



新手入门



逐步进阶



实战提高



图解教学



范例练习



视频光盘

XML应用 入门与提高

祝红涛 陈军红 编著

影响百万人的经典清华版
全新改版震撼上市



光盘超值赠送
全部案例的源文件
多媒体视频演示

清华大学出版社



软件入门与提高丛书

XML 应用入门与提高

祝红涛 陈军红 编著



清华大学出版社
北京

内 容 简 介

本书从初学者的角度出发,由浅入深、循序渐进地介绍了 XML 文档应用与开发的相关知识,书中不但提供了大量 XML 的示例,还提供了大量的实战和上机练习用于演练。

本书共分 13 章,主要内容包括标记语言的类型和历史,XML 的概念、特点和用途,XML 与其他语言的区别,XML 的相关标准技术,XML 的使用优势、应用领域和开发工具,XML 的构成元素、文档声明、元素、标记、属性、字符、实体和命名空间,DTD、XSD、XPath、XQuery、XSLT、XLink、XPather、Web 服务和 WSDL,还包括 DOM 解析器和 SAX 解析器对 XML 文档的解析,以及在 C#语言中处理 XML 文档数据等。在本书最后一章提供了一个综合项目——XML 通信录作为结束。

本书涉及面广泛,几乎涉及 XML 应用与开发的所有重要知识,适合所有 XML 初学者使用。另外,对于大中专和培训班的学生来说,本书更是一本不可多得的教材。

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 应用入门与提高/祝红涛,陈军红编著.——北京:清华大学出版社,2015

(软件入门与提高丛书)

ISBN 978-7-302-37541-8

I. ①X… II. ①祝… ②陈… 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(2014)第 174630 号

责任编辑:杨作梅 李玉萍

装帧设计:刘孝琼

责任校对:王 晖

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>, 010-62791865

印 刷 者:清华大学印刷厂

装 订 者:三河市吉祥印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:26.25 字 数:626 千字

(附 DVD1 张)

版 次:2015 年 1 月第 1 版

印 次:2015 年 1 月第 1 次印刷

印 数:1~3000

定 价:58.00 元

产品编号:057494-01

前 言

XML 中文被称为可扩展标记语言, 是 SGML(标准通用标记语言)的子集。使用 XML 可以用来标记数据、定义数据类型, 它是一种允许用户对自己的标记语言进行定义的源语言。XML 非常适合万维网传输, 它提供了统一的方法来描述和交换独立于应用程序或供应商的结构化数据。

使用 XML 不仅可以从 HTML 中分离数据, 还可以交换数据和共享数据。另外, 许多新的 Internet 语言都是通过 XML 进行创建的, 本书将详细介绍与 XML 相关的知识, 包括与 XML 有关的技术。

本书内容

全书共分 13 章, 主要内容如下。

第 1 章 XML 概述。本章着重介绍 XML 的基本知识, 包括它的概念、特点、用途、与其他语言的区别、相关技术以及使用优势和应用领域等内容。

第 2 章 XML 入门。本章介绍 XML 基本语法和文档组成, 包括标记、属性和元素的使用, 字符的使用, 注释的使用, 以及实体引用和命名空间等内容。

第 3 章 XML 与 DTD。本章首先介绍 DTD 的概念和构成, 接着介绍 DTD 文件中的元素和属性声明, 然后介绍如何通过不同的方式引用 DTD 文件, 最后再了解实体声明和引用的相关知识。

第 4 章 XML 与 XSD。XSD 的英文全称是 XML Schema Definition, 有时也被称为 XML Schema。本章详细介绍 XSD 技术, 包括它的优点、文档制定、构成元素、数据类型以及校验工具等内容。

第 5 章 XPath 和 XQuery 技术。本章从 XSL 常用标记开始介绍, 然后依次介绍 XPath 技术和 XQuery 技术。

第 6 章 文档显示技术。本章着重介绍 XSLT 的知识, 其内容包括 XSLT 模板、XSLT 标记、基本判断、条件判断、循环处理、XSLT 函数, 以及在 XSLT 中如何使用 CSS 等。

第 7 章 XLink 和 XPointer 技术。本章首先介绍 XLink 技术, 包括它的语法、属性、简单链接和扩展链接, 然后介绍 XPointer 技术, 包括它的概念、形式、位置路径、位置集合、位置步以及对 XPath 的扩展内容等。

