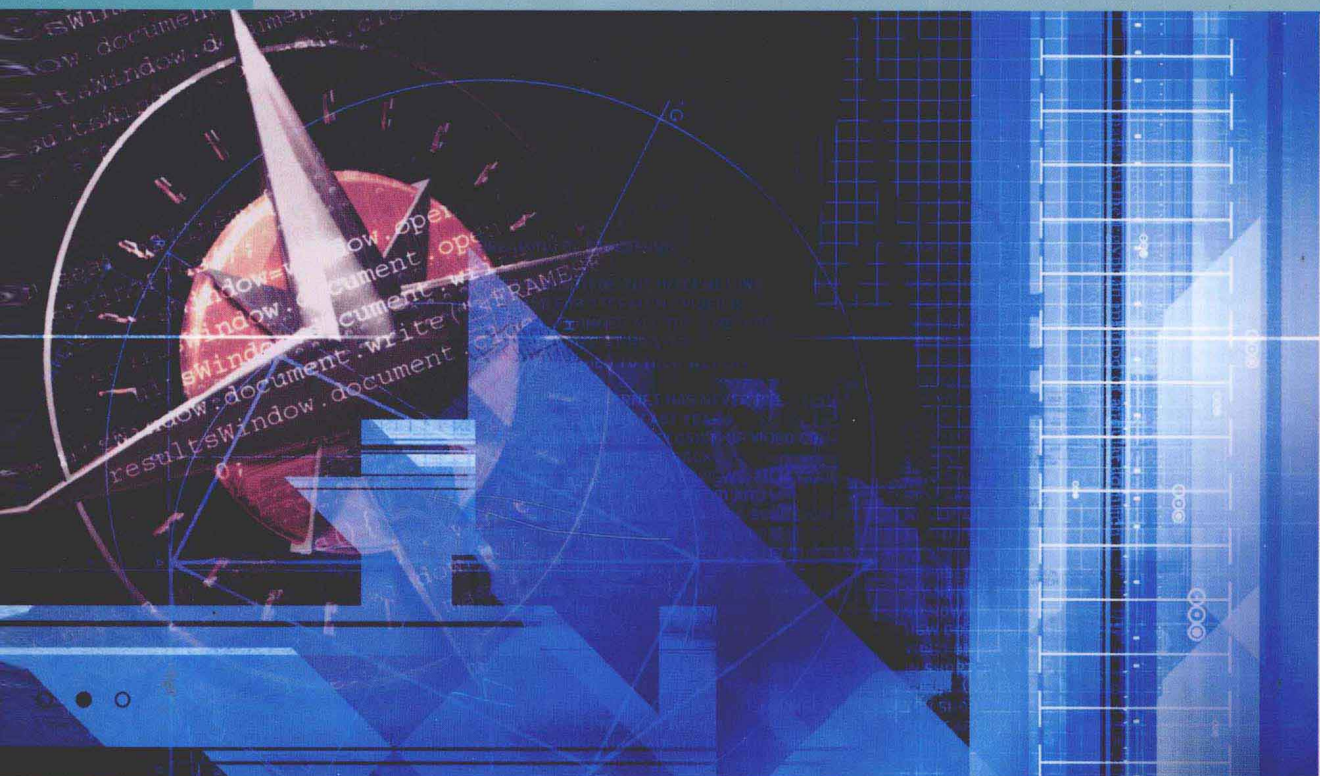


职业技术教育软件人才培养模式改革项目成果教材



# XML 基础

吴扬扬 颜毛智



高等教育出版社

职业技术教育软件人才培养模式改革项目成果教材

# **XML 基 础**

吴扬扬 颜毛智

高等教育出版社

## 内容提要

本书是职业技术教育软件人才培养模式改革项目成果教材。

本书共 7 章,主要内容包括 XML 概述、XML 语法、XML 模型、样式语言、XML 链接、解析器和应用程序接口以及应用案例介绍。通过回答:“XML 是什么”,“为什么需要使用 XML”以及“如何使用 XML”这几个问题来介绍目前最重要和最实用的 XML 技术。

本书适用于高等职业学校、高等专科学校、成人高校、本科院校举办的二级职业技术学院,也可供示范性软件职业技术学院、继续教育学院、民办高校、技能型紧缺人才培养使用,还可供本科院校师生、计算机从业人员和爱好者参考使用。

## 图书在版编目(CIP)数据

XML 基础/吴扬扬,颜毛智. —北京:高等教育出版社,  
2005.2

ISBN 7-04-016414-0

I. X... II. ①吴...②颜... III. 可扩充语言,XML  
—程序设计—高等学校;技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2005)第 003361 号

策划编辑 冯 英 责任编辑 焦建虹 封面设计 王凌波 责任绘图 尹文军  
版式设计 马静如 责任校对 杨凤玲 责任印制 朱学忠

出版发行 高等教育出版社  
社 址 北京市西城区德外大街 4 号  
邮政编码 100011  
总 机 010-58581000

经 销 北京蓝色畅想图书发行有限公司  
印 刷 河北省财政厅印刷厂

开 本 787×1092 1/16  
印 张 17.75  
字 数 430 000

购书热线 010-58581118  
免费咨询 800-810-0598  
网 址 <http://www.hep.edu.cn>  
<http://www.hep.com.cn>  
网上订购 <http://www.landaco.com>  
<http://www.landaco.com.cn>

