

XML XSL XHTML

可扩展

网

页语言

— XML
XSL
XHTML

使用详解

黄斯伟 编著

人民邮电出版社
www.pptph.com.cn

00128508

T-393.092
37/

可扩展 **网** 页语言



— XML XSL XHTML

使用详解

黄斯伟 编著



北航 C0546818

人民邮电出版社

图书在版编目(CIP)数据

可扩展网页语言:XML、XSL、XHTML 使用详解/黄斯伟编著. —北京:人民邮电出版社,2001.4

ISBN 7-115-09242-7

I.可... II.黄... III.①可扩展语言,XML—程序设计②可扩展语言,XSL—程序设计
③超文本标记语言,HTML—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2001)第 18910 号

可扩展网页语言——XML、XSL、XHTML 使用详解

◆ 编 著 黄斯伟

责任编辑 杨 璐

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@pptph.com.cn

网址 <http://www.pptph.com.cn>

读者热线:010-67129212 010-67129211(传真)

北京汉魂图文设计有限公司制作

北京顺义向阳胶印厂印刷

新华书店总店北京发行所经销

◆ 开本:787×1092 1/16

印张:16.75

字数:411 千字

2001 年 4 月第 1 版

印数:1—6 000 册

2001 年 4 月北京第 1 次印刷

ISBN 7-115-09242-7/TP·2174

定价:28.00 元

内容提要

本书对于可扩展标识语言（XML）、可扩展样式语言（XSL）和可扩展超文本标记语言（XHTML）的用法做了详细的介绍。

作者结合实际的操作和国际标准，对 XML 的创建、浏览、应用进行了深入细致的探讨，运用了大量的实例对 XSL 进行解释，还详细介绍了 XHTML 的应用和语言规范。书中将三者有机地结合在一起，难度由浅入深，逐步提高。

在附录中，还提供了各个语言规范的速查表，便于读者参考。

本书结构清晰、难度适中，适合从事网站建设的专业人士参考借鉴，也可以作为广大网络爱好者制作网页的参考教材。

编者的话

随着网络的发展，在各种媒体和出版物中越来越多地谈论到 XML 语言。

作为一种新兴的标识语言，XML 不是独立的，而是同许多语言有着千丝万缕的联系。例如，XML 和 JavaScript、动态 HTML 就能很好地结合使用。

在网站建设过程中，我们希望网站的数据内容能和显示样式直接分离，而 XML 和 XSL 正是控制这两个方面的最新语言。同时，XML 和 XSL 具有的“剥离数据”、“独立显示”的特点令数据在不同应用程序，甚至不同系统之间的相互传输成为可能。

XHTML 可以有效地把 XML 和 XSL 结合起来，它有比 HTML 更严格的、更规范的语法。

兼容性和易用性是网页设计的首要原则，XHTML 严格的语法规范令网页的兼容性大大提高。

本书将 XML、XSL 和 XHTML 三者有机地结合在一起，参考国际标准编写了大量的实例，并对它们进行了详细的介绍。为加强本书的可读性，本书特别注意简化叙述纯理论知识，而重点利用例子对知识点进行说明。书中所选的实例来源于生活，如用户信息注册、股票牌价、天气预报系统等，让读者更容易接受和理解，并且可以作为日后创建站点的模板。

本书是学习 XML、XSL 和 XHTML 的基础教材，读者能够在本书大量实例的带领下，循序渐进地领会其中的奥秘。

书中正文部分由黄斯伟编纂。胡萍负责编写附录，同时为书稿的修改和校对做了大量工作。

本书特为读者建立了讨论信箱，如果读者在阅读和学习的过程中有任何问题或者想法及建议，都可以通过电子邮件与我们联系，邮件请寄：css@email.com。另外，本书特开通 <http://flowaway.topcool.net/> 网站，提供与书本相关的资料，欢迎浏览。

