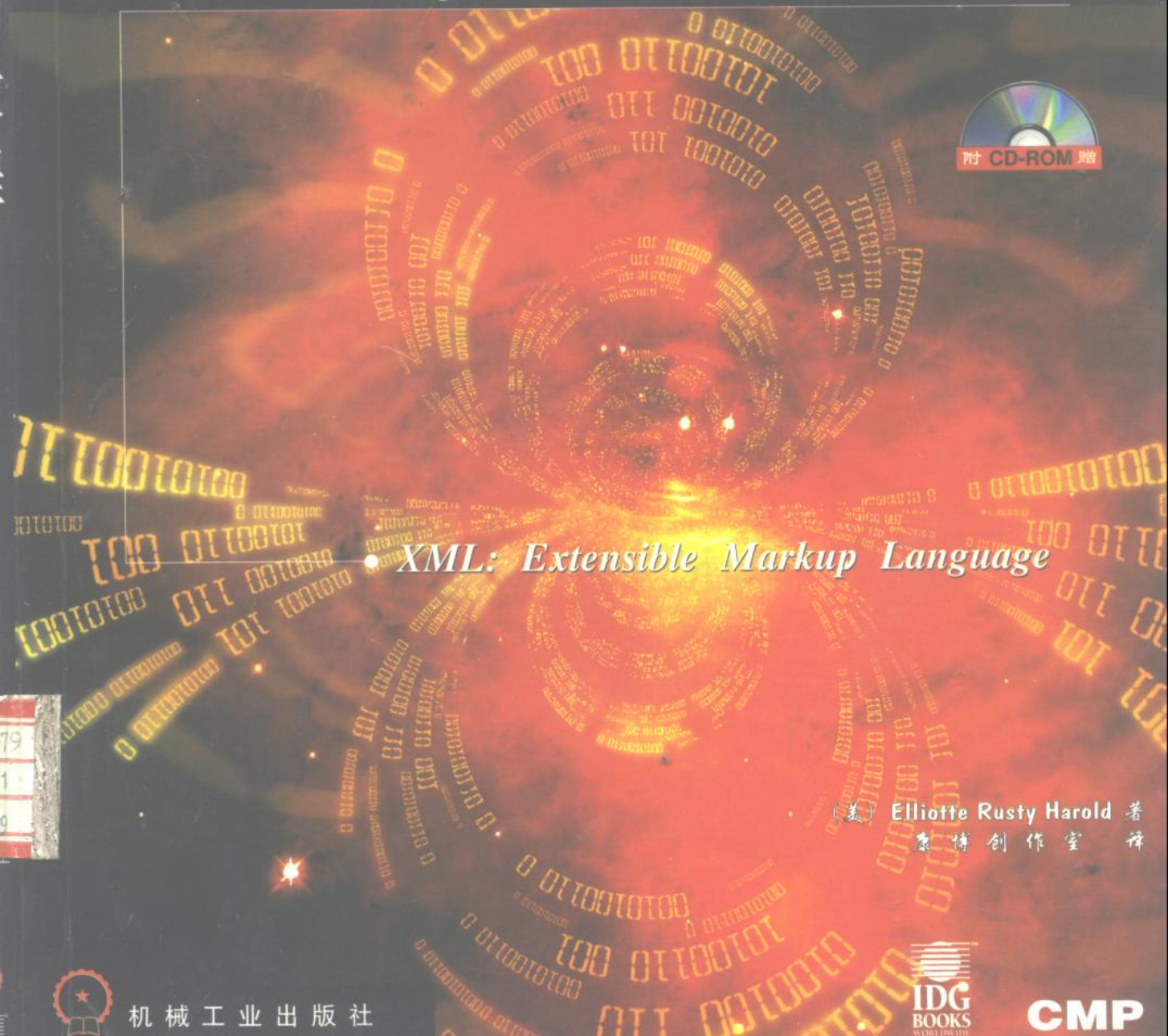
最新 Internet 技术基础与应用系列丛书

# XML

# 实用教程




XML: Extensible Markup Language

[美] Elliotte Rusty Harold 著
康博创作室 译

机械工业出版社
IDG BOOKS WORLDWIDE
CMP

本书详细介绍了可扩展标记语言(XML)的应用技术。全书共分11章，第一部分是XML基础篇，介绍XML的基础知识；第二部分是XML提高篇，着重介绍DTD技术、属性、字符集、XLinks和 XPointers；第三部分是XML应用篇，介绍如何从头开发一个DTD。本书结构严谨、语言流畅、示例详实，适用于从事网络工作开发人员和系统维护人员。

