XML

基础教程

（第二版）

XML A Prime

[美] Simon St. Laurent 著
康晓林 伊希荣 陈维义 等译
林义雄 审校

IDG BOOKS WORLDWIDE
M&T BOOKS
PHEI
電子工業出版社
Publishing House Of Electronics Industry
URL:http://www.phei.com.cn

XML 基础教程

(第二版)

XML: A Primer, 2nd Edition

[美] Simon St. Laurent 著

康晓林 伊希荣 陈维义 等译

林义雄 审校

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

可扩展标记语言(XML)是国际互联网联盟(W3C)开发的一门用于网页设计和数据交换的新技术,具有良好的应用和发展前景。本书介绍了 XML 以及如何运用 XML 提供的信息管理和网页格式化的强大功能来创建站点和网页。内容包括"所见即所得"存在的问题、级联样式表、建造基本的 XML 文档、规划文档设计、文档类型定义及高级 XML 语法、用 XML 重建传统 Web 站点和网页文档、XML 用于商业、用 XML 进行文档管理、XML 和非文档数据应用、与 XML 连接、XML 应用的体系结构及前景。本书语言平实易懂,解释详细周到,示例深入浅出,不论初学者还是有经验的开发人员都会发现这是一本不可多得的好书。

图书在版编目(CIP)数据

XML 基础教程:第 2 版/(美)劳伦特(Laurent, S. S.)著;康晓林等译 .-北京:电子工业出版社,2000.1
ISBN 7-5053-5716-6
Ⅰ.X… Ⅱ.①劳…②康… Ⅲ.计算机网络-可扩充语言,XML-教材 Ⅳ.TP393
中国版本图书馆 CIP 数据核字(1999) 第 74125 号

书 名:XML 基础教程(第二版)
著 者:[美]Simon St. Laurent
译 者:康晓林 伊希荣 陈维义 等译
审 校 者:林义雄
责任编辑:嘉 益
特约编辑:赤 耳
印 刷 者:北京东光印刷厂
出版发行:电子工业出版社 URL: http: //www.phei.com.cn
北京市海淀区万寿路 173 信箱 邮编 100036
经 销:各地新华书店经销
开 本:797×1092 1/16 印张: 17.5 字数: 420 千字
版 次:2000 年 1 月第一版 2000 年 1 月第一次印刷
书 号:ISBN 7-5053-5716-6/TP·2945 著作权合同登记号: 图字:01-1999-2977
定 价:29.00 元
凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换。联系电话:68214070

译 者 序

随着人类步入信息社会和知识经济的到来，网络与人们生活、工作、娱乐的关系越来越密切。近几年 Internet 的飞速发展就充分地说明了这一点。Web 浏览是 Internet 的最主要功能之一，是人们在网上获取、发布信息的最主要手段。现在 Web 网页最普遍的数据格式是 HTML(超文本标记语言)，大多数信息提供者甚至不知道除 HTML 之外还有其它选择。但随着 XML(可扩展标记语言)的出现和发展，这种状况马上就要改变了！

XML 是 SGML(标准通用标记语言)的一个子集，或者说是 SGML 的一种压缩形式。SGML 出现于 Web 和 HTML 之前，其创建者旨在向信息管理人员提供更大的灵活性。XML 继承了这一原则并将它带到 Web 当中，允许 Web 的开发者和设计者们创建自定义的标记。因此 XML 比早期的 HTML 文档编码系统在组织和表现信息时具有更大的灵活性和更强大的功能。此外，XML 还将这种巨大的灵活性与语法的严格性(比 HTML 严格得多)结合起来，使开发人员能够突破 HTML 能力的限制创建出可重复使用的文档。

当前 XML 是国际互联网联盟(W3C)的“推荐标准”，它是在 1998 年 2 月 10 日通过 W3C 成员的鉴定和批准而获得 W3C 文档所能被授予的这一最高级别的。这意味着 XML 已被 W3C 首肯并很快就能得到广泛应用。实际上，目前国际互联网联盟正在围绕着 XML 创建一系列新的标记规范。其它一些组织也正在利用 XML 的能力为文档和数据标记创建标准，以 Microsoft IE 和 Netscape 为代表的浏览器开发商正在陆续推出对 XML 的支持，而且 HTML 的下一个版本很可能被重新构造成一个 XML 应用，因此业界将 XML 称为第二代 Web。

