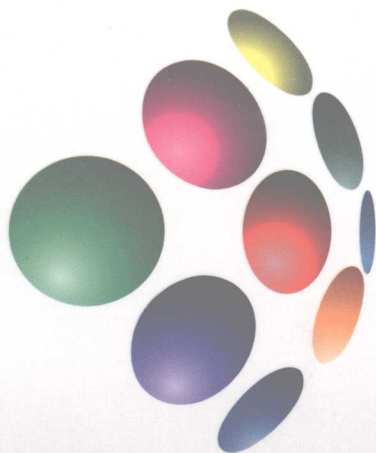


HTML + CSS 网页布局

开发指南

- ◎ 严格依照CSS最新规范并结合Web标准
- ◎ 涵盖层叠、继承、盒模型、布局和定位等CSS核心技术
- ◎ 详细讲解CSS各种样式属性的功能，并配有典型实例
- ◎ 代码兼顾多种浏览器，力求获得最好的兼容性
- ◎ 介绍圆角、图像替换、滑动门等CSS高级技巧，并提供大量参考资料以便深入学习



清华大学出版社

HTML+CSS 网页布局开发指南

贾 铮 王 韡 雷奇文 编著

HTML+CSS
网页布局开发指南

清华大学出版社

北京

内 容 简 介

本书由浅入深、循序渐进地介绍了 HTML、XHTML 和 CSS 的语法、元素、属性的使用方法,介绍了如何使用(X)HTML 和 CSS 编写符合标准的 Web 页面。

本书共分六篇:第一篇从基本概念讲起,包括(X)HTML 基础知识、CSS 基本概念和语法规则;第二篇为 CSS 核心原理,内容涉及选择符、度量、样式层叠与继承,以及如何处理浏览器兼容性问题;第三篇是样式设计,介绍 CSS 如何控制文本、链接、图像、列表、表格和表单元素的样式;第四篇讨论 CSS 布局技术,包括浮动和定位的概念,列举常见的页面布局形式并提供实现方案;第五篇是 CSS 高级主题,介绍 CSS 在 XML/打印媒介中的应用、用户界面元素的样式以及特定于 IE 和 Firefox 浏览器的 CSS;第六篇通过一个较完整的页面实例,介绍如何使用 XHTML 构建页面内容,并综合运用各种 CSS 技术进行页面设计。

本书注重基础、讲究实用、选材精当、深入浅出,适合初、中级读者学习使用,也适合具有 HTML 和 CSS 基本知识的任何网页设计和开发人员阅读及参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

HTML+CSS 网页布局开发指南/贾铮,王韡,雷奇文编著.—北京:清华大学出版社,2008.9

ISBN 978-7-302-18503-1

I. H… II. ①贾… ②王… ③雷… III. ①超文本标记语言, HTML—主页制作—程序设计 ②主页制作—软件工具, CSS IV. TP312 TP393.092

中国版本图书馆 CIP 数据核字(2008)第 136219 号

责任编辑:邹 杰 宋延清

装帧设计:杨玉兰

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京鑫丰华彩印有限公司

装 订 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:190×260 印 张:25.25 字 数:625 千字

版 次:2008 年 9 月第 1 版 印 次:2008 年 9 月第 1 次印刷

印 数:1~4000

定 价:40.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:030217-01

前 言

本书作者在 9 年前开始接触互联网，体验到网络给人们带来的便捷。又经过 1 年，开始使用 HTML 制作 Web 页面，那时还是表格布局一统天下，使用 CSS 进行布局的人少之又少。作者当然也使用表格布局页面。这样做的结果是，页面充斥着大量的 table 标记和很多控制样式的 HTML 属性和元素，增加了维护工作的难度。后来，随着 Web 标准的逐步推广，要求页面内容与其表现形式分离，而 CSS 就是控制页面表现的语言。于是，作者开始仔细研究 CSS，很快就领略了它的强大功能，后来便一直使用 CSS 进行 Web 开发。

CSS 入门非常容易，它本身并不涉及过于复杂的语法，其代码具有很好的自我说明的特性，即使从未接触过 CSS 的人也能理解大部分代码的含义。

希望通过本书将作者所掌握的 CSS 知识和多年开发中积累的经验与读者分享，帮助读者学习、理解和运用好 CSS。

1. 如何阅读本书

本书采用循序渐进的编写方式，初学者可按顺序阅读每一章的内容。每章的开始部分会告诉你本章所涉及的主要内容，便于快速查阅和了解章节内容。如果读者对某些内容已经掌握，则可以跳过这一章继续阅读。中级读者可以根据需要，挑选某些章节进行阅读。