版 次 2005 年 2 月第 1 版  
印 次 2005 年 2 月第 1 次印刷  
定 价 22.40 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 16414-00

# 出版说明

信息产业是国民经济和社会发展基础性、战略性产业。加快发展信息技术和信息产业,以信息化带动工业化,以信息化促进工业化,是当前和今后我国产业结构调整发展的战略重点。软件产业是信息产业的核心,加快软件人才培养是加快软件产业发展的先决条件。为适应经济结构战略性调整及软件产业发展的需要,加快培养各类软件应用性人才,在国家改革和发展委员会、教育部的指导和支持下,福建省从2002年开始,在全国率先举办软件类高等职业技术教育,拟以办学模式和人才培养模式改革为重点,积极探索有水平、有质量、有特色的软件高职教育发展的新路子。

在软件类高等职业技术教育改革和建设过程中,福建省坚持教育创新,把改革教学内容和课程体系,加强专业建设、教材建设和教学队伍建设作为工作的重点。目前,根据软件行业发展趋势、就业环境和软件高等职业技术教育的办学特点,经组织专家论证和审定,福建省高校首批开设了可视化编程、Web应用程序设计、软件测试、网络系统管理员、网络构建技术、数据库管理员、图形图像制作、多媒体制作、计算机办公应用等9个软件高职专业,制订了较为科学合理的人才培养方案。为配合支持软件类高职教育的改革和建设,福建省教育厅聘请软件教育有关专家、学者和著名软件企业的高级工程技术人员成立了“职业技术教育软件人才培养模式改革项目成果教材编审委员会”,以“抓好试点规划,实施精品战略”为指导方针,认真吸取国内外软件技术发展成果,根据软件企业对人才培养提出的新要求和软件高职的办学特点,认真处理好教材的统一性与多样化、基本教材与辅助教材、学历教育教材与认证培训教材的关系,以组织开展软件高职公共基础课、专业基础课和专业主干课教材的建设为重点,同时扩大品种,实现教材系列配套,在此基础上形成特色鲜明、优化配套的软件高等职业技术教育教材体系。

本软件系列教材适合于本科院校、高职高专院校、成人高校及继续教育学院的软件高职类专业及相关专业使用。

职业技术教育软件人才培养模式改革项目成果教材编审委员会

2003年5月

# 前 言

可扩展标记语言 XML(eXtensible Markup Language)是环球网络委员会 W3C(World Wide Web Consortium)的一个工作组特别设计的、用于网上传送信息的新标准,它使用一个简单又灵活的标准格式,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。XML 的诸多优点及先进性备受网络开发者的青睐,Microsoft、IBM、ORACLE、Sun 等公司都积极支持并参与 XML 的研究和产品化工作,先后推出了支持 XML 的产品或者改造原有的产品以支持 XML。XML 技术的发展如火如荼。

根据软件高职高专培养计算机应用高等技术人才的需要,我们编写了本书,目的是通过回答初学者关于 XML 的基本问题:“XML 是什么”,“为什么需要使用 XML”以及“如何使用 XML”这几个问题来介绍目前最重要和最实用的 XML 技术。在编写过程中,我们一方面注意跟踪 XML 技术的发展,保证教材内容的新颖性;另一方面注重实用性,以一个计算机公司的应用案例(包括公司产品目录、产品信息发布以及业务数据交换等)贯穿全书,循序渐进地介绍 XML 的语法、规范、功能及应用。

本书分 7 章详细介绍 XML 标准的语法和应用。第 1 章简要介绍 XML 的历史起源,XML 的编辑浏览工具以及 XML 的应用,并将 XML 与 HTML 进行比较,让初学者了解 XML 是什么以及为什么需要使用 XML;第 2 章、第 3 章和第 4 章分别介绍 XML 的 3 个基本要素,即 XML 语法、XML 模型和样式语言,并结合大量的例题向读者详细讲述如何使用 XML。第 5 章和第 6 章分别介绍与 XML 相关的 XML 链接、XML 解析器和应用程序接口等更深入的内容,为读者今后进行更高层次的 XML 应用开发打下基础。最后,在第 7 章讨论几个 XML 应用案例,让读者进一步了解如何在实际应用中应用 XML。由于 XML 相关的技术和标准在不断发展、不断更新中,因此在一些章节中我们还提供一些线索以帮助读者获得更多信息。

学时分配大致是这样的:第 1 章 6 学时,第 2 章 6 学时,第 3 章 18 学时,第 4 章 12 学时,第 5 章 6 学时,第 6 章 8 学时,第 7 章 8 学时,共 64 学时左右。第 6 章涉及 ASP、Java 和 C++ 编程,可根据学生的实际情况来选择。