第 8 章 Web 服务和 WSDL 技术。本章着重介绍 WSDL 的知识, 在介绍 WSDL 技术之前, 首先会了解 Web 服务。

第 9 章 DOM 解析器操作 XML 文档。本章重点介绍如何使用 DOM 解析器对 XML 文档进行解析。其主要内容包括 XML DOM 的作用、XML DOM 与 W3C 的关系、常用的操作对象、DOM 节点类型以及节点操作等。



第 10 章 SAX 解析 XML 文档。本章将详细介绍如何使用 SAX 解析器来处理 XML 文档。其主要内容包括 SAX 的历史版本、工作原理、监听器接口、解析 API、解析步骤以及接口应用等。

第 11 章 C# 中的 XML 应用。本章介绍如何使用 C# 语言处理 XML 文档，包括 XML 与 DataSet 的交互、System.xml 命名空间和处理 XML 数据常用类等内容。

第 12 章 XML 中的 Web 应用。本章介绍 XML 在 Web 中的应用，使用 ASP.NET 环境和 SQL Server 数据库，包括 XML 数据绑定控件的使用、XML 数据与 SQL Server 之间的转换、站点地图中的 XML 以及 ASP.NET 配置文件的介绍等。

第 13 章 通讯录。本章主要介绍使用 ASP 结合 XML 实现通讯录的过程，包括定义 XML 文件、添加联系人、查看联系人、修改和删除联系人。

本书特色

本书内容详细、实例丰富，知识面广，全面地讲解了 XML 的应用和开发。与已经出版的相关图书相比，这本图书的最大特点体现在以下几个方面。

- 知识全面，内容丰富。本书紧紧围绕 XML 的相关知识展开详细的讲解，涵盖了实际开发中的具体应用代码。
- 理论和示例结合。本书中几乎每一个知识点都有丰富而典型的练习，而且每一章最后都会通过一个或多个综合实战总结本章的知识。作为一本 XML 入门书，理论与练习很好地结合起来进行讲解，最容易让读者快速掌握。
- 应用广泛，提供文档。对于大多数的精选实战案例，都会向读者提供详细的实现步骤，结构清晰简明，分析深入浅出，并且有些实战很贴近实际。
- 随书光盘。本书配备了视频教学文件，包括每个章节所涉及的源代码、开发环境的安装演示等。读者可以通过视频文件更加直观地学习 XML 的知识。
- 网站技术支持。读者在学习或者工作的过程中，如果遇到问题，可以直接登录 www.itzcn.com 与我们取得联系，作者会在第一时间内给予回复。
- 贴心的提示。为了便于读者阅读，全书还穿插了一些技巧、提示等小贴士，体例约定如下。
 - ◆ 提示：通常是一些贴心的提醒，让读者加深印象或提供建议，或者是解决问题的方法。
 - ◆ 注意：提出学习过程中需要特别注意的一些知识点和内容，或者相关信息。
 - ◆ 技巧：通过简短的文字，指出知识点在应用时的一些小窍门。

读者对象

本书适合作为软件开发入门者的自学用书，也适合作为高等院校相关专业的教学参考书，还可供开发人员查阅、参考。

- XML 开发入门者；
- XML 初学者以及在校学生；

- 各大中专院校的在校学生和相关授课老师;
- 准备从事与 XML 应用相关的人员。

本书由祝红涛和陈军红编著,参与本书编写的人员还有程朝斌、王咏梅、郝军启、王慧、郑小营、张浩华、王超英、张凡、赵振方、张艳梅等,在此特表示感谢。在本书的编写过程中,我们力求精益求精,但书中难免会存在一些疏漏或不足之处,敬请广大读者批评、指正。