本书的配套光盘上有书中出现的全部示例，可以与书配套学习使用。



**本书版权登记号:图字:01-98-1913**

**图书在版编目(CIP)数据**

XML实用教程/(美)哈罗德(Harold,E.R.)著；康博创作室译.—北京：机械工业出版社，1999.1

(最新Internet技术基础与应用系列丛书)

书名原文：XML Extensible Markup Language

ISBN 7-111-06952-8

Ⅰ.X… Ⅱ.①哈…②康… Ⅲ.计算机网络-可扩充语言，XML-程序设计-教材 Ⅳ.TP393

中国版本图书馆CIP数据核字(98)第34376号

出 版 人:马九荣 (北京市百万庄大街22号 邮政编码100037)

责任编辑:温莉芳

北京市密云县印刷厂印刷 · 新华书店北京发行所发行

1999 年 1 月第 1 版第 1 次印刷

787mm × 1092mm 1/16 · 18.25印张

印数：0001-5000册

定价:46.00 元 (附光盘)

# 译者的话

计算机技术和信息浪潮席卷了整个世界，改变了人们早已熟悉了的生活、工作和学习方式，并且还大大改变了人们的思维方式。新思想、新概念、新创造、新技术层出不穷，令人目不暇接。Internet的发展就是其中的一项卓越的技术，人们之间的距离日益缩短，互相交流日益增加，思想更加活跃，人们的要求也因而更大更高。XML作为一项新技术，从1996年开始，发展十分迅捷。

XML是Extensible Markup Language的缩写，含义是可扩展标记语言。XML是一种元标记语言，它没有许多固定的标记，为开发人员提供了更大的活动范围，满足了人们日益增长的需要。全书共分三部分11章。第一部分是XML基础篇，包括4章内容，第1章是XML入门介绍，第2章是开始XML，第3章是正规化XML，第4章介绍XSL。第二部分是XML提高篇，由5章组成，第5章是在XML文档中使用DTD，第6章是从多个数据源汇编文档，第7章是用属性描述元素，第8章是介绍国际字符集，第9章是介绍XLinks和XPointers技术。第三部分是XML提高篇，包括第10章用CDF技术推出Web站点，第11章是从头开发一个DTD。本书结构严谨、语言流畅、示例详实，是一本难得的好书。

本书由康博创作室统一策划，闪四清先生主译和校对。由于时间和水平所限，难免会出现一些疏漏，敬请读者批评更正。

1998 年 11 月于北京

# 前　言

欢迎学习XML语言。读完本书之后，我希望读者会同意这种说法，即从Java出现以后，XML是Internet上最激动人心的发展，它使Web站点的开发更容易、效率更高、更奇妙。

本书是向读者介绍XML的基本内容及其发展。在本书中，读者将学习如何用XML编写文档，并且使用XSL样式表把这些文档转换成HTML语言，以便以前的浏览器可以读懂这些文档。读者还将学习如何使用文档类型定义(DTDs)来描述并验证其有效性。这种技术随着越来越多的浏览器，例如Netscape和Internet Exploer 5.0，提供了对XML的专门支持，将会变得越来越重要。

本书是针对Web页作者而不是软件开发者使用XML的第一本书。因此，本书没有花费很多篇幅讨论BNF语法和分析元素树结构。相反，本书将向读者展示如何使用XML语言和目前已有的工具来更加有效地制作强大的Web站点。

## 本书读者范围

本书内容主要是针对Web站点开发者而写的。在本书中，假设读者打算使用XML语言制作那些使用HTML语言很难创建或者根本不可能创建的Web站点，读者可能会惊奇地发现，由于使用XSL样式表和一些免费工具，读者可以完成这些以前对每个开发人员都要花费成千上万资金来定制软件或者要求具有像Perl一样的高深编程语言技术才能完成的任务。本书所讨论的软件仅需花费数分钟的下载时间，并且需要的编程技术不超出剪切和粘贴JavaScript的技巧。