本书作者西蒙·圣·劳伦特是计算机网络，特别是 Internet 方面的专家，具有丰富的 Web 站点开发和网络管理的实践经验，而且有关于 XML 和网页创作的多部著作。作者以平实易懂的语言向读者深入浅出地展现 XML 的由来、特性、功能和使用方法等信息。对 XML 语法的解释力求结合实用的示例，以便读者能够尽快地理解和掌握。另外，虽然 XML 本质上是 SGML 的一个简化的子集，但本书并不要求读者事先具备 SGML 的任何知识，只要具备 HTML 的基本常识就能顺利地读懂本书。因此，不论初学者还是有经验的开发人员都适合阅读本书。

参加本书翻译工作的有：康晓林、伊希荣、陈维义(第一～第四章)、刘颂(第五～第六章)、李乃超(第七～第十章)、郝蕴、牛顺亭、林红、张启、刘立军、王英祥(第十一～第十三章及其他部分)。

本书最后由林义雄教授作了统稿和审校。

我们有幸参加了这本书的翻译工作，非常希望能为您掌握 XML 这一极富应用前景的 Web 创作利器奉献微薄之力，但因水平有限，疏漏在所难免，恳请不吝赐教。

译 者

1999 年 12 月

关于作者

西蒙·圣·劳伦特，Web站点开发人员、网络管理员、计算机书籍撰写者、XML专家，现居住在纽约州伊萨卡市。主要著作有：《建造XML应用》、《XML文档类型定义内幕》、《动态HTML入门》、《Cookies与共享带宽》。此外，西蒙·圣·劳伦特还利用业余时间编写用于XML处理的开放式Java源代码，请见站点http://www.simonstl.com。

前　言

可扩展标记语言(XML)早已被新闻媒体大肆炒作。本书《XML 基础教程》(第二版)将展示 XML 的真实面目,论述其目前现状及将来的几个发展方向。

本书面向的读者

本书适合于开始学习 XML 的任何一位读者。虽然它侧重于信息处理的文档方面,但是那些对如何使用 XML 进行原始数据交换感兴趣的读者同样会在本书中寻找到有用的信息。HTML 开发人员已经具备一定的基础,他们在接触 XML 时已对基本的标记有一定的理解,但我希望任何一位具有一定 Web 基础知识的读者都能够理解本书。本书是一本入门读物,书中不会对一些不常见的 XML 代码和面向程序员的代码示例作过多的解释,但这并不妨碍从中获得 XML 的基础知识及对 XML 标记的通盘了解。

本书的组织结构

本书的第一版在组织编排上力求向具有一定 HTML 开发经验的人员提供一条学习 XML 的尽可能快的途径。这种倾向仍然在一定程度上存在于本版中,具有其它背景的开发人员可以根据自己的实际情况跳过其中的一些章节。

本书前三章介绍 XML 的基本情况。第一章着重介绍 HTML,目前 HTML 仍旧是在国际互联网传输信息的主导方式。第二章论述显示信息的工具,以及实现标记以使标记可以集中面向内容的样式表。第三章讨论基本的 XML 语法并介绍分析程序;分析程序是 XML 的一个关键的基础工具。

接下来的六章讨论文档类型定义(DTD),文档类型定义是 XML 中一个更加复杂同时又更加强大的部分。第四章描述数据建模中涉及到的一些任务,从而为后面的工作做好准备。第五章向你提供建造 XML 文档类型定义所必需的工具。第六、

七、八、九章展示如何一步一步地建造 XML 文档类型定义。第九章可能会令程序员更感兴趣。

再接下来的两章讨论 XLink 和 XPointer，二者虽然放在最后讨论，但对 XML 来说是两个很有发展前途的补充工具。这两个标准将有助于建造下一代 Web 导航，其涵义将远远地超过目前的狭窄范畴。

