



Internet 新技术丛书

XML 技术大全

*XML:
The Complete
Reference*

(美) Heather Williamson 著

智慧东方工作室 译



机械工业出版社
China Machine Press



Education

Internet新技术丛书

XML技术大全

(美) Heather Williamson 著

智慧东方工作室 译



机械工业出版社
China Machine Press

本书详细介绍 XML 技术, 涉及 XML 的方方面面, 例如解析器、校验器、大纲、文档类型定义、样式表、名字空间、XLink、XPointer 等内容。本书内容翔实, 实例丰富, 不仅介绍 XML 的基本概念, 而且涉及 XML 高级功能, 例如脚本语言的应用、文档对象模型、XML 子语言等。本书既适合 Web 应用开发人员参考, 也适合想学习 XML 语言的技术人员阅读。

Heather Williamson: XML: The Complete Reference (ISBN 0-07-212734-1) .

Copyright © 2001 by the McGraw-Hill Companies, Inc.

Authorized translation from the English language edition published by McGraw-Hill, Inc.

All rights reserved. For sale in the People's Republic of China.

本书中文简体字版由机械工业出版社和美国麦格劳-希尔国际公司合作出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

本书版权登记号: 图字: 01-2001-3947

图书在版编目 (CIP) 数据

XML 技术大全/ (美) 威廉逊 (Williamson, H.) 著; 智慧东方工作室译.-北京: 机械工业出版社, 2002.1

(Internet 新技术丛书)

书名原文: XML: The Complete Reference

ISBN 7-111-09433-6

I .X... II .①威... ②智... 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 (2001) 第 071606 号

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: 宋 宏 张鸿斌

北京牛山世兴印刷厂印刷 · 新华书店北京发行所发行

2002 年 1 月第 1 版第 1 次印刷

787mm×1092 mm 1/16·39.5 印张

印数: 0 001-4000 册

定价: 59.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

前 言

还有什么行业的发展速度比计算机行业快呢？如果你的汽车每年必须更换一次，而今年的汽油和去年的车不兼容，那么如何是好呢？这正是 Internet 开发者经常面临的问题：今年最热门的 Internet 开发语言不能在去年的 Web 浏览器上发挥出完全功效。

但使用 XML，你不必担心自己的文档同下一次、最新的以及最伟大的开发不兼容。只要你有基本的 XML 相容软件，便能看到你的基本 XML 文档。本书将帮助你熟练运用自己的 XML，而软件开发的工作则留给像微软和网景这样的公司去做。

本书面向的读者

本书的读者可以是 Web 迷，希望利用 XML 及其同类语言的高级特性和能力创建自己的 Web 站点，也可以是经验丰富的 Web 开发者，只是想快速参考语言及其用法；也可以是刚开始入门的 XML 开发者，需要解决在自己学习过程中遇到的难题。

但是，不管你属于哪类人，都应该掌握下述技能：

- 知道以 SGML 为基础的语言（如 HTML）的基本语法。
- 知道创建一个 HTML 页的基本过程。
- 对于使用浏览器、链接、URL 等时涉及到的基本概念有一定程度的了解。
- 理解 SGML 文档的元素-属性结构。

本书的读者不必理解、甚至不必知道任何 XML 子语言，但对于本书的内容，应该有一定程度的兴趣。

本书的内容

当前，XML 已成为 Internet 上最热门的主题之一，它成为程序员必须掌握的一种技能。本书的目标是向你提供掌握 XML 所需的“一切东西”。从编写你的第一个 XML 文档时涉及到的基本概念，一直到 XML 领域内的一些高深主题，包括解析器、校验器、大纲、文档类型定义（DTD）、样式表（CSS 和 XSL）、名字空间、XLink、XPointer、Xbase 和 XML Query 等等。除此以外，书中还覆盖了 XML 的大量支持标准——资源描述框架（RDF）、XHTML、同步标记语言（SYMM）（如 SMIL）以及可扩展矢量图形（SVG）格式等等。

本书首先探讨 XML 的基础，比如创建简单文档，以便由当前的 XML 解析器正确进行解析。在你学习为 XML 文档创建 DTD，以及对文档及其 DTD 进行校验的时候，这些信息会随着阅读的深入而不断扩展。之后，随着你从简单的层叠样式表（CSS）过渡到更复杂的扩展样式表语言（XSL）和 XSL 转换（XSLT），样式表会不断地为文档添加更多的装饰和功能。在你用样式表完成了一个 XML 文档之后，将学习 XML 大纲、XHTML、XLink、XPointer 和 RDF。还将学习使用脚本（JavaScript 和 Perl）对一个 XML 文档中包含的信息进行处理，并操作同文档相关的文本文件数据库，从而探索文档对象模型（DOM）。最后，学习流行的、使用非常广泛的 XML 子语言，比如 MathML 和 SMIL。下面对各章和各部分进行介绍。