编 者

目 录

第 1 章 XML 概述	1	2.1.1 XML 文档构成	24
1.1 标记语言概述	2	2.1.2 XML 文档声明	25
1.1.1 标记语言的类型	2	2.1.3 XML 文档注释	26
1.1.2 标记语言的历史	3	2.1.4 XML 处理指令	27
1.2 XML 的基础知识	4	2.2 XML 元素和标记	28
1.2.1 XML 的概念	4	2.2.1 元素和标记	28
1.2.2 XML 的特点	5	2.2.2 元素定义	30
1.2.3 XML 的用途	6	2.2.3 元素特性	31
1.3 XML 与 HTML 语言	7	2.3 XML 元素属性	32
1.3.1 语法要求不同	7	2.3.1 属性的使用	32
1.3.2 标记不同	7	2.3.2 元素和属性	33
1.3.3 作用不同	8	2.4 XML 中的字符和实体	34
1.3.4 其他不同	8	2.4.1 实体概述	35
1.4 XML 的相关技术	9	2.4.2 CDATA 文本段	36
1.4.1 W3C 组织	9	2.4.3 文档规范	37
1.4.2 技术标准	9	2.5 XML 命名空间	38
1.5 XML 的优势	12	2.5.1 命名空间概述	38
1.5.1 技术优势	12	2.5.2 自定义命名空间	38
1.5.2 商业优势	14	2.5.3 命名空间的特点	41
1.6 XML 的应用领域	15	2.5.4 默认的命名空间	42
1.6.1 电子商务	15	2.6 实战——系统用户信息管理	42
1.6.2 网络出版	16	2.7 本章习题	44
1.6.3 移动通信	16	第 3 章 XML 与 DTD	47
1.6.4 Web 服务	16	3.1 DTD 简介	48
1.6.5 Web 集成	17	3.1.1 DTD 概述	48
1.6.6 配置文件	17	3.1.2 DTD 的构成	48
1.6.7 分布式计算	17	3.1.3 DTD 文档定义	49
1.6.8 前景观望	18	3.2 元素声明	51
1.7 XML 开发工具	18	3.2.1 元素声明简介	51
1.8 实战——XML 文档引入 CSS 文件	19	3.2.2 元素声明分类	52
1.9 思考与练习	21	3.3 属性	55
第 2 章 XML 入门	23	3.3.1 属性声明	55
2.1 XML 文档	24	3.3.2 属性类型	56



3.3.3 属性附加声明.....	61	4.5 实战——通过代码验证 XML 文档.....	110
3.4 实战——水果信息 DTD.....	63	4.6 思考与练习.....	114
3.5 DTD 引用.....	64	第 5 章 XPath 和 XQuery 语言.....	117
3.5.1 引用内部 DTD.....	64	5.1 XSL 常用标记.....	118
3.5.2 引用外部 DTD.....	65	5.2 XPath 简介.....	120
3.5.3 混合引用 DTD.....	65	5.3 XPath 定位技术.....	121
3.6 实体声明与引用.....	66	5.3.1 路径表达式.....	122
3.6.1 实体的概念与分类.....	66	5.3.2 XPath 轴.....	126
3.6.2 内部普通实体.....	68	5.3.3 节点测试.....	129
3.6.3 外部普通实体.....	70	5.3.4 XPath 运算符.....	131
3.6.4 参数实体.....	73	5.4 XPath 函数.....	133
3.7 实战——衣帽配饰信息管理.....	74	5.4.1 String 类函数.....	133
3.8 本章习题.....	76	5.4.2 Number 类函数.....	134
第 4 章 XML 与 XSD.....	79	5.4.3 Boolean 类函数.....	135
4.1 了解 XSD 技术.....	80	5.4.4 Node-Set 类函数.....	137
4.1.1 XSD 的概念.....	80	5.5 XQuery 技术.....	138
4.1.2 XSD 的好处.....	80	5.5.1 XQuery 简介.....	138
4.1.3 XSD 与 DTD.....	82	5.5.2 XQuery 表达式基础.....	139
4.1.4 制定 XSD 文档.....	82	5.5.3 FLWOR 表达式.....	141
4.2 XSD 的构成元素.....	83	5.5.4 XQuery 函数.....	142
4.2.1 XSD 的元素.....	83	5.6 实战——数码信息显示.....	143
4.2.2 创建根元素.....	84	5.7 本章习题.....	145
4.2.3 创建简单元素.....	86	第 6 章 文档显示技术.....	149
4.2.4 简单类型元素.....	87	6.1 XSLT 简介.....	150
4.2.5 限定数据约束.....	89	6.1.1 XSL 与 XSLT.....	150
4.2.6 创建属性元素.....	92	6.1.2 XSLT 转换技术.....	151
4.2.7 创建复合元素.....	93	6.1.3 CSS 样式技术.....	155
4.2.8 指示器.....	97	6.2 XSLT 模板.....	156
4.2.9 any 和 anyAttribute 元素.....	100	6.2.1 模板概述.....	157
4.3 XSD 的数据类型.....	101	6.2.2 模板匹配.....	158
4.3.1 字符串数据类型.....	101	6.2.3 子模板调用.....	160
4.3.2 日期和时间数据类型.....	102	6.3 XSLT 基础语法.....	163
4.3.3 数值数据类型.....	104	6.3.1 XSLT 标记.....	163
4.3.4 其他数据类型.....	106	6.3.2 简单判断.....	164
4.4 XSD 的常用验证工具.....	106	6.3.3 多条件判断.....	169
4.4.1 XMLWriter 工具.....	106	6.3.4 循环处理.....	171
4.4.2 XMLSpy 工具.....	109		