编者

2000 年 3 月

目 录

第 1 章 XML 概述.....	1
1.1 XML 的起源和发展.....	1
1.1.1 标记语言.....	1
1.1.2 推出 XML.....	2
1.1.3 XML 简介.....	3
1.2 XML 的编辑工具.....	4
1.2.1 通用编辑工具.....	4
1.2.2 专用 XML 编辑器.....	6
1.3 支持 XML 的浏览器.....	8
1.3.1 两大常用浏览器.....	8
1.3.2 其他浏览器.....	10
第 2 章 开始学习 XML.....	13
2.1 设计 XML 元素.....	13
2.1.1 元素.....	13
2.1.2 属性.....	15
2.1.3 元素生成器.....	18
2.1.4 检查语法.....	19
2.2 设计 XML 文档.....	21
2.2.1 设计规则.....	21
2.2.2 语法检查工具.....	22
2.3 浏览 XML.....	23
2.3.1 两种浏览器.....	23
2.3.2 用 IE 5.0 显示 XML.....	24
2.4 加入显示信息.....	26
2.4.1 用 CSS 格式化 XML.....	26
2.4.2 XML 不能替代 HTML.....	29
2.4.3 用 XSL 格式化 XML.....	31
2.5 设计 XML 数据岛.....	34
2.5.1 XML 数据岛.....	34
2.5.2 数据捆绑.....	35
2.6 解析器和对象模型.....	39

2.6.1 XML 解析器	39
2.6.2 XML 对象模型	43
第 3 章 进一步学习 XML	47
3.1 XML 元素命名	47
3.1.1 命名冲突	47
3.1.2 解决命名冲突	47
3.1.3 命名规则的应用	50
3.2 PCDATA 和 CDATA	50
3.2.1 PCDATA	50
3.2.2 字符实体	51
3.2.3 CDATA	53
3.3 XML 编码	54
3.3.1 Windows 95/98 记事本	54
3.3.2 Windows 2000 记事本	55
3.3.3 错误信息	57
3.4 XML 应用程序	58
3.4.1 用数据岛调用	58
3.4.2 数据岛与表格捆绑	59
3.4.3 与或<div>捆绑	60
3.4.4 加入导航脚本	62
3.5 XML 行为	63
3.5.1 XML 行为的概念	63
3.5.2 XML 行为实例	66
第 4 章 XML 高级技术	69
4.1 简单的 XML 服务器	69
4.1.1 硬盘上的服务器	69
4.1.2 安装 IIS	70
4.1.3 安装 PWS	73
4.2 利用 ASP 文件	74
4.2.1 生成 HTML	74
4.2.2 生成 XML	75
4.3 从数据库获取 XML	76
4.3.1 创建 Access 数据库	76
4.3.2 数据库与 XML 结合	80
第 5 章 XSL 技术	83
5.1 XSL 简介	83
5.1.1 CSS 格式化 HTML	83

5.1.2	CSS 格式化 XML	84
5.1.3	XSL 格式化 XML	87
5.1.4	XSL 浏览器	88
5.2	XSL 的基本使用	89
5.2.1	HTML 模板	89
5.2.2	设计规范的 HTML	94
5.2.3	访问和输出属性	96
5.3	发掘 XSL 的潜能	100
5.3.1	筛选数据	100
5.3.2	数据排列	104
第 6 章	进一步学习 XSL	107
6.1	XSL 元素	107
6.1.1	W3C 的 XSLT 元素	107
6.1.2	IE 5.0 的 XSLT 元素	109
6.2	XSL Patterns	110
6.2.1	XSL Patterns 语法介绍	111
6.2.2	XSL Patterns 演示工具	112
6.2.3	DOM 定义 XSL 查询	117
6.2.4	XSL 定义 XSL 查询	118
6.2.5	定义 XSL 查询实例	119
第 7 章	更多 XSL 实例	125
7.1	高级 XSL 特性	125
7.1.1	处理异常数据	125
7.1.2	设计“match”属性	130
7.1.3	自动编号	132
7.1.4	自动创建链接	134
7.2	创建动态 HTML	136
7.2.1	运行脚本	136
7.2.2	脚本互动	139
7.3	“context()”方法	141
7.3.1	网格显示	141
7.3.2	有条件选择输出	143
7.3.3	上下文关联输出	144
7.3.4	逗号分隔输出	146
7.3.5	排序的分隔输出	146
第 8 章	拓展 XSL 功能	149
8.1	条形图	149