第一部分 XML 基础

第一部分讲解了 XML 文档的基本结构。学习如何标识 XML 元素和属性，探索对 XML 文档进行校验的用意以及了解一些用来进行校验的软件。有专门一章讲述 XML 名字空间，以及如何在文档中将它们用于校验目的。

第 1 章对 XML 的所有相关技术和要求进行了常规性的概述。你可在此了解该语言的历史、与当前标准配合的软件以及与未来标准相容的软件。除此以外，该章还解释了 XML 应用程序的要求、它们能做什么以及它们能对你的开发项目起到什么帮助。

第 2 章讨论了与 XML 协同使用的各种软件，包括解析器、编辑器和校验器。

第 3 章解释了进行正确信息解析的需求，以及如何随你的文档协同使用校验软件。在此你将开始构建一个简单的 XML 文档，学习如何保存、查看和添加注释，以及如何查实它是否已正确解析。

第 4 章探讨了如何组织你的数据，以便最终能顺利转换成 XML 应用程序。该章全面探讨了 DOM，让你知道用户在计划对数据进行格式化时打算创建的是什么。另外，你还会了解样式表如何影响你的数据展示，并知道何时该将数据用作属性，何时将数据用作元素。

第 5 章讨论了同 XML 元素关联在一起的规则，如何开始和结束标记，如何标识元素及其子元素。另外，它还解释了运用空标记和非空标记和 `<?xml>` 语句所需的一切信息，另外还探讨了根元素的重要性。

第 6 章探讨了 XML 属性特定的实现方式，并揭示了属性同元素的关系，以及应该在文档内的什么地方使用属性。

第 7 章探讨了基本的 XML 名字空间概念和用途，并解释了用于引用名字空间的语法。

第二部分 DTD 的运用

文档类型定义 (DTD) 提供了对 XML 文档的配置进行控制的指示。在该部分的 4 章中，介绍如何使用 DTD 来定义 XML 元素和属性，以及如何在 XML 文档内部使用实体和记号。

第 8 章探讨了嵌入和外部 DTD，其中包括特定的校验软件、何谓校验以及为何需要它。还探讨定义 DTD 语句和使用公共 DTD 的一般规则。

第 9 章强调了 XML 实体的标识，以及如何使用常规和参数实体。你将在在此探索如何使用来自外部文档的一个实体引用在 XML 文档中创建一个实体引用。

第 10 章对 DTD 中的属性进行了探讨，并解释了如何标识它们及其默认值。讨论的其他主题包括属性数据类型、预定义属性以及如何为一个元素同时声明多项属性。

第 11 章探讨了如何在 DTD 中添加注释、标记和未解析实体。还强调了如何为数据使用处理指令，并对 DTD 中的条件语句进行了讨论。

第三部分 添加样式

该部分的重点在于样式表在 XML 文档中的使用和实现，包括层叠样式表及可扩展样式表语言。

第 12 章解释了在什么时候、什么地方使用样式表，其中包括何时该用 CSS，何时该用 XSL。

第 13 章探讨了如何在 XML 文档中使用外部样式表, 其中包括它们在 XML 文档中的创建与保存。如何用 CSS 来控制一个文档的基本外观。其中涉及字体、背景、边框、页边距、填充和其他标准外观设置的运用。

第 14 章探讨了用于控制可视性、溢出、索引、对象大小和对象显示样式的 CSS 属性。该章讨论了听觉样式表属性——包括 first-line 和 first-letter 伪类, !important 的使用, 以及对列表元素的外观进行控制的属性。

第 15 章简要地介绍了 XSL, 解释了 XSL 和 XSLT 的用途, 以及用于创建、查看和校验 XSL 和 XSLT 文档的软件。

第 16 章探讨了 XSLT 语言, 详尽解释了什么是 XSLT、如何工作以及它的原理。在此还有对 XPath 的进一步讨论。

第 17 章探讨了 XML 内容的格式化概念, 解释了如何使用附录 C 总结的 XSL-FO 元素。

第四部分 使用大纲

在这一部分, 我们将讨论 XML 大纲的问题, 说明大纲规范当前的状态。因为 XML 大纲这个主题得用整本书来讨论, 所以这里肯定无法面面俱到。但是, 刚开始编写和开发自己的大纲时, 你需要掌握的一切信息都可在这里找到。

第 18 章探讨了 XML 大纲以及 XML 大纲和 XML 名字空间的差别, 其中包括理解 XML 大纲所需的信息。同时提供了基本大纲文档和支持软件的例子。

第 19 章对 XML 大纲元素的使用进行了扩展, 展示了如何定义 complexType 和 simpleType 元素。还探讨了如何在你的文档中对元素进行分组。

第 20 章进一步扩展了 XML 大纲属性的使用, 其中包括标识属性、创建属性以及一些简单的数据类型。

第 21 章对其他大纲章节中没包括的一些信息进行了讨论。讨论了如何使用目标名字空间, 如何设置大纲的限制, 本地与全局变量的使用, 以及其他同大纲相关的主题。

第五部分 使用 XML 查询

由于 XML 本身在不断地发展, 所以你随时都可对文档中包含的信息进行扩充, 并可改变获取信息的方式。第五部分探讨了许多同 XML 相关的“热门话题”。在 XML 文档的世界中, 这些技术正变得越来越成熟。

