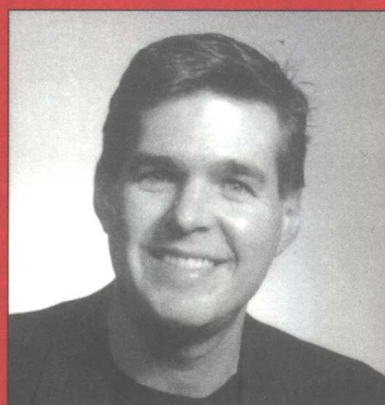
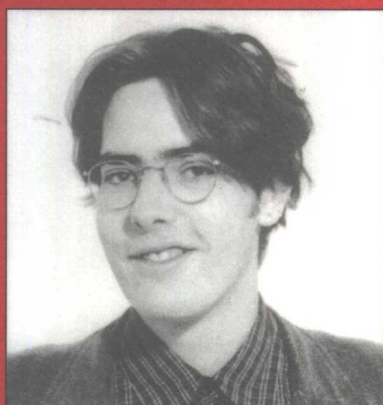
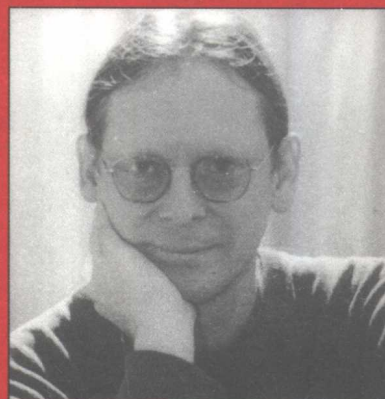
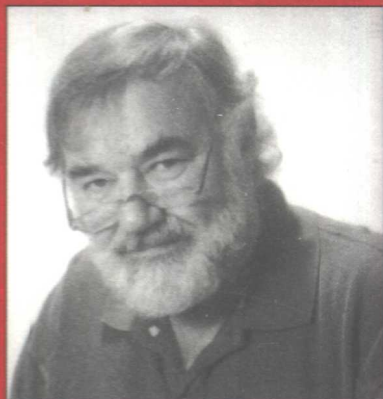




PROGRAMMER TO PROGRAMMER™

Beginning XHTML

Wrox 程序员参考系列



XHTML

初级编程

(英) Frank Boumphrey 等著

马朝晖 等译

p2p.wrox.com 提供在线支持



机械工业出版社
China Machine Press



乐思

Wrox程序员参考系列

XHTML初级编程

(英) Frank Boumphrey 等著

马朝晖 等译



机械工业出版社
China Machine Press

本书由浅入深详细地介绍了与XHTML相关的所有主题。主要包括：XHTML的基础知识，XHTML页面的设计问题，介绍表单及脚本编程等。本书实例丰富、简单实用，适合于学习网页制作和站点设计的初、中级读者。

Frank Boumphrey, et al: Beginning XHTML.

Authorized translation from the English language edition published by Wrox Press.

Original copyright © 2000 by Wrox Press. All rights reserved.

Chinese simplified language edition published by China Machine Press.

本书中文简体字版由英国乐思出版公司授权机械工业出版社独家出版。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：01-2000-4102

图书在版编目（CIP）数据

XHTML初级编程/（英）邦弗瑞（Boumphrey, F.）等著；马朝晖等译. —北京：机械工业出版社，2001.11

（Wrox程序员参考系列）

书名原文：Beginning XHTML

ISBN 7-111-09334-8

I. X… II. ①邦… ②马… III. 超文本标记语言，-XHTML-程序设计 IV. TP312

中国版本图书馆CIP数据核字（2001）第045578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：张鸿斌 谢君英

北京第二外国语学院印刷厂印刷·新华书店北京发行所发行

2002年1月第1版第1次印刷

787mm×1092mm 1/16·35.75印张

印数：0 001-5 000册

定价：55.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

前言

本书的主题是XHTML（可扩展超文本标记语言）。HTML（超文本标记语言）是推动Internet的发展并将全世界数百万用户连接起来的语言，而XHTML是HTML之后的下一代标记语言。

学完本书后，你将了解如何使用XHTML建立Web页面。在此过程中，你将学习一些XML（XHTML的基础）的背景知识，但是大体上本书是通过提供实践练习的方式讲解如何建立Web页面。本书涉及以下基本问题：

- XHTML到底是什么？
- XHTML与HTML和XML有什么差别？
- 如何建立Web页面和Web站点？
- 如何使Web页面具有交互性？

我们将采用彻底全面的方式配合大量完整的实例来回答这些问题。所以，即使你对Web技术完全陌生，通过学习本书，你也将深刻地理解XHTML究竟是什么，以及如何利用它建立强大的Web页面。

