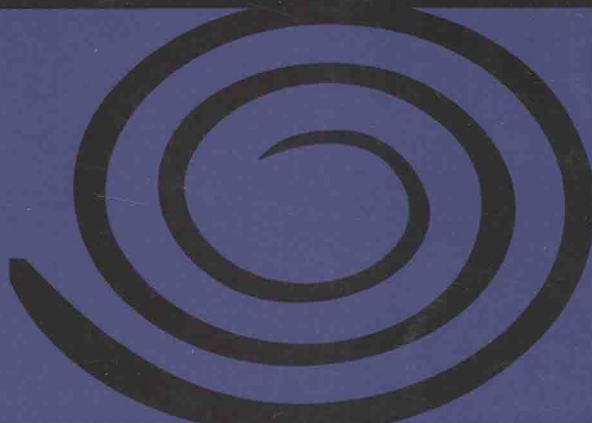
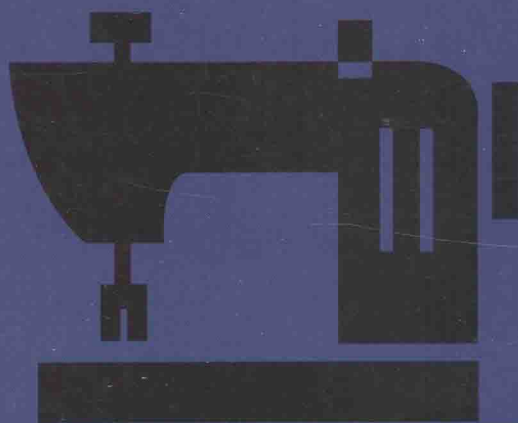




XML

实用教程



理论结合实践，加强动手能力培养

通过6个综合案例演示在实际项目开发中XML的应用

附赠多媒体教学视频、PPT教学课件，以及全书实例的源程序和相关素材

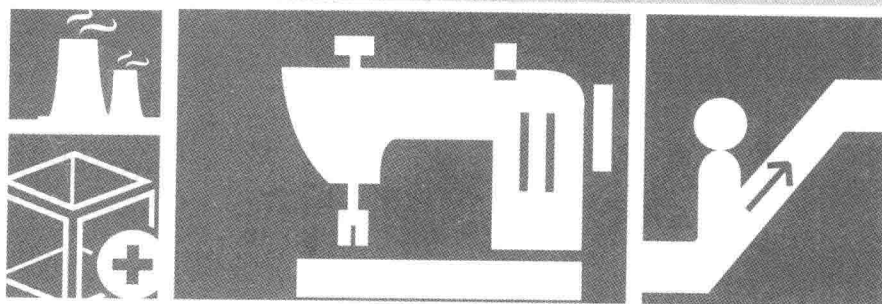
陈作聪 苏静 王龙 等编著



机械工业出版社
China Machine Press

XML

实用教程



陈作聪 苏 静 王 龙 等编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

XML实用教程 / 陈作聪, 苏静, 王龙等编著. -- 北京: 机械工业出版社, 2014.4

ISBN 978-7-111-46082-4

I. ①X… II. ①陈… ②苏… 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 (2014) 第045228号

可扩展标记语言 (XML) 是一种新的 Web 开发辅助语言, 利用它可通过 Internet 进行信息的描述、交换和显示, XML 已经成为不同系统之间数据交换的基础。本书是学习和应用 XML 语言的实用教材, 书中详细阐述了 XML 的基本概念、语法规则、文档类型定义、模式定义、级联样式表、可扩展样式表、文档对象模型, 此外还介绍了 XML 在 Java 和 .NET 中的编程, 最后通过 6 个综合案例 (第 11 章及附录 A~附录 E) 演示了在实际项目开发中 XML 的应用。

