

信息科学与技术丛书

孟庆昌 王津涛 编著

HTML5, CSS3, JavaScript 开发手册

- 全面呈现 HTML5
- 深入讲解 CSS3
- 详细介绍 JavaScript
- 精心设计 HTML5 与 CSS3 的典型范例



机械工业出版社
CHINA MACHINE PRESS

013069646

TP393.092
2501

信息科学与技术丛书

HTML5, CSS3, JavaScript 开发手册

孟庆昌 王津涛 编著



机械工业出版社



北航

C1677801

TP393.092
2501

312230610

本书集 HTML5、CSS3、JavaScript 技术于一体，详细地介绍了客户端程序设计。

全书共分为四部分。第一部分详细介绍了 HTML5 的各方面知识，第二部分详细介绍了 CSS3 相关知识，第三部分介绍了 JavaScript 动态程序，第四部分列举了 HTML5 与 CSS3 的典型范例。

本书实例丰富，为每个知识点精心设计了案例，且采用代码与截图相结合的方式编写。无论是刚刚接触 Web 前端的新人，还是对 HTML5 及未来 Web 应用技术有兴趣的读者均可阅读本书。

书中实例代码可在 <http://www.cmpbook.com> 免费下载。

图书在版编目 (CIP) 数据

HTML5, CSS3, JavaScript 开发手册 / 孟庆昌, 王津涛编著. —北京: 机械工业出版社, 2013.7

(信息科学与技术丛书)

ISBN 978-7-111-43399-6

I. ①H… II. ①孟… ②王… III. ①超文本标记语言—程序设计②网页制作工具③JAVA 语言—程序设计 IV. ①TP312②TP393.092

中国版本图书馆 CIP 数据核字 (2013) 第 165485 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 车 忱

责任编辑: 车 忱

责任印制: 李 洋

三河市国英印刷有限公司印刷

2013 年 9 月第 1 版·第 1 次印刷

184mm×260mm·23 印张·569 千字

0001—4000 册

标准书号: ISBN 978-7-111-43399-6

定价: 59.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

社服务中心: (010) 88361066

教材网: <http://www.cmpedu.com>

销售一部: (010) 68326294

机工官网: <http://www.cmpbook.com>

销售二部: (010) 88379649

机工官博: <http://weibo.com/cmp1952>

读者购书热线: (010) 88379203

封面无防伪标均为盗版

出版说明

随着信息科学技术的迅速发展，人类每时每刻都会面对层出不穷的新技术和新概念。毫无疑问，在节奏越来越快的工作和生活中，人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书来获取新知识和新技能，从而不断提高自身素质，紧跟信息化时代发展的步伐。

众所周知，在计算机硬件方面，高性价比的解决方案和新型技术的应用一直备受青睐；在软件技术方面，随着计算机软件的规模和复杂性与日俱增，软件技术不断地受到挑战，人们一直在为寻求更先进的软件技术而奋斗不止。目前，计算机和互联网在社会生活中日益普及，掌握计算机网络技术和理论已成为大众的文化需求。由于信息科学技术在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用，所以这些专业领域中的研究人员和工程技术人员越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们了解和掌握新知识、新技能的热切期待，以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状，机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络 and 工程应用等内容，注重理论与实践的结合，内容实用、层次分明、语言流畅，是信息科学与技术领域专业人员不可或缺的参考书。

目前，信息科学技术的发展可谓一日千里，机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作，为推进我国的信息化建设做出贡献。

机械工业出版社

前言

对于当今整个 Web 开发领域来说, HTML5 可谓最热门的话题之一。在已故苹果公司掌门乔布斯看来, 未来客户端将是 HTML5 的天下。

HTML5 相较于 HTML4 做出了一定程度的修改。这些修改包括一些标签的增加/删减、语法结构的简化以及新的 API (Application Programming Interface, 应用程序编程接口)。例如, HTML5 为了更符合内容与表现的分离, 增加了 header 这样的专用标签。而在 HTML4 中, 如果要定义“头部”区块, 只能用 `<div class/id="header"></div>`。