本书主要由吴扬扬执笔,颜毛智负责书中大部分例题的设计、编写和调试。本书由傅清祥教授审校。由于时间仓促,而作者水平有限,难免存在缺点和错误,恳请广大读者来信(E-mail: wuyy@hqu.edu.cn)批评指正。

作者

2004 年 10 月

# 目 录

|                           |    |                                 |     |
|---------------------------|----|---------------------------------|-----|
| <b>第 1 章 XML 概述</b> ..... | 1  | 2.9 本章小结                        | 58  |
| 1.1 标记语言简介                | 1  | 习题                              | 59  |
| 1.2 了解 XML                | 5  | <b>第 3 章 XML 模型</b> .....       | 60  |
| 1.3 需要 XML 的原因            | 8  | 3.1 定义 XML 文档结构的意义              | 60  |
| 1.3.1 HTML 的缺点            | 9  | 3.2 DTD 语法                      | 61  |
| 1.3.2 XML 的设计目标           | 11 | 3.2.1 DTD 声明                    | 61  |
| 1.3.3 XML 与 HTML 的比较      | 12 | 3.2.2 根据 DTD 对 XML 文档进行验证       | 64  |
| 1.4 XML 的应用简介             | 13 | 3.2.3 定义元素                      | 66  |
| 1.5 XML 的编辑浏览工具简介         | 15 | 3.2.4 定义属性                      | 73  |
| 1.5.1 XML 编辑器             | 15 | 3.2.5 参数实体                      | 84  |
| 1.5.2 XML 解析器             | 25 | 3.2.6 命名空间                      | 84  |
| 1.5.3 XML 浏览器             | 26 | 3.3 W3C XML Schema 语法           | 88  |
| 1.6 相关标准                  | 30 | 3.3.1 Schema 文档的结构              | 89  |
| 1.7 本章小结                  | 32 | 3.3.2 在 XML Schema 定义中使用命名空间    | 90  |
| 习题                        | 32 | 3.3.3 在 XML 文档中引用 XML Schema 定义 | 91  |
| <b>第 2 章 XML 语法</b> ..... | 33 | 3.3.4 定义元素                      | 93  |
| 2.1 XML 文档结构              | 33 | 3.3.5 定义元素属性                    | 107 |
| 2.1.1 文档的序言               | 34 | 3.4 XML 建模                      | 112 |
| 2.1.2 文档的主体               | 35 | 3.4.1 信息建模                      | 113 |
| 2.1.3 文档的尾声               | 36 | 3.4.2 设计 XML 文档                 | 116 |
| 2.2 元素和标记                 | 36 | 3.5 本章小结                        | 123 |
| 2.2.1 元素的基本形式             | 36 | 习题                              | 123 |
| 2.2.2 标记                  | 37 | <b>第 4 章 样式语言</b> .....         | 125 |
| 2.2.3 元素内容                | 38 | 4.1 需要样式单的原因                    | 125 |
| 2.3 属性                    | 41 | 4.2 层叠样式单 CSS                   | 126 |
| 2.4 实体                    | 43 | 4.2.1 在 XML 文档中引用 CSS 样式单       | 127 |
| 2.4.1 实体的两种类型             | 44 | 4.2.2 CSS 的样式规则                 | 128 |
| 2.4.2 实体的使用方法             | 45 | 4.2.3 添加注解                      | 142 |
| 2.5 注释                    | 50 |                                 |     |
| 2.6 CDATA 段               | 55 |                                 |     |
| 2.7 处理指令                  | 56 |                                 |     |
| 2.8 实例分析                  | 56 |                                 |     |