本书适合的读者

本书是Wrox Beginning系列书籍之一，所以我们将从头开始以节奏均匀的教程方式向你讲解XHTML。如果你是位经验丰富的Web开发人员，那么本书可能不适合你。

本书适合下面两类初学者：

- Web编程的初学者并选择了XHTML作为起点。如果喜欢计算机技术而且希望学习速度快，那么本书是合适的。但是本书的内容很多，你要有思想准备。
- 具有一些编写HTML的经验，但不是专家级的程序员，而且实际上不了解XML、XHTML或相关的其他技术。本书会介绍相关的背景知识，但是在代码细节实现和创建Web页面时的思维方式方面有许多差异。本书还会涉及较深入的技术。

作为学习本书的起点，你只需要知道基本的知识以及如何在文本编辑器中编写文本和如何运行Web浏览器。如果你从未写过一程序代码，那也没什么害怕的——本书就是为你准备的。本书的基本目标是教会你编写令人激动的动态Web页面，并为你打开通向其他技术的大门。

本书的内容

本书在概念上分为三个部分，覆盖了所有与XHTML相关的主题。你将高兴地发现，我们采用了符合逻辑、次序合理的方式来讲解，这使你能够充分理解XHTML的工作原理、作用以及如何使用它设计交互式的Web页面。

MSD 8/02

第一部分使用大量实例讲解XHTML的基本知识:

第1章和第2章解释XHTML如何配合其他Web技术, 以及XHTML与HTML之间的差异。

第3~8章讲解XHTML语言的基本概念。我们从使用XHTML创建Web页面这样的简单实例开始, 然后学习增加Web页面之间的链接, 增加图像、列表和表格的方法。

第9章将更详细地讨论样式单, 并讲解如何将内容与样式规则分离。

第10章通过讨论XHTML的父语言XML, 讲解了XHTML语言背后的一些理论。

第二部分讨论与构造XHTML页面相关的设计问题。

第11章和第12章涉及站点和页面设计, 关于如何构造站点以及使页面易于使用的一些问题。

第13章讨论了如何为不同的计算机HTML浏览器和其他类型的用户代理编写页面。

第14章讨论如何在页面中包含多媒体内容。

第三部分介绍表单和脚本编程, 向你讲解如何使Web页面具有交互性:

第15章讲解如何使用表单获取用户的输入。

第16~18章讲解JavaScript的基本知识, 以及如何使用它为Web页面增加趣味性。

第19章解释如何使用框架来编写应用于多个Web页面的脚本。

第20章介绍如何使用Mozquito工具创建交互式的表单。

最后, 附录提供了关于XHTML标记、CSS特性、JavaScript对象和其他信息的有价值的参考资料。

使用本书的条件

你只需要做两件事:

- 需要一个文本编辑器以便编辑Web页面。你可能有一些新奇的HTML编辑器, 但是既然你要学习XHTML (而且有些工具编写的HTML代码质量很差), 所以我们建议使用简单的文本编辑器来编写Web页面。
- 需要一个Web浏览器以便查看Web页面。我们在本书中使用微软的Internet Explorer 5.0 (IE 5)。尽管不必使用多个Web浏览器, 但我们仍建议你使用多种Web浏览器, 这使你能够看到Web页面在不同浏览器中的表现有何差异。

本书约定

在本书中, 我们使用了许多不同的文本样式和布局, 以便协助区分不同种类的信息。下面是我们采用的例子样式及其含义的解释:

实践——它们如何工作?

- 1) 每一步都有一个编号。
 - 2) 按这些步骤进行操作, 使例子运行。
 - 3) 然后阅读“工作原理”, 了解发生的事情。
- 以不加框的楷体字表示建议、提示和背景信息。

作者介绍

Frank Boumphrey

Frank当前就职于Cormorant Consulting，这是一家专门从事医学和法律文档的公司。

早在穿孔卡和机器语言的时代，他就开始进行程序设计。他编写的第一个程序用于区分前苏联进口的ICBM和鹅绒。

后来，他放弃了编程并成为一名医师，最终在美国中西部的大型机构中担任脊椎外科的教授和主任医师。此间他参与了将MRI引入医学界的工作。

进入半退休状态后，他又恢复了对计算技术的兴趣。现在，他正在尝试帮助医学机构将他们的病例记录以一种半相关的方式组织起来，还向医学人员和保健人员讲授文档问题。有趣的是，他还应法律公司的请求协助他们将医学数据库进行重新组织！

Frank参加了W3C的XHTML工作组，他的主要任务是协助进行Web上的XML开发。

Cassandra Greer

Cassandra Greer 1966年出生于加利福尼亚的 Glendale，但是她一直在其他地方生活。她声称来自旧金山，因为这是她最喜欢的美国城市，而且她与姨妈和表兄妹们住在这里。现在她住在德国慕尼黑。