8.1.1	调用脚本	149
8.1.2	定义变量和函数	150
8.1.3	完整的例子	151
8.1.4	脚本影响转换	155
8.2	附加方法	156
8.3	两种脚本语言	159
8.3.1	JavaScript 脚本	162
8.3.2	VBScript 脚本	166
第 9 章	XHTML 基础	171
9.1	XHTML 简介	171
9.2	XHTML 和 HTML 的区别	171
9.2.1	小写代码	171
9.2.2	必须有结尾标记	172
9.2.3	属性值必须用引号	173
9.2.4	最小化属性	173
9.2.5	放弃“name”属性	174
9.2.6	脚本和样式元素	174
9.2.7	遵循 XML 规范	174
9.2.8	文档声明	175
9.3	XHTML 语法	176
9.3.1	三大部分	176
9.3.2	文档类型定义	177
9.4	转换现有的 HTML	177
9.4.1	手工转换	177
9.4.2	转换软件	178
9.5	验证 XHTML 代码	179
第 10 章	XHTML 规范	181
10.1	XHTML 标记	181
10.1.1	基本标记	181
10.1.2	字符格式标记	183
10.1.3	输出标记	185
10.1.4	块标记	187
10.1.5	链接标记	188
10.1.6	框架标记	188
10.1.7	表单标记	189
10.1.8	列表标记	192
10.1.9	图像标记	194
10.1.10	表格标记	195

10.1.11 样式标记	197
10.1.12 元信息标记	199
10.1.13 脚本标记	199
10.2 XHTML 属性	201
10.2.1 核心属性	201
10.2.2 语言属性	202
10.2.3 键盘属性	203
10.3 XHTML 事件属性	203
10.3.1 窗口事件	203
10.3.2 控件元素事件	204
10.3.3 键盘事件	204
10.3.4 鼠标事件	207
第 11 章 常见问题解答	211
11.1 一般问题	211
11.1.1 XML 概念	211
11.1.2 XML 解析器	212
11.1.3 HTML 不可替代	214
11.1.4 使用 XML 的好处	214
11.1.5 初学 XML	215
11.1.6 开发 XML 环境	215
11.1.7 XML 的应用范围	215
11.1.8 忽略 XML	216
11.1.9 IE 4.0 的支持	216
11.1.10 IE 5.0 的支持	216
11.1.11 ML 之间的关系	217
11.1.12 XML 的重要性	218
11.1.13 压缩 XML	218
11.1.14 XML 安全性	218
11.2 XML 验证	219
11.2.1 DTD 用途	219
11.2.2 DTD 的必要性	219
11.2.3 XML 架构	219
11.2.4 XML 命名	220
11.3 XSLT 和 XPath	220
11.3.1 XSLT 概念	220
11.3.2 XPath 概念	221
11.3.3 XSLT 的重要性	221
11.3.4 XSLT 和 CSS	221
11.4 标准	221

11.4.1 XML 和 W3C.....	221
11.4.2 DOM 和 W3C	222
11.5 工具支持	222
11.5.1 SQL Server 和 ADO	222
11.5.2 BizTalk Server	222
11.5.3 SOAP	223
11.5.4 Windows DNA	223
11.6 具体疑问	223
11.6.1 “Load()” 方法	223
11.6.2 字符实体	224
11.6.3 使用 COM 组件	226
11.6.4 HTML 实体	226
11.6.5 元素中的空白字符	227
11.6.6 属性中的空白字符	230
11.6.7 OM 中的空白字符	230
11.6.8 XML 声明	231
11.6.9 打印 XML	232
11.6.10 DTD 中的命名	233
11.6.11 在 VB 中使用 DSO	234
11.6.12 Java 中的 DOM	235
附录 1 CSS 属性列表	237
附录 2 XHTML 的标记	239
附录 3 XHTML 的属性	241
附录 4 字符实体表	249
附录 5 语言解码	255