本书内容严格依照 CSS 规范编写，但并不是对规范的生搬硬套。书中将一些晦涩难懂的专业术语转化为通俗易懂的语言进行讲解，同时配合适当的插图和丰富的代码示例供读者学习和参考。希望读者在阅读过程中能够亲自试验，以加深理解。为了让读者能了解国外相关领域的发展状况，进一步地深入学习，书中还提供了一些英文资料的参考网址。

2. 本书针对的读者

本书主要面向初、中级读者，也适合广大 Web 设计人员和爱好者阅读。对于刚刚步入 Web 设计领域的读者，本书将从最基本的概念开始，逐步介绍使用 CSS 进行 Web 开发的方法。如果读者对 CSS 已有一定的了解，本书也将介绍一些 CSS 开发的高级技巧，并提供进一步学习和研究的参考资料。对于从事 Web 开发工作的读者，本书可作为一本 CSS 参考手册。

3. 致谢

感谢小晖子工作室的翁烨晖、王韡、雷奇文对本书编写所做的贡献，感谢作者家人和朋友的支持，感谢清华大学出版社与编辑，有了他们的帮助本书才能得以顺利出版。

限于作者的水平，书中难免存在错误与疏漏之处，希望读者及 Web 领域的同行给予批评和指正。

编 者

目 录

第一篇 初识 CSS

第 1 章 Web 设计基础.....	1	2.2.2 元素类型	15
1.1 因特网和万维网	1	2.2.3 属性	16
1.1.1 何为因特网	1	2.3 注释、空白和特殊字符	18
1.1.2 何为万维网	1	2.3.1 注释	18
1.2 Web 设计技术	2	2.3.2 空白的处理	18
1.2.1 HTML	2	2.3.3 特殊字符	20
1.2.2 XML	2	2.4 (X)HTML 文档结构	21
1.2.3 XHTML	3	2.4.1 文档类型声明	21
1.2.4 CSS	3	2.4.2 文档元素	22
1.2.5 DOM 与 ECMAScript	3	2.4.3 文档头	23
1.3 Web 标准概述	3	2.4.4 文档体	24
1.3.1 什么是 Web 标准	3	2.4.5 文档树	28
1.3.2 使用语义化的(X)HTML	4	2.5 编写符合标准的(X)HTML	29
1.3.3 Web 文档的三层结构	4	2.5.1 使用语义化标记	29
1.3.4 Web 标准的优势	4	2.5.2 避免使用具有表现功能的元素 和属性	30
1.4 浏览器的故事	5	2.6 小结	30
1.4.1 Netscape Navigator	6	第 3 章 CSS 的基本概念	32
1.4.2 Microsoft Internet Explorer	6	3.1 什么是 CSS	32
1.4.3 Mozilla Firefox	7	3.1.1 何为样式	32
1.4.4 Opera	8	3.1.2 何为层叠	33
1.4.5 Safari	9	3.2 CSS 的作用	33
1.4.6 其他浏览器	9	3.2.1 排版与风格设计	33
1.5 Web 技术的未来	10	3.2.2 简化的 Web 开发	36
1.5.1 Microsoft Silverlight	10	3.3 CSS 的起源及发展	37
1.5.2 Adobe Flex	10	3.4 应用到 Web 页面	38
1.6 小结	11	3.4.1 内联样式	39
第 2 章 HTML 和 XHTML 基础	12	3.4.2 嵌入样式	39
2.1 页面设计初探	12	3.4.3 外部样式	40
2.1.1 编写第一个 Web 页面	12	3.5 管理 CSS	41
2.1.2 first.html 中都有些什么	13	3.6 编写第一个 CSS 样式	41
2.2 标记、元素和属性	14	3.6.1 增加样式	41
2.2.1 标记、内容和元素	14		