第 22 章讨论了 XML 查询语言, 其中包括语言的设计用途, 它的使用方式, 以及 XML Query 工作草案的最新状态。并对 2001 年 2 月起的每一份 XML Query 工作草案进行了讨论。

第 23 章提供了对 XML 链接语言的常规性介绍。解释了 XLink 的结构, 以及如何使用和创建 XLink 语句。该章还探讨了如何为你的链接元素(例如属性)创建一个 DTD, 以及定义链接中使用的本地和远程资源。

第 24 章对 XML Pointer 语言进行了常规性介绍, 解释了为什么要使用 XPointer, 以及它们的结构是什么。本章还提供了 XPointer 的应用实例。

第 25 章讨论了元数据在 XML 文档中的使用, 并探讨了所用的框架、为什么要用它, 并提供了一些 RDF 的使用实例。

第六部分 实践 XML

如果不经常应用，那么 XML 对你而言也没有多大的用处。该部分将指导你把 XML 投入实用。

第 26 章探讨了如何从 HTML 转换成 XHTML，并展示了一些例子，解释具体的转换过程，并深入讨论 XHTML 的模块化特性。

第 27 章解释了如何使用 DOM 和 JavaScript / JScript 来修改 XML 文档中的信息的外观和可视性。

第 28 章提供了一个加上详细注释的例子，解释如何使用 CGI/Perl 从一个文本文件数据库中收集信息，以及如何将其转换成一个 XML 文档。

第七部分 XML 子语言

随着 XML 的流行，而且由于它在大量领域都易于应用，所以 XML 派生出了大量的子语言。从第 29 章到第 31 章，我们主要讨论了 XML 子语言的三大类别。

第 29 章探讨了 SMIL 的基本用法及其语法，以及用基于 XML 的语言来创建通行同步多媒体文档的目的。它还讨论了适用于阅读和查看这些文档的软件。

第 30 章探讨了 SVG 的基本用法和语法，以及用基于 XML 的语言创建矢量图形，以便在网络系统中使用。

第 31 章探讨了 MathML 和 CML 的基本用法和语法，其中包括如何使用基于 XML 的语言在一系列系统中共享科学数据。XML 使 MathML 和 CML 能够标识科学领域使用的各种特殊符号。

附录

本书在附录中提供了大量材料，其中包括 CSS 和 XSL 参考，以及编写一个标准 XML DTD 的要点。这些附录采用易于阅读的表格或列表形式，以便读者快速和轻松地参考信息。附录 A 提供了 DTD 关键代码和一个示范 DTD。附录 B 是一份完整的 CSS 语言参考，附录 C 是针对格式化元素的 XSL 格式化对象参考，而附录 D 提供了 Unicode 字符集，除了提供 MathML 和 CML 参考之外，还对其他 UNICODE 语言进行了说明。

获取最新信息

与 XML 有关的信息经常会发生改变，所以你需要经常访问万维网协会的网站，地址是 <http://www.w3.org>，以查找许多 XML 相关规范的当前状态。另外还有大量与 XML 相关的信息站点，比如：

- <http://www.xmlhack.com>
- <http://www.xml.com>
- <http://www.xslt.com>
- <http://www.internet.com>
- <http://www.xmlinfo.com>
- <http://www.xslinfo.com>

■ <http://www.microsoft.com/xml>

■ <http://www.schema.net>

■ <http://www.xmlsoftware.com>

本书英文版原书书名: XML: The Complete Reference

英文版原书书号: ISBN 0-07-212734-1

英文版出版社网址: www.osborne.com

目 录

前 言

第一部分 XML 基础

第 1 章 XML 综述	1
1.1 早期岁月	1
1.1.1 SGML 基础	1
1.1.2 XML 的要求	2
1.2 当前的标准	5
1.2.1 XML 1.0	6
1.2.2 XML 名字空间	6
1.2.3 文档类型定义	6
1.2.4 层叠样式表	6
1.2.5 可扩展样式表语言	7
1.2.6 XML 大纲	7
1.2.7 XML 查询语言	8
1.2.8 XLink	8
1.2.9 XPath	8
1.2.10 XPath	9
1.2.11 XML 数字签名	9
1.2.12 资源描述框架大纲	9
1.2.13 正规 XML	9
1.2.14 XHTML	10
1.3 XML 软件	10
1.3.1 非校验解析器	10
1.3.2 校验解析器	11
1.4 DOM	12
1.5 小结	12
第 2 章 XML 校验和非校验解析器	13
2.1 开始编写 XML 文档	13
2.2 XML 非校验解析器	16
2.2.1 从一个基本的解析器开始	16
2.2.2 在线解析器	17
2.3 XML 校验解析器	18
2.3.1 校验的需求	19