6.3.5 输出内容的排序.....	171	8.1.1 Web 服务介绍	214
6.4 XSLT 函数.....	174	8.1.2 Web 服务的技术支持	215
6.4.1 函数.....	174	8.1.3 查看和创建 Web 服务	216
6.4.2 Current()函数.....	174	8.2 认识 WSDL 技术.....	219
6.4.3 document()函数	175	8.2.1 WSDL 概述.....	219
6.4.4 format-number()函数	175	8.2.2 WSDL 的基本结构.....	220
6.4.5 NaN 值	176	8.2.3 创建 definitions 元素.....	222
6.5 在 XSLT 中使用 CSS.....	176	8.2.4 创建 types 元素.....	223
6.6 实战——网购信息显示.....	178	8.2.5 创建 message 元素	224
6.7 本章习题.....	180	8.2.6 创建 portType 元素	225
第 7 章 XLink 和 XPointer 技术	183	8.2.7 创建 binding 元素.....	226
7.1 XLink 基础知识	184	8.2.8 创建 service 元素.....	227
7.1.1 XLink 介绍	184	8.3 WSDL 的其他知识.....	228
7.1.2 XLink 语法	184	8.3.1 WSDL 端口.....	228
7.1.3 XLink 属性	185	8.3.2 WSDL 与 UDDI.....	229
7.2 XML 的链接类型.....	186	8.3.3 WSDL 的工作方式.....	231
7.2.1 XML 链接介绍.....	187	8.4 实战——不同项目之间调用 Web 服务	232
7.2.2 简单链接.....	187	8.5 思考与练习	235
7.2.3 扩展链接.....	190	第 9 章 DOM 解析器操作 XML 文档	237
7.2.4 扩展链接的类型.....	191	9.1 常用的 4 种 XML 解析器	238
7.2.5 扩展链接示例.....	195	9.1.1 DOM 解析器.....	238
7.2.6 外联链接.....	197	9.1.2 SAX 解析器.....	238
7.3 XPointer 基础知识	199	9.1.3 JDOM 解析器	238
7.3.1 XPointer 介绍	199	9.1.4 DOM4J 解析器	239
7.3.2 XPointer 形式	199	9.1.5 比较解析器	239
7.3.3 位置路径和集合	201	9.2 关于文档对象模型	240
7.3.4 位置步	201	9.2.1 XML DOM 的作用.....	240
7.3.5 XPointer 示例	204	9.2.2 XML DOM 与 W3C	241
7.4 XPointer 对 XPath 的扩展	205	9.2.3 XML DOM 的文档结构.....	241
7.4.1 新增函数.....	205	9.3 常用的 XML DOM 的操作对象.....	243
7.4.2 点的扩展.....	206	9.3.1 XML DOM 的加载和解析.....	244
7.4.3 区域扩展.....	206	9.3.2 常用的 3 种操作对象	246
7.5 实战——简单链接实现的图像 嵌入链接.....	207	9.3.3 了解 Node 对象	246
7.6 思考与练习	210	9.3.4 了解 Document 对象	248
第 8 章 Web 服务和 WSDL 技术	213	9.3.5 了解 Element 对象.....	250
8.1 了解 Web 服务	214	9.3.6 获取 Attr 对象.....	252
		9.4 DOM 节点类型.....	254