她在Brigham Young大学的德语系得到了文学士学位，在语言学系得到了文学硕士学位（TESOL）。同时，她在BYU英语中心工作，向各国的学生教授英语，而且在计算机实验室工作并提出了一项HyperCard方案。毕业后，她继续在慕尼黑从事教学工作，并且在BMW、Siemens、Xerox Engineering Systems、CompuNet、慕尼黑技术大学和慕尼黑大学物理系等处教授技术英语并从事翻译工作多年。

最近，Cassandra偶然结识了Mozquito.com软件公司的成员，在那里不但她的英语教学经验 and 写作技巧派上了用场，而且在Mac机和网络方面长达10年的经验也有了用武之地。

Dave Raggett

Dave Raggett博士是从HP实验室转到W3C工作的，是W3C在HTML、表单、语音浏览器和数学方面的联系人。他曾经参与对HTML进行模块化以满足移动设备和电视的需求，还参与下一代Web表单的开发，研究用于访问Web的语音交互技术和下一代脚本语言。他从牛津大学毕业后，结婚并生了两个孩子，现在住在英国西部的Bath。

以有框的楷体表示不应该被忽视的重要信息。

读者支持

我们尽可能使本书准确、令人愉快，但是实际效果要由你来评价。请把你的看法通过电子邮件地址feedback@wrox.com告诉我们。

下载源代码

当你使用本书中的例子时，可能喜欢自己亲手键入所有代码。许多读者喜欢这么做，因为这是熟悉所用的编码技术的好办法。

无论你愿否键入代码，我们的Web站点上都提供了本书的所有源代码，网址为<http://www.wrox.com>。

如果你喜欢键入代码，那么可以使用代码文件检验结果。如果你不喜欢键入代码，那么从我们的Web站点下载源代码就是必须的！

无论采用哪种方式，这些源代码都会帮助你更新和调试代码。

勘误

我们已经尽力确保文本和代码中没有错误。但是难免会出错，你可以从<http://www.wrox.com>得到我们所有书籍的勘误表。如果你发现了一个未被报告的错误，请告诉我们。

我们的Web站点提供其他信息和支持，包括我们的所有书籍的代码、样章、即将出版的书籍的预览以及相关主题的文章和意见。

本书英文原书书名：Beginning XHTML

英文原书书号：ISBN 1-861003-43-9

本书主要由马朝晖、陈美红翻译，参与翻译、录入、审校的还有刘丽珍、王建芬、杨帆、邹辉、潘浩、楼涵、董小蕾、王悦、李军、罗伟、鲍广华、瞿兰、陆明、宋丽、杨立军、李鸣、马晓云、成荣光等。

Jenny Raggett

Jenny Raggett是一位技术方面的自由作家，其专长是向外行解释技术信息。她最近参与W3C的工作，为非技术人员编写许多关于Web新技术的文档。新的工程包括为环保组织规划Web站点，她参与这个活动是因为对环保有浓厚的兴趣。

Sebastian Schnitzenbaumer

Sebastian Schnitzenbaumer 1977年出生于慕尼黑，他的童年是在印度洋中的一个岛上度过的。他十几岁时回到慕尼黑后，完成了学业并开始从事Web设计咨询服务。多年来，建立Web应用程序一直是他的主要专业，其范围从公众Web站点到大公司的IT管理。随着时间的流逝，对更交互性的Web应用程序的需求与日俱增：通过Web管理公司数据库，在浏览器内编辑Web站点，进行投票和调查，等等。Web上到处都有交互活动和反馈，而Web表单是其中起主要作用的构件。

1998年，他与其同事兼朋友Malte Wedel合作建立了Mozquito.com（这家国际软件公司最初称为Stack Overflow）。他们开发了更高级的HTML Web表单的方案，而后加入了W3C，Sebastian成了W3C HTML工作组的成员，参与开发XHTML。

Ted Wugofski

Ted Wugofski曾经对多种即将出现的技术进行技术指导：人工智能、计算机电话通信、数字电视和移动Internet。Ted致力于开发将Internet引入非传统计算设备的产品。Ted参加了各种W3C工作组，并领导高级电视系统委员会中的一个组为数字电视定义一种标记语言。你可以通过ted.wugofski@ieee.org与他联系。