第十二和十三章是全书的总结，将论述 XML 给几个技术领域带来的影响。这些领域包括开发 Web 浏览器和客户-服务器体系结构。

本书使用的图标

本书大部分是文本和图片的一般性混和编排，但经常有一些重要的信息需要单独列出，这些信息全部放在提示、参照或警告中。

"注意"提供有关正在讨论的话题的额外细节。当然，这些细节并非人人都感兴趣。"注意"中提供的信息是有用的，但不是关键性的。

"提示"中提供的信息通常不是传统文档中出现的，而是通过经验得来的。任何情形下都可以使用"提示"，但当"提示"出现时，它一定能够为你节省时间。

"当心"图标很重要。你可以认为不存在问题，但看上去很正常的行为完全可以带来可怕的结果。为避免潜在的灾难发生，请务必阅读这些警告。

本书中会经常提到某些问题在其它地方讨论会更好。按照"交叉参考"所指向的地方，你会找到有关某话题的更进一步的信息，这些信息通常是非常有用的。

第　一　章

XML 的兴起

* *

XML 的目标就是要改变 Web 的基本结构，超越 HTML 并代之以更强大、更具有可扩展性的体系结构。XML 旨在使 Web 返回到基于内容的结构，而不再是开发人员强加给它的基于格式的结构，因为开发人员已被不成熟的 Web 设计工具弄得灰心丧气。XML 还可以通过结束浏览器开发巨头在 Web 元素开发和实现上的垄断地位而将 Web 从他们的手中解放出来。同时，XML 还向应用开发人员承诺，不论他们是否在 Web 上工作，XML 都会向他们提供储存不同种类信息的非常方便的格式。

国际互联网联盟(简称 W3C，见站点 http://www.w3.org)以一种很有前途的新的标记方法而远远地走在商业浏览器开发商的前面。XML(可扩展标记语言)使开发人员创建自己可交互操作式标记语言成为可能，它包括 HTML 但又不局限于 HTML。由于新增加的特征转移到组件模型而不是单个程序，XML 的使用可能会导致 Netscape 和 Microsoft 之间浏览器大战的停火，甚至可能会促使新的浏览技术的出现。XML 带来的更直接的影响是，它使得开发人员能够创建以逻辑内容而不是以格式为基础的标记结构。这将使人和计算机能够更容易地在文档内搜索具体的基于内容的信息，而不是仅仅搜索一个页面上的文本。与样式技术相一致的 XML 将使作者能够创建出很容易管理的漂亮网页，使开发人员能够更好地控制信息，并使这种控制具有极大的灵活性。

“所见即所得”带来的麻烦

我曾经使用的第一个文字处理程序是一个非常简单的文本编辑器。当时看到通过在屏幕上四处移动光标使只能显示 40 列字符的屏幕可以将具有 80 列字符的页面显示出来。但它仅仅有助于做家庭作业，或编写其它类似的需要在点阵打印机上输出的枯燥文档。在使用计算机(一边使用计算机编程一边咒骂计算机)许多年后，我抛弃计算机而购买了一台电动打字机。它让我能够做一些有趣的事情，如不需要键入奇异的换码字符就可以给文本加上下划线。尽管没有敲打黑体文本的好方法，但

我不用再担心因为编排换码字符出错而浪费大量的纸张。打字机以传统的纸张和墨水的方式给了我“所见即所得”的感性认识。

我坚持使用这台打字机好几年，直到发现 Macintosh 计算机。当 Mac 计算机第一次出现时，我对它并没有什么好感，因为当时发行的许多杂志都以我不曾拥有的这种昂贵计算机做封面。而它甚至连一个象样的编程软件包都没有。但四年后当再一次遇到 Mac 计算机时，我被惊呆了。用它书写论文实在太棒了！因为我可以锁定使用所有样式信息，可以将页面分成多栏，甚至可以偶尔使用 72 点字型。尽管在我的 ImageWriter 打印机上输出的页面看上去不是很好，但比原来的点阵计算机文本强多了。在呈交的论文上，我使用了斜体字的标题和参考书目，将页面分成几栏，甚至还使用了一两张图片。书写再也不是简单的组织语句，而完全可以创建标题、子标题、表格和脚注，还可以使用其它各种格式，使得即使一篇很短的论文也可以包含多种结构，从而看上去很漂亮。通过使用样式，可以一次应用一组格式化工具，给这组格式化工具命名，以后可以对它进行调用。这看上去简直就跟魔术一样。

十年后我仍旧使用标题和子标题为我的文档安排格式，我并不关心脚注。但这时出现了一个新问题：很难重新使用老文档。当我在撰写某学年的论文时这无关紧要——写完交上去之后就再也不用想它了。但我现在整天跟若干年前由距离几千里外的人所写的大量信息打交道时，我很少需要关心如何将这些文件转化成相同的文字处理格式。我发现自己经常花费数小时时间所做的并不是编辑这些材料，而是重新格式化它们，这当然不是因为我喜欢这样做。滥用制表符和空格符（使用打字机养成的习惯）成长起来的一代人创建的文档无法剪切和粘贴到其它文档，因为所有东西都断开了。行断点彻底错误，文本全部堆积到左边或右边，表格已不成形，甚至象行间隔等简单的东西也出现问题。给创建具有漂亮外观的文档带来极大方便的神奇的格式化手段现在却带来了这么多严重的问题。