2.3.2 在线校验器	19
2.3.3 大纲检查器	20
2.4 小结	20
第 3 章 XML 初步	21
3.1 XML 结构	21
3.2 XML 元素	23
3.3 开始写第一个文档	24
3.4 解析文档	24
3.5 使用元素层	25
3.6 给 XML 代码加上注释	26
3.7 数据结构和组织	27
3.8 小结	28
第 4 章 XML 数据组织	29
4.1 根据数据创建信息层	29
4.1.1 标识信息的重要性	29
4.1.2 为列表添加细节	30
4.2 对数据进行结构化	31
4.2.1 标识出元素	31
4.2.2 标识出属性	34
4.3 保证数据兼容于文档对象模型	35
4.4 小结	35
第 5 章 创建良构的 XML	36
5.1 文档基础	36
5.1.1 <? xml>声明	36
5.1.2 根元素	38
5.2 创建和描述元素	38
5.2.1 开始和结束元素标记	38
5.2.2 元素内容	39
5.2.3 空元素标记	40
5.2.4 为元素命名	40
5.3 子元素	42
5.4 小结	43
第 6 章 增添属性	44
6.1 用属性增强元素	47

6.2 属性规则	49	9.2.1 内部常规实体	76
6.2.1 属性的命名	50	9.2.2 在实体内使用实体	78
6.2.2 属性值	50	9.2.3 常规实体限制	78
6.3 用属性改进空元素	51	9.2.4 组装	78
6.4 属性的共享	51	9.2.5 外部常规实体	79
6.5 样式表和属性配合使用	51	9.3 定义参数实体	83
6.5.1 添加一个样式表	53	9.3.1 内部参数实体	83
6.5.2 一个快速层叠样式表	53	9.3.2 外部参数实体	85
6.5.3 一个快速 XSL 样式表	54	9.4 小结	86
6.6 小结	55	第 10 章 运用属性	88
第 7 章 XML 名字空间	56	10.1 探究属性	88
7.1 什么是名字空间	56	10.2 属性还是元素	89
7.2 在文档内部使用名字空间	57	10.3 属性的定义	89
7.2.1 标识名字空间	57	10.3.1 控制属性类型	90
7.2.2 语法	59	10.3.2 设置默认值	94
7.2.3 默认名字空间	59	10.4 定义多个属性	96
7.3 将名字空间添加到 DTD	60	10.5 使用预定义的属性	99
7.4 小结	62	10.5.1 xml:space	99
		10.5.2 xml:lang	100
		10.6 小结	100
		第 11 章 添加其他数据	101
		11.1 探究标记	101
		11.2 未解析的外部实体	103
		11.3 条件语句	105
		11.4 添加处理指令	107
		11.5 小结	108
		第三部分 添加样式	
		第 12 章 何时使用样式表	111
		12.1 什么是样式表	111
		12.2 CSS 的优点	111
		12.3 XSL 的优点	114
		12.4 如何选择样式表	116
		12.5 小结	117
		第 13 章 CSS 基础	118
		13.1 基本 CSS 语句	118
		13.2 把 CSS 添加到文档	119
		13.3 CSS 选择	119
		13.3.1 选择元素	120
第二部分 DTD 的运用			
第 8 章 XML 文档校验	63		
8.1 DTD 和校验	63		
8.2 将 DTD 添加到文档	64		
8.2.1 链接到文档类型定义	64		
8.2.2 嵌入文档类型定义	65		
8.3 PUBLIC 和 SYSTEM DTD	66		
8.3.1 引用 SYSTEM DTD	66		
8.3.2 引用 PUBLIC DTD	67		
8.4 声明 Element 类型	67		
8.5 控制元素内容	70		
8.6 添加注释	73		
8.7 小结	73		
第 9 章 定义 DTD 实体	74		
9.1 理解实体	74		
9.1.1 内部实体	75		
9.1.2 外部实体	75		
9.1.3 常规实体	75		
9.1.4 参数实体	76		
9.2 定义常规实体	76		