第 1 章 XML 概述

1.1 XML 的起源和发展

1.1.1 标记语言

XML 是 eXtensible Markup Language（可扩展标识语言）的简称，和 HTML（HyperText Markup Language）一样，它也是一种“标记”语言。

```
<html>
  <head>
    <title>标记语言例子</title>
  </head>

  <body>
    <h2>这里是二级标题</h2>
    <p>在这里写上段落主体内容，并且查看效果。</p>
  </body>
</html>
```

这是一个简单的标记语言的例子。可以看出，标记语言文件由两个最基本的部分组成，一部分就是“标记（tags）”，它们都被一对尖括号“<>”括起来，如图 1.1 所示，这些内容并不在浏览器上显示。另一部分就是实际的内容，这些内容是自由书写的，在浏览器上显示，但是受到标记的约束。实际内容可以使用各种语言，但是标记必须是英语。

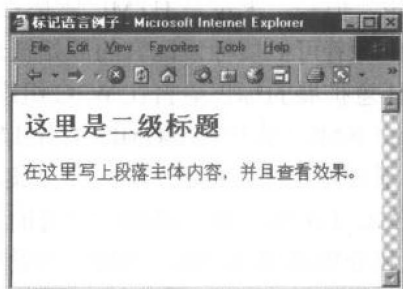


图 1.1 浏览器显示标记语言

HTML 语言出现在 20 世纪 90 年代初,到 1998 年,HTML 4.0 版本的推出将 HTML 语言扩展到一个全新的领域。尽管如此,HTML 语言仍然带有一些前期遗留的缺陷。在早期的网络中,HTML 语言只需要负担一些简单的信息发布功能,例如文字、图像、表格等简单对象。但是,因特网的飞速发展使 HTML 在样式表现、多媒体演示、数据库处理、大型站点控制维护等方面的不足表现得越来越明显。

HTML 的标记部分是由国际标准化组织统一制订的,还有少数标记由各个浏览器生产商自行制订。HTML 标记的数量是非常有限的,W3C 在最新的 HTML 4.01 版本中规定了 90 多个标记元素,目前,网络上常用的 HTML 标记总共达到 100 多个。因为 HTML 的局限性最早反映在样式的表现上,所以 W3C 在 1996 年推出 CSS 1,即“层叠样式表规则 1”。从此,网页编写的过程便一分为二,HTML 负责控制结构,即表格、文字、图像等部分;而 CSS 则负责控制样式,即字体、颜色、间距等。有些功能不足的 HTML 标记逐渐被淘汰,例如、<strike>和<center>等样式控制元素。

1.1.2 推出 XML

HTML 的局限性突出表现在自身标记的数量和功能方面。因此,急需一种功能强大的标记语言。举例来说,如果要显示一段红色的行楷文字,用传统 HTML 可以这样编写:

```
<p><font face="华文行楷" color="red">这里是红色的行楷汉字,使用传统 HTML 方法编写。</font></p>
```

用 CSS 样式表可以这样编写:

```
<p style="font-family: 华文行楷; color: red;">这里是红色的行楷汉字,使用 CSS 样式表方法编写。</p>
```

理想的设计方法是:

```
<HwXingkai><red>这里是红色的行楷汉字,使用最理想的标记状态编写。</red></HwXingkai>
```

显然,最后一种方法最理想,但是,无论是 HTML 还是 CSS 都无法达到这样的要求。

其实最后一种设计方法就包含着可扩展标识语言的原始概念。网络标准的制定者注意到这样的情况,于是,一种可以任意扩展的标识语言在 W3C 组织的带领下应运而生。

eXtensible Markup Language (XML) 是 W3C SGML (World Wide Web Consortium Standard Generalized Markup Language, 通用标记语言标准) 的一个子集,它是为网上传输数据而特别优化定制的。当前的版本是 XML 1.0 第二版。推出它的目的是为了能够在网络上按照现在 HTML 所用的方法利用、接收和处理通用 SGML。XML 的设计宗旨是让网络应用程序的执行更容易,并且增强 SGML 和 HTML 的互动操作性能。