目 录

前言	
作者介绍	
第1章 Web、HTML和标记语言	1
1.1 Web是什么	1
1.2 Web的历史	2
1.3 浏览器	2
1.3.1 浏览器的历史	3
1.3.2 浏览器和HTML	4
1.4 文本编辑器	5
1.4.1 微软的FrontPage	5
1.4.2 Allaire的Homesite	6
1.4.3 Notepad	7
1.5 浏览Web页面	7
1.6 HTML	9
1.6.1 元素和标记	10
1.6.2 标记属性	11
1.7 标记和标记语言	11
1.7.1 语义标记	12
1.7.2 HTML标记	12
1.7.3 结构性的、样式性的和描述性的标记	14
1.8 总结	15
第2章 从HTML到XHTML	16
2.1 SGML	16
2.2 XML	18
2.2.1 良构的XML文档	18
2.2.2 XML分析器	20
2.3 XHTML	23
2.3.1 XHTML与HTML的差别	24
2.3.2 向后兼容原则	27
2.3.3 W3C	28
2.4 总结	29
第3章 起步	30
3.1 文本文档的结构化	30
3.2 文档的总体结构	31
3.2.1 文档内容	31
3.2.2 文档结构	31
3.2.3 文档样式	31
3.2.4 样式和结构	32
3.2.5 文档树	33
3.2.6 分析文档	33
3.3 XHTML文档的结构	34
3.4 XHTML文档的组成	36
3.4.1 版本声明	37
3.4.2 DOCTYPE声明	37
3.4.3 文档头部	37
3.4.4 文档主体	38
3.4.5 注释	38
3.5 浏览器和内置的语义	38
3.6 基本的XHTML行内描述性元素	42
3.7 基本的XHTML行内样式元素	44
3.8 列表	46
3.8.1 无次序列表	46
3.8.2 有次序列表	48
3.8.3 定义列表	49
3.8.4 列表和不同的媒体类型	50
3.9 属性	51
3.9.1 通用的属性	51
3.9.2 表示性的属性	54
3.10 级联样式单	55
3.10.1 什么是级联样式单	55
3.10.2 CSS类	58
3.10.3 链接到外部样式单	59
3.10.4 真正级联的级联样式单	61
3.11 总结	61
第4章 链接和嵌入的对象	62
4.1 超文本链接	62

4.2 <a>元素	63	5.10 可伸缩的矢量图形	116
4.2.1 锚标记作为链接源	63	5.11 总结	116
4.2.2 相对链接——相同文件夹中的文件	64	第6章 表格	117
4.2.3 相对链接——不同文件夹中的文件	65	6.1 表格结构	118
4.2.4 绝对链接——不同服务器上的文件	67	6.2 XHTML和表格结构	119
4.2.5 指针: 使用<a>作为目标	69	6.2.1 建立简单的表格	119
4.2.6 其他文件中的目标	72	6.2.2 增加列标题和表格标题	120
4.2.7 使用图像作为源链接	75	6.2.3 为表格增加进一步的结构	122
4.3 使用不同的协议进行链接	76	6.3 跨行和跨列	123
4.3.1 ftp协议	76	6.3.1 使用colspan和rowspan	123
4.3.2 mailto: 指令	76	6.3.2 summary属性	126
4.4 其他连接元素	78	6.4 表格的样式化	126
4.5 总结	83	6.4.1 样式单和XHTML属性	126
第5章 图像	84	6.4.2 边框、框架和规则	127
5.1 位图和矢量图形	84	6.4.3 使用CSS特性画框架和边框	128
5.1.1 位图	85	6.4.4 单元格间距和填充	130
5.1.2 矢量图形	85	6.4.5 CSS padding特性	133
5.1.3 无损压缩和有损压缩	86	6.4.6 背景色	134
5.2 文件格式	86	6.4.7 高度和宽度	135
5.2.1 GIF文件格式	87	6.4.8 相对于文档对齐表格	138
5.2.2 JPEG文件格式	88	6.4.9 对齐表格与其他表格	140
5.2.3 PNG文件格式	90	6.4.10 嵌套表格	141
5.3 元素	90	6.4.11 在表格内对齐文本	142
5.4 图像属性	92	6.5 使用表格进行布局	144
5.5 图像效果	100	6.5.1 标题和导航条	145
5.5.1 使用透明度	101	6.5.2 增加公告栏	145
5.5.2 使用动画	103	6.5.3 使用表格设置页边距	146
5.6 图像映像	103	6.6 在非可视和不能滚动的媒体中的表格	147
5.6.1 坐标	104	6.7 总结	148
5.6.2 区域、形状和单位	104	第7章 框架	149
5.6.3 热点区域	105	7.1 布置框架集	149
5.6.4 其他area属性	108	7.1.1 框架集: 分割窗口	149
5.7 有效地使用图像	109	7.1.2 <frameset>元素	151
5.7.1 改变图像的大小	109	7.1.3 <frame>元素	155
5.7.2 低带宽图形	113	7.2 嵌套框架	156
5.8 到何处寻找图形	115	7.2.1 嵌套<frameset>元素	156
5.9 有用的图形工具	115	7.2.2 导入“框架集”文件	157