13.3.2 选择类	121	14.2.1 使用:first-child	152
13.3.3 选择 ID	121	14.2.2 使用:first-line	152
13.3.4 根据情况选择	122	14.2.3 使用:first-letter	154
13.3.5 使用伪元素	122	14.3 对象的可视格式化	155
13.3.6 使用伪类	123	14.3.1 display 属性	156
13.4 字体控制	124	14.3.2 设置对象位置	159
13.4.1 设置字体族科	124	14.3.3 设置对象大小	163
13.4.2 设置字体大小	125	14.3.4 控制对象的可视性	164
13.4.3 设置字体伸缩	125	14.3.5 控制对象的 overflow 和 clip 属性	164
13.4.4 设置字体样式	126	14.4 控制光标	165
13.4.5 设置字体变型	126	14.5 设置对象的重要性	166
13.4.6 设置字体粗细度	126	14.6 使用媒体类型	166
13.4.7 设置快捷字体	127	14.6.1 使用@media	166
13.4.8 设置字体颜色	128	14.6.2 使用@import	167
13.5 文本控制	128	14.7 了解听觉样式表	168
13.5.1 设置字母字距	128	14.8 小结	170
13.5.2 设置行高	129	第 15 章 XSL 基础	171
13.5.3 设置文本对齐方式	129	15.1 背景	171
13.5.4 设置单词间距	130	15.1.1 对 XSL 的需要	171
13.5.5 设置自己的对齐方式	130	15.1.2 XSL 的历史和现状	172
13.5.6 文字缩排	131	15.2 XSL 基础	174
13.5.7 使用文本装饰	131	15.2.1 通过转换定义样式: XSLT	174
13.5.8 使用文本阴影	131	15.2.2 通过 XML 实现 CSS: XSL-FO	175
13.5.9 使用文字变形	132	15.3 XSL 软件和 XSLT 软件	178
13.6 对象框控制	132	15.3.1 编辑 / 创作工具	178
13.6.1 设置边框	132	15.3.2 处理器	179
13.6.2 设置边框折叠	134	15.4 XSL-FO 软件	181
13.6.3 设置轮廓	135	15.4.1 固有 XSL-FO 表现器	182
13.6.4 设置边距	137	15.4.2 格式化对象到 PDF 转换器	183
13.6.5 添加填充	138	15.5 小结	184
13.6.6 设置背景	139	第 16 章 XSL 转换	185
13.7 组装样式表	143	16.1 XSLT 的概念	185
13.8 小结	146	16.1.1 源和结果树	185
第 14 章 CSS: 应用	148	16.1.2 节点和节点类型	187
14.1 添加内容	148	16.2 XPath 标准	188
14.1.1 添加:before	148	16.2.1 环境	190
14.1.2 添加:after	149	16.2.2 位置步进语法	191
14.1.3 创建计数器和自动编号	151	16.2.3 XPath 位置路径示例	195
14.2 使用 CSS 伪类和伪元素	152		

16.2.4 节点值	196	18.4.1 大纲编辑器	293
16.2.5 XPath 函数	196	18.4.2 大纲校验器和解析器	295
16.3 用 XSLT 转换 XML	202	18.5 小结	296
16.3.1 样式表结构	202	第 19 章 大纲的元素、类型和组	297
16.3.2 模板规则	208	19.1 大纲元素说明	297
16.3.3 条件处理	219	19.2 元素类型	300
16.3.4 排序	221	19.2.1 复杂类型	300
16.3.5 在结果树中生成内容	224	19.2.2 简单类型	302
16.3.6 变量和参数	229	19.2.3 定义自己的简单类型	304
16.3.7 导入和包容样式表	236	19.3 使用元素组	306
16.3.8 特殊结果树输出	239	19.4 元素内容	306
16.3.9 将节点从源复制到结果树	241	19.4.1 属性和简单值	306
16.3.10 模板规则模式	244	19.4.2 混合内容	307
16.3.11 从样式表发出消息	246	19.4.3 空内容	307
16.3.12 XSLT 函数	248	19.5 小结	308
16.4 小结	267	第 20 章 定义大纲的属性	309
第 17 章 XSL 应用	268	20.1 大纲属性说明	309
17.1 XSL-FO 文档结构	268	20.2 属性组	310
17.2 布局主控集: 页面布局和序列	269	20.3 使用大纲注解	311
17.2.1 页面布局	269	20.4 使用约束	312
17.2.2 页面序列	272	20.4.1 规定一个<choice>	312
17.3 添加内容: 页面序列	273	20.4.2 创建一个<sequence>	313
17.3.1 块和内联格式化	275	20.4.3 使用<all>	314
17.3.2 一个页序列中的固定内容	275	20.5 创建惟一元素和属性	314
17.3.3 页和页之间不同的内容	277	20.6 大纲和名字空间	315
17.4 查看结果文档	277	20.7 使用实体	316
17.5 修改区域的显示属性	279	20.8 小结	317
17.6 插入图像	280	第 21 章 大纲的高级概念	318
17.7 XSL-FO 表	280	21.1 目标名字空间和大纲	318
17.8 小结	286	21.2 未声明的目标名字空间	320
		21.3 大纲约束与 ID 属性	320
		21.4 全局和局部声明	321
		21.5 将大纲当作多个文档使用	322
		21.6 替换组	323
		21.7 抽象元素和类型	324
		21.8 小结	324
		第五部分 使用 XML 查询	
		第 22 章 XML 查询入门	325
第四部分 使用大纲			
第 18 章 大纲入门	287		
18.1 什么是大纲	287		
18.2 DTD 和大纲的比较	288		
18.3 写一个基本的大纲	290		
18.3.1 一个分级的大纲	290		
18.3.2 大纲元素的平面目录	292		
18.4 目前可用的软件	293		