9.5 操作 XML 文档	257	10.9 思考与练习	307
9.5.1 动态创建 XML 文档	257	第 11 章 C# 中的 XML 应用	311
9.5.2 添加和删除元素节点	261	11.1 C# 中的 XML 应用	312
9.5.3 添加和删除属性节点	263	11.2 XML 与 DataSet 的交互	312
9.5.4 添加和修改文本节点	266	11.2.1 DataSet 对象简介	312
9.5.5 替换和复制节点	267	11.2.2 DataTable 对象简介	316
9.6 实战——读取 XML 文档实现级联 菜单	269	11.2.3 ReadXml() 读取 XML	318
9.7 思考与练习	272	11.2.4 数据筛选	320
第 10 章 SAX 解析 XML 文档	275	11.2.5 WriteXml() 写入文件	322
10.1 了解 SAX 解析器	276	11.3 System.Xml 命名空间	324
10.1.1 SAX 解析器概述	276	11.4 XML 数据类及节点类	325
10.1.2 SAX 的历史版本	277	11.4.1 XmlDocument 类	325
10.1.3 SAX 的工作原理	278	11.4.2 XmlNodeList 类	326
10.1.4 SAX 和 DOM 的区别	279	11.4.3 XmlNode 类	327
10.2 SAX 解析 XML 文档	280	11.4.4 实战——下拉框绑定 XML 节点	328
10.2.1 解析 XML 文档的 API	280	11.5 XML 读取类	329
10.2.2 SAX 的监听器接口	281	11.5.1 XmlReader 简介	330
10.2.3 SAX 解析 XML 文档	282	11.5.2 XmlReader 的属性和方法	331
10.3 ContentHandler 接口	285	11.5.3 XmlReader 的使用	333
10.3.1 了解 ContentHandler 接口	285	11.6 XML 写入类	334
10.3.2 处理文档的开始和结束	285	11.6.1 XmlWriter 简介	334
10.3.3 处理特殊指令	287	11.6.2 XmlWriter 应用	337
10.3.4 处理元素的开始和结束	288	11.7 实战——空调信息管理	339
10.3.5 处理文本数据	290	11.8 本章习题	342
10.3.6 处理空白字符	291	第 12 章 XML 中的 Web 应用	345
10.3.7 处理命名空间	292	12.1 XML 数据源绑定控件	346
10.4 DTDHandler 接口	293	12.1.1 XmlDataSource 控件	346
10.4.1 处理注释声明	293	12.1.2 TreeView 控件	348
10.4.2 处理未解析实体	294	12.1.3 GridView 控件	351
10.5 EntityResolver 接口	296	12.2 XML 与 SQL Server 转换	354
10.5.1 了解 EntityResolver 接口	296	12.2.1 数据转换技术	354
10.5.2 解析实体和忽略实体	296	12.2.2 SqlConnection 对象	356
10.6 ErrorHandler 接口	298	12.2.3 SqlCommand 对象	357
10.6.1 了解 ErrorHandler 接口	298	12.2.4 SqlDataAdapter 对象	359
10.6.2 解析错误信息	299	12.2.5 将 XML 写入数据库	361
10.7 其他的监听器接口	300	12.3 站点地图	363
10.8 实战——解析学生 XML 文档	302		

12.3.1 站点地图简介	364	13.3.2 公共普通函数	384
12.3.2 SiteMapPath 控件	365	13.3.3 XML 处理函数	386
12.4 Web 配置文件	367	13.3.4 其他代码	388
12.5 实战——新闻管理	371	13.4 系统实现	389
12.6 本章习题	374	13.4.1 添加联系人	389
第 13 章 通讯录	377	13.4.2 查看所有联系人	391
13.1 系统概述	378	13.4.3 修改联系人	394
13.2 系统的数据存储	378	13.4.4 删除联系人	395
13.3 通用模块实现	379	习题答案	400
13.3.1 联系人实体类	379		



第 1 章

XML 概 述

XML 是一个在 Internet 上到处可见的流行术语，同时它也是一项日趋成熟的技术。在真实世界中许多应用，特别是在数据管理、显示和组织等方面。毫不夸张地说，XML 正在成为今天一切 Web 服务和大多数面向服务架构(Service Oriented Architecture, SOA)的基础。XML 是整天与数据打交道的使用者的重要工具，不管是在 Internet 上公开的数据还是某个机构内部的私有数据。