然而，XML以HTML和Internet的基础设施为基础创建。因此，本书假设读者已经具备了在所选择的Web浏览器中用ftp发送文件、发送电子邮件和加载URL的技术。另外，本书还假设读者具有 Netscape 1.0水平的基本知识。在本书中，当讨论新知识时，像Cascading Style Sheets(层叠样式表)或者〈SPAN〉和〈DIV〉标记一类的基本HTML内容就不详细解释，而只是讨论较深的话题。

本书假设用户具有下列技术:

- 能够使用文本编辑器书写包括链接、图像和文本内容的基本HTML页。
- 能够在Web服务器上放置自己编写的HTML页。

另一方面，本书假设读者不具备下列技术:

- 读者不需要任何SGML知识。事实上，在整本书中，几乎仅是在前言中提及了SGML。(在第1章中介绍XML的历史中，简要讨论了SGML)。通过设计，XML比SGML更加简单，应用范围更广，如果首先必须学习SGML，那么就失去了这些优点。
- 不像其他许多介绍XML的书籍那样假设XML是一种标记语言，而不是一种编程语言，这里读者不必是Java、Perl、C或者其他程序设计语言的程序员。

下面这些领域的知识并非本书的必备知识，但是，如果具备了这些知识，那么在开始编写XML文件时，这些知识是非常有用的。

- Web服务器配置和HTTP协议
- JavaScript
- 数据库理论，特别是正规化规则
- SGML
- 一些国际字符集知识

就像编写HTML一样，读者不必具备学习XML文件的预备技术。然而，那些懂得这些知识的读者将比不懂这些知识的读者更容易接受书中的一些内容。如果读者已经熟悉了一部分内容，那么就更容易采纳和解释某些实际操作。如果读者不熟悉这些知识，那么就必须接受一些表面看来杂乱无章的XML语言规则。

本书假设读者正在使用Windows 95或者Windows NT4.0或者更高版本。对于一个长期使用Mac和Unix的用户，作出这种假设，我觉得很遗撼。像Java一样，XML应该是与平台无关的。同样像Java一样，这种特点在宣传上多少有点不足。虽然XML代码是可以用任何编辑器书写的纯文本形式，但是有些关键性的工具只能在Windows上使用。我希望，在不远的将来，这些关键性工具将能在Macintosh和UNIX上使用。到那时，XML的开发将仍然主要是基于PC的工作。

## 本书主要内容

本书的主要目标是教会读者编写Web的XML文档。幸运的是，XML有一条明确平缓的学习曲线，与HTML非常类似，但是与SGML不同。当读者学了一点儿XML内容时，就会编写一些XML语言。当读者再多学一些XML内容时，就会多编写一些XML语言。因此，本书各章内容环环相扣，逐步加深，读者应该循序阅读。根据这种方法，读者将学习下列内容：

- 如何进行标记，使XML文档比同样的HTML文档的维护和开发更简单容易
- 如何以人人都能阅读的格式在Web服务器上发送XML文档
- 如何确保XML格式正确
- 如何使用XSL样式格式化读者的XML文档
- 如何使用DTD验证XML文档的有效性
- 如何使用实体创建由较小部分组合而成的大文档
- 如何描述数据属性
- 如何在XML文档中使用像ζ和√的国际字符
- 如何使用XLinks和XPointers连接XML文档
- 如何使用CDF推出Web站点
- 如何从头开发DTD

当读者掌握了这些内容后，就可以使用XML来创建迷人的Web页了。

## 本书结构

### 第一部分：XML基础篇

第一部分介绍XML的用途、结构和语法以及与其相关的样式表语言XSL。第1章介绍XML，包括XML的历史和理论、XML的目标以及一些有趣的用途和应用。第2章XML入门，

向读者介绍一些简单的XML文档，并且教会读者如何使用文本编辑器编写这些简单的XML文档，如何把它们转换成HTML文档，以及如何在Web服务器上使用这些文档。第3章正规化XML，讨论XML如何采取简单性和可扩展性而不是增加过多的标记来实现其强大的功能。XML预定义几乎根本没有标记，而只是使读者定义自己所需的标记。第4章XSL，主要讨论如何使用XSL样式表提供数据，这种样式表描述如何在自己的文档中格式化单个元素。本章还介绍如何使用样式表提供定制使Web站点看起来和感觉到是一个统一整体的外表。