22.1 XML 查询要求	325
22.2 XML 查询数据模型	326
22.3 XML 查询代数	326
22.3.1 数据和类型	328
22.3.2 映射	329
22.3.3 原子数据	330
22.3.4 反复	330
22.3.5 选择	330
22.3.6 限定	331
22.3.7 联结	332
22.3.8 排序	332
22.4 利用 XML 查询	333
22.5 小结	334
第 23 章 XLink	336
23.1 理解 XLink	336
23.2 XLink 语法	337
23.2.1 XLink 名字空间	337
23.2.2 XLink 属性	338
23.2.3 链接类型	341
23.3 Xbase 概览	347
23.4 小结	348
第 24 章 XPointer 的使用	349
24.1 理解 XPointer	349
24.2 XPointer 大纲	349
24.3 XPointer 语法	350
24.3.1 完全形式	351
24.3.2 裸名	351
24.3.3 子序列	352
24.4 Xpointer 转义	352
24.4.1 转义 XPointer 特殊字符	353
24.4.2 转义 URI 中的字符	353
24.4.3 转义在 XML 文本中使用的字符	353
24.5 XPointer 函数	354
24.5.1 标识一个 location-set	354
24.5.2 点位置类型	355
24.5.3 范围位置类型	355
24.5.4 string-range 函数	355
24.5.5 range-to 函数	356
24.5.6 here 函数	357

24.5.7 origin 函数	357
24.5.8 start-point 函数	357
24.5.9 end-point 函数	358
24.6 小结	358
第 25 章 资源描述框架	360
25.1 什么是 RDF 和 RDF 大纲	360
25.1.1 RDF	360
25.1.2 RDF 大纲	361
25.2 实现 RDF 基本模型	361
25.2.1 资源	361
25.2.2 属性	361
25.2.3 语句	362
25.2.4 RDF 基本语法	362
25.3 使用 RDF 容器	364
25.3.1 RDF 容器语法	364
25.3.2 容器对象	365
25.3.3 语句	366
25.3.4 URI 范式	367
25.4 小结	368

第六部分 实践 XML

第 26 章 XHTML	369
26.1 什么是 XHTML	369
26.2 HTML 到 XHTML 的转换	369
26.2.1 元素转换	370
26.2.2 从 HTML 到 XHTML 的快速转换	371
26.3 未来的 XHTML 模块	372
26.3.1 结构模块	373
26.3.2 文本模块	373
26.3.3 超文本模块	375
26.3.4 列表模块	375
26.3.5 展示模块	376
26.3.6 编辑模块	377
26.3.7 双向文本模块	378
26.3.8 表单模块	378
26.3.9 表格模块	379
26.3.10 图像模块	381
26.3.11 客户端图像地图模块	381
26.4 小结	381

第 27 章 随 JavaScript 来实现 XML	383
27.1 过程	383
27.1.1 XML 文档	383
27.1.2 大纲	384
27.1.3 样式表	385
27.1.4 编写脚本	386
27.2 组装	387
27.3 小结	391
第 28 章 用 CGI 收集和写入数据	392
28.1 设置变量和打开脚本	392
28.2 文本文件数据库的格式	393
28.3 创建 XML 文件	393
28.4 打开数据库和开始循环	394
28.5 读入数据	394
28.6 写入数据	394
28.7 保存 XML 文件并结束程序	395
28.8 完整的脚本	395
28.9 小结	397

第七部分 XML 子语言

第 29 章 SYMM 和 SMIL	399
29.1 SYMM 基础	399
29.2 理解 SMIL	399
29.3 SMIL 编辑器和播放机	400
29.4 SMIL 和 HTML	401
29.5 SMIL 媒体元素	401
29.5.1 未知媒体类型	402
29.5.2 SMIL 媒体属性	403
29.6 SMIL 布局和格式化元素	404
29.6.1 设置演示文稿: <smil>、 <head>和<body>	404
29.6.2 使用元信息	405
29.6.3 使用<layout>	407
29.6.4 使用<root-layout>	408
29.6.5 使用<region>	408

29.6.6 使用<viewport>	409
29.7 设置 SMIL 计时	412
29.7.1 站点同步	412
29.7.2 计时和同步属性	414
29.8 对象和演示文稿之间的链接	414
29.8.1 使用<a>	414
29.8.2 使用<area>	415
29.8.3 链接属性	415
29.9 创建 SMIL 动画和变形	416
29.9.1 动画效果	416
29.9.2 变形效果	417
29.10 小结	418
第 30 章 使用可扩展矢量图形	419
30.1 SVG 入门	419
30.2 SVG 语法	420
30.2.1 SVG 数据类型	421
30.2.2 SVG 元素	422
30.2.3 SVG 属性	430
30.3 一个 SVG 的例子	435
30.4 小结	437
第 31 章 科学语言: MathML 和 CML	438
31.1 数学语言: MathML	438
31.1.1 MathML 元素	438
31.1.2 MathML 示例	452
31.2 化学语言: CML	453
31.2.1 CML 元素	453
31.2.2 CML 示例	454
31.3 小结	460