7.3 目标框架	159	9.5 如何继承CSS特性	193
7.4 行内框架	162	9.6 设置字体颜色	194
7.5 CSS和框架	163	9.7 设置超文本链接的颜色	199
7.6 <noframes>元素	163	9.8 设置背景色	200
7.7 使用框架进行显示	164	9.9 设置背景纹理	201
7.8 总结	165	9.10 边框、空白边和空白间距	205
第8章 元数据	166	9.10.1 设置页面空白边	206
8.1 元数据	166	9.10.2 特定标题前后的间距	207
8.2 文档信息	167	9.10.3 首行缩进	208
8.2.1 <head>元素	168	9.10.4 边框	209
8.2.2 <title>元素	168	9.11 用于控制字体的CSS特性	209
8.2.3 <meta>元素	168	9.11.1 设置字体尺寸	210
8.2.4 <link>元素	171	9.11.2 设置特定XHTML元素的字体尺寸	211
8.2.5 <base>元素	171	9.11.3 字体灰度	212
8.2.6 PICS	172	9.11.4 字体样式	213
8.2.7 RDF	172	9.11.5 使文本成为大写或小写	214
8.3 名称空间	174	9.11.6 行高	215
8.3.1 以URI作为惟一名称	174	9.11.7 设置字体名称	215
8.3.2 名称空间声明	175	9.12 创作自己的命名样式	218
8.3.3 名称空间缺省设置	175	9.13 总结	220
8.4 HTTP头	176	第10章 XML简介	221
8.4.1 客户-请求HTTP头	176	10.1 什么是XML	221
8.4.2 MIME类型	178	10.2 数据革命	225
8.5 总结	182	10.3 数据显示以外的问题	225
第9章 样式单	183	10.4 如何创建XML文档	226
9.1 将样式应用于Web	183	10.4.1 良构的文档	226
9.1.1 出现样式单以前的情况	183	10.4.2 合法的文档	230
9.1.2 样式单的出现	184	10.4.3 编写DTD	230
9.1.3 使用样式单的优点	184	10.4.4 XML的样式化	233
9.1.4 级联样式单语言	185	10.4.5 XSL	237
9.1.5 支持CSS	186	10.5 总结	237
9.2 CSS样式单如何工作	187	第11章 站点设计概念	239
9.2.1 格式化对象	187	11.1 一般概念	239
9.2.2 样式单为什么“级联”	187	11.2 规划	240
9.2.3 样式单特性的继承	187	11.2.1 市场分析	240
9.3 如何编写CSS	187	11.2.2 故事板	240
9.4 使用style属性应用本地样式	191	11.2.3 可伸缩性	241

11.3 链接策略	241	第13章 不同的媒体类型	273
11.3.1 链接范例	241	13.1 什么是媒体类型	273
11.3.2 链接的注意事项	243	13.1.1 已知的媒体类型和媒体类型描述符	273
11.3.3 行内链接	243	13.1.2 适合于媒体类型的样式单特性	274
11.4 链接和元信息	244	13.2 将样式单与媒体类型相关联	275
11.4.1 <link>元素	244	13.2.1 使用<style>元素	275
11.4.2 链接的标签	245	13.2.2 使用<link>元素	281
11.5 构造站点结构	246	13.2.3 使用@media规则	283
11.5.1 线性结构	246	13.2.4 使用@import规则	285
11.5.2 层次化结构	246	13.3 打印媒体和页面式媒体	286
11.5.3 复合结构	247	13.3.1 分页特性	286
11.5.4 模块化站点设计	248	13.3.2 @page规则	292
11.5.5 Internet应用程序	249	13.4 处理媒体类型的策略	297
11.6 品牌	250	13.4.1 设备的限制	298
11.6.1 商标和徽标	250	13.4.2 用户代理的限制	299
11.6.2 颜色方案	250	13.4.3 人机界面的限制	300
11.6.3 公司样式	251	13.4.4 策略	300
11.7 页面设计	251	13.5 总结	306
11.7.1 合法性检验	252	第14章 多媒体	308
11.7.2 个性化	252	14.1 播放音频	308
11.8 站点管理	252	14.1.1 音频插件	308
11.9 总结	253	14.1.2 在文档内放置音频	310
第12章 页面设计	254	14.2 播放视频	318
12.1 在Web页面上使用颜色	254	14.3 流式的音频和视频	321
12.1.1 背景色	254	14.3.1 Web服务器	321
12.1.2 文本的颜色	254	14.3.2 建立你的流式内容	322
12.1.3 Web页面上颜色的组合	255	14.3.3 引用你的流式内容	323
12.2 超文本链接	255	14.4 集成多媒体和SMIL语言	324
12.3 背景图像	256	14.5 总结	335
12.4 使用字体时的考虑因素	256	第15章 XHTML表单	337
12.5 文本的布局	259	15.1 什么是表单	337
12.6 使用CSS进行页面布局的一些例子	261	15.1.1 客户/服务器交互和表单	337
12.7 关于浏览器的考虑因素	265	15.1.2 表单的提交	338
12.7.1 HTML浏览器	265	15.2 <form>元素	340
12.7.2 应付不同的浏览器	267	15.3 基本表单	341
12.8 可访问性准则	270	15.4 使用控件	343
12.9 总结	272	15.5 常见属性	343