**第二部分：XML提高篇**

第二部分的内容是在XML基础知识上的提高，介绍文档类型定义(DTD)、在XML文档中使用DTD和可扩展链接语言(XLL)。第5章在XML文档中使用DTD，浏览XML这种元标记语言，这种语言可以通过文档类型定义。在本章中，读者将学会如何创建在特殊领域中使用的标记语言，这些领域例如音乐、数学、航天、电子、家谱以及其他能够想象到的领域。第6章汇编来自多个数据源的文档，介绍一个单个XML文档如何由数据和声明两部分组成，这两个部分来自许多源实体和参数实体。第7章用属性描述元素，向读者介绍如何在XML标记内使用和声明XML属性。属性就是与某种元素(例如标识号)相关联的额外信息，这些信息仅由阅读和编写文件的程序使用，而不是为了使人们阅读和编写元素内容。第8章国际字符集，介绍如何使用非英语语言编写XML文档，国际文本在计算机应用程序中是如何表示的，XML如何理解文本，以及如何利用必须用非英语语言阅读和编写的软件。第9章XLinks和XPointers，介绍XLL(可扩展链接语言)，这是一种链接文档的新方法。XLinks和XPointers除了提供基于URL的HTML的超链接和固定点(anchor,锚点)以外，还提供附加的功能。

**第三部分：XML实践篇**

第三部分向读者介绍XML在不同领域中的两种实际用途：Web站点和家谱。第10章用CDF推出Web站点，研究微软的通道定义格式(CDF)。本章还介绍如何把Web站点转换成CDF通道。第11章从头开发一个DTD，引导用户循序开发几个用于家谱数据的DTD。按照这种方法，读者将会学会如何使用XML标记以及为什么和何时选择这些标记。

## 快速参考

快速参考提供了简洁完整的全部XML关键字用来构造说明和DTD语法的清单。在这一章中，可以找到某个标记的准确语法。

## 附录

最后，附加了3个附录，这些内容不适于增加到书中的主要部分中。附录A是国际文本，包括了各种不同的国际字符集和这些字符集在XML中应用的详细信息。附录B是XML说明书，包含了由W3C出版的完整XML1.0说明。附录C是附加资源，提供了一些介绍XML更深内容的资源，这些资源包括邮件清单、说明书和软件。术语，介绍了本书中所用的主要术语。还有有关本书配套光盘内容的描述。

## 如何使用本书

本书的结构按照阅读顺序设计，每一章的内容都建立在前面章节的基础上。当然，欢迎

读者跳过那些熟悉的内容。

希望读者经常停下来，试验一下示例程序和编写自己的XML文档。要想学习好，不能仅靠阅读，还得进行实际练习。

在开始学习之前，请读者注意在本书使用的下面语法约定。

与HTML不同的是，XML对字母大小写敏感。〈FATHER〉与〈Father〉或者〈father〉是不同的。father元素不同于Father元素或者FATHER元素。遗撼的是，大小写敏感标记语言与使用标准英语语言的习惯有冲突。在极少情况下，读者才会碰上没有用大写字母开头的句子。在大多数情况下，读者会看到在句子中间使用了大写字母化的字。请莫担心这些实例，文章对这种常用的大写字母化进行了解释。

## 与作者联系

如果读者发现书中的错误，或者喜欢与作者讨论有关XML的问题，请发送电子邮件：elharo @ sunsite.unc.edu。读者还可以浏览作者的Web站点：http://sunsite.unc.edu/xml//。

# 目　录

## 第三部分 XML实践篇



## 第四部分 附 录

# 第一部分 XML基础篇

## 第1章 介绍XML

本章介绍以下内容:

- 什么是XML
- 为什么用XML
- XML历史
- XML程序
- 相关技术
- XML应用程序

本章向读者介绍XML和一些有趣的用途和应用程序。读者将学习如何把XML使用在诸如化学、数学、台球、多媒体装置等形形色色的领域中。

### 1.1 什么是XML

XML是Extensible Markup Language(可扩展标记语言)的简称，因为Extensible Markup Language常错拼为eXtensible Markup Language，因此简称XML。XML是一组用来形成语义标记的规则集，这些标记可把一篇文档分割成许多部分或验证文档中的不同部分。