|                                |     |                               |     |
|--------------------------------|-----|-------------------------------|-----|
| 4.2.4 继承性 .....                | 142 | 6.2.1 DOM 的定义 .....           | 202 |
| 4.3 可扩展样式单语言 XSL .....         | 143 | 6.2.2 DOM 解析器 .....           | 203 |
| 4.3.1 XSLT 转换的概述 .....         | 144 | 6.2.3 使用 DOM 接口访问文档 .....     | 204 |
| 4.3.2 XSL 文档格式 .....           | 146 | 6.3 SAX .....                 | 215 |
| 4.3.3 XSLT 基本元素 .....          | 147 | 6.3.1 SAX 的定义 .....           | 215 |
| 4.3.4 匹配模式 .....               | 160 | 6.3.2 常用的 SAX 接口和类 .....      | 217 |
| 4.3.5 格式化对象简介 .....            | 166 | 6.3.3 创建 SAX 应用程序 .....       | 220 |
| 4.4 本章小结 .....                 | 172 | 6.4. 本章小结 .....               | 237 |
| 习题 .....                       | 173 | 习题 .....                      | 238 |
| 第 5 章 XML 链接 .....             | 175 | 第 7 章 应用案例介绍 .....            | 239 |
| 5.1 XLink .....                | 175 | 7.1 数学标记语言 MathML .....       | 239 |
| 5.1.1 XLink 和 HTML 链接的比较 ..... | 176 | 7.1.1 让计算机理解数学 .....          | 240 |
| 5.1.2 链接元素 .....               | 177 | 7.1.2 MathML 基本组成 .....       | 240 |
| 5.1.3 简单链接 .....               | 180 | 7.1.3 编码风格 .....              | 242 |
| 5.1.4 扩展链接 .....               | 183 | 7.2 Web 分布式数据交换 WDDX .....    | 250 |
| 5.1.5 链接库 .....                | 187 | 7.2.1 数据传输与集成的业务问题 .....      | 250 |
| 5.2 XPointer .....             | 188 | 7.2.2 WDDX 解决方案 .....         | 250 |
| 5.2.1 使用 XPointer 的原因 .....    | 188 | 7.3 构建在线拍卖网站 .....            | 253 |
| 5.2.2 标识片断的三种形式 .....          | 188 | 7.3.1 三层 Web 体系结构 .....       | 253 |
| 5.2.3 扩展 XPath 表达式 .....       | 191 | 7.3.2 从中间层获取数据 .....          | 254 |
| 5.2.4 XPointer 转义 .....        | 196 | 7.3.3 创建用户页面 .....            | 257 |
| 5.3 XML Base .....             | 197 | 7.3.4 从客户端更新数据 .....          | 258 |
| 5.4 本章小结 .....                 | 197 | 7.4 推播频道 .....                | 259 |
| 习题 .....                       | 198 | 7.4.1 创建推播频道 .....            | 259 |
| 第 6 章 解析器和应用程序接口 .....         | 199 | 7.4.2 Microsoft 的频道定义格式 ..... | 261 |
| 6.1 解析器 .....                  | 199 | 7.4.3 RSS .....               | 267 |
| 6.1.1 了解 XML 解析器 .....         | 200 | 7.5 本章小结 .....                | 272 |
| 6.1.2 验证解析器和非验证解析器 .....       | 201 | 参考文献 .....                    | 273 |
| 6.1.3 解析器和应用程序 .....           | 201 | 参考网站 .....                    | 274 |
| 6.2 文档对象模型 DOM .....           | 202 |                               |     |

# 第 1 章 XML 概 述

---

## 本章学习目标

- 理解标记语言
  - 了解 XML 的产生和发展
  - 了解 XML 的用途
  - 掌握 XML 编辑器、解析器和浏览器的使用方法
  - 了解 XML 相关标准
- 

自从 1998 年 2 月成为推荐标准后,XML 一直受到广泛的关注。许多厂商包括 Microsoft、IBM、Oracle、Sun 等,都相继推出了支持 XML 的产品或改造了原有的产品以支持 XML。W3C 也一直在致力于完善 XML 的标准体系。近几年来,许许多多与 XML 相关的新名词、新技术如潮水般不断涌来。那么,XML 是什么?为什么它有这么大的冲击力?是什么原因使得这么多的厂商、这么多的应用开发人员对它充满热情?

可扩展标记语言 XML(Extensible Markup Language)是由 W3C 的 XML 工作组制订的。这个工作组是这样描述 XML 的:“可扩展标记语言 XML 是标准通用标记语言 SGML(Standard Generalized Markup Language)的一个子集。其目标是允许普通的 SGML 在 Web 上以目前超文本标记语言 HTML(HyperText Markup Language)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。”

通过上面的描述可知:XML 是一种通用的标记语言(Markup Language),它是特别设计的一组定义语义标记的规则。根据这些规则,开发人员可以创建各种标记来标注文档的结构和语义,以便于在 Web 上传送信息和交换数据。遵循 XML 语法规则,在文本中添加标记,这些标记就构成了一个 XML 文档。XML 有两个先驱——SGML 和 HTML。SGML 和 HTML 这两个标记语言都非常成功,那么,XML 只是 SGML 的子集,为何不直接用 SGML 来描述 Web 网页呢?而 HTML 以其简单精练的语法、极易掌握的通用性与易学性,已为大众所接受,成为用来建构网页的标准语言,为何还要设计 XML?下面的讨论将解答这些问题。

## 1.1 标记语言简介

标记语言起源于传统印刷。印刷之前必须排版。在电子出版业尚未出现之前,人们需要用手抄或者打字的方式先拷贝手稿。然后,再对副本以人工方式标记,并加上编辑说明,以告诉印

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、如何对齐等,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,加“格式描述码”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、Wordpad 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、Wordpad 等文本编辑器都支持的一种文件存储格式 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。

步骤 1:打开写字板 Wordpad,输入“Welcome to Innovation Computer Company!”,选择字体 Times New Roman 和字号 20,如图 1.1.1 所示。