另外，还有一些其它细微的问题。那些年我认为自己是在创建标题和子标题，而实际上却不然，我实际上创建的是格式上象标题的文本。我们可以将这些样式叫做“标题”，但对计算机来说它们只不过是另外一组没有内在意义的字符。“所见即所得”也改变了人们。那些可能连五年级美术班都没有毕业的人也可以在页面上使用多达三十种字体。当这些字体渐渐失去新意后，他们中的许多人开始采用较为保守的方法来排布格式，其意图仍旧是使他们的文档看上去符合他们所想要的样子。设计人员已经习惯于指定位置时精确到千分之一英寸——仿佛人人可以分辨出这么小的尺寸。

在“所见即所得”之前，文档无疑是不美观的，但它们也有一些其它优点，尽管这些优点往往不被注意。有一些创建文档管理和文档标记系统以使计算机能够有效地管理大规模文档库的行动曾被酝酿过。简单的纯文本虽然单调，但与常见的文字处理程序或桌面出版程序的输出相比，纯文本更容易管理。这些文档管理工具在“所见

即所得"的早期还是很新的,直到后来用户逐渐习惯于使用基于纸张的媒体的系统之后,这些工具才慢慢地变得不算昂贵,也慢慢地多起来。目前市场上大多数程序的最终目标仍旧是文档的打印输出结果。

HTML的快速成长

当国际互联网第一次在1994年受到广泛的关注时,一小伙业余爱好者和专业设计人员开始着手创建他们能够创建的最激动人心的网页。但许多人很快就打了退堂鼓,因为习惯于完全"所见即所得"环境的他们实在因HTML格式化工具的缺乏而失望。各浏览器之间莫名其妙的差异使得很难预测一个网页看上去会使什么样子,公司用户需要象控制纸张页面文档那样控制他们的电子文档。在一定时期内这些抱怨和需求刺激了HTML的发展。象之前许多Internet技术一样,HTML是因为它本身有趣而被出于兴趣的爱好者们传播开来的。HTML很简单,只需一两天时间就能够学会,它向人们提供了一种全新的阅读和写作体验。这些早期的爱好者们所产生的冲力以及新闻媒体的相关报导,为HTML发展成后来的引人注目的事物提供了巨大的推动力。

虽然Web的发展已使其在经济上成为一块很有生命力的市场,但HTML必须改变自身以满足用户的需求。设计人员和他们的雇主希望能够创建出看上去完全符合他们需要的网页,而且希望能够达到象一般桌面出版系统所提供的那样的控制水平。经常爆发的浏览器大战(这时受伤的通常是网页开发人员)使HTML变得更加强大(尽管不兼容),国际互联网联盟(W3C)已或多或少地兼顾各方,出台了HTML 4.0。尽管准确地讲,HTML对普通用户来说仍旧不是一个简单的页面排版系统,但它提供的工具已方便多了。Web设计已经成为遍布全球的设计人员和通信专家的一项专门性工作,使公司和个人创建高级(尽管并非总是在视觉上令人愉悦)站点成为可能。

表格是HTML设计的一大进步,尽管它们的广泛使用推翻了关于应用表格的许多观点。设计人员使用表格创建与许多打印设计中使用的传统单元格相似的文档,使用这些表格的情形远远地超过了一般的成行成列的表格状信息显示。可点图技术的不断进步,使设计人员在HTML不能产生出他们所需要的界面时,可以使用可点图创建出自己的指向-点击界面。框架和弹出式窗口使开发人员可以将精力集中于页面元素,而不需要每次想改动某个小地方时都得重建整个屏幕的信息。〈FONT〉标记使得能够比基于结构的格式化更加精确地指定文本显示。随着Microsoft和Netscape争夺市场份额的竞争不断升级,两个公司都向调色板添加了各种工具。Netscape创建了〈BLINK〉,Microsoft则创建了与之相对应的〈MARQUEE〉。两个公司都创建了HTML元素及其属性的扩展,这使开发人员大感困惑,需要花费许多时间和设备在多个浏览器中检查站点。更糟糕的是,两个公司都不以完全相同的方式