#### 1.1.1 XML是一种元标记语言

XML不是像HTML或troff一样的标记语言，标记语言定义了描述一定数目元素的许多固定标记。如果读者所选的标记语言没有包含自己所希望的或者所需要的标记，那么读者太不走运了。这时，只能等待该标记语言的下一个版本的发行，并且希望新版本中包括了所需的标记，实际上，读者完全受到供应商所选内容的控制和支持的。

然而，XML是一种元标记语言。在这种标记语言中，读者可以按照工作要求构造自己所需的标记。虽然这些标记必须根据一定的常规原则来组织，但是这些标记在其含义上是非常灵活的。例如，如果读者正在制作家谱，需要描述人，即出生、死亡、墓地、家庭、结婚、离婚等等，那么就应该为这些元素中的每一个元素创建标记。读者不必使这些标记适应段落、列表项、强调重点或者其他常用分类。

所创建的标记可以用文档类型定义(DTD)来制作成文档。在本书的第二部分中，读者将学到更多的有关DTD知识。但目前，可以先把DTD看作是某种文档的术语和语法。例如，Peter Murray-Rust的MDL.DTD描述了分子学的术语和语法：化学、晶体学、固体物理学等等。这种DTD可以由该领域中的许多不同的人共享和使用。其他DTD适用于其他规则，而您也可以创建您自己的DTD。

读者可能会认为，虽然这种方法在理论上是很不错的，但是在实际上，Netscape和Internet Explorer是不可能支持数千种不同的语言的。然而，当把XML用作这些语言的模板时其长处才能准确地展示出来。XML定义了一种像CML、MathML和其他指定的标记语言必须遵循的元语法。如果某种应用程序理解这种元语法，那么它就能自动理解所有由此元语言创建的语言。

浏览器没有必要硬拷贝每一个可能被数千种不同的语言使用的标记。相反，当浏览器阅读给定的文档或者给定文档的DTD时，就会识别由这种文档使用的标记。关于如何提供这些标记的详细说明在附加在该文档的样式表中提供。

例如，考虑下面的Schrodinger公式

$$ih\frac{\partial\psi(r,t)}{\partial t}=-\frac{h^2}{2m}\frac{\partial^2\psi(r,t)}{\partial x^2}+V(r)\psi(r,t)$$

许多科学论文中充满了这些公式，科学家们一直等待了许多年，希望浏览器支持写这些公式的标记。音乐家们的境遇与此类似，因为Netscape和Internet Explorer不支持工作单音乐。

XML的使用意味着没必要将等到浏览器供应商理解用户的想法。当自己需要标记时，就可以创造出这些标记，然后告诉浏览器如何显示这些标记。

### 1.1.2 XML是一种语义／结构化标记语言

XML描述文档的结构和含义，它不描述页面上元素的格式。虽然格式可以通过样式表增加到文档中，但是文档本身仅包含描述文档内容的标记，而不是文档的外观。

相对而言，HTML包含了格式、结构和语义标记。〈B〉是一种使其内容字体变粗的格式标记，〈STRONG〉是一种使其内容着重强调的语义标记，〈TD〉表示其内容是某个表格结构的一部分。事实上，有些标记可以有全部这三种含义。〈H1〉标题表示页面标题并且字体为20磅变粗的Helvetica体。

例如，在HTML中，书目中的一本书就像下面例子一样:

```
<dt> Java Secrets
<dd> by Elliotte Rusty Harold
<ul>
<li> Publisher: IDG Books Worldwide
<li> ISBN:0-764-58007-8
<li> Pages:900
<li> Price:$59.99
<li> Publication Date:May,1997
<li> Bottom Line:Buy It
< / ul><P>
The Jave virtual machine,byte code,the sun packages,native
methods,stand-alone applications,and a few more naughty bits.<P>
```