与 Flash 相比, HTML5 的优点是无需插件、对搜索引擎友好, 且在性能与稳定性方面的表现更优。但这并不意味着 Flash 在短时间内会被 HTML5 完全取代。目前, 在普及率、兼容性以及开发群体方面 HTML5 还无法超越 Flash。况且 Adobe 公司也会及时对 Flash 进行更新升级。罗马不是一天建成的, HTML5 要想完全取代 Flash 还有很长的路要走。

随着 W3C 对 HTML5 的推进, 各大浏览器开发公司 (如谷歌、苹果、Opera、Mozilla 等) 不断更新自己的浏览器, 以提供对 HTML5 各项功能的支持。在这种形势下, 学习 HTML5 无疑相当于掌握了未来的 Web 开发利器。目前, 许多 Web 开发人员正在转向 HTML5 的学习。

本书集 HTML5、CSS3、JavaScript 技术于一体, 详细地介绍了客户端程序设计。书中配备了大量的实例, 可操作性极强。

全书共分为 21 章, 通过渐进的方式介绍 HTML5、CSS3、JavaScript 技术, 内容安排如下:

第 1 章介绍 HTML5 的新特性、文档结构以及语义化标记等。

第 2 章介绍 HTML5 的视频和音频。

第 3 章介绍 HTML5 表单 (包括表单类型、表单特性以及表单的验证等)。

第 4 章介绍如何创建 HTML5 画布。

第 5 章介绍 HTML5 中涉及的事件 (包括表单事件、媒介事件、鼠标事件、键盘事件以及 window 事件等)。

第 6 章介绍如何使用 HTML5 的 `querySelector` 与 `querySelectorAll` 方法遍历文档元素。

第 7 章介绍构建可伸缩矢量图形 (SVG)。

第 8 章介绍如何使用 HTML5 Web Storage 实现存储。

第 9 章介绍如何使用 HTML5 Web SQL Database 实现存储。

第 10 章介绍如何使用 HTML5 IndexedDB 实现存储。

第 11 章介绍 HTML5 文件处理。

第 12 章介绍如何实现 HTML5 的拖放功能。

第 13 章介绍 HTML5 地理定位。

第 14 章介绍 HTML5 通信 (包括 `PostMessage`、`Server-Sent Events` 以及 `WebSocket`)。

第 15 章介绍 HTML5 Web Worker。

目 录

出版说明

前言

第 1 章 HTML5 概述	1	4.3 使用图像	33
1.1 认识 HTML5	1	4.3.1 将图像复制到画布	33
1.2 HTML5 新特性	1	4.3.2 图像缩放	34
1.2.1 新的元素	1	4.3.3 图像剪切	35
1.2.2 新的 API	2	4.4 Canvas 文本	35
1.3 文档结构与字符集	2	4.5 样式与色彩	36
1.4 语义化标记	2	4.5.1 色彩属性	36
1.5 检测技术	7	4.5.2 设置透明度	37
第 2 章 使用视频与音频元素	8	4.5.3 渐变	37
2.1 设置视频元素	8	4.5.4 阴影	38
2.1.1 声明视频元素	8	4.6 变形	39
2.1.2 浏览器支持性检测	9	4.6.1 保持与恢复	39
2.2 设置音频元素	11	4.6.2 移动	40
2.3 媒体特性	11	4.6.3 缩放	41
第 3 章 表单元素与相关属性	15	4.6.4 旋转	41
3.1 输入类型	15	4.7 遮盖	42
3.2 浏览器支持检测	18	4.8 绘制简单动画	43
3.2.1 输入类型检测	18	第 5 章 事件处理	45
3.2.2 表单特性检测	19	5.1 事件处理方式	45
3.3 output 元素与 datalist 元素	20	5.1.1 监听方式	45
3.3.1 output 元素	20	5.1.2 赋值方式	47
3.3.2 datalist 元素	20	5.2 HTML5 常用事件	47
3.4 表单特性	21	5.2.1 表单事件	47
3.5 表单验证类型	22	5.2.2 键盘事件	49
3.6 验证反馈	24	5.2.3 鼠标事件	50
第 4 章 画布处理	26	5.2.4 媒体事件	51
4.1 HTML5 画布基础	26	5.2.5 window 事件	52
4.1.1 声明 canvas	26	第 6 章 HTML5 选择器	54
4.1.2 渲染上下文	26	6.1 传统的 DOM 遍历方式	54
4.1.3 浏览器支持检测	27	6.1.1 使用 JavaScript 方法查找	54
4.2 绘制图形	27	6.1.2 使用元素节点属性查找	56
4.2.1 绘制矩形	27	6.2 新的 Selectors API	57
4.2.2 绘制路径	28	6.3 使用 Selectors API	57
4.2.3 绘制各种线条	30	6.3.1 querySelector 方法	57