本书由浅入深, 循序渐进, 理论结合实践, 侧重于实用。在讲解基础知识的同时给出大量实例, 便于读者亲手实践, 加强动手能力的培养。代码注释详尽, 功能分析透彻, 便于读者深层次理解程序功能和实现方法。每章还给出了课后扩展练习, 便于读者巩固所学知识。

本书可作为高等院校计算机、电子商务以及信息类相关专业的教材, 也可供相关技术人员、广大 Web 应用程序开发者和用户参考学习。

XML实用教程

陈作聪, 苏静, 王龙 等编著

出版发行: 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: 夏非彼 迟振春

印 刷: 中国电影出版社印刷厂

开 本: 188mm×260mm 1/16

书 号: ISBN 978-7-111-46082-4

ISBN 978-7-89405-335-0 (光盘)

版 次: 2014年6月第1版第1次印刷

印 张: 23.25

定 价: 49.00元 (附光盘)

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 82728184 88379604

读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

前言

XML 是 eXtensible Markup Language 的简写, 它是由万维网联盟 (W3C) 制定的一种标记语言, 也是表示结构化数据的事实上的工业标准, 被广泛应用于结构化数据的存储和交换中。XML 提供了直接在数据上工作的通用方法, 其优势在于将结构化数据的存储和显示相分离, 允许不同来源数据的无缝集成和对同一数据的多种处理。

目前在软件开发、电子商务等领域, 几乎处处都能看见 XML 的身影。很多人可能觉得 XML 非常简单, 因为他们每天都会接触 XML 文档, 但是许多人仅仅知道 XML 可以用作配置文件, 至于更多、更深的内容就知之甚少。实际上, XML 无论对于 Java 平台还是其他如 .NET 平台都具有非常重要的价值。XML 在电子商务、移动应用开发、Web Service、云计算等技术和领域中也起着非常重要的作用。特别是随着云计算的兴起, 本地和云端之间通过 Internet 进行数据传输和数据交换成为必然的趋势, 在这之中 XML 承担着数据存储和数据交换的重任。

本书系统介绍了设计和使用 XML 文档所必需的基本技术, 以及很多现在非常重要的相关技术。通过本书的学习, 读者可以对 XML 语言有一个全面的了解, 掌握相关的主要知识点, 在软件设计中运用 XML 技术灵活地进行信息的描述、交换和显示。全书共分为 11 章, 具体内容如下。

第 1 章 XML 概述

本章主要介绍了标记语言、SGML 语言、HTML 超文本语言产生的原因和发展, 最后讲解 XML 语言的概念、优势和相关技术。

第 2 章 XML 语法

本章主要介绍 XML 的语法, 重点介绍了 XML 文档结构、XML 格式的约束规则、如何声明元素和属性, 以及怎样使用命名空间等。

第 3 章 XML 应用开发工具

本章主要介绍在整个 XML 技术的发展阶段出现过的编辑器、解析器、浏览器及其他工具软件, 并对 XMLSpy 的安装、主界面、主要功能和基本使用进行详细介绍。

第 4 章 规范 XML 文档的 DTD

本章重点介绍了 DTD 中的元素、属性、实体的声明语法和格式, 以及在 XML 文档中使用 DTD 的方式。

第 5 章 XML 模式

本章重点介绍了模式的文档结构、XML 模式中元素及属性的声明语法、使用



XML 模式的方式。

第 6 章 修饰 XML 文档的 CSS

本章首先介绍了 CSS 的作用，然后详解 CSS 语法及常见的 CSS 属性设置，并以实例的方式讲述了如何使用 CSS。

第 7 章 转换 XML 文档的 XSL

本章着重讲解 XSL 转换语言、XSL 文档的编写、模板的规则、XSL 的转换执行过程、实现 XSL 的方式。

第 8 章 数据岛

本章详细介绍了数据岛的使用方式、数据源对象 DSO 的操作，并给出了遍历和分页显示 XML 文档的实例。

第 9 章 DOM 和 SAX

本章重点介绍文档对象模型 DOM 的基本概念、常用接口、常用对象、DOM 和 XML 如何进行交互等。此外，还讲述了操作 XML 文档的另外一种方式——SAX，包括其工作原理、常用的事件和接口，以及对 XML 的解析。