这本书是通过使用定义标题、定义数据、无序列表、列表项和段落来描述的。在这些标记元素中，没有一种元素与书有关。在XML中，同样的数据可能如下所示:

```
<book>
  <title> Java Secrets< / title>
  <author> Elliotte Rusty Harold< / author>
  <publisher> IDG Books Worldwide< / publisher>
  <isbn> 0-764-58007-8< / isbn>
  <pages> 900< / pages>
  <price> 59.99< / price>
  <publication_date> May,1997< / publication_date>
  <recommendation> Buy It < / recommendation>
  <blurb>
    The Java virtual machine,byte code,the sun packages,native methods,stand-alone
applications,and a few more naughty bits.
  < / blurb>
< / book>
```

在这个程序清单中，使用了有含义的标记名称如<book>、<title>、<author>、<price>来代替了象<dt>和<li>一样的常用标记。采用这种方法，有很多优点，至少使人们阅读源代码来理解作者的本意更加容易。

XML标记也使非人类、自动化机器人定位文档中的全部书籍更加容易。在HTML中，机器人只能确定<dt>是一个元素，他们无法确定<dt>是表示书的标题呢，还是一种定义，或者是设计人员喜欢的一种缩进文字的方法。实际上，一篇文档可以包含有全部三种意义的<dt>元素。

XML元素名称可以组织成超过文章内容的特殊含义，例如，它们可以是数据库的字段名称。在HTML中，对于各种用途有固定的标记，而在XML中，对于各种用途没有固定的标记，这使得XML对于各种应用更灵活和更易被接受。

## 1.2 为什么用XML

为什么人们热衷于XML呢?使用XML可以做哪些用目前的技术很难做到的事情呢?由于XML中的可扩展，使得人们可以用XML做各种各样的事情。一旦读者学会了XML，那么读者就可能找到了解决目前正需要克服的许多难题的解决方法。

本节考察了一些可以使用XML的用途。在这之后，读者会看到一些已经用XML开发成功的特殊应用程序。

### 1.2.1 特殊的域标记语言

通过允许专家们开发自己通用的标记语言，XML可以使该领域中的每个人可以交流注释、数据和信息，而不必担心接收端的用户是否具有用来创建这些数据的特殊专用许可。他们甚至可以把这些文档发送给该专业之外的人们，有理由相信他们至少可以查看这些文档，即使不理解这些文档也可以查看。

对于那些专业领域之外的人们，不必创建庞大的过于复杂的程序，就可以使上述过程发生。虽然读者可能对描述电子工程图的通用方法不感兴趣，但是电子工程师感兴趣。虽然读者没有必要在自己的Web页中包括工作表音乐，但是作曲家则很有必要。XML允许电子工程师描述他们的电路，允许音乐家标注他们的音符，而不互相冲突。这两种领域都不需要来自浏览器制造商或者复杂的插件的特殊支持。

### 1.2.2 通用数据格式

最近40年来，许多计算机数据遭到了不可弥补的损失，这不是因为自然灾害或者损坏了备份介质(尽管这些也是问题，但是XML对此无能为力)，而是简单地因为没有人想到人们如何阅读数据介质和文档的格式。如果在现代的许多公司中没有巨额的时间和资源的投资，那么一个在十年前5 1/4英寸软盘上的Lotus 1-2-3文件可能就无法使用了。象Lotus Jazz的二进制格式的数据可能被人永远地遗弃了。

从低级观点来看，XML是一种极其简单的数据格式。XML是用百分之百的纯ASCII文本和一些人们普遍使用的格式来编写的。ASCII是抵抗破坏的删除部分即使删除了很大的一部分字节，都不会破坏文本的剩余部分。这种格式的优点是相对于其他许多格式而言，这些格式例如压缩数据格式或串行Java对象，当遭到破坏甚至仅有一个字节就会使该文件剩余部分是不可阅读的。

更进一步而言，XML更宜归档。W3C的XML说明书和浩翰的论文书籍(例如本书)都准确地告诉读者如何阅读XML数据。采取这种格式，不会有什么毛病。

从高级观点来看，XML是一种自描述语言。假设读者是一个23世纪的信息考古专家，碰到了在一张老式软盘上的一块XML代码，这张软盘是经过了几个世纪幸存下来的，其XML代码如下：

```
<person id = "p1100"sex = "M">
  <name>
    <given>Judson< / given>
    <surname> McDaniel< / surname>
  </name>
  <birth>
    <date>21 Feb 1834< / date>< / birth>
  <death>
    <date>9 Dec 1905< / date> < / death>
< / person>
```