3.6.2 本书约定	42
3.7 CSS 规则详解	43
3.8 注释和风格	44
3.8.1 代码注释	44
3.8.2 代码风格	46

3.9 工具的重要性	47
3.9.1 开发工具	47
3.9.2 辅助工具	51
3.10 小结	56

第二篇 CSS 核心原理

第 4 章 选择符

4.1 选择符的模式和语法	57
4.1.1 模式	57
4.1.2 语法	57
4.2 基本选择符	58
4.2.1 通用选择符	58
4.2.2 类型选择符	58
4.3 组选择	59
4.4 id 和 class 选择符	59
4.4.1 id 选择符	60
4.4.2 class 选择符	61
4.4.3 多重 class	62
4.4.4 用 id 还是 class	65
4.4.5 class 和 id 属性的命名规则	65
4.5 伪元素和伪类	67
4.5.1 :first-line 和 :first-letter 伪元素	67
4.5.2 :before 和 :after 伪元素	69
4.5.3 :first-child 伪类	70
4.5.4 有关链接的伪类	70
4.5.5 有关用户动态行为的伪类	71
4.5.6 :lang 伪类	71
4.6 与元素关系相关的选择符	71
4.6.1 后代选择符	71
4.6.2 子选择符	72
4.6.3 邻接兄弟选择符	73
4.7 属性选择符	75
4.7.1 属性选择符的匹配方式	75
4.7.2 属性选择符示例	75
4.8 小结	77

第 5 章 CSS 中的度量

5.1 值的类型	78
5.1.1 整数和实数	78
5.1.2 长度	78
5.1.3 百分数	79
5.1.4 URL 或 URI	79
5.1.5 颜色	79
5.1.6 字符串	81
5.2 单位	81
5.2.1 表示绝对长度的单位	82
5.2.2 表示相对长度的单位	82
5.3 小结	85

第 6 章 层叠和继承

6.1 继承	86
6.1.1 什么是继承	86
6.1.2 利用继承	87
6.1.3 所有的规则都能继承吗	88
6.1.4 inherit 指定继承	90
6.2 @import 规则	91
6.2.1 @import 规则的用途	91
6.2.2 @import 规则的使用	92
6.3 层叠的含义	92
6.3.1 确定度	93
6.3.2 !important 声明	94
6.3.3 层叠顺序	95
6.4 小结	96

第 7 章 盒模型

7.1 盒模型概述	98
7.2 边框	99

7.2.1 边框样式风格	99
7.2.2 边框颜色和粗细程度	100
7.2.3 边框样式缩写形式	101
7.3 填充和边距	103
7.3.1 填充	103
7.3.2 边距	105
7.4 宽度和高度	106
7.4.1 width 和 height	106
7.4.2 最大值和最小值	110
7.5 盒模型相关内容及高级主题	112
7.5.1 盒模型的维度	112
7.5.2 盒模型与背景	113
7.5.3 边距重叠问题	114
7.5.4 边距实现对齐功能	115
7.6 overflow 属性	115
7.7 元素类型和 display 属性	117
7.7.1 block 和 inline	117

7.7.2 inline-block	118
7.7.3 run-in	119
7.8 小结	119

第 8 章 解决浏览器兼容性问题

8.1 条件注释	120
8.1.1 条件注释的基本结构	120
8.1.2 条件注释举例	121
8.2 CSS Hacks	123
8.2.1 利用选择符	123
8.2.2 利用!important 声明	124
8.2.3 利用@import 规则	124
8.3 正确使用 Hacks 技术	125
8.3.1 避免乱用 Hacks	125
8.3.2 良好的习惯	125
8.4 小结	126

第三篇 CSS 样式设计

第 9 章 文字处理

9.1 字体族和字号	127
9.1.1 ClearType 技术	127
9.1.2 字体族	130
9.1.3 文字度量	135
9.1.4 字体大小	136
9.1.5 颜色	139
9.2 基本文字样式	139
9.2.1 斜体和粗体	139
9.2.2 大小写	140
9.2.3 文字装饰	141
9.2.4 font 属性	142
9.3 段落文字样式	144
9.3.1 字词间距	144
9.3.2 行高	145
9.3.3 缩进	148
9.3.4 对齐方式	149
9.3.5 强制换行	151

9.3.6 其他相关属性	152
--------------------	-----

9.4 文字样式实战	156
------------------	-----

9.5 小结	158
--------------	-----

第 10 章 链接处理

10.1 认识链接	159
10.1.1 锚元素	159
10.1.2 链接状态	159
10.1.3 默认效果	160
10.2 添加链接样式	160
10.2.1 LVHA, 爱和恨	160
10.2.2 下划线	161
10.3 链接实战	163
10.3.1 简单方式	163
10.3.2 图像方式	164
10.4 小结	169