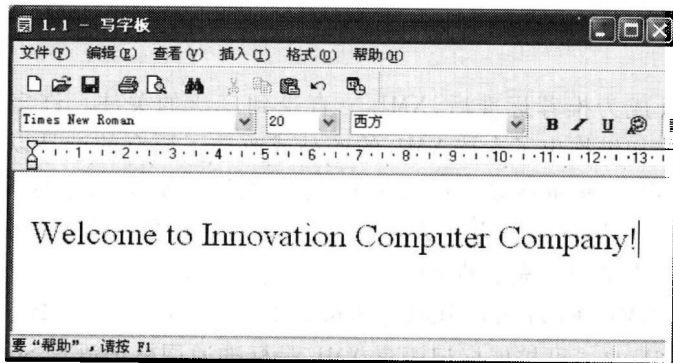


图 1.1.1 在写字板中输入一段文字

步骤 2:单击“保存”按钮,在“保存为”对话框中,选择保存在桌面“例题”文件夹,输入文件名“1.1”,选择保存类型“RTF”,如图 1.1.2 所示,然后点击“保存”按钮。

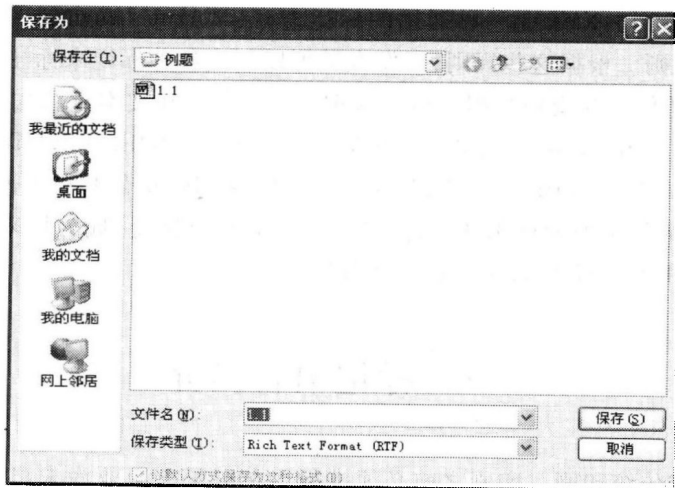


图 1.1.2 保存 RTF 文档

注意:保存类型必须选择“RTF”!

步骤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。

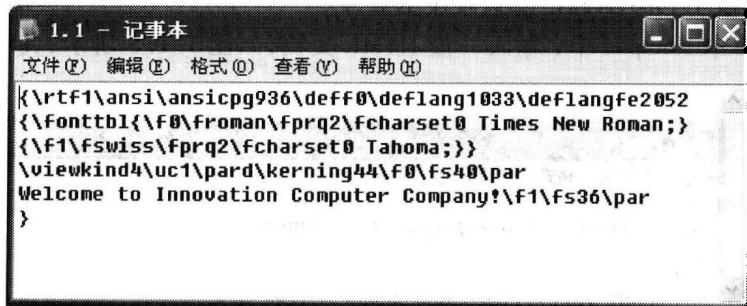


图 1.1.3 用记事本打开“1.1.RTF”文件

可以看到在图 1.1.3 中显示的内容除了刚才输入的“Welcome to Innovation Computer Company!”外,还有其他代码,这些代码就是 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。

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 1.1.4 给出了用 Word 打开“1.1.RTF”文件的显示结果,与图 1.1.1 中 WordPad 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“Welcome to Innovation Computer Company!”。

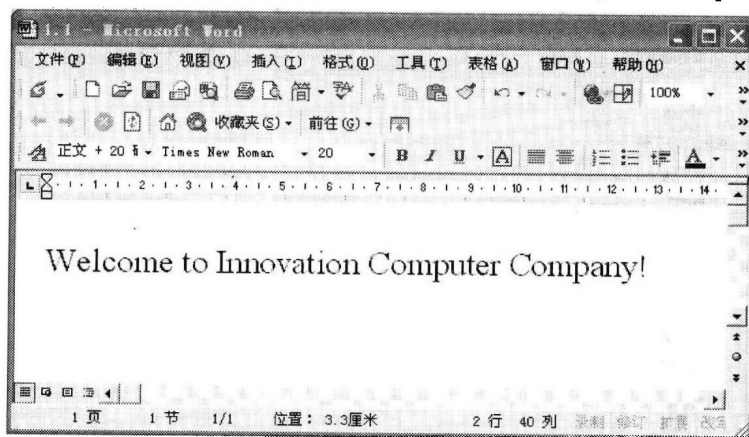


图 1.1.4 在 Microsoft Word 中打开“1.1.RTF”文件

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 Notepad 等文本编辑工具来手工编写,也可以用 DreamWeaver 和 FrontPage 等可视化的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。