第 10 章 .NET 和 XML

本章主要介绍了在 .NET 平台上操作和处理 XML 文件需要用到的具体技术及相关对象，重点介绍了 .NET 平台上的 DOM 编程方法，以及 XML 与 ADO.NET 技术的关系。

第 11 章 基于 XML 的在线视频网站

在本章中，通过一个综合案例演示了在实际项目开发中 XML 的具体应用，通过对案例的剖析，体现 XML 技术在 Web 应用方面的优势，并进一步掌握利用 DOM 操作 XML 文档的方法和技巧。

本书由浅入深，循序渐进，理论结合实践，侧重于实用。在讲解基础知识的同时给出大量实例，便于读者亲手实践，加强动手能力的培养。代码注释详尽，功能分析透彻，便于读者深层次理解程序功能和实现方法。章后还给出了扩展练习，便于读者巩固所学知识。书中的应用实例均来自于实际开发，读者对其稍加修改后可直接应用到实践中。

本书可作为高等院校计算机、电子商务以及信息类相关专业的教材，也可供相关技术研究人员、广大 Web 应用程序开发者和用户参考学习。

本书主要由陈作聪、苏静、王龙编写，此外，刘小松、胡书敏、王相羽、袁润非、杨阳、王炳乾、苏宝保、张琴、李龙、王华、李辉、刘峰、徐浩、李建国、马建军、唐爱华、苏小平、朱丽云、马淑娟、周毅、张乐、李大勇、许小荣、魏勇、王云等同志也参与了本书的部分编写工作。

由于作者水平有限，书中错误、纰漏之处难免，欢迎广大读者、同仁批评斧正。

编 者

2014 年 3 月

目 录

前言

第 1 章 XML 概述.....	1
1.1 标记语言	1
1.1.1 标记语言的发展.....	1
1.1.2 标记语言的理解.....	2
1.2 SGML 通用标记语言	3
1.3 HTML 超文本标记语言	4
1.3.1 基本的 HTML 结构	4
1.3.2 常用的 HTML 标记	6
1.4 初识 XML	8
1.4.1 XML 的产生.....	9
1.4.2 XML 的简介.....	9
1.4.3 XML 与 SGML、HTML 的关系	11
1.4.4 XML 的举例.....	12
1.4.5 XML 的优点.....	14
1.4.6 XML 的应用及发展.....	16
1.5 XML 相关技术	19
1.5.1 文档类型定义 (DTD)	19
1.5.2 层叠样式表 (CSS)	19
1.5.3 可扩展样式语言 (XSL)	20
1.5.4 XSLT (XSL Transformation)	20
1.5.5 文档对象模型 (DOM)	21
1.5.6 XML 模式 (XML Schema)	21
1.5.7 XPath (XML Path Language)	21
1.5.8 XLink (XML Link)	22
1.5.9 SAX (Simple API for XML)	22
1.5.10 XML 查询语言 (XQL)	22
1.5.11 XPointer	23
1.6 扩展练习	23
第 2 章 XML 语法.....	25
2.1 XML 文档的结构	25