本章将向读者介绍 XML 的基本概念、特点、用途和应用领域等知识，XML 的结构以及相关技术将在本书后面的章节进行讨论。

本章学习目标：

- 了解常用的标记语言和分类
- 熟悉 XML 的概念和特点
- 熟悉 XML 的用途
- 掌握 XML 与 HTML 的区别
- 了解 W3C 组织
- 掌握 XML 的技术标准
- 熟悉 XML 的技术优势和商业优势
- 了解 XML 的开发工具
- 掌握 XML 与 CSS 如何结合使用



1.1 标记语言概述

XML 通常被称为可扩展性标记语言，它是标记语言的一种，因此本书在介绍 XML 之前会首先介绍一下标记语言的知识。本节的内容包括两部分，首先了解标记语言的分类，然后再了解它的发展历史。

1.1.1 标记语言的类型

标记语言是一种将文本以及文本相关的其他信息结合起来，展现出关于文档结构和数据处理细节的计算机文字编码。文本相关的其他信息(包括例如文本的结构和表示信息等)与原来的文本结合在一起，但是需要使用标记进行标识。

标记语言不仅仅是一种语言，就像许多语言一样，需要一个运行时环境让它有用。提供运行时环境的元素称为用户代理。通常会将标记语言称为置标语言，一般情况下它包括以下 3 种类型。

1. 标识性标记语言

标识性标记语言(Presentational Markup)定义了编码过程中标记文档的结构信息。例如在文本文件中，文件的标题可能需要用特定的格式表示(例如居中或者放大等)，这时就需要标记文件的标题。字处理以及桌面出版产品有时候能够自动地推断出这类结构信息，但是绝大多数的编辑器(像 Wiki 这样的纯文本编辑器)还不能解决这个问题。

2. 过程性标记语言

过程性标记语言(Procedural Markup)一般都专注于文字的表达，但是通常对于文本编辑者可见，并且能够被软件根据出现的顺序依次解读。为了格式化一个标题，在标题文本前会紧接着插入一系列的格式标识符，以指示计算机转换到居中的显示模式，同时加大及加粗字体。在标题文本后会紧接着缀上格式中止标识，对于更高级的系统宏命令或者堆栈模式会让这一过程的实现方式更加丰富。

大多数情况下，过程性标记能力包含一个 Turing-complete 编程语言。过程性标记语言的范例包含 nroff、troff、TeX、Lout 以及 PostScript。过程性标记语言被广泛应用在专业出版领域，专业的出版商会根据要求使用不同的标记语言以达到出版要求。

3. 描述性标记语言

描述性标记有时被称为逻辑标记或概念上的标记，鼓励作者将编写的方式描述在概念上，而非视觉上。

下面对目前常见的几种标记语言进行简单说明。

- 标准通用标记语言(SGML)，是一种专门的置标语言，被用作编写《牛津英语词典》的电子版本。由于它较为复杂，因此难以普及。
- 超文本标记语言(HTML)，是为网页创建和其他可在网页浏览器中看到的信息设

计的一种标记语言。

- 可扩展标记语言(XML), 又称可扩展置标语言。其中“置标”是指计算机所能理解的信息符号, 通过这种标记, 计算机之间可以处理包含各种信息的文章等。
- 可延伸超文本标记语言(Extensible HyperText Markup Language, XHTML), 也是一种标记语言, 其表现方式与超文本标记语言类似, 不过在语法上更加严格。

1.1.2 标记语言的历史

标记语言用一系列约定好的标记来对电子文档进行标记, 以实现电子文档的语义、结构和格式的定义。这些标记必须很容易和内容区分, 并且易于识别。

1969年, IBM公司开发了一种文档描述语言, 用来解决不同系统中文档格式不同的问题。这样一来, 文档的显示和打印可能更少或更多地依赖特殊的硬件, 不过这样的系统需要不同的计算机系统, 提供专门的软件来显示和打印文档。IBM公司把这种标识语言称作通用标记语言, 英文缩写为GML, 全称是Generalized Markup Language。GML是IBM公司许多文档系统的基础, 包括Script和Bookmaster。IBM公司提出在各文档之间共享一些相似的属性, 如字体大小和版面, 并且设计了一种文档系统, 通过在文档中辅加一种标记, 从而可以标识文档中的每种元素。