即使不熟悉XML语言，也能指出这些数据描述了一个名叫Judson McDaniel的人，出生于1834年2月21日，死于1905年12月9日。事实上，即使该数据有漏洞或者破坏了，仍然可能获得大部分信息。而对于大多数专用电子表格或者字处理格式，则不能这样说。

### 1.2.3 数据交换

因为XML易理解、非专有、易读写，所以对于在不同的应用程序之间的数据交换来讲，XML是一种极好的格式。正在开发的一种XML格式是开放金融交换格式(OFX)，OFX用来让个人金融程序如Microsoft Money和Quicken交易数据。这种数据可以在程序之间来回发送，并且可以与银行、交易所等此类机构交换数据。

XML是一个非专有格式，无版权、专利、商业秘密或者智力产权的制约。当XML富有表现力时，XML还极易使人类和计算机程序读和写。因此说，对于这些交换语言来说，XML是一种不可置疑的选择。

XML有助于用户避免陷入这些特殊的程序，因为用户的数据可以用这种程序编写，或者因为能接收这种程序的专有格式。在计算机图书中，大多数出版商要求以Microsoft Word格式提交书稿。因此，大多数计算机图书作者即使喜欢用WordPerfect或者Nisus Writer，也还必须

使用Word。对于其他公司来说，除非他们的程序能读写Word文件，否则想挤进该市场是相当困难的。在工程方面，整理未归档整理的Word文件格式是一种有限时间和资源的巨大投资。大多数其他字处理软件对读写Word文件有些限制，而且一般地会丢失掉Word文件的风格、修订标记和其他重要特征(因为Word的文档格式是不能制作文档的、专有的和经常改变的)。即使图书作者喜欢使用其他简单的程序，Word仍趋向取得竞争成功。如果某种通用的字处理格式是由XML开发的，那么图书的作者就可以自己选择字处理软件，而不必受到出版商的要求限制。

### 1.2.4 结构化数据

对于大型复杂的文档，XML是一种理想语言。它不仅允许指定文档中的词汇，还允许指定元素之间的关系。例如，如果把许多销售合同放在一个Web中，那么就需要每个合同都有一个电话号码和电子邮件地址。如果是正在向数据库中输入数据，那么就能保证不会遗漏字段。甚至当没有输入数据时，还可以提供一个缺省值来使用。

XML还提供一种集成多个数据源数据并且作为单个文档显示的客户机端包括机制。数据位置甚至可以重新安排。根据用户的操作，部分数据可以被显示或者隐藏。当使用大型信息仓库如关系数据库时，这种结构非常有用。

## 1.3 XML简史

像许多标准一样，XML是从同一时期的许多地方逐渐形成的。要想准确地指出是某人某时创建或者说“这就是XML开始使用的地方”，这是非常困难的。然而，有一点儿很清楚，即XML有两个主要先驱者：SGML和HTML。这两种语言都是非常成功的标记语言，并且在标记语言的发展史中，HTML可能是最广泛使用的语言。然而，SGML和HTML都有很明显的缺点。为了把这两种标记语言的优点结合起来并消除其缺点，出现了一种新的解决方法，即XML。

首先推动XML发展的是SGML，SGML是标准通用化标记语言的简称。SGML是一种文档语义标记的国际标准，已经使用了十余年。SGML的语义标记有助于计算机分类和索引，并且可以扩展成处理新数据格式的各种方法。

SGML主要应用于需要处理大量高度结构化数据的防卫区和各种其他工业领域。然而，SGML相当复杂，更不用说其昂贵的价格。例如，Adobe FrameMaker的标准版本价格为850美元。Adobe FrameMaker+SGML是以价格1 995美元推出的，并且是比较便宜的SGML软件之一。SGML文档管理软件常常耗资数万美元。虽然SGML有其优点，但是由于其过于复杂和昂贵，使得有些人无法把他们的家族树放置Internet上。

XML的第二个来源是HTML。1990年，Tim Berners-Lee在CERN开发出了HTML，这是作为SGML的简单替代品而设计的，可以被普通人理解，不用支付昂贵的授权工具费用。HTML获得的成功超出了其最初的设想。