15.5.1 name属性	343	16.9.2 填充数组 (一)	384
15.5.2 value属性	344	16.9.3 填充数组 (二)	386
15.5.3 disabled属性	344	16.10 循环和多重操作	386
15.5.4 <input>元素	344	16.10.1 for循环	386
15.5.5 <textarea>元素	350	16.10.2 while循环	389
15.5.6 <select>和<option>元素	350	16.11 用户定义的函数	393
15.5.7 表单中的XHTML标记	352	16.11.1 声明函数	393
15.6 表单和可用性	352	16.11.2 调用函数	393
15.7 总结	353	16.11.3 执行动作的函数	395
第16章 JavaScript	354	16.11.4 返回值的函数	396
16.1 什么是脚本语言	355	16.11.5 变量范围和函数	397
16.2 在文档中包含脚本	355	16.12 总结	400
16.3 对象、特性、事件和方法	358	第17章 在Web页面中使用脚本	401
16.3.1 浏览器对象	358	17.1 浏览器对象	401
16.3.2 方法	358	17.1.1 window对象	402
16.3.3 参数	360	17.1.2 navigator对象	408
16.3.4 方法和函数	361	17.1.3 document对象	409
16.3.5 特性	362	17.2 JavaScript对象	410
16.3.6 事件	364	17.2.1 String和Math对象	411
16.4 JavaScript语言	366	17.2.2 Date对象	411
16.4.1 大小写敏感	366	17.2.3 form对象	411
16.4.2 标志	366	17.3 浏览器事件	412
16.4.3 空格	366	17.3.1 事件和事件处理器: 事件捕获	412
16.4.4 注释	367	17.3.2 事件处理器	413
16.4.5 直接量	367	17.3.3 事件例子	414
16.4.6 标识符	367	17.4 浏览器探测	417
16.5 JavaScript中的输出	368	17.4.1 探测出浏览器	418
16.6 变量	372	17.4.2 探测出色深	420
16.7 变量类型	374	17.5 cookie	421
16.7.1 字符串	375	17.5.1 cookie包含什么	421
16.7.2 数字	376	17.5.2 设置cookie	422
16.7.3 将字符串转换为数字	377	17.5.3 谁可以读cookie	422
16.7.4 布尔值	379	17.5.4 获取cookie	423
16.7.5 对象	380	17.5.5 修改cookie	424
16.8 控制程序流程	380	17.5.6 cookie的限制	424
16.9 数组	383	17.5.7 cookie例子	424
16.9.1 创建数组	384	17.6 总结	427

第18章 文档对象模型	428	20.1 Mosquito Factory入门	472
18.1 文档对象模型的基本原理	428	20.1.1 Mosquito的诞生	473
18.1.1 什么是文档对象模型	428	20.1.2 在Web出版项目中使用Mosquito	474
18.1.2 DOM的级	429	20.1.3 安装Mosquito	474
18.1.3 文档的树型结构	430	20.1.4 Mosquito Factory编辑器和检验器	474
18.2 第0级DOM	432	20.1.5 使用Mosquito和其他创作工具	476
18.3 第1级DOM	440	20.2 表单标记语言	477
18.4 工厂方法	447	20.2.1 XHTML对现有浏览器的可扩展性	477
18.4.1 创建元素和文本节点	448	20.2.2 FML的好处	478
18.4.2 编辑文本节点	450	20.3 用FML创建电子商务定单表单	479
18.4.3 创建属性和值	450	20.3.1 基本的地址表单	479
18.5 总结	453	20.3.2 创建产品目录	491
第19章 多框架JavaScript应用程序	454	20.4 总结	504
19.1 链接多框架	455	附录A XHTML标记和属性	505
19.2 在框架之间传递信息	457	附录B 使用HTML Tidy	513
19.3 在页面中存储数据	460	附录C 在Web页面中使用颜色	518
19.4 写新页面	465	附录D 样式单特性	522
19.4.1 创建新窗口	466	附录E JavaScript参考	536
19.4.2 写到框架	468	附录F 资源	543
19.5 总结	470	附录G 支持和勘误	552
第20章 通过Mosquito使用XHTML	471		