6.3.2 querySelectorAll 方法	59	10.4 IndexedDB API	86
第 7 章 使用 HTML5 SVG	61	10.4.1 初始化数据库连接	86
7.1 SVG 形状	61	10.4.2 建立存储对象	87
7.1.1 SVG 圆形	61	10.4.3 向存储对象中添加数据	88
7.1.2 SVG 矩形	61	10.4.4 查询存储的数据	89
7.1.3 SVG 椭圆	62	10.5 应用举例	90
7.1.4 SVG 线条	62	第 11 章 文件处理	94
7.1.5 SVG 多边形	63	11.1 File API	94
7.1.6 SVG 折线	63	11.1.1 FileList 与 File 接口	94
7.1.7 SVG 路径	63	11.1.2 Blob 接口	95
7.2 SVG 渐变	63	11.1.3 FileReader 接口	99
7.2.1 线性渐变	64	11.2 FileWriter API	101
7.2.2 放射性渐变	64	11.2.1 BlobBuilder 接口	102
7.3 SVG 滤镜	65	11.2.2 FileSaver 接口	103
第 8 章 Web 存储	66	11.2.3 FileWriter 接口	103
8.1 认识 Web Storage	66	11.3 FileSystem API	104
8.2 Web Storage 组成	66	11.3.1 FileSystem 初始化	104
8.3 Web Storage API	67	11.3.2 FileEntry 接口	106
8.3.1 检测浏览器支持性	67	11.3.3 DirectoryEntry 接口	111
8.3.2 存储和获取数据	68	11.3.4 FileSystem URL	116
8.3.3 转换存储数据	71	第 12 章 拖放操作	117
8.3.4 Storage 事件	71	12.1 drag and drop 简介	117
8.4 存储限制	72	12.2 drag and drop API	117
8.5 使用 JSON 处理数据	73	12.2.2 浏览器支持性检测	117
8.5.1 认识 JSON	73	12.2.2 drag and drop 事件	117
8.5.2 存储 JSON 对象	75	12.2.3 dataTransfer 对象	118
第 9 章 使用 Web SQL Database	77	12.3 drag and drop 过程	118
9.1 核心方法	77	第 13 章 地理定位	121
9.2 执行操作	78	13.1 浏览器支持性检测	121
9.2.1 打开连接	78	13.2 单次位置请求	121
9.2.2 创建数据表	78	13.2.1 获取位置信息	122
9.2.3 插入数据	79	13.2.2 错误处理	123
9.2.4 查询数据	80	13.2.3 可选位置特性	125
9.2.5 修改与删除	81	13.3 持续获取位置信息	126
9.3 完整的操作实例	81	13.4 在 Google Map 中显示 位置	127
第 10 章 使用 IndexedDB	84	第 14 章 通信应用	129
10.1 认识 IndexedDB	84	14.1 PostMessage	129
10.2 IndexedDB 存储结构	84	14.1.1 浏览器支持性检测	129
10.3 适应多内核浏览器	85		