常见问题

提问：什么是 SGML？

回答：SGML 是一种描述标记语言的国际标准，更确切地说，它是专门用来描述标记语言的。它和 HTML、XML 有着密切的联系，因此，理解 SGML 对于理解 HTML 和 XML 有着一定的帮助。

在“<http://www.w3.org/MarkUp/SGML/>”里，可以看到有关 SGML 的大量信息。

1998 年 2 月 10 日，W3C 推出了 XML 1.0 的第一版，经过两年多的发展，2000 年 10 月 6 日推出了第二版。然而，第二版并不是 XML 的全新版本，它只是对第一版的一些错误进行了修正，使读者更容易理解。

1.1.3 XML 简介

依照国际网络标准化组织的说法，“XML 描述了一类数据对象的文档（即 XML 文档），并且部分描述了处理这些文档的计算机程序行为”。读者可以把这个概念简单地理解为 XML 是一个严格符合 SGML 编写格式的，被应用程序所利用的文档。随着学习的深入，读者对于这句话的理解将会逐渐清晰。

XML 文档是由许多“储存单位(storage units)”组成的，它们被称为“字符实体(entities)”，字符实体可以包括解析的和未解析的数据。解析的数据由“字符(characters)”组成，有一部分来自于“字符数据(character data)”，其他则来自于“标记(markup)”。其中，标记部分将文档的存储布局和逻辑结构进行描述编码。XML 提供了加强存储布局和逻辑结构约束的机制。

XML 提供了一种结构化的数据显示，这种结构化的数据功能强大且易于使用。为了让读者对 XML 的基本概念有一个直观的理解，这里举两个生活中的例子。

第一个例子是关于电脑配件采购单的，打开记事本编辑一些 HTML。

```
<h1><font color=red>电脑配件采购单</font></h1>
<p><font color=green>硬盘</font></p>
<p><font color=red>内存</font></p>
<p><font color=red>主板</font></p>
<p><font color=green>显示器</font></p>
```

用浏览器查看这个文件，可以看到 1 个大标题和 4 个小项，假设其中绿色的项目代表已经选购的配件，而红色标识表示还未选购的配件。如果将此表提供给采购员，那么设计者必须在采购前对采购员发出附加指令，“请找出所有在<p>中，的物品，并立即去采购！”

这样说无疑是在浪费时间。因为，现实应用中的这个“采购员”可能是一个搜索引擎，这样搜索会大大增加搜索的时间。那么，有了 XML 会是怎样的情况呢？

```
<stock>
  <title>电脑配件采购单</title>
  <prepared>硬盘、显示器</prepared>
  <unprepared>内存、主板</unprepared>
</stock>
```

在 XML 的世界里, 标记可以任意“发明”, 设计者可以找出一种最适合于描述内容的单词、词组作为标记。任何人要搜索在“<unprepared>”标记里的内存或主板时, 都会找到这个采购单。另外, 如果其他信息内容按照同样的标记形式编写, 设计者也可以告诉其他程序如何处理这些数据。

第二个例子是关于寄送电子邮件的。

```
<e_mail>
  <to>托尼</to>
  <from>珍妮</from>
  <subject>备忘录</subject>
  <body>这个周末可千万别忘了我! </body>
</e_mail>
```

由自定义标记“<e_mail>”可以一目了然地知道这是一个电子邮件的内容。其中的 4 个自定义标记也非常容易理解, 信件由珍妮寄给托尼, 标题为“备忘录”, “<body>”中就是信件的主体内容。

这就是 XML, 通过以上两个例子, 读者应该对 XML 有了一个初步的概念。XML 不像 HTML 一样使用预定义的标记, 而是使用由设计者自行“发明”的标记, 这些标记所标识的内容既能够被常人阅读, 又能够被机器阅读。



注意事项

注意: 必须明确的是, 既然 XML 只是数据的载体, 那么 XML 文档不会去做任何事情, 以上面的例子为例, 这个 XML 本身并不做什么, 它仅仅是包含在 XML 标记中的纯信息。要发送这个电子邮件, 必须另写一个小程序来发送、接收或者显示它。
这方面内容将在以后介绍。