实现标记。间隔可能会改变,颜色也可能改变,就是精心对齐的元素也可能会散乱在页面上。

注意

如果有关标准的争论或者销售商明显的无意遵循标准令你烦恼,可以查看互联网标准工程站点(http://www.webstandards.org)。HTML 的半标准地位销售商对其它标准(包括 CSS 和 XML 本身)的部分实现已经驱使许多 Web 开发人员以实际行动来捍卫标准。

同时,大型 Web 站点上的网页数目在不断膨长,站点经常膨长到包括 10,000 个以上网页。这些网页经常由于开发人员对超文本知之甚少,对组织管理知道得更少,而在层次模式方面组织得很松散。许多站点是根据混乱的目录结构进行组织的,建造这些目录结构的开发人员往往习惯于原来的 FTP 档案库和 Gopher 站点的结构,而 FTP 和 Gopher 都是 Web 的先驱。大型站点将接下来的困难交给必须维护它们的管理人员和试图阅读它们的用户。导航用的超文本本身是一种奇怪的艺术形式,它是组织技能、记忆、好的设计和运气的结合。设计导航超文本尤为困难。搜索引擎可以帮助用户寻找其路径,但很快就会发现即使拥有强大计算能力的管理人员也很难跟得上这种新媒体(站点)的爆炸式膨长。

诸如 Crawler 和 Robot 之类的自动化工具已开始搜索大量的 Web 文档。有一些只是索引标题,更高级的搜索工具则开始索引页面的整个内容。AltaVista(http://www.altavista.com)最初是由 Digital 公司创建来演示和促销其 Alpha 处理器一个搜索引擎,它应用共享千兆内存和巨大带宽的多个处理器,以"暴力"方法索引 Web。尽管 AltaVista 和其它许多搜索引擎能够提供服务,但它们的工作标准最宽松:一个文档的整个内容。我们设法给搜索引擎赋予一定的智能,甚至让它们区分语言和处理文字的形状,但要让它们在我们不用指出哪一部分是哪一部分的情况下读取、分类和组织文档,还要走很长的路。大多数情况下,人们用于识别文档之关键部分的格式化信息通常是被搜索引擎抛弃的。

这种容量和日益复杂的格式化的结合使开发人员想知道是否有更好的标记方法。HTML 已经走了很长的路,它走得很快,但其本身作为一门被设计为用于格式化的标记语言的局限性已日益明显。随着浏览器大战进入一个新的阶段,开发人员开始需要一种替代者,以使浏览器能够决定如何显示某个标记。每次 Web 站点在规模上翻一番时,搜索引擎的局限性都显得越来越明显。最后,随着 Web 的逐渐普及,HTML 在显示不容易符合标准文本和图形模型的信息时,表现出来的局限性越来越突出。开发人员需要能够创建他们自己的标记集,而且需要在创建自己的标记集时,所采用的方式能够与其客户使用的浏览器相一致。

追溯起源:结构和 SGML

当吉姆·伯纳斯·李于 1991 年创建 HTML 时,他将 HTML 基于一门更加强大但却更加复杂的标记语言,它就是标准通用标记语言,简称 SGML。是时 SGML 已经以各种形式存在了 20 年,但由于其复杂性,除出版、政府和大规模信息处理等部门之外,很少有其它组织采用它。因此,SGML 标记、管理和处理是专门的技术,只有一少部分政府、公司和学术界的用户能够掌握。

那些包括觉得 HTML 标准发展太慢的开发人员应该回过头来看看 SGML 发展的艰难步伐。最早形成于 19 世纪 60 年代的通用标记语言(GML),于 1969 年在 IBM 公司由研究人员 Goldfard、Mosher 和 Lorris 同时创建。后来 Charles Goldfarb 于 1978 年出任美国国家标准协会(ANSI)的文本处理计算机语言委员会的主持人,此时 GML 已经成为出版行业中的一个重要标准。该委员会在 1980 年公布了它的第一个工作草案,截止 1983 年,它的第六个工作草案已被国内税务局和国防部等用户采购,它们又命令其合同商也使用 SGML。在 1984 年,该委员会发展成一组协作共事的子委员会,它们为国际标准组织(ISO)和美国国家标准协会(ANSI)开发标准。在 1986 年,也就是 SGML 的标准化进程进行了八年之后,SGML 成为国际标准组织的 ISO 8879:1986 号标准。当然,有关 SGML 的工作仍在继续。有一组委员会定期评估 SGML 的发展变化,包括脚本样式表、多媒体、链接扩展和各种文档的管理问题。