附 录

附录 A DTD 关键代码和一个示范 DTD	461
附录 B CSS 语言参考	480
附录 C XSL 格式化对象参考	526
附录 D UNICODE 字符集	567

第一部分 XML 基础

第 1 章 XML 综 述

可扩展标记语言 (XML) 只是一系列规则的集合。这一系列规则说明了如何定义标记 (文本标记), 这些标记可以将文档分为多个部分和子部分。XML 是类似 HTML 的标记语言, 但和 HTML 不同的是, 根据 XML 的定义, 它并不限于一个特定的词汇表、行业或者用途。XML 是集互联网精华于一身的扩展。针对不同的公司, 或者自己希望的一种目的, 可对它进行完全的定制。

XML 的标记规则基于一种早期的标记语言, 即标准通用标记语言 (Standard Generalized Markup Language, SGML)。XML 采用 SGML 做为基础, 并借鉴了一些 HTML (另一种基于 SGML 的语言) 的知识, 创建了存储结构化信息的文本文件。不要着急, 使用 XML 比使用 SGML 更为简单, 而且 XML 允许我们 (文档创建者) 进行自由的创建。XML 非常易于使用、易于由计算机进行读取 (对人来说也不难)、易于调试, 而且易于创建适用于各行业 (这些行业应使用结构化数据, 例如电子表格、数据库、财务数据和技术图形) 的可扩展标记语言。

1.1 早期岁月

自从人类在世界上出现后, 就开始创造解决困难的方法。亚历山大·格雷姆·贝尔发明第一部电话时, 他试图解决那个时代的通信难题。现今, 我们有非常专业的技术开发语言 (例如 XML), 可以和任何地点的任何人一起共享和显示信息, 无论他们正在使用什么软件、无论他们是连接到互联网还是在办公厅交换磁盘。这就是 XML 的功能。最棒的是, XML 依靠自身强壮的 SGML 基础, 实现了标识文档碎片的一系列需求, 以便该文档可以进行共享, 或部分共享 (任何地点、任何人, 采用任何软件), 无论用户是连接到网上, 还是简单地从事公司项目数据库文件格式的修改。

1.1.1 SGML 基础

XML 的基础是标准通用标记语言。开发 SGML 是为了提供标识文档部分和文档内容的方法。该方法并非通过实际的内容或行号进行标识, 而是通过文档包含的信息类型进行标识。例如, 我们可以使用 SGML 标记文档, 并搜索一个字符串 (例如 “h1”), 以便标识出一级标题, 无论该标题的内容是什么。我们随后可以采用脚本, 通过寻找 <h1> 文本标记, 找出所有的一级标题, 以便将它们显示出来, 或者复制到另一个文档形成内容列表, 也可以挑选出所有不是一级标题的文本。

SGML 是由通用标记语言 (General Markup Language, GML) 演化发展而来的, GML 是 IBM 在 20 世纪 60 年代末开发的语言。GML 在 1986 年成为标识文档标准之前, 只有很

少的大组织、教育学院和政府办公室采用该语言。SGML 被采纳为官方标准的巨大推动力，来自于美国国税局和美国国防部对它的官方兼并。在这两个组织接收该标准之后，他们要求所有的承包商也要采用该标准，SGML 获得了它成为标准的强大推动力。可以通过网络共享文件的概念出现后，互联网文档就开始进行酝酿，开发者将 SGML 做为创建 HTML 的模型。此后伴随着互联网的高速发展，SGML 在它最新的“孩子”XML 上获得了新生。

HTML 与 SGML 和 XML 不同，它控制信息的表达。HTML 是严谨的格式语言。SGML 和 XML 都是一系列规则，这些规则用于控制标记语言的创建，标记语言可以标识文档内容，以便全书的信息可以保持格式一致。SGML 和 XML 都适用于含有大量相似组织信息的文档，例如目录、地址簿、数学功能和账目记录。SGML 和 XML 也非常适用于数据库结构中信息的显示。

SGML 和 XML 最好的地方在于，它们允许设计团队标识出可用于所有文档的结构。想一想本书，最初的文档是用 Microsoft Word 编写的，所有的小节标题、页眉、页脚和定义的段落文本或字体，都采用了 Word 的样式。如果本书由 SGML 或 XML 编写，那么这些语言的标记将用于创建页眉、页脚、段落、代码段、图形说明、图形编号和其他文档结构的使用。我们可能要问，如果 Word 程序已经完成了这些功能，为什么我们还要对 XML 感兴趣？答案很简单：Word 程序不能共享样式信息。例如，无论是 Corel 公司的 WordPerfect，还是 Sun 公司的 StarOffice 字处理软件，都不能立即读取我们在 Microsoft Word 中创建的样式。该文档必须经过转换过程，以便其他类型的软件可以读取这些信息，而且文档样式的转换也并非总是很准确。SGML 和 XML 允许该文档由任何类型的软件进行读取，文档的作者可以指定文档的内容部分如何由所有的程序进行解释。然而，由于 XML 只描述文档的内容，如果我们希望安排超越正规文本的格式，就需要提供一个样式表。两种主要用于 XML 的样式表语言是层叠样式表 (Cascading Style Sheets, CCS) 和可扩展转换样式表 (Extensible Style Sheet Transformations, XSLT)，这两者都将在本章后面进行讨论。