经过若干年的发展, 在标记语言的概念达成共识的基础上, IBM公司的研究人员Charles Goldfarb带领的开发团队一直在完善GML, 并将其称为标准通用语言(Standard Generalized Markup Language, SGML)。SGML成为IBM公司内部格式化和维护合法化文件的手段。后来它被不断地扩展和修改, 作为一种全面的信息标准以适应工业范围的广泛应用。1986年, SGML被国际标准化组织(ISO)所采纳。尽管SGML的功能非常强大, 但是由于它非常复杂, 需要一大堆昂贵的软件配合运行, 因而很长一段时间都没有被推广。

1989年, 欧洲物理量子实验室(CERN)的信息专家发明了万维网, 通过超文本链接语言(HTML)人们可以随便浏览网页的内容, 并把原来技术专家之间的网络通信, 普及到了寻常百姓家。随着蒂姆·伯纳斯德·李的不懈努力, 他在缔造了互联网的同时也完善了HTML。

HTML当初仅仅提供了一种对静态文本信息表现的方法, 但是它显然并不能适应越来越多的需要。随后越来越多的标记诞生了, 例如: 使用描述图片; 使用<script>为网页加入脚本支持, 提供动态网页内容。两大浏览器厂商微软和网景公司甚至创建了自己的产品兼容的标记。于是HTML成为了功能强大, 但是失去了严谨结构化的事物。

1996年, 万维网联盟(W3C)寻找一种在Web中应用SGML的灵活性和强大功能的方法, 这导致了1998年2月XML 1.0版本规范的发布。XML具备了SGML的核心特性, 它的规范内容甚至不到SGML的十分之一, 因此其简洁性可见一斑。随着XML的发布, 它的应用如雨后春笋般的出现, 因此它被看作是IT的明日之星。



1.2 XML 的基础知识

可扩展的标记语言(eXtensible Markup Language, XML), 是 Internet 环境中跨平台、依赖于内容的技术, 也是当今处理分布式结构信息的有效工具。本节将简单介绍 XML 的基础知识, 包括它的特点和用途等。

1.2.1 XML 的概念

HTML 被设计用来显示数据, 而 XML 则是被设计用来传输和存储数据。那么到底什么是 XML 呢? 下面几点 XML 进行了简单的总结。

- XML 是可扩展标记语言。
- XML 是一种标记语言, 类似于 HTML。
- XML 的设计宗旨是传输数据。
- XML 标记没有被预定义, 使用者需要自行定义标记。
- XML 被设计为具有自我描述性。
- XML 是 W3C 的推荐标准。

XML 的使用方法非常简单, 只要使用者遵循一定的规则即可, 以下列出的几点用于对 XML 的详细说明。

1. XML 是不作为的

XML 是不会做任何事情的, 它被设计用来结构化、存储和传输信息。例如, 下面是 John 写给 Lily 的一个便签。XML 格式如下。

```
<message>
<from>John</from>
<to>Lily</to>
<heading>Reminder</heading>
<body>Don't forget the meeting!</body>
</message>
```

上述的一条便签具有自我描述性, 它拥有标题和留言信息, 同时包含了发送者和接收者的信息。但是, 这个 XML 文档仍然没有做任何事情, 它仅仅是包装在 XML 标记中的纯粹的信息。使用者需要编写软件或者程序, 才能传送、接收和显示出这个文档。

2. XML 仅仅是纯文本

XML 并没有什么特别的, 它仅仅是纯文本而已。有能力处理纯文本的软件都可以处理 XML。不过, 能够读懂 XML 的应用程序可以有针对性地处理 XML 标记, 标记的功能性意义依赖于应用程序的特性。

3. 通过 XML 可以发明自己的标记

在前面所示的一段代码中, <from>、<to>和<heading>等标记没有在任何 XML 标准中

定义过，这些标记都是由文档使用者发明的，这是因为 XML 没有预定义的标记。

在 HTML 中使用的标记都是预定义的，HTML 文档只使用在 HTML 标记中定义过的标记，例如<p>、和等。XML 与 HTML 不同，它允许使用者定义自己的标记和文档结构。