2.1.1 XML 文档的组成.....	25
2.1.2 XML 文档的声明.....	26
2.1.3 XML 文档的处理指令.....	28
2.1.4 XML 文档的注释.....	28
2.2 XML 文档的元素.....	30
2.2.1 元素的定义.....	30
2.2.2 元素的命名规则.....	32
2.2.3 元素的内容构成.....	32
2.2.4 元素的特征.....	33
2.3 XML 元素的属性.....	35
2.3.1 属性的定义.....	35
2.3.2 属性值的定义.....	36
2.3.3 元素和属性的转换.....	37
2.4 CDATA 段.....	38
2.4.1 XML 中的特殊字符.....	38
2.4.2 CDATA 的格式.....	39
2.4.3 CDATA 的位置.....	40
2.5 XML 命名空间.....	41
2.5.1 概述命名空间.....	41
2.5.2 定义命名空间.....	42
2.5.3 默认命名空间.....	44
2.6 格式良好的 XML 文档.....	45
2.6.1 创建格式良好的 XML 文档.....	46
2.6.2 有效的 XML 文档.....	46
2.7 扩展练习.....	47
第 3 章 XML 应用开发工具.....	50
3.1 XML 软件.....	50
3.1.1 XML 浏览器.....	50
3.1.2 XML 解析器.....	52
3.1.3 XML 编辑器.....	54
3.2 Altova XMLSpy.....	56
3.2.1 Altova XMLSpy 的安装.....	56
3.2.2 Altova XMLSpy 的主界面.....	59
3.2.3 Altova XMLSpy 的主要功能.....	62
3.2.4 Altova XMLSpy 的使用.....	65
3.3 扩展练习.....	67

第 4 章	规范 XML 文档的 DTD	70
4.1	认识 DTD	70
4.1.1	DTD 的概念	70
4.1.2	DTD 的基本结构	72
4.2	DTD 中的定义元素	75
4.2.1	声明元素类型	75
4.2.2	元素基本类型	76
4.2.3	元素限定符	79
4.3	DTD 中的定义属性	82
4.3.1	属性的声明	82
4.3.2	属性的类型	84
4.3.3	属性的附加声明	94
4.4	DTD 中的引用	96
4.4.1	内部 DTD 的引用	96
4.4.2	外部 DTD 的引用	96
4.4.3	DTD 的混合引用	100
4.5	DTD 中的实体	101
4.5.1	什么是实体	101
4.5.2	普通实体	102
4.5.3	参数实体	105
4.6	扩展练习	107
第 5 章	XML 模式	111
5.1	Schema 概述	111
5.2	Schema 的格式	113
5.3	Schema 的元素声明	117
5.3.1	schema 根元素	118
5.3.2	element 元素	118
5.3.3	默认值和固定值	120
5.3.4	引用和替代元素	121
5.4	Schema 的属性声明	124
5.5	Schema 的数据类型	125
5.5.1	简单类型	125
5.5.2	复杂类型	130
5.6	Schema 的名称空间	135
5.6.1	名称空间的类型	136
5.6.2	名称空间的引用	136
5.7	创建可视化的 XML Schema	137
5.8	扩展练习	140



第 6 章 修饰 XML 文档的 CSS	144
6.1 CSS 入门	144
6.1.1 CSS 基础知识	144
6.1.2 CSS 举例	146
6.2 CSS 语法	148
6.3 CSS 选择器	148
6.4 CSS 属性	154
6.4.1 显示属性的设置	154
6.4.2 字体属性的设置	156
6.4.3 色彩属性的设置	160
6.4.4 边框属性的设置	163
6.4.5 布局属性的设置	164
6.4.6 背景属性的设置	167
6.4.7 文本属性的设置	169
6.5 CSS 链接	172
6.5.1 一般链接指令	172
6.5.2 @import 指令	173
6.5.3 无需链接的内部 CSS	173
6.6 扩展练习	175
第 7 章 转换 XML 文档的 XSL	177
7.1 XSL 入门	177
7.1.1 什么是 XSL	177
7.1.2 XSL 原理	178
7.1.3 XSL 举例	179
7.1.4 XSL 与 CSS 的区别	182
7.2 XSL 模板	182
7.2.1 XSL 模板的规则	183
7.2.2 XSL 模板的调用	184
7.2.3 其他常用元素	184
7.3 XSL 选择节点	186
7.3.1 使用元素名选择节点	186
7.3.2 使用路径选择节点	187
7.3.3 使用匹配符选择节点	187
7.3.4 使用元素属性选择节点	188
7.4 XSL 控制指令	191
7.4.1 简单判断指令	191
7.4.2 多条件判断指令	192
7.4.3 循环处理指令	194