注意

要想了解 SGML 的所有规范(差不多过几年就有一些改动),请查阅可以称得上是 SGML 宝典的下面这本书:《SGML 手册》,作者是 Charles F. Goldfarb,牛津大学出版社 1992 年出版。

与 HTML 不同,SGML 并不指定文本如何显示。SGML 不是一门格式化语言,甚至不是一门特定的标记语言。SGML 是允许人们创建他们自己的标记语言的一套规范。它规定的内容识别符使文本的一致性格式化非常容易,从而使文档管理系统能够快速地给信息定位。SGML 非常适合于涉及大量结构相似的数据工程,如目录、手册、清单、转录和统计摘要等。SGML 最受联邦政府、IBM 和其它大公司的喜欢。它使开发人员能够集合在一起方便地开发数据结构规范,创建文档类型定义(DTD),然后将它运用到整个组织内的文档。

更重要的是,在许多情况下,用 SGML 创建的文档能够容易地移植到不同的格式。因为 SGML 使用基于内容的标记,而不是基于格式的标记,所以改变格式化规

则很容易，仅取决于文档输出到点阵式直行打印机、激光打印机、四色印刷机、CD-ROM、Web 站点或者甚至是音频喇叭。设计小组决定使用的格式与开始时 DTD 开发人员所做的工作相匹配，从而以适合具体输出媒体的样式显示信息。计算机化的存储系统也可以把文档当作小的数据库来对待，使用基于内容标记和索引信息的搜索对它们进行查询。重复使用相同信息的公司也因为事先制作好了适用于新文档的文本而受益匪浅。SGML 不会使所编写的东西更加漂亮，但可以使它们更容易管理。

HTML 的根源

SGML 对 HTML 的贡献大部分是语法，即使用“<标记 属性=值>内容</标记>”这一形式的标记语言。SGML 将内容与格式分离的意图有一些也保留在 HTML 中，从不同的浏览器对同一标记的完全不同的解释可以看出这一点。用<EM>表示强调，用<ADDRESS>表示地址信息，通过这种描述元素的方式，伯纳斯·李创建了一门足以灵活应付许多种不同信息的简单的格式化语言。从<H1>到<H6>这六个标记描述不同层次的标题，向文档提供一种比较自然的结构。<HEAD>和<BODY>标记将元信息（首先是<TITLE>与文档中的可见文本分离开来。最重要的是，<A HREF="目标.html">标记为超文本链接提供了一个简单而强大的结构。

HTML 为简化 SGML 做了很好的工作，使业余爱好者也能够使用标记，这是扩大标记的影响力所必需做的。极具讽刺意味的是，伯纳斯·李当初从没有打算让用户必须手工键入代码。最初在欧洲核研究委员会所做的试验，使用的是一个不可视地管理代码的简单的标记处理器。虽然 HTML 只是一小部分具有少量属性的标记集合，但 Web 却是用来交换网页和共享信息的。

但是，正如我们已经看到的，面对设计人员借助“所见即所得”工具而毁坏的这个世界，HTML 显得很不适合。尽管超链接和很容易使用的 Web 都是好东西，但从一开始，就有异议认为这些空谈到底能为无用的格式化语言带来什么起色。在浏览器大战刚开始时，想要在 Web 上创建出与在纸上同样细致（不考虑屏幕分辨率的限制）的文档的设计人员和开发人员，从长远看肯定不会接受 HTML 最初极其简单的格式化工具。简单标准固有的灵活性缺乏号召力。唯一职责是格式化的标记充斥着 HTML 领域，从<B>和<I>直到<FONT>都远远超过<EM>和<ADDRESS>的使用频率。设计人员动手制作 HTML，将各种标记混合搭配在一起创建出他们想要的外观，而不考虑文档结构。因为 HTML 标记只是用来指定格式而没有可替代的格式化结构，所以它注定是一门格式化语言，而不是用于文档的结构化框架。早期所有的问题都慢慢地暴露出来，而这些问题在一开始时，只是光彩夺目的 HTML 中一些基本的瑕疵。