步骤 1:在记事本 Notepad 中,输入一段 HTML 文档,如图 1.1.5 所示,其中,用尖括号括起来的是标记。

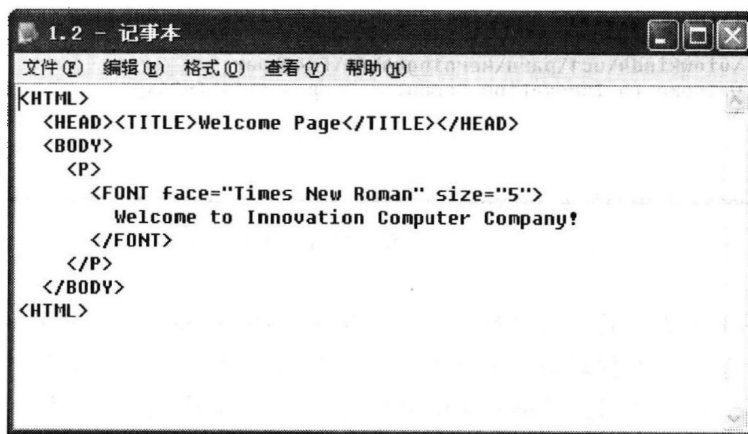


图 1.1.5 用记事本编辑 HTML 文档

步骤 2:选择“文件”菜单中的“保存”。在“另存为”对话框中,选择保存在 C 盘的“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 1.1.6 所示。

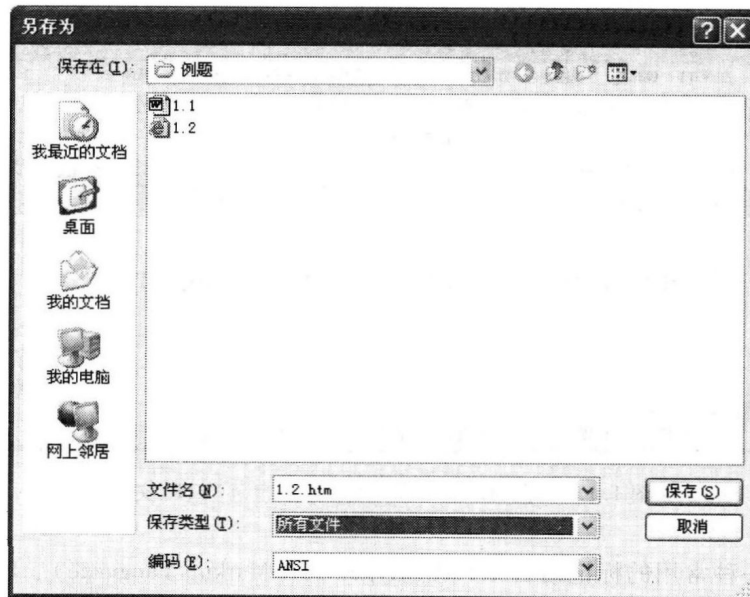


图 1.1.6 保存 HTML 文档

注意:文件名必须包含扩展名.htm,并且保存类型必须选择“所有文件”!

步骤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1.1.7所示。

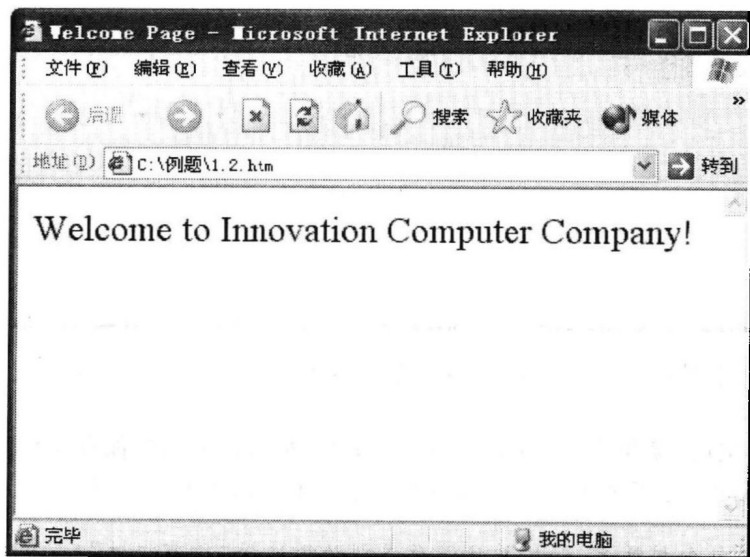


图 1.1.7 在 IE 中显示“1.2.htm”文件

IE 浏览器用字体 Times New Roman 显示“Welcome to Innovation Computer Company!”。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“Welcome to Innovation Computer Company!”。

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 和 RTF、HTML 有何不同呢?下面的内容将回答这个问题。

## 1.2 了 解 XML

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 描述任意类型的文档。例如,假设要描述一个计算机公司的主要业务、地址、邮编和联系电话等,这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,在本书的第3章中将会详细介绍 DTD。