第 11 章 图像和背景

11.1 图像格式	171
-----------------	-----

11.1.1	GIF	171
11.1.2	JPEG	172
11.1.3	PNG	173
11.2	给图像添加样式	173
11.2.1	img 元素	174
11.2.2	盒模型相关样式	174
11.2.3	图文混排	175
11.3	背景和图像	183
11.3.1	设置背景图像	183
11.3.2	平铺背景图像	185
11.3.3	背景图像位置	188
11.3.4	background 属性	193
11.4	图像实战	194
11.4.1	网络相册	194
11.4.2	圆角设计	199
11.4.3	图像替换	202
11.5	小结	205
第 12 章	列表	206
12.1	列表元素	206
12.2	列表相关样式	209
12.2.1	列表样式类型	209
12.2.2	列表样式图像	211
12.2.3	列表样式位置	212
12.2.4	list-style 属性	213
12.3	其他相关样式	213
12.3.1	边框、填充和边距	213
12.3.2	更改布局方式	215
12.4	列表实战	215
12.4.1	新闻列表	215
12.4.2	导航菜单	219
12.5	小结	223

第 13 章 表格

13.1	使用表格	224
13.1.1	表格用途	224
13.1.2	表格元素	224
13.2	边框样式	227
13.2.1	border 属性	227
13.2.2	border-spacing 属性	229
13.2.3	边框模式	229
13.3	与表格相关的样式	231
13.3.1	caption-side 属性	231
13.3.2	添加填充	232
13.4	表格大小	232
13.4.1	表格宽度计算方式	233
13.4.2	高度	237
13.5	对齐	238
13.6	表格实战	240
13.6.1	斑马线效果	240
13.6.2	日历	242
13.7	小结	247

第 14 章 表单

14.1	表单概述	248
14.1.1	表单元素	249
14.1.2	表单元素的显示	250
14.2	添加样式	252
14.2.1	盒模型相关属性	253
14.2.2	文字相关属性	254
14.2.3	背景和图片	255
14.2.4	表单元素的布局	257
14.3	表单实战	259
14.4	小结	266

第四篇 CSS 布局技术

第 15 章 浮动与定位

15.1	float 属性	266
15.1.1	(X)HTML 文档流	266
15.1.2	float 属性的作用	268

15.2	clear 属性	271
15.3	使用 float 属性布局页面	274
15.4	IE 的浮动问题	276
15.4.1	边距加倍问题	276

15.4.2	3px 间隔问题	277
15.4.3	捉迷藏问题	279
15.4.4	断头台问题	280
15.5	position 属性与定位	282
15.5.1	静态定位和相对定位	282
15.5.2	绝对定位	284
15.5.3	固定定位	289
15.6	利用定位实现布局	290
15.7	控制元素的深度	292
15.8	元素的可见性	294
15.9	小结	296

第 16 章 常见的页面布局方式

16.1	布局类型概述	297
16.2	float 还是 position	299
16.2.1	float 布局	299
16.2.2	position 布局	299
16.3	布局实战	300
16.3.1	二分栏固定式布局	300
16.3.2	三分栏流动式布局	305
16.3.3	弹性布局	316
16.4	小结	318

第五篇 CSS 高级主题

第 17 章 CSS 高级应用

17.1	CSS 在 XML 中的应用	320
17.1.1	XML 概述	320
17.1.2	使用 CSS	321
17.1.3	为 XML 文档添加样式	322
17.1.4	CSS 布局	327
17.2	用于打印的 CSS	331
17.2.1	为打印媒介指定 CSS	331
17.2.2	分页处理	334
17.3	用户界面元素	335
17.3.1	鼠标指针	335
17.3.2	系统颜色	340
17.3.3	轮廓线	342
17.4	滤镜与转场	343
17.4.1	程序生成面	343

17.4.2	静态滤镜	345
17.4.3	转场	348
17.5	behavior 属性与 CSS 表达式	350
17.5.1	behavior 属性	350
17.5.2	CSS 表达式	351
17.6	微软对 CSS 的扩展	351
17.6.1	控制滚动条外观	351
17.6.2	缩放功能	352
17.7	Mozilla 扩展	354
17.7.1	at 规则	354
17.7.2	伪类和伪元素	354
17.7.3	属性	356
17.7.4	属性值	358
17.8	小结	361

第六篇 CSS 实战

第 18 章 MyBlog 实例

18.1	实例说明	363
18.2	从布局开始	364
18.2.1	结构分析	364
18.2.2	准备 XHTML 代码	365
18.3	准备样式表	365

18.4	添加标题和导航	366
18.4.1	准备 XHTML 代码	366
18.4.2	标题样式	367
18.4.3	导航设计	368
18.5	左栏内容设计	370
18.5.1	处理 #middle	370

18.5.2 处理左栏.....	370	18.7.2 添加样式.....	379
18.5.3 添加样式.....	371	18.8 底脚处理.....	381
18.6 文章显示.....	374	18.9 完整代码.....	381
18.6.1 处理#content.....	374	18.9.1 XHTML 文档代码.....	381
18.6.2 准备代码.....	374	18.9.2 样式表文档代码.....	384
18.6.3 添加样式.....	375	18.10 小结.....	390
18.7 留言表单.....	379	参考文献.....	391
18.7.1 XHTML 代码.....	379		