14.1.2	发送与监听	129	17.1.1	浏览器支持性检测	159
14.1.3	创建 postMessage 应用	130	17.1.2	创建 Notifications	159
14.2	Server-Sent Events	132	17.1.3	显示提示许可	160
14.2.1	浏览器支持性检测	132	17.1.4	添加回调和其他操作	160
14.2.2	Server-Sent Events API	133	17.1.5	实例代码	160
14.2.3	服务器端代码设置	133	17.2	HTML5 Progress	161
14.2.4	创建 Server-Sent Events 应用	134	17.2.1	语法与属性	161
14.3	WebSocket	136	17.2.2	自定义 HTML5 的样式	162
14.3.1	WebSocket API	136	17.2.3	Progress 实例	162
14.3.2	浏览器支持性检测	137	第 18 章	CSS3 属性	165
14.3.3	WebSocket API 使用方法	137	18.1	边框属性	165
第 15 章	使用 Web Worker		18.1.1	border-colors 属性	165
处理线程		139	18.1.2	border-image 属性	166
15.1	Web Worker API	139	18.1.3	border-radius 属性	171
15.1.1	浏览器支持性检查	139	18.1.4	box-shadow 属性	175
15.1.2	构建 Web Worker	139	18.2	背景属性	176
15.1.3	Web Worker 传递信息	140	18.2.1	background-clip 属性	176
15.1.4	终止 Worker	140	18.2.2	background-origin 属性	178
15.1.5	错误处理	140	18.2.3	background-size 属性	180
15.1.6	加载外部脚本	141	18.2.4	multiple background 属性	182
15.2	Worker 简单操作流程	141	18.3	颜色属性	184
15.3	使用 Worker 线程做后台		18.3.1	HSL colors 属性	184
数值计算		143	18.3.2	HSLA colors 属性	185
15.4	Worker 线程嵌套	145	18.3.3	RGBA colors 属性	187
15.5	共享线程	146	18.3.4	opacity 属性	188
15.5.1	创建共享线程	146	18.4	文本属性	189
15.5.2	与共享线程通信	146	18.4.1	text-shadow 属性	189
15.5.3	共享线程的使用	147	18.4.2	text-overflow 属性	191
第 16 章	离线 Web 应用	149	18.4.3	word-wrap 属性	192
16.1	认识 Offline Web Application	149	18.4.4	word-break 属性	193
16.2	manifest 文件	149	18.5	字体属性	195
16.2.1	引用清单文件	149	18.5.1	@font-face 属性	195
16.2.2	清单文件结构	150	18.5.2	字体格式	195
16.3	在线状态检测	151	18.5.3	应用举例	196
16.4	Application cache API	152	18.6	用户界面属性	197
16.4.1	浏览器支持性检测	152	18.6.1	box-sizing 属性	197
16.4.2	ApplicationCache 对象	152	18.6.2	resize 属性	199
第 17 章	HTML5 的其他应用	159	18.6.3	outline-offset 属性	200
17.1	HTML5 Notifications	159	18.6.4	nav-index 属性	201