1.1.2 XML 的要求

创建 XML 词汇表和文档结构时，为了创建正确的文档，我们有两项任务（关于校验 XML 文档的更多信息，请参见第 5 章）。首先，我们需要标识出所有的信息类型，并标识放入文档中的内容。无论是希望进行文档校验，还是进行简单的解析，我们都需要完成任务，以便浏览器可以对文档进行浏览。其次，对于需要在文档中进行标识的信息，我们要提供一个结构定义，该结构可以将这些信息包起来。如果没有用文档定义（文档类型定义）描述 XML 文档中的内容结构，它就不能进行校验。由于可以创建自己的内容标识（标记）和内容描述，可以采用 XML 描述信息的任何类型，这些类型可以归入特定的内容类型。例如，我们可以简单使用 XML 跟踪全美牛仔竞技总决赛的结果，就像我们可以标识出复杂数学方程式的所有部分一样。我们可以通过识别内容类型和描述内容，简单地提供标识标记结构。

HTML 是第一种采用标识语言安排文档格式，并得到广泛运用的语言，因此它可以由任何人读出，但 HTML 有一个主要的弱点：它是格式语言，并不注重文档的结构或内容。这意味着文档设计者可以使用各种 HTML 标记，以便控制他们文档的外观，但他们没有方法控制文档的物理结构。该系统可以创建专业的文档，但不能帮我们创建格式规范的、基

于 XML 规则的文档，也不能创建含有常用可标识部分的文档。

层叠样式表的发展是从内容控制向格式化产生分化的第一步。样式表允许所有指定类型的标记按指定的方式进行格式化。样式表同样允许对含有指定标识符的标记按指定的方式进行格式化。这是在 HTML 文档标题和段落周围去掉成组 HTML 标记<P>的第一步。我们可以采用样式表对信息进行格式化，如果效果不好，可以采用 XML 进行格式化（就像采用 HTML 一样）。这在很大程度上要归功于 XML 严谨的核心内容。参见第 13 章和第 14 章中关于层叠样式表的更多信息。

对于 XML 实现正确标识文档内容的工作，我们应该了解它是如何工作的。我们需要了解如何进行信息标记，以便其他软件或样式表可以解释文档的结构，并对这些信息进行格式化。

1. 结构描述

XML 的标记描述了文档的结构，也标识了文档的内容。XML 标记（类似 HTML）采用单括号“<”和“>”表示开始和结束。括号间是内容的识别名称或内容的标记。在开始括弧后，我们将放置由 XML 标记的实际内容。最后，XML 需要有一个结束的标记，该标记与开始标记（XML 是指定的情况）共用相同的名称，除了开始单括号后附加了一个反斜线：“</”。

```
<TAGNAME>内容</TAGNAME>
```

这些标记包括属性，这些属性提供了标记内容或者标记本身的附加信息。在 name = "value"的形式中，这些属性符合名 / 值对的设置。属性位于标记中，如下面代码片断所示：

```
<TAGNAME 属性名称 = "属性值"> 内容 </TAGNAME>
```

XML 仅仅是格式类似于 HTML 的样式，不要愚蠢的认为它的规则像 HTML 一样不明确。如果曾经编过 HTML，就会知道即使在大多数属性周围忘掉了引用标记，文档也可以加载到浏览器中。甚至在元素后面，我们也可以忘掉添加结束标记，例如<P>和
。XML 要严格得多。各属性值必须装入引用标记中，每一个开始标记必须拥有一个结束标记，除非它是一个空标记，我们可以在第 5 章读到更多的相关内容。读取 XML 文档的应用程序不允许忽略错误，或者尝试预测文档作者的意图；它们必须停止在文档错误处，并将错误信息发给文档的察看者。这使得 XML 相关文档容易进行跟踪，因为在察看文档时，我们准确知道是哪一行导致该故障。这些规则涉及文档如何设标记，以便控制文档的良构性。根据 XML 软件，如果文档符合 XML 规格中所有标准的验证，该文档便是良构的文档。我们可以在第 5 章找到更多有关良构文档的信息。

如果在类似 IE 5 的互联网浏览器上察看 XML 文档，我们将发现文档没有执行格式化。所有信息以大纲结构显示（参见图 1-1）。软件内嵌的控制允许我们打开和关闭数据段，但它们并不格式化信息。

当用互联网浏览器或其他文档阅读器察看 XML 文档时，XML 中没有固有信息或子语言控制信息的格式化。采用 XML（与 HTML 不同），除非格式化控制已经内嵌到察看文档的软件，样式表控制所有的格式化，类似于采用同步多媒体综合语言（Synchronize Multimedia Integration Language, SMIL）的实际播放器，例如 Real Network RealPlayer 8。