**【例 1.3】** 创建一个计算机公司简介的 XML 文档。

步骤1:在记事本中,输入创新计算机公司简介的 XML 文档,如图 1.2.1 所示。

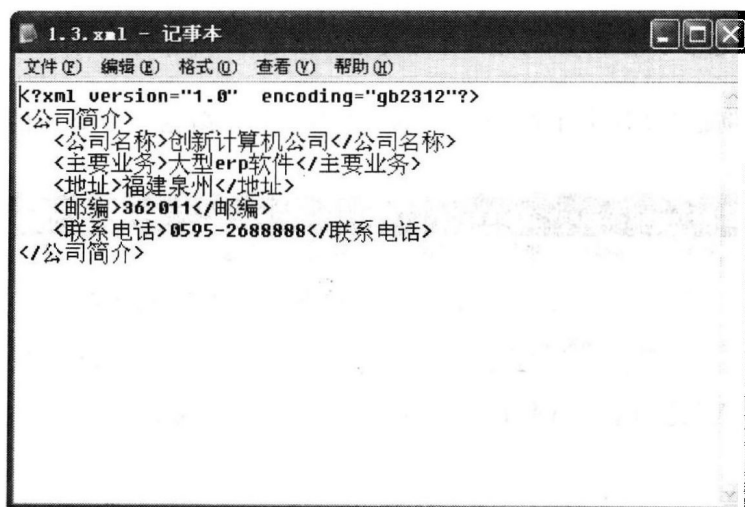


图 1.2.1 在记事本中输入创新计算机公司简介的 XML 文档

步骤 2: 选择“文件”菜单中的“保存”, 在“另存为”对话框中选择保存在桌面“例题”文件夹, 输入文件名“1.3.xml”, 然后选择保存类型为“所有文件”, 如图 1.2.2 所示。

**注意:** 文件名必须包含扩展名.xml, 并且保存类型必须选择“所有文件”!



图 1.2.2 保存“1.3.xml”文件

双击“例题”文件夹中的“1.3.xml”文件, 在 IE 浏览器中的显示结果如图 1.2.3 所示。IE 5.0 浏览器以上的版本均支持 XML, 能解读 XML 文档。

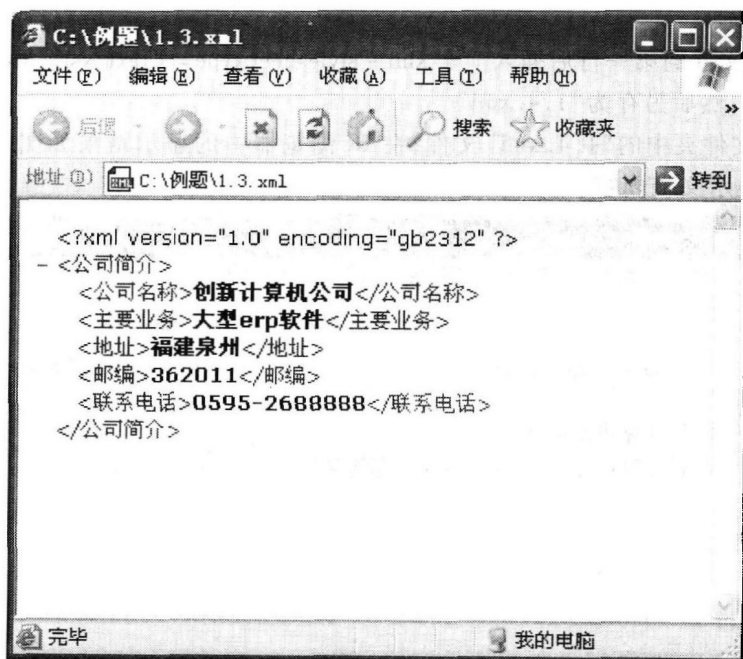


图 1.2.3 在 IE 中显示“1.3.xml”文件

XML 文档看上去和 HTML 非常相像,但这里的标记表示的不再是显示格式,而是指明了数据语意,它们是由用户根据 XML 所提供的规则自行定义的。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,通过指定和该 XML 文档相连的样式单文件,为 XML 文档定义显示格式。

**【例 1.4】** 为公司简介 XML 文档指定样式单文件“1.4style.xml”,样式单文件内容如下:

```
<? xml version = "1.0" encoding = "GB2312" ?>
<xsl:stylesheet version = "1.0" xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
  <xsl:template match = "/">
    <html>
      <body>
        <table border = "1" >
          <tr>
            <td>创新计算机公司简介:</td>
          </tr>
          <tr>
            <td> <xsl:value-of select = "公司简介" /></td>
          </tr>
        </table>
      </body>
    </html>
  </xsl:template>
```

</xsl:stylesheet>

在“1.3.xml”文件的第一行后插入“<? xml - stylesheet type = "text/xsl" href = "1.4style.xsl" ?>”，然后另存为“1.4.xml”。