18.6.5 定义方向键控制顺序	202	18.11.8 animation-play-state 属性	257
18.7 基础盒模型	204	18.12 渐变	259
18.7.1 overflow 属性	204	18.12.1 linear-gradient	259
18.7.2 overflow-x 与 overflow-y 属性	206	18.12.2 radial-gradient	261
18.8 多栏属性	208	18.12.3 repeating-linear-gradient	265
18.8.1 columns 属性	208	18.12.4 repeating-radial-gradient	266
18.8.2 column-width 属性	210	18.13 其他模块	268
18.8.3 column-count 属性	211	18.13.1 Speech	268
18.8.4 column-gap 属性	212	18.13.2 Media queries	270
18.8.5 column-rule 属性	214	第 19 章 CSS3 选择器	273
18.8.6 column-rule-color 属性	215	19.1 属性选择器	273
18.8.7 column-rule-style 属性	215	19.2 伪类选择器	274
18.8.8 column-rule-width 属性	216	19.2.1 结构性伪类选择器	274
18.8.9 column-span 属性	217	19.2.2 UI 元素状态伪类选择器	284
18.8.10 column-fill 属性	219	19.2.3 目标伪类选择器	286
18.9 变换属性	221	19.2.4 否定伪类选择器	287
18.9.1 transform 属性	221	19.3 兄弟选择器	288
18.9.2 transform-origin 属性	233	第 20 章 JavaScript 基础	290
18.9.3 transform-style 属性	235	20.1 JavaScript 起步	290
18.9.4 perspective 属性	236	20.1.1 JavaScript 简介	290
18.9.5 perspective-origin 属性	238	20.1.2 JavaScript 的功能特性	290
18.9.6 backface-visibility 属性	241	20.1.3 JavaScript 的使用	291
18.10 过渡属性	243	20.2 注释	292
18.10.1 transition 属性	243	20.3 变量	293
18.10.2 transition-property 属性	244	20.4 运算符与表达式	295
18.10.3 transition-duration 属性	245	20.5 流程控制	298
18.10.4 transition-timing-function 属性	247	20.5.1 if 条件语句	299
18.10.5 transition-delay 属性	248	20.5.2 switch 语句	301
18.11 动画属性	249	20.5.3 for 循环语句	301
18.11.1 animation 属性	250	20.5.4 for... in 语句	302
18.11.2 animation-name 属性	251	20.5.5 while 语句和 do-while 语句	303
18.11.3 animation-duration 属性	253	20.5.6 break 语句和 continue 语句	304
18.11.4 animation-timing-function 属性	253	20.6 函数	305
18.11.5 animation-delay 属性	253	20.6.1 定义函数	306
18.11.6 animation-iteration-count 属性	254	20.6.2 return 语句	306
18.11.7 animation-direction 属性	254	20.6.3 函数的调用	306
		20.7 消息对话框	309
		20.8 异常处理	311
		20.8.1 捕获异常 (try-catch-finally)	311

20.8.2 声明异常 (throw)	311	20.11.2 Cookie 的存储、读取和删除 ...	329
20.9 内置对象	312	第 21 章 HTML5 与 CSS3 范例	332
20.9.1 字符串内置对象方法	312	21.1 垃圾收集箱	332
20.9.2 布尔型内置对象方法	317	21.2 拖放上传图片	335
20.9.3 数组内置对象方法	317	21.3 监控文件读取进度	337
20.9.4 数学内置对象方法	319	21.4 基于 EasyWebSocket 的实时	
20.9.5 日期内置对象方法	321	聊天室	340
20.10 校验	326	21.5 图片墙	344
20.11 Cookie 脚本	328	21.6 幻灯片播放	349
20.11.1 Cookie 的属性	329	21.7 手风琴式垂直菜单	354

第1章 HTML5 概述

1.1 认识 HTML5

HTML 的上一个版本诞生于 1999 年。从那以后, Web 世界经历了巨变。目前, HTML5 已经成为炙手可热的前端语言。

HTML5 实际指的是包括 HTML、CSS 和 JavaScript 在内的一套技术组合。它希望能够减少浏览器对于丰富网络应用服务插件 (plug-in-based rich internet application, RIA), 如 Adobe Flash、Microsoft Silverlight 以及 Oracle JavaFX 的依赖, 并且提供更多能有效增强网络应用的标准集。

HTML5 是 W3C 与 WHATWG 合作的结果, 将成为 HTML、XHTML 以及 HTML DOM 的新标准。HTML5 仍处于完善之中。不过, 大部分现代浏览器已经具备了某些 HTML5 支持。

1.2 HTML5 新特性

1.2.1 新的元素

HTML5 提供了一些新的元素和属性, 一些过时的 HTML4 标记将被取消, 其中包括纯粹用做显示效果的标记, 如和<center>, 因为它们已经被 CSS 取代。