第一篇 初识 CSS

第 1 章 Web 设计基础

本章首先介绍什么是因特网和万维网，它们之间的区别和联系是什么。接着会介绍一些与 Web 设计相关的基本知识，包括 Web 设计所使用的各种技术、什么是 Web 标准以及它的优势所在。浏览器是用户访问 Web 应用的途径，本章将介绍几款目前较为流行的浏览器，它们各自的发展历程和功能特点。最后让读者了解两种较新的 Web 开发技术。

本章主要内容

- 什么是因特网和万维网，二者是什么关系
- Web 设计技术概述
- 什么是 Web 标准
- Web 标准涵盖哪些内容
- 目前比较流行的浏览器
- Microsoft Silverlight 和 Adobe Flex 技术

1.1 因特网和万维网

1.1.1 何为因特网

各位读者可能对因特网(Internet)这个词不会感到陌生。因特网是由许许多多计算机连接在一起构成的一个计算机网络。在这个网络中，人们可以使用各自的计算机互相传递信息；可以在自己的计算机上存储文件供别人访问；可以利用多台计算机实现分布式应用。Internet 规模庞大，遍及全球。一旦上网，你就与全球无以计数的计算机连成了一体。

1.1.2 何为万维网

有上网经历的读者会注意到，在浏览器的地址栏中经常出现“www”，比如要访问百度网站就需输入“www.baidu.com”。其中的三个“w”是英文 World Wide Web 的缩写，中文译为万维网。万维网是无数个网络站点和网页的集合，它们在一起构成了因特网最主要的部分(因特网还包括电子邮件、Usenet 以及新闻组等应用)。

万维网实际上是多媒体的集合，各个部分由超级链接连接而成。我们通常使用浏览器上网观看的内容，就是万维网的内容。比如在浏览器中输入“www.sina.com.cn”，就可以访问新浪

网站的首页(见图 1-1)。网页也称作 Web 页面或 Web 文档,它包含了文字、图像、动画和一些具有交互功能的控件。



图 1-1 使用浏览器访问新浪网站的首页

1.2 Web 设计技术

Web 设计涵盖的范围相当广,本节只介绍一些构建 Web 文档所需的几项最为基本的技术。

1.2.1 HTML

HTML, 英文全称为 HyperText Markup Language, 中文译为超文本标记语言。它是用来创建 Web 文档的语言。页面元素是由特定的标记来确定的, 这些标记告诉浏览器该如何显示这个页面。所谓超文本, 就是一种含有指向其他文档链接的文本, 即我们俗称的链接。超文本把存放于不同计算机中的文档链接在一起。访问者不必关心链接所指的内容到底位于何处, 只需要在链接上单击一下鼠标, 页面就会马上转到所指的文档中去。

1.2.2 XML

XML, 即 eXtensible Markup Language(可扩展标记语言)。XML 是一种定义其他语言的语言, 是一种元语言(Meta Language)。使用 XML 可以创建自己定制的标记语言, 然后使用这种语言对自己的文档进行格式化。

XML 与 HTML 相比, 对语法的要求更加严格, 也比 HTML 灵活。于是 XML 和 HTML 相

结合产生了 XHTML。

1.2.3 XHTML

XHTML 中的 X 和 XML 中的 X 含义一致,它是可扩展超文本标记语言。XHTML 是由 XML 创建出来的,是对 HTML 的重新改造。XHTML 更健壮、更灵活、更强大,将来可能得到更加完善的支持。

1.2.4 CSS

HTML 可以将内容、结构和格式化信息都包含在同一个 Web 文档中,这样做虽然简单易行,但也存在很多问题。各种信息存放在一起不利于文档的维护,修改起来费时费力。因此 HTML 应只负责文档的内容和结构,而格式化信息交给一套新的语言来完成,这就是 CSS,它也是本书的主角。

CSS 全称为 Cascading Style Sheet,中文译为层叠样式表(也有译作级联样式表的)。其作用就是要控制 HTML 的页面布局和外观样式,使 Web 文档内容结构和表现形式完全分离。

1.2.5 DOM 与 ECMAScript

DOM(Document Object Model, 文档对象模型)是 HTML 和 XML 文档的一个应用程序接口(Application Programming Interface, API)。它提供了一种结构化的文档表示方式,从而使开发人员可以修改它的内容以及最终的表达方式。总而言之,它把网页内容与脚本或编程语言联系在一起。