XML的出现

当初Web标准的创建者和他们的接班组织W3C(国际互联网联盟)一时蜂拥而上,试图赶上商用浏览器开发商的步伐。"Netscape扩展"向设计人员提供了HTML早期版本所缺乏的控制,促进了Netscape的壮大。直到最近W3C才追赶上浏览器开发商并以以下这套强大的标准拦住了他们的去路。级联样式表(CSS)是W3C发出的第一击。CSS使开发人员能够为文档声明他们自己想要的格式,而无须将一堆标记和图形混合到一起。GSS使标记从携带格式信息的重担下解放出来,允许它们再次携带内容信息。通过提供一套完整的描述格式的术语(比HTML4.0全面),CSS提供了一种能够与HTML和其它任何标记语言相兼容的样式工具。

一旦有哪种格式化标准与标记不匹配,W3C完全可以创建一种利用SGML灵活性的标记标准。XML强调提内容信息的重要性,它使设计人员能够创建和管理他们自己的元素集。设计人员可以使用这些与CSS相一致的元素集来创建产生格式的标签(如果愿意的话,他们当然可以这么做),但主要的重点在于管理内容,包括超文本链接,使它们本身得到极大增强。XML的崛起来自于W3C的SGML社评委员会的关注,因为他们感觉到HTML在朝着错误的方向发展。为了改善现状,他们提出了一种与现有Web技术相一致的标记语言,使用的一些工具仍旧是为使用HTML开发的,但以其更加容易管理的技术而向前迈进了一大步。XML提供的是SGML功能的一个子集合,而不仅仅是使用SGML语法的一套标记。它虽然是SGML的简化(有一些SGML用户认为简化的程度不够高),但发誓要恢复Web最初的承诺,将混乱复杂的网页创作现状简单化,若想经获得了多大的成功,只需要看看W3C为下一代HTML的产生所订的计划(见http://www.w3.org/MarUp/Activity.html),这一计划将使当前巨大的HTML工程分解为许多用XML定义的子模块。

XML由从SGML社评委员会演变而来的W3C的XML工作组控制。XML工作组(见http://www.w3.org/XML/)控制与XML直接相关的大部分规范,包括XML语法、XML扩展链接和XML分段使用。同时,W3C的其它工作组控制其它支持性标准,如文档对象模型(DOM)、级联样式表(CSS)和可扩展样式语言(XSL),以及补充或应用XML的标准,如数学标记语言(MathML)、多媒体同步一体化语言(SMIL)以及资源描述框架(ROF)等。虽然以上这些由W3C控制,但XML还有许多其它应用(它们当然也有自己的名称缩写)由其它组织、公司和个人所控制。尽管很有希望在将来由开发人员坚持XML标准,但XML现在是由销售商实现的。

XML的实现已花了一段时间。当W3C于1998年2月发布XML1.0语法的推荐标准(官方标准)时,对XML的支持大部分停留在程序员能够使用的水平,但一般的HTML开发人员还不能够使用。这种状况现在正在慢慢改变,特别是Microsoft公司在Internet Explorer5中为XML提供了较高级的支持,以及Netscape的Mozilla

工程最近也提供了对 XML 的巨大支持。但是，由于当时安装时没有此设置（人们并不经常升级他们的浏览器），还要等一段时间才能人人可以得到启用 XML 的阅读器、XML 编辑器及其它 XML 应用程序。

在本书中，我们要详细讨论 XML 提供的工具，以及开发人员怎样将它们用于常见的任务。HTML 开发人员将会发现许多信息是面熟的，但是也有一些内容（如创建 DTD）比较陌生。这本书不是面向 SGML 开发人员的，但熟悉 SGML 的读者会看到，许多熟识的概念与来自更广阔的 Web 世界的概念结合在一起。尽管本书着重讲述使用 XML 创建文档，我们也会论及管理 XML 以及将 XML 与其它 Web 技术相结合的一些技术。软件开发成为 XML 的重要市场，所以本书讨论的一些工程也会对软件开发人员有益。XML 可能乍看上去有些抽象，但随着你进一步阅读本书和体会书中的示例，其含义会越来越清晰。我们将看到，尽管 XML 文档与严格定义的层次结构或表式结构非常不相像，但 XML 文档集合仍旧可以当作数据库来对待；我们将探讨 XML 使之成为可能的新的文档结构和数据。

第 二 章

内容与形式分离:标记和样式

* *

在讲述 XML 如何为内容建模之前,先得简单介绍一下 XML 是如何把我们从 HTML 的展示形式中解脱出来的。尽管我们已经在前一章指出了 HTML 的一些不完善的地方,但是从总体上说来 HTML 使标记得以普及,只是它的现有结构值得改进。我们将看一看 HTML 的一些不完善的地方,但是从总体上说来 HTML 使标记得以普及,只是它的现有结构值得改进。我们将看一看 HTML 在样式表之前(浏览器 3.0 及以前的版本)和之后(4.0 以后的版本)是如何被使用的。从 XML 的角度看 HTML,你会理解 XML 即将带来的新东西。样式表对 HTML 和 XML 都很重要,它将标记从束缚 HTML 很久的格式化结构中解放出来,使得标记能够专注于内容和重复使用,而不用将注意力集中于如何展示。