7.4.4	输出内容排序指令.....	196
7.5	XPath 查询.....	198
7.5.1	XPath 概述.....	198
7.5.2	XPath 的类型和运算符.....	200
7.5.3	XPath 的位置路径表达式.....	201
7.5.4	XPath 的标准函数.....	204
7.6	扩展练习.....	207
第 8 章	数据岛.....	210
8.1	数据岛概述.....	210
8.1.1	定义数据岛.....	210
8.1.2	绑定数据岛.....	211
8.2	应用数据岛.....	214
8.2.1	数据岛的使用形式.....	214
8.2.2	使用数据岛访问 XML 元素内容.....	216
8.2.3	使用数据岛访问 XML 标记属性.....	221
8.3	数据源对象 DSO.....	223
8.3.1	什么是 DSO.....	223
8.3.2	XML 数据岛的操作.....	226
8.3.3	使用 DSO 显示图片.....	230
8.4	切换 DSO 和 XSL.....	232
8.5	扩展练习.....	235
第 9 章	DOM 和 SAX.....	239
9.1	解析 XML 文档.....	239
9.2	什么是 DOM.....	241
9.3	DOM 文档树模型.....	242
9.4	DOM 接口和对象.....	244
9.4.1	DOM 接口.....	244
9.4.2	DOM 对象.....	246
9.5	DOM 的使用.....	247
9.5.1	Java 中的 DOM.....	247
9.5.2	DOM 中的类和方法.....	248
9.5.3	遍历 XML 文档.....	251
9.5.4	访问属性节点.....	253
9.5.5	访问节点内容.....	254
9.5.6	删除元素和属性.....	256
9.5.7	动态创建 XML 文档.....	258
9.6	SAX 概述.....	261



9.6.1 SAX 的优缺点.....	261
9.6.2 SAX 的事件.....	263
9.6.3 SAX 的常用接口.....	265
9.7 使用 SAX 解析器.....	266
9.7.1 创建步骤.....	266
9.7.2 解析 XML 文档.....	267
9.8 扩展练习.....	270
第 10 章 .NET 和 XML	273
10.1 .NET 简介.....	273
10.2 .NET 的 XML 分析器.....	275
10.3 使用流模式处理 XML.....	275
10.3.1 处理 XML 的类.....	275
10.3.2 读取 XML.....	277
10.3.3 写出 XML.....	281
10.4 使用 DOM 处理 XML.....	285
10.4.1 使用 DOM 加载及保存 XML 数据.....	286
10.4.2 使用 DOM 访问 XML.....	288
10.5 ADO.NET 与 XML.....	300
10.5.1 ADO.NET 简介.....	300
10.5.2 XML 与 DataSet 对象的关系.....	301
10.5.3 使用 DataSet 对象访问 XML.....	302
10.6 扩展练习.....	305
第 11 章 基于 XML 的在线视频网站.....	307
11.1 系统分析与设计.....	307
11.1.1 系统需求分析.....	307
11.1.2 系统模块设计.....	308
11.2 系统 XML 文件设计.....	309
11.2.1 user.xml.....	309
11.2.2 film.xml.....	310
11.3 系统运行演示.....	312
11.4 前台管理模块设计.....	314
11.4.1 显示前台首页.....	315
11.4.2 显示视频类别信息列表.....	317
11.4.3 显示视频详情.....	321
11.4.4 查询视频.....	322
11.4.5 观看视频.....	323
11.5 后台管理模块设计.....	325

11.5.1	管理视频.....	325
11.5.2	删除视频.....	328
11.5.3	添加视频.....	329
11.5.4	用户管理.....	331
11.6	登录模块设计	334
附录 A	新闻发布系统	337
附录 B	公共信息交流网.....	343
附录 C	班级管理系统	349
附录 D	在线 RSS 阅读器	353
附录 E	在线同学录.....	357