ECMAScript 是由欧洲计算机制造商协会(European Computer Manufacturers Association, ECMA)通过 ECMA-262 予以标准化的脚本程序设计语言。通过 DOM 可对页面中的对象进行访问和操作。目前 Web 页面中经常使用的 JavaScript 和 JScript 与 ECMAScript 兼容,但也包含超出 ECMAScript 的功能。

1.3 Web 标准概述

1.3.1 什么是 Web 标准

Web 标准是由 W3C(World Wide Web Consortium)和其他标准化组织指定的一系列规范的集合,其中包含了上文提到的(X)HTML、XML 和 CSS。Web 标准的目的在于创建一个统一的用于 Web 表现层的技术标准,以便通过不同浏览器或终端设备向最终用户展示信息内容。

1.3.2 使用语义化的(X)HTML

首先我们应该了解(X)HTML 的用途是什么, 根据 W3C 的解释, “HTML 是一种在万维网上发布超文本的通用语言……HTML 使用诸如<h1>和</h1>一类的标记来将文本结构化为主题、段落、列表、超文本等。”

所谓语义化, 就是指(X)HTML 可以给普通的文本增加结构和含义。比如某段文本是一个普通段落, 另一段文本是一个标题或是列表中的一项等。在第 2 章, 我们还要详细介绍如何使用语义化标记。

1.3.3 Web 文档的三层结构

1. 结构、表现和行为

Web 文档通常包含三个不同的层次(见图 1-2)。首先是结构层, 该内容是由(X)HTML 编写的文本文档。它包含文档的内容, 以及由(X)HTML 添加的语义化的标记。第二层为表现层, 这是本书的重点。该层内容是 CSS 样式代码。它描述了文档该以何种样式呈现在访问者面前, 样式包括页面各部分的布局、文字段落排版、背景图像和颜色等。第三层是行为层, 这里不对它进行过多的介绍。该层定义了文档模型以及如何与用户进行交互, 所涉及技术主要是 DOM 以及 ECMAScript 脚本语言。

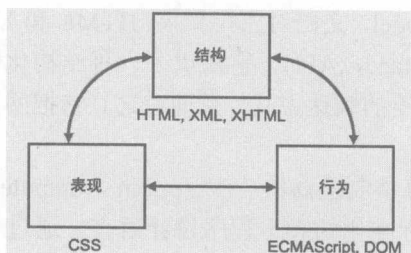


图 1-2 结构、表现和行为及各部分所涉及的技术

2. 模型、视图和控制器

模型、视图和控制器(Model-View-Controller, MVC)模式是面向对象开发中常用的一种设计结构, Web 文档的三层结构也可看作是一种 MVC 结构。模型表示数据结构, 也就是(X)HTML 所编写的文档内容。视图表示模型的展现形式, 即用 CSS 控制页面的样式。控制器对应着程序的动作行为, 因此 Web 中的交互结构可视为控制器。

1.3.4 Web 标准的优势

1. 易于开发和维护

一个大型的网站往往需要很多人员的参与, 他们的分工也不同, 有负责页面样式设计的, 有负责页面编码的, 有负责样式编写的等。由于内容结构和表现的分离, 不同开发人员可以独

立工作,专注于自己负责的内容。

样式信息和内容是相互独立的,因此同一个样式信息可用于不同的内容中,从而实现代码重用。这种做法可以减少重复编码,加快开发进度。

样式信息重用也使得维护工作大大减轻,只需要修改一小部分样式代码,就可以使所有用到该样式的区域同时改变外观。

2. 高兼容性

由于 W3C 对 Web 标准的推动,越来越多的浏览器支持由 W3C 制定的各种标准,从而使根据标准编写的 Web 页面在不同的浏览器中均能获得一致的效果。

3. 高灵活性

现在,越来越多的人使用手机或 PDA 访问网站,通常这些设备的屏幕要远远小于 PC 机的显示器。网页的内容和表现的分离使得我们可以针对不同平台和设备应用不同的样式文件,而网页内容无须改动。对于需要打印输出的页面,我们也无须再提供另一份“适合打印”的页面,取而代之的只是新的 CSS 样式。

4. 提高访问速度

内容与表现的分离,使得页面中不再包含冗余的样式代码,而独立出来的样式表可以充分地进行重用,网页整体代码量大大减少。这样不仅能占用更少的网络带宽,提高页面载入速度,同时浏览器也能对页面进行快速解析,显示给用户。