1.2 XML 的编辑工具

1.2.1 通用编辑工具

XML 文件实质上是一个纯文本文件。也就是说, 可以用最简单的 Windows 记事本 (Notepad) 编辑 XML 文件 (“开始” → “程序” → “附件” → “记事本”)。如图 1.2 所示的就是用记事本编写的 XML 文件。

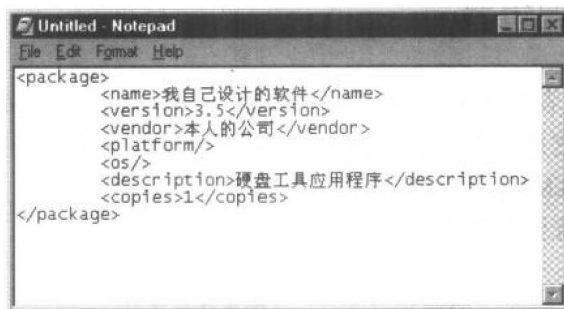


图 1.2 用记事本编辑 XML 文件

这个例子的源代码如下：

```
<package>
<name>我自己设计的软件</name>
<version>3.5</version>
<vendor>本人的公司</vendor>
<platform/>
<os/>
<description>硬盘工具应用程序</description>
<copies>1</copies>
</package>
```

除了 Windows 的记事本，其他通用纯文本编辑器也可以作为 XML 的编辑工具，图 1.3 所示的就是第三方程序 UltraEdit-32 的编辑界面。



图 1.3 Ultraedit-32 的编辑界面

此类编辑器不但可以编辑任意类型的文件，而且有的还能够把不同种类的代码用不同的颜色表示出来，极大地方便了设计者。

1.2.2 专用 XML 编辑器

微软为 XML 的设计者提供了一款免费的 XML 专用编辑器，即 Microsoft XML Notepad。（微软 XML 记事本）。它是一个非常简单易用的编辑工具，个人及小型站点开发者可以利用它快速有效地编写基于 XML 的数据结构。

XML Notepad 的系统要求见表 1.1。

表 1.1 安装 XML Notepad 的系统要求

项目	要求
处理器	至少 486/66MHz 的处理器
操作系统	Windows 95、Windows 98、Windows 2000、Windows NT 4.0，运行 IE 4.01 以上版本的浏览器以及安装了 Service Pack 1 的 Windows NT 4.0。最佳运行浏览器是 IE 5.0，它可以提供一些额外的特性支持
内存	3.8 MB 内存。这是 IE 4.0 所有必需的 DLL 动态链接库和 UI 用户界面组件等一起打开一个空文件所必须的内存需求。XML 数据所需的额外内存需求要视 XML 数据的量而定。一般而言，每 64kB 的 XML 数据文件需要 0.75 MB 的内存。如果运行 IE 5.0 和 XML Notepad，则需要更多的内存
磁盘空间	需要 1 MB 的磁盘空间，用于安装 XML Notepad 文件，0.5 MB 用来运行 XML Notepad

首先到 MSDN 站点“<http://msdn.microsoft.com/xml/notepad/xpsetup.exe>”下载这个大小为 343.5 kB 的可执行文件，双击可以自动解压安装，如图 1.4 所示。

注意：所下载的“xpsetup.exe”仍旧是一个安装文件，它需要从网上下载数据完成安装。因此，在 IE 浏览器提醒用户下载的时候，应该选择“在当前位置打开文件”选项，这样，安装文件下载完毕后，就会自动运行，完成后面的安装。
所有安装文件的大小在 600 kB 左右。

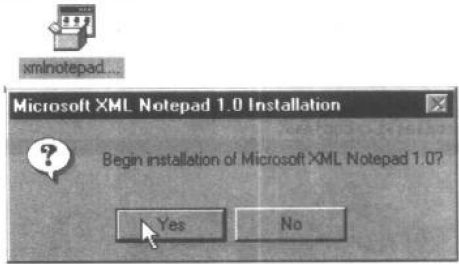


图 1.4 运行 XML 记事本安装程序

安装完毕后，应用程序自动在“开始”菜单中添加“Microsoft XML Notepad”程序项，如图 1.5 所示。