第 1 章 XML 概述

XML 是 W3C（万维网联盟）提出的一种可扩展标记语言，其全称是 eXtensible Markup Language，它是随着人们对信息传播要求的不断提高而产生的一种技术。XML 的出现，把网络表达语言集合推进了一大步，使得标记语言可以由编程人员按需要进行扩展，其扩展能力是无限的，这就给 Web 编程带来了革命性的进展。本章将首先介绍标记语言产生的背景、发展历史，并对目前的两种常用标记语言：HTML 和 XML 做了一个概括性的描述，最后对 XML 的优势和与 XML 密切相关的技术进行介绍。

1.1 标记语言

标记语言（Markup Language）也被称为“标签语言”或“置标语言”，它是由一些标识码（Codes）或控制标签（Tags）所组成，用于对电子文档进行标记，以实现电子文档的语义、结构及格式的定义。

1.1.1 标记语言的发展

电子数据交换是早期计算机网络发展的一个主要目的，而结构化信息就是使数据交换成为可能的重要方法。因为，如果在不同的行业中需要交互使用的信息采用统一的模型标识，那么信息就能方便和高效地进行共享。

在上述背景下，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，文档的显示和打印可能更少或更多地依赖于特殊的硬件，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 把自己这种标识语言称作通用标记语言（Generalized Markup Language，即 GML）。GML 是 IBM 许多文档系统的基础，包括 Script 等。IBM 提出在各文档之间共享一些相似的属性，诸如字体大小和版面，并且设计了一种文档系统，通过文档中辅加一种标签，从而可以标识文档中的每种元素。

经过多年的发展，1984 年国际标准化组织 ISO 开始对此提案进行讨论，1986 年国际标准化组织 ISO 认为 IBM 提出的概念很好，并发布了为生成标准化文档而定义的标记语言标准 ISO 8897——称为新的语言 SGML，即标准通用标记语言。SGML（Standard Generalized Markup Language）从 IBM 的 GML 演化而来，是第一个标准化的信息结构化技术。SGML 是结构化的、可扩展的语言，这些特点使它在很多公司受到欢迎，被用来创建、处理和发布



大量的文本信息。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，因而在很长一段时间都没有得到推广。

在 1989 年，欧洲物理量子实验室（CERN）的专家发明了万维网，通过超文本链接语言 HTML，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普通家庭。随着蒂姆·伯纳斯·李的不懈努力，他在缔造了互联网的同时也完善了 HTML 语言。

HTML 当初仅仅提供了一种对静态文本信息表现的方法，但并不能适应越来越多的需要。随后诞生了越来越多的标签，如使用用于描述图片，通过<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，都创建了由自己产品兼容的标签。于是，HTML 成了功能强大但失去了严谨结构化的事物。

1996 年，W3C 寻找一种在 Web 中应用 SGML 的灵活性和强大功能的方法，这促使了 1998 年 2 月 XML 1.0 规范的发布。XML (eXtensible Markup Language, 可扩展标记语言) 具备 SGML 的核心特性，又非常简洁，XML 规范的内容甚至不到 SGML 的十分之一。XML 的应用随即如雨后春笋般展开，并被看作是 IT 界的重大变革。

1.1.2 标记语言的理解

标记语言是将文本（Text）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电脑文字编码，包括：文本的结构和表示信息等，它们与原来的文本结合在一起，使用标记进行标识，标记最早用于出版业，是作者、编辑以及出版商之间用于描述出版作品的排版格式。

标记语言是可以让电子文档变得更具有结构性，而这种结构性能够便于应用软件解析和应用。但是，标记如果独立存在则是毫无意义的，它们必须结合数据后才能形成一份有用的电子文档，而所谓有用的电子文档则是针对应用软件而言的，也就是说应用软件能够解读这份电子文档中的标记语言，并且由标记语言的意义来对该份电子文档作特定的处理。这些标记必须能够容易地和内容相区分、易于识别。标记语言必须定义什么样的标记是允许的，什么样的标记是必须的，标记是如何与文档的内容相区分的，以及标记的含义是什么。