遗憾的是，HTML还不能很好地广泛使用。首先，它由数量较少的固定的标记集如〈P〉、〈STRONG〉和〈TITLE〉所限制。如果开发人员需要一种〈BOOK〉，或者〈FATHER〉或者〈MOLEC ULE〉标记，那么他就太不走运了。HTML缺少SGML的柔性和适应性。

90年代初，Marc Andreesen和Eric Bina以及其他几个人在伊利诺思大学开发了Mosaic。在

Illinois大学他们只作为NCSA廉价劳动力，每小时有几美元。在Mosaic中，他们增加了HTML中对〈IMG〉的标记的支持，这时Web迅捷发展起来了。两年后，Andreesen和Bina以及其他人在Netscape公司开始在Navigator中飞速地增加标记，所有这些标记都成了每一个新beta版本的标准。然而，在这整个过程中，没有人担心维护作为一种SGML应用程序的HTML纯度，特别是，像〈CODE〉，〈HL〉和〈TABLE〉的语义标记可以自由地与像〈CENTER〉，〈I〉和众所周知的〈BLINK〉的类型标记混合一起。

到了1996年，HTML的语义根被图形设计人员的需求完全破坏了。虽然HTML标记不在描述其内容的含义，但是当使用Web浏览器显示时，还可以描述其内容的形式。这种方法似乎解决了索引机器人、禁止访问和其他许多用途的难题。而实际上，这些问题是数年前SGML已经解决的问题 。在这一点上，SGML用户组会议开始散布了有关HTML是如何糟糕的谣言。然而，对于大多数用途方面，SGML仍然过于复杂和昂贵。

1994年末和整个1995年，Web站点开始发展起来了。像Time Warner一样的老守护介质公司和像clnet一样的启动程序开始以一种显然不清楚的规模在公共Web站点工作。许多公司中的雇员开始把其内部文档放在公司局域网络上的专用Web服务器上，这仅是因为它比用公文或者像Lotus Notes专用系统更容易分布。

当这些Web站点发展到由成千上万个Web页面组成时，许多公司发现他们需要使用像Informix或者Oracle一样的大型关系数据库来存储Web内容，然后这些内容都堆到使用标记的模板中。较小的站点开始使用像Frontier或者Cold Fusion产品来完成类似的工作。实际上，在不同工业领域的许多不同的公司正在步入使用类SGML方法的王国中，他们正在或多或少独立地重新开发自己的SGML版本。除此之外，为了进行这项工作，这些公司需耗费数万美元。显然，迫切需要一种新方法。

1996年夏天，Sun Microsystems的Jon Bosak开始开发W3C SGML工作组(现在称为XML工作组)。该工作组的目标是创建一种SGML，使其在Web中，既利用SGML的长处，又保留HTML的简单性。最终，这种目标软件逐渐被视为XML，尽管对其还有各种各样的称法。另外一些名称包括MAGMA(通用化标记应用程序的最小体系结构)、SLIM(Interet标记结构化语言)和MGML(最小通用化标记语言)。在这几种情况中，缩略语名称似乎比全名更重要。

XML被认为是一种SGML，不仅保持了原有的功能性和可扩展性，而且更加易用价廉，对普通用户已经足够了。对于一些重要目标，XML已经达到了。在简单性方面，XML软件达到或者稍逊于SGML。在功能方面，仅有很少的SGML功能XML不能实现。在费用方面，许多与XML一同使用的免费工具已经可以使用了，并且还在不断地增加，Adobe公司已经宣布，打算在其FrameMaker中增加对XML的支持。

Tim Bray是开发XML的主要推动者，并且还是XML1.0版本的两位主要作者之一，Bray在管理新牛津英语字典项目(OED)这项“世界上最好的工作”方面已经花费了几年宝贵时光。他从事OED项目的经验已经用在了XML的工作上。其实，他正在努力开发标记语言，并且希望能够有在1987年始OED项目时的工作热情。特别是，Bray希望达到下面一些目标:

- 一种编程人员成够实现的简单语言
- 一种不限制于美国英语的语言
- 易于搜索引擎索引的文档

Tim Bray和伊利诺思大学的C.M.Sperberg-McQueen编写了XML规则文档的大部分原始内