4. XML 不是对 HTML 的替代

XML 是对 HTML 的补充，它不会替代 HTML，理解这一点很重要。在大多数的 Web 应用程序中，XML 用于传输数据，而 HTML 则用于格式化并显示数据。对 XML 最好的描述是，XML 是独立于软件和硬件的信息传输工具。

5. XML 无所不在

当用户看到 XML 标准突飞猛进的开发进度，以及大批软件开发商采用这个标准日新月异的速度时，不禁感叹它的强大。

目前，XML 在 Web 中起到的作用不会亚于一直作为 Web 基石的 HTML。XML 无所不在，它是各种应用程序之间进行数据传输的最常用的工具，并且在信息存储和描述领域变得越来越流行。

1.2.2 XML 的特点

XML 是 W3C 推荐参考通用标记语言，同样也是 SGML 的子类，使用者可以自定义一组标记。它有多个特点，下面是它的 3 个基本特点。

(1) XML 可以从 HTML 中分离数据

简单来说，就是能够在 HTML 文件之外将数据存储于 XML 文档中，这样能够使开发者集中精力使用 HTML 做好数据的显示和布局，并且确保数据改动时不会导致 HTML 文件也进行改动，从而可以方便地维护页面。

(2) XML 可用于交换数据

把数据转换成 XML 格式存储时将大大减少交换数据时的复杂性，还可以使这些数据能被不同的程序读取。

(3) 利用 XML 可以共享数据

XML 数据以纯文本格式存储，这使得 XML 更易读，更便于记录，更便于调试，使不同系统、不同程序之间的数据共享变得更加简单。

XML 具有良好的格式和丰富的显示样式，实现了数据的共享和通用。用户在遵循严格的语言要求创建 XML 文档时，还方便了 XML 在不同系统之间信息的传输。另外，XML 并不是万能的，它存在着以下缺陷。

- 语义表达能力上略显不足。
- XML 的有些技术尚未形成统一标准。
- 应用程序/浏览器对 XML 的支持也是有限的。



1.2.3 XML 的用途

XML 应用于 Web 开发的许多方面，常用于简化数据的存储和共享，以下通过 6 点说明了 XML 的用途。

1. XML 把数据从 HTML 分离

如果使用者需要在 HTML 文档中显示动态数据，那么每当数据改变时将花费大量的时间来编辑 HTML。通过使用 XML，数据能够存储在独立的 XML 文档中，这样使用者就可以专注使用 HTML 进行布局和显示，并且确保修改底层数据无须对 HTML 进行任何改变。

2. XML 简化数据共享

在真实世界中，计算机系统和数据使用不兼容的格式来存储数据。XML 数据以纯文本格式进行存储，因此提供了一种独立于软件和硬件的数据存储方法，这让创建不同应用程序可以共享的数据变得更加容易。

3. XML 简化数据传输

通过使用 XML 可以在不兼容的系统之间轻松地交换数据。对于使用者来说，其中一项最费时的挑战就是 Internet 上不兼容系统之间交换数据。由于可以通过各种不兼容的应用程序来读取数据，以 XML 交换数据降低了这种复杂性。

4. XML 简化平台的变更

升级到新的系统(硬件或软件平台)总是非常费时的，这必须转换大量数据，不兼容的数据经常会丢失。

XML 数据以文本格式存储，这使得 XML 可以在不损失数据的情况下，更容易扩展或升级到新的操作系统、新应用程序或新浏览器。

5. XML 使开发者的数据更有用

由于 XML 独立于硬件、软件以及应用程序，XML 使开发者的数据更可用，也更有用。不同的应用程序都能够访问开发者的数据，不仅仅在 HTML 页面中，也可以从 XML 数据源中进行访问。

通过使用 XML，开发者的数据可以供各种阅读设备使用(例如手持计算机、语音设备、新闻阅读器等)，还可以供盲人或其他残障人士使用。

6. XML 用于创建新的 Internet 语言

许多新的 Internet 语言是通过 XML 创建的，下面列出了几个示例。

- XHTML，这是最新的 HTML 版本。
- WSDL，用于描述可用的 Web Service，即 Web 服务。
- WAP 和 WML，用于手持设备的标记语言。