双击“例题”文件夹中的“1.4.xml”文件，在 IE 浏览器中的显示结果如图 1.2.4 所示，按样式单指定的格式显示文档内容。

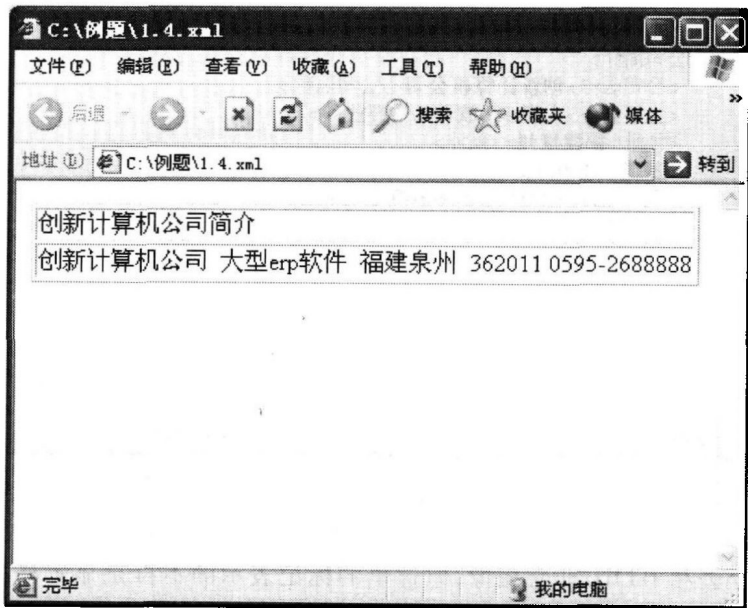


图 1.2.4 按样式单指定的格式显示

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。

XML 并不是一种新的语言，它只是 SGML 的一个子集。事实上，XML 能做的，SGML 都能做到，甚至 SGML 能做得更多。SGML 确定了一套严格一致的、独立于平台的表达信息的格式，这种格式在 1986 被国际标准化组织 (International Organization for Standardization, ISO) 所采纳，成为一个标准。它是很多标记语言的母语言，HTML 和 XML 都派生自 SGML (尽管派生的方法不同)。

### 1.3 需要 XML 的原因

SGML 虽然功能强大，但太庞大、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的含义也被削弱了。现在，HTML 实际上已演变为专门用于 Netscape 和 Microsoft IE 两大浏览器的页面显示语言。HTML 确实解决了浏览页面最基本的问题——显示问题，为互

联网的蓬勃发展立下了赫赫战功。但由于自身的缺陷,使它难以满足网络进一步发展的需要。

### 1.3.1 HTML 的缺点

HTML 的缺点主要有:

① HTML 的 tag 集合是固定的,它定义了一套固定的标记,用来描述一定数目的元素,用户不能自行新增有意义的、能供他人使用的 tag。如果标记语言中没有所需的标记,用户就没有办法了,这时只好等待标记语言的下一个版本,希望在新版本中能够包括所需的标记,但是这样一来就得依赖于软件开发商的选择了。

② HTML 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,它并不能揭示 HTML 文档中元素内容的含义。尽管 HTML 在网上信息发布方面发挥了很大作用,但却不利于计算机自动识别,不利于计算机之间的相互交流和传递信息。

下面,以 1.2 小节中例 1.3 的计算机公司简介的网页内容为例,分析一段相应的 HTML 代码。

**【例 1.5】** 一个计算机公司简介的 HTML 文档。

创新计算机公司简介的 HTML 代码如下:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312">
    <title>创新计算机公司简介</title>
  </head>
  <body>
    <h2>创新计算机公司</h2>
    <table border="1">
      <tr>
        <td>主要业务</td>
        <td>大型 erp 软件</td>
      </tr>
      <tr>
        <td>地址</td>
        <td>福建泉州</td>
      </tr>
      <tr>
        <td>邮编</td>
        <td>362011</td>
      </tr>
      <tr>
        <td>联系电话</td>
        <td>0595-2688888</td>
      </tr>
    </table>
  </body>
</html>
```

```
</tr>
</table>
</body>
</html>
```

上述 HTML 文档在 IE 浏览器中的显示结果如图 1.3.1 所示。



图 1.3.1 计算机公司简介 HTML 文档的显示结果

在创新计算机公司简介的 HTML 文档中,“<h2>创新计算机公司</h2>”中的标记“h2”指定了在浏览器中“创新计算机公司”有特定的表现形式,但是 HTML 文档并没有指出它到底是什么。当然,阅读是没什么问题的,因为人能理解其中的含义。但问题在于计算机无法真正理解文本的含义,也就不能识别它。解决办法是让标记带有语意信息,帮助计算机尽可能多地了解文档。

③ 缺乏严格的结构,HTML 中的标签可以不配对出现。例如:去掉例 1.5 中创新计算机公司简介的 HTML 文档中的结束标记“</h2>”,浏览器照样能显示它。这对浏览器的开发者是件头痛的事,他们不得不把大量的精力耗费在文法错误的包容上,相应地,浏览器的程序也要加大,甚至牺牲浏览器的时间效率和空间效率。

然而,电子商务、电子图书、远程教育等全新领域如异军突起,迅猛发展并逐渐成为互联网世界必不可少且愈发重要的组成部分,随之而来的是 Web 文件的复杂化、多样化、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、表达形式和再现出来。这就要求允许 Web 文件具有高度的复杂性以及表达的高度灵活性,以应付如此复杂多变的具体应用。

1996 年人们开始致力于引进一个新的标记语言,旨在将 Web 从发布媒体转变成应用处理媒体。这个新的标记语言应既具有 SGML 的强大功能和可扩展性,同时又具有 HTML 的简单性。国际互联网论坛 W3C 专门成立一个 SGML 专家小组来从事此项工作,由 Sun 公司的 Jon Bosak