5. 提高用户体验

从 Web 应用的角度看,用户体验(User Experience, UE)指的是 Web 应用程序能够提供直观简洁的用户界面、简便的操作以及有效的交互方式。只有当用户亲自使用时才能体验到它们。用户体验包含了多方面的内容,其中一致性、可用性等方面均可通过标准化开发来实现。比如各个页面使用统一的 CSS 样式,利用行为层技术改善交互设计等。

 **延伸:** 傅捷(网络名为阿捷)是国内较早的 Web 标准化执行者,他参与翻译的《网站重构》一书将 Web 标准化思想引入到国内的网页设计行业。在他的网站上有一篇讲述 Web 标准的文章,读者可以访问如下地址来阅读:

<http://www.w3cn.org/article/tips/2007/123.html>

1.4 浏览器的故事

网页和浏览器是分不开的,用户正是通过浏览器来达到访问网页的目的。浏览器(Browser)是一种软件程序,可以和网络建立连接并与之通信。它可以将存在于万维网中的特定网页获取下来,对网页中的内容进行分析,并按照一定的规则显示出来。

目前主流浏览器有 Internet Explorer、Firefox、Opera、Safari 等,它们适用于各种不同的平台环境。其中最为流行和普及的是 Internet Explorer,它的市场占有率接近于 90%。

1.4.1 Netscape Navigator

1994 年,网景通信公司(Netscape Communications Corporation)推出了 Netscape Navigator(以下简称 NN)浏览器。作为全球第一款商业浏览器,NN 大规模地引领人们开始在网络世界中漫游,正好印证了 Navigator(领航员)这个名字。此后,NN 浏览器迅速走红,一度曾经占据了超过 90%的浏览器市场份额。如图 1-3 所示为以 Netscape Navigator 9 浏览器浏览网页。

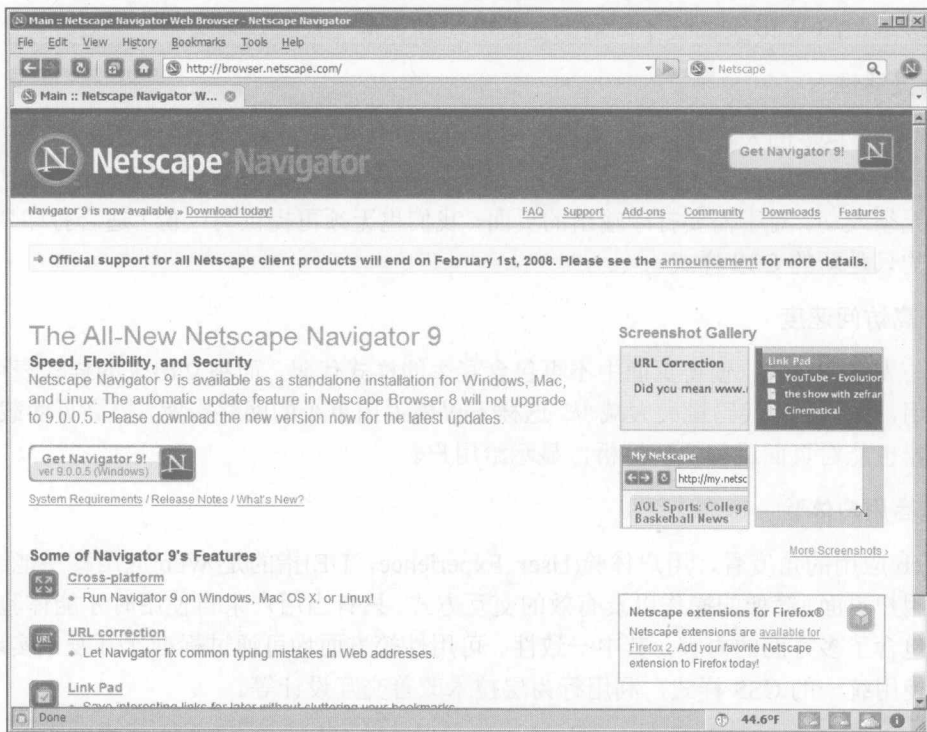


图 1-3 Netscape Navigator 9 浏览器(注意页面中提到停止提供任何技术支持的声明)

NN 浏览器的出现推动了网络的普及,网络的普及也使浏览器市场得到扩大。随着微软公司将自己的 Internet Explorer 浏览器与 Windows 操作系统实行捆绑销售,网景通信公司不得不开始面对挑战。从那时起,NN 浏览器便开始走向没落。