例如，HTML 中可以使用
标签，
可以被应用软件解释为换行，但是如果前后没有数据，那么这个换行是没有意义的。

标记语言实际上是一种类似于说明性质的语言，它只是告诉它的解释者代表什么样的含义。通过标记语言，可以对一个段落或一篇文章的文字布局和显示样式进行设定。下面是一段标记语言所修饰的代码：

```
<h4>标记语言</h4>
```

```
<p>h4 标记表示一个标题</p>
```

```
<ul>
```

```
  <li>通用标记语言</li>
```

```
  <li>超文本链接语言</li>
```

```
  <li>可扩展标记语言</li>
```



在上面的代码中，标签由小于号(<)和大于号(>)括起来的文字组成，很容易就和实际的内容区分开来。浏览器能够解读这些标签，并按照定义的格式显示。文中的标记主要是修饰标记中间内容的显示方式。

1.2 SGML 通用标记语言

SGML 是国际上定义电子文件结构和内容描述的标准。早在 1969 年，IBM 公司的工作人员 Ed Mosher、Ray Lorie 和 Charles F. Goldfarb 就开发了一种文档描述语言 GML，用来解决不同系统中文档格式不同的问题。GML 是 IBM 许多文档系统的基础，是一种自参考的语言，可以描述任何其他语言的语法和词汇。1974 年，Goldfarb 提出了在 GML 中实现语法分析器的概念。经过多年的发展，1984 年国际标准化协会(ISO)开始对此提案进行讨论，终于在 1986 年演变成一个国际标准(ISO 8897)，并被称为 SGML。

SGML 实际上是一种通用的文档结构描述符号化语言，主要用来定义文献模型的逻辑和物理类结构。一个 SGML 规定了在文档中插入描述标记的标准格式，指定了描述文档结构的标准方法，简而言之，可以使用 SGML 创作每一种类型的文档设置层次结构模型。

SGML 具有以下主要特点：

- 可支持无数的文档结构类型，例如公告、技术手册、章节目录、设计规范、各种报告、信函和备忘录等。
- 可以创建与特定的软硬件无关的文档，因此很容易与不同计算机系统的用户交换文档。

一个典型的文档可以分成三个部分：结构、内容和样式。SGML 主要是处理结构和内容之间的关系。

1. 结构

为了描述文档的结构，SGML 定义了一种称为“文档类型定义(Document Type Definition, DTD)”的文件，它为组织的文档元素(章节和主题等)提供了一个框架。此外，DTD 还为文档元素之间的相互关系制定了规则，DTD 定义的这些规则可以确保文档的一致性。

2. 内容

这里指的内容就是信息本身，内容包括信息名称(标题)、段落、项目列表和表格中的具体内容、具体的图形和声音等。确定内容在 DTD 结构中的位置的方法称为“加标签”，而创建 SGML 文档实际上就是围绕内容插入相应的标签。这些标签就是给结构中每一部分的开始和结束做标记。

3. 样式

SGML 本身就是在定义样式时的设置标准，即文档样式语义学和规范语言(Document Style Semantics and Specification Language, DSSSL)。



SGML 是一个庞大复杂的系统, 其应用标准包括超媒体文档标记方面的超媒体语言 HyTime、文献样式语义和规范说明语言 DSSSL、可扩展的样式语言 XSL 和可扩展的链接语言 XLL 等。SGML 规模庞大、功能丰富, 具有各种选项, 用来标记文献以使文献信息不依赖于特定的软硬件, 而且具有方便的互操作性和格式的转换功能, 以适合多种应用或达到反复使用的目的。

SGML 是很多大型组织, 例如电信、飞机、汽车公司、化工企业和军队的文档标准, 它是语言无关的、结构化的、可扩展的语言, 这些特点使它在很多公司受到欢迎, 被用来创建、处理和发布大量的文本信息。