注意

如果你不关心展示的形式,或者不是具有开发人员的背景,那么可以跳过这一章直接进入使用 XML 为内容建模。本章最适合于将 XML 用于 Web 网页或打印文档等供人阅读的媒体的那些用户,而对于使用 XML 连接计算机和数据库则不太适合。本章有关 XSL 转换的讨论可能会对后者有用(尽管 XSL 转换现在还非常不稳定),但建议这些读者还是先跳过本章直接阅读后面的内容,等到发现需要这方面的信息时再返回来查阅。

HTML 是 SGML 的许多应用中的一个,国际互联网联盟(W3C)使用 SGML 文档类型定义为创建 HTML 的规则提供正式的定义。尽管有些 HTML 元素(例如指示行断开的 BR 元素)很难在结构更紧凑的 SGML 世界中有意义地表达出来,在至今出现的所有 HTML 版本中都可以使用文档类型定义。HTML 的语法一直比较宽松,倾向于 SGML 的多种可能性。例如,闭端标记传统上总是可用可不用的。直近随着 HTML 创建工具的发展,关闭每个标记才变成很寻常的事,而且仍有许多 HTML 创建工具依然不关闭标记。

浏览器长期以来一直忍受着语法使用中的多样性，尽管它们经常会因具体使用的语法而在显示时略有不同，通常它们会计算出一个元素在哪里结束以及另一个元素从哪里开始。这种语法的多样性产生了许多后果——一种浏览器显示文档的方式往往与同样的文档在其它浏览器中显示的样子完全不同。再加上浏览器如何格式化特定标记的定义很松散，这种语法上的灵活性，害得设计人员经常彻夜不眠地在多个浏览器中重复创建同样的格式，而从没找到给他们带来如此痛苦的那些标记。

HTML 的根：原来的规范

在表格、框架、字体标记和客户端可点图，以及 HTML 现在所拥有的所有神奇的工具出现之前，已经有一少部分标记提供常见的学术论文结构所具有的格式（毕竟，当初欧洲核研究委员会的研究人员创建 Web 时就是用它作为物理学家们交流学术的场所）。跟 SGML 不同，HTML 为其标记定义了格式用途，但是它们不象一般的“所见即所得”文字处理程序或桌面出版软件包中的提供的格式那样具体。

嵌入在开端 HTML 标记中的是 HTML 文档的两个重要的部分：HEAD 元素和 BODY 元素，二者分别携带不同的信息。HEAD 元素包含有关文档的数据，如：TITLE 元素，为文档中的所有超级链接设置基址 URL 的 BASE 元素，以及 META 元素。META 元素可以包含信息（由于它是有关数据的数据，因此也叫做元数据），这些信息涉及文档的作者、创建它的组织、供搜索引擎查找的关键字，以及页面创建和文档管理软件在跟踪该网页在更大的站点组织中的位置时使用的信息。同样可以出现在 HEAD 元素中的 LINK 标记将文档与样式表等外部资源连接在一起。

注意

在本章（乃至全书中）我所使用的 HTML 元素名称是常见的大写形式，如：使用 BODY 而不是 body，HTML 而不是 html。W3C 好象要在 HTML 的下一个版本中使用小写，而且随着 XML 的影响不断扩大，大小写的区分将对 HTML 的使用更加重要。

BODY 元素几乎是所有内容出现的地方。除了位于浏览器窗口顶部的标题之外，HEAD 的信息通常对用户来说是不可见的。BODY 部分中的信息产生 Web 页面的真正外观并吸引用户的大部分注意力。例如，大多数 Microsoft Word 用户让文件的属性框（其作用类似于 HEAD 元素）处于关闭状态，因为太没有必要为每个文件都键入搜索关键字。一般用户关心的是文档中的文本内容，再就是其格式。HTML 的 BODY 元素与文字处理文档中的正文相似。在 BODY 元素中，所有文本的标记是按顺序排列的，其顺序是常见的从左到右，从上到下（但是，如果你使用提非欧洲字符内