1998 年 11 月网景通信公司被美国在线收购,而美国在线之后又成为时代华纳的一部分。2003 年 7 月 15 日时代华纳解散了网景通信公司。2007 年 12 月 28 日,据网景官方博客消息,NN 浏览器将在 2008 年 2 月 1 日停止更新并且不再提供任何技术支持,1994 年问世的 Netscape Navigator 正式退出历史舞台。

1.4.2 Microsoft Internet Explorer

1995 年,微软公司(Microsoft Corporation)推出了自己的 Internet Explorer(IE)浏览器,并与 Windows 95 操作系统进行捆绑销售,1997 年推出 IE4,增加了活动桌面、频道、FrontPage

Express、Web 发布向导等功能,后来与 Windows 98 捆绑在一起销售。

目前国内使用率最高的是 2001 年发布的 IE6,它集成在 Windows XP 操作系统中。由于 Windows 在操作系统领域占有相当高的市场份额,随之一起销售的 IE 系列浏览器也获得了相当高的市场占有率。但随后微软公司放慢了浏览器开发的步伐,随后的几年中并没有新版本的浏览器问世。鉴于 Firefox、Opera 等浏览器带来新的竞争压力,微软公司后来决定加大对浏览器的开发力度。

2006 年 10 月 18 日,IE7 浏览器发布,它支持标记式浏览、RSS 聚合,增加了快速选项卡、反钓鱼攻击以及页面整体缩放等功能(目前 beta 版的 IE8 浏览器也已经发布)。

专门介绍 IE 浏览器的微软官方页面的地址如下:

<http://www.microsoft.com/china/windows/products/winfamily/ie/default.mspx>

该页面当前介绍的 IE 版本为 IE7,如图 1-4 所示。

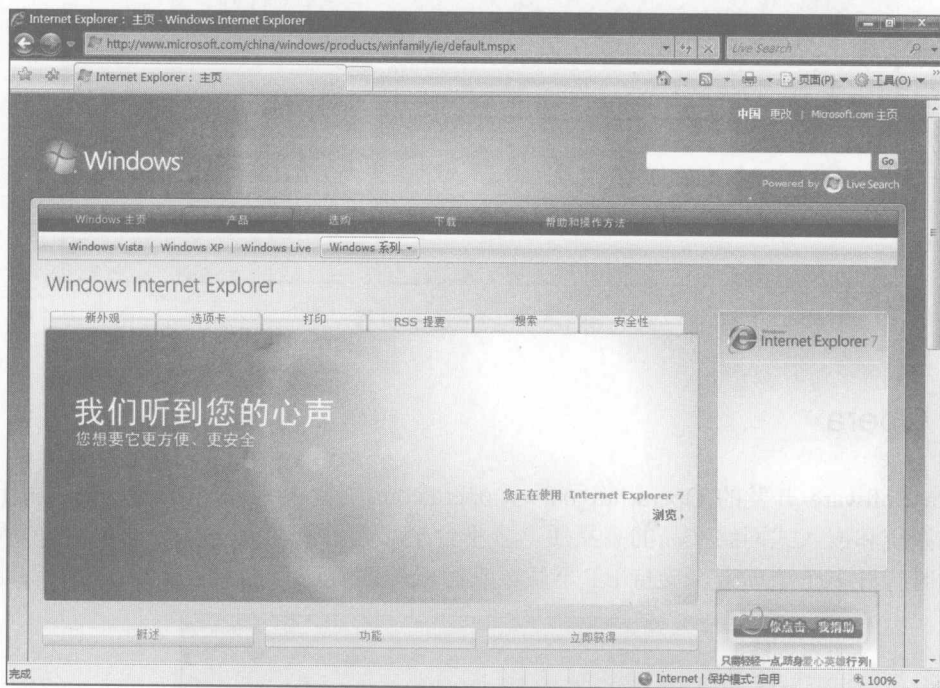


图 1-4 使用 Internet Explorer 7 浏览器

1.4.3 Mozilla Firefox

Mozilla Firefox(<http://www.mozilla.org.cn>)是由 Mozilla 基金会与数百个志愿者所开发的网页浏览器,如图 1-5 所示。Mozilla 基金会的前身是网景公司内部成立的 Mozilla 组织,为的是在和微软竞争中扭转颓势,Mozilla 组织专门负责维护包括 Netscape Navigator 浏览器在内的多种网络应用软件。后来,Mozilla 组织注册成为非营利机构,并正式更名为 Mozilla 基金会。

Firefox 1.0 正式发布于 2004 年 11 月 9 日,它支持标记浏览、RSS 和多种扩展。2006 年 10 月 24 日,Firefox 2.0 版发布。现在它的 3.0 beta 版也已经发布。