1.3 HTML 超文本标记语言

HTML 的全称是“Hyper Text Markup Language”, 意思是“超文本标记语言”, 它是专门用来编写网页的一种编程语言。

1989 年, 欧洲粒子物理实验室的研究人员开发了一个基于 SGML 的超文本简化版本, 被称为超文本标记语言 HTML。HTML 是 SGML 的一个子集, 继承了 SGML 的许多重要特点。HTML 只使用了 SGML 中很小一部分标记, 并且为了便于在计算机上实现, HTML 可使用的标记是固定的, 也不需要包含 DTD, 因而其语法是不可扩展的。HTML 这种固定的语法使其易于使用, 开发用来处理 HTML 文档的浏览器也十分容易。正是由于 HTML 的简单性和易用性, 使 Web 技术及其应用得到了空前的发展, 并很快从计算机界走向全社会, 并最终走向千家万户。

大多数网页的构成基础都是 HTML 语句。我们上网浏览一个网页时, 实际上是把该网页所对应的 HTML 文件下载到我们的计算机中, 然后由本地计算机中的浏览器(如 IE)再解释、显示。

1.3.1 基本的 HTML 结构

标记是 HTML 语言中最基本的单位。HTML 的标记分为单独标记和成对标记两种。

- 成对标记是由首标记<标记名>和尾标记</标记名>组成的, 成对标记的作用域只作用于这对标记中的文档。
- 单独标记的格式为<标记名>, 单独标记在相应的位置中插入元素就可以了。

大多数的标记都有自己的一些属性, 属性用于进一步改变显示的效果, 各属性之间无先后次序, 属性是可选的, 属性也可以省略而采用默认值。当浏览器收到 HTML 文件后, 就会解释里面的标记符, 然后把标记符相对应的功能表达出来。

HTML 文本是由 html、head、title 和 body 共 4 个标记组成, 这 4 个标记能构成整个网页的效果。它们属于基本标记, 其基本格式如下:

```
<html>  
<head>
```



```
<title></title>
</head>
<body></body>
</html>
```

【实例 1-1】构建网页最基本的结构

通过这个实例，我们使用最简单的 HTML 编辑工具来创建一个简单的 HTML 页面来显示一些文本，具体步骤如下：

01 打开 Windows 中的记事本，创建一个 html 文件，命名为“web.html”，在其中输入如下的代码：

```
1. <html>
2.   <head>
3.     <title>一个最简单的网页</title>
4.   </head>
5.   <body>使用 HTML 标记的网页！</body>
6. </html>
```

代码解析：

上面的第 1 行与第 6 行代码中的<html>和</html>是放在文档的最外层，文档中的所有文本和 html 标记都包含在其中，它表示该文档是以超文本标识语言（HTML）编写的。

第 2 行与第 4 行中的< head ></ head >是 HTML 文档的头部标记，在浏览器窗口中，头部信息是不被显示在正文中的，在此标记中可以插入其他标记，用以说明文件的标题和整个文件的一些公共属性。

第 3 行中的<title>和</title>是嵌套在< head >标记中的，标记之间的文本是文档标题，它被显示在浏览器窗口的标题栏。

第 5 行中的< body ></ body >标记一般不省略，标记之间的文本是正文部分，一个网页所要显示的内容都是放在这里的，即我们所要设计的主体部分，也是一个 HTML 页面最重要的部分。

上面的这几对标记在文档中都是唯一的，head 标记和 body 标记是嵌套在 HTML 标记中的。

02 使用 IE 浏览器运行该文件，可以看到如图 1-1 所示的运行结果。

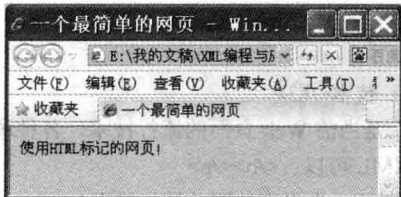


图 1-1 HTML 页面效果