第1章 Web、HTML和标记语言

本书的主题是XHTML，即可扩展超文本标记语言(eXtensible HyperText Markup Language)。XHTML基于HTML，而且与XML（可扩展标记语言，eXtensible Markup Language）的原则相一致。本书第1章引出了许多定义，特别是针对那些对Web比较陌生的人。因此，我们将对XHTML和XML的讨论留到第2章，而在本章中讨论Web基本知识和HTML。对Web和HTML非常熟悉的读者可以跳过本章，或者快速阅读本章。

在本章中，我们将讨论以下主题：

- Web是什么。
- 不同的浏览器和HTML编辑器。
- HTML语言的基础知识。
- 浏览Web页面时发生的事。
- 标记和分析是什么。

1.1 Web是什么

Web和Internet是两个常常被混用的术语，有必要先在头脑中将这两个概念分开。Internet是由相互链接的节点（计算机）组成的网络，它的起源是1960年美国国防部（DOD）的一个工程。在受核威胁的时代里，DOD担心他们的通信会在一次核打击中被摧毁，所以他们开发了一个计算机网络，可以在部分网络路由受到破坏的情况下继续工作。

所以，Internet是由相互连接的节点组成的网络，它采用的方式就像大城市的地铁系统（地铁系统是由相互连接的地铁站组成的网络）。地铁系统用于将人从一个地方运送到另一个地方，而Internet用于将信息从一个地方运送到另一个地方。

地铁系统建立在钢铁（和其他材料）的基础之上，而Internet使用一套连网协议来传输信息，即TCP/IP（传输控制协议/网际协议）。连网协议只是描述信息包的一种方法，它使信息包可以沿电话线、电缆或T1线从一个节点发送到另一个节点，直至到达预定的目的地。

当用户让浏览器去得到一个Web页面时，浏览器使用TCP协议将这个指令打包。TCP是一个传输协议，它为指令提供可靠的传输格式。它确保整个消息被正确地打包以便传输（以及在消息到达目的地后，进行正确的解包和组装）。

在数据片段经网络发送前，它们需要被加上地址，就像你在发送信件前要在信封上写地址一样。所以，第二个协议——HTTP（超文本传输协议），将地址标签放在数据片段上。WWW使用HTTP协议将信息从一台机器传输到另一台机器。当你看到一个URL的前缀是http://时，就可以知道所用的Internet协议是HTTP。HTTP所做的只是使不同的计算机可以相互通信。

另一方面，WWW是在Internet上运行的软件。Web的体系结构基于客户/服务器模型。用户的浏览器是客户，它获取远程计算机（这台远程计算机被称为Web服务器，它存储供访问的Web

页面和文件)上保存的信息。这个Web服务器可以位于本地网络,也可以位于地球上的其他地方,只要浏览器可以使用HTTP找到它,它的位置就无关紧要。当Web第一次出现时,它的独特之处是:它可以为Internet上的不同服务提供图形的(而非基于文本的)界面。

1.2 Web的历史

可以说Web起源于书籍写作。文档或书籍具有结构,如段落和页,而且它的文本可以包含对其他文档和书籍的引用。索引帮助在书中定位特定的信息,而章节间的交叉引用使我们可以沿某条路径阅读此书。编号的行和句子使读者可以直接跳到感兴趣的某一点。这个基本的文档模型也可以用于Web上的信息:文本的页面具有到其他页面或页面中的部分的链接。

尽管我们知道Web是新生事物,但电子链接的信息这个概念出现得很早。例如,40年代,Vanevar Bush在“*As We May Think*”一文中描述了一个称为memex的系统。其他梦想者包括Douglas Englebart(他参与开发了鼠标和超文本)和Ted Nelson(他提出了“超文本”这个术语)。

讲到现代Web时,我们不能不提到CERN,这是一个位于瑞士日内瓦附近的国际高能物理研究中心。1989年,Tim Berners-Lee和Robert Caillau在CERN联合发明了一个链接的信息系统,这个系统可以跨大范围的不同计算机系统进行访问。当时,许多人使用TEX(一种基于命令的文档建立系统)和Postscript来建立文档,极少人使用SGML(标准通用标记语言,将在第2章中讨论)。Tim意识到需要用某些更简单的东西来应付所有类型的机器(从“哑终端”到高端图形X Windows工作站)。HTML(超文本标记语言)被设计为一个非常简单的解决方案,并且与HTTP网络协议相配合。尽管这两者后来加入了一些较复杂的内容,但其最初的简单性使编写浏览器和服务器的(主要是后者)非常简单,而且促进了Web最初的发展。

CERN在1991年通过一个称为www-talk的讨论列表公布了Web概念。不久,认同此思路的他人参加进来,通过建立Web站点和实现浏览器(如Cello、Viola和MidasWWW),协助发展Web。当位于Urbana-Champaign的超级计算机应用国家中心(NCSA)支持Marc Andreessen和Eric Bina开发Mosaic浏览器时,出现了突破。后来,它被移植到PC和Mac机上并获得了巨大成功。Web呈指数式增长,这使其他基于Internet的信息系统(如WAIS、Hytelnet、Gopher和UseNet)黯然失色。

标准化的目的主要是使现有的做法变得正规,或者在新的特性被广泛部署前对其进行定义。1994年年中,Internet工程任务组(IETF)开发了HTML 2.0。1996年年初,WWW协会(W3C)开发了HTML 3.2。1997年出现的HTML 4.0中加入了表格、表单和对象等新特性。

最近,HTML 4.01修正了HTML 4.0规范中的许多bug,并为XHTML规范提供了基础。XHTML 1.0是用XML重写的HTML 4.01规范。在本章开头我们曾经提到,我们将在下一章中单独讨论XML和XHTML,所以我们先略过对这些题目的详细讨论。WWW协会还在继续研究XHTML,新的规范将会陆续推出。你可以访问<http://www.w3.org/MarkUp/>以了解此工作的新进展。

1.3 浏览器

正如你从以上的历史中看到的,HTML标准的开发过程并不完全是线性的。它从标准的1.0和2.0版本直接到了3.2版,然后到4.0版,没有版本3.0。其原因是浏览器中非标准特性的快速开

发。这些开发工作走到了标准3.0版的前面，以致于此版本还没有完成起草阶段就被放弃了。下面，让我们较详细地了解一下浏览器的发展史及其功能的变化。

1.3.1 浏览器的历史

我们先看看早期的Web浏览器，由Marc Andreessen和Eric Bina开发的NCSA Mosaic浏览器。它于1993年在伊利诺斯大学开发出来，它改变了Web的发展道路。它第一次使用户可以通过使用图形用户界面（GUI）来访问Web页面。Mosaic是为Windows和Macintosh平台开发的，它的增长非常迅速，但不久后就被Netscape Navigator取代了（见图1-1）。Netscape于1994年推出了他们的Navigator浏览器的0.9版，当时已经有超过20种参与竞争的浏览器，并且这个数量还在快速增加。当时HTML标准已经形成，而Netscape通过在标准中增加非标准特性获得了成功，这些特性包括Navigator 1.1中的背景图像和闪烁文本。

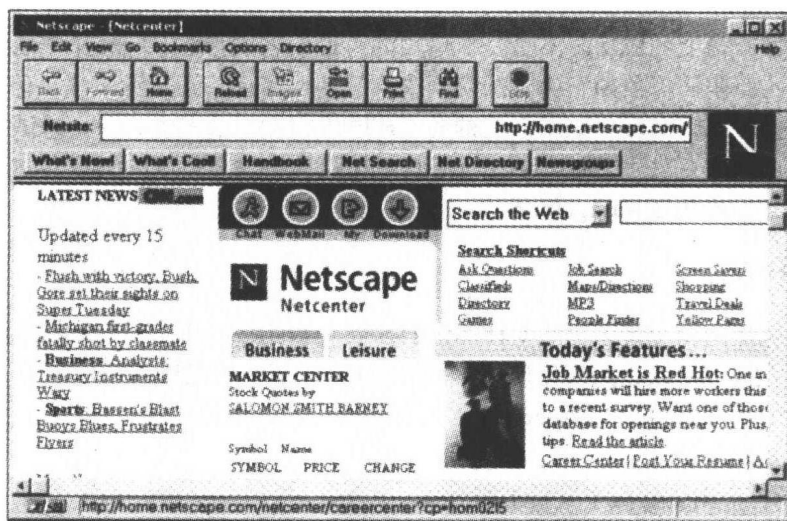


图 1-1

到了2.0版，它可以支持Java applet以及它自己的脚本语言，称为JavaScript。Navigator浏览器中的许多增强功能后来被吸收进了HTML标准的新版本，因为HTML标准要努力跟上革新的步伐。但是，这些革新中的一部分不符合标准的精神和HTML的目标。可以说，HTML原本只致力于管理Web文档的结构，但是突然发现它被卷入了表示方式和样式标记之中，如和<frame>。实际上，样式和结构是Web页面截然不同的两个方面。编写语句时，应该可以不考虑页面上文本的字体和背景的颜色。样式对底层的结构毫无影响，可以从实际的Web页面中完全分离出来。实际上，在后面的章节中我们将讨论如何分离样式和内容。

直到1995年12月，微软才认识到了Internet的重要性，并且宣布了他们的Internet策略，而当时Netscape已经拥有了75%的市场份额。但是，尽管微软让Netscape领先了许多，1996年3月他们还是承诺提供一整套Internet技术，当年晚些时候，他们开发了自己的浏览器Internet Explorer的2.0版。这个浏览器不支持HTML框架和Netscape Navigator 2.0的Java支持。直到Internet