HTML5 中定义的新元素见表 1-1。

表 1-1 HTML5 中的新元素

标 签	说 明	标 签	说 明
<article>	定义 article	<mark>	定义有记号的文本
<aside>	定义页面内容之外的内容	<meter>	定义预定义范围内的度量
<audio>	定义声音内容	<nav>	定义导航链接
<canvas>	定义画布	<output>	定义输出的一些类型
<command>	定义命令按钮	<progress>	定义任何类型的任务的进度
<datalist>	定义下拉列表	<rp>	定义若浏览器不支持 ruby 元素则显示的内容
<details>	定义元素的细节	<rt>	定义 ruby 注释的解释
<embed>	定义外部交互内容或插件	<ruby>	定义 ruby 注释
<figcaption>	定义 figure 元素的标题	<section>	定义 section
<figure>	定义媒介内容的分组以及它们的标题	<source>	定义媒介源
<footer>	定义 section 或 page 的页脚	<summary>	定义 details 元素的标题
<header>	定义 section 或 page 的页眉	<time>	定义日期/时间
<hgroup>	定义文档中的 section 的信息	<video>	定义视频
<keygen>	定义生成密钥		

1.2.2 新的 API

除了原先的 DOM 接口, HTML5 增加了更多样化的 API。下面是 HTML5 中一些有趣的新特性。

- (1) Canvas API: 动态生成图形、图表、图像以及动画。
- (2) Audio 与 Video API: 为开发者提供了一套通用的、集成的、脚本式的处理音频与视频的 API, 而无需安装任何插件。
- (3) Form API: 包含多个新的表单输入类型。这些新特性提供了更好的输入控制和验证。
- (4) Web Storage API: 在客户端浏览器中以键值的形式在本地存储数据, 无论用户离开站点还是关闭浏览器后再次打开时存储数据都会存在。
- (5) Communication API: 构建实时 (real-time) 和跨源 (cross-origin) 通信。
- (6) Geolocation API: 用户可共享地理位置, 并在 Web 应用的协助下享用位置感知服务。
- (7) Drag and Drop API: 通过鼠标对目标元素进行拖放操作。
- (8) File API: 处理文件上传和操纵文件。
- (9) Web SQL Database API: 允许应用程序通过一个异步的 JavaScript 接口访问数据库。

1.3 文档结构与字符集

说到文档结构, 请读者首先看一下 HTML4 的文档:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

是否既麻烦又难记? 如果是这样的话, 现在该切换到新的 HTML5 文档结构了。HTML5 文档非常容易建立, 只需要改变文档头部的 DOCTYPE 声明即可。

```
<!DOCTYPE html>
```

与 DOCTYPE 一样, 字符集的声明也简化了。原来的声明是

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
```

现在简化为

```
<meta charset=gb2312" />
```

1.4 语义化标记

HTML5 不仅把现有的标记变得更简洁, 还引入了很多新的语义化标记元素。Google 分析了上百万的页面, 发现 div 标签的 ID 名称重复量很大, 如程序员习惯将页脚定义为<div

`id="footer">`，所以 HTML5 引入了一组新的语义化标记元素。表 1-2 列出了新增的语义化标记元素。

表 1-2 HTML5 新增的语义化标记元素

元素名	说 明
header	定义文档的页眉，header 元素可以用来包裹搜索框、目录列表或任何相关的 logo
section	定义文档中的节，比如章节、页眉、页脚或文档中的其他部分
article	定义外部的内容。外部内容可以是来自一个外部的新闻提供者的一篇新的文章，或者是来自 blog 的文本，或者是来自论坛的文本，还可以是来自其他外部源的内容
nav	定义导航链接的部分
aside	定义 article 以外的内容。aside 的内容应该与 article 的内容相关。aside 的内容可用做文档的侧栏
footer	定义 section 或 document 的页脚。它包含创作者的姓名、文档的创作日期以及联系信息

HTML5 的语义化标记元素改变了以往单纯使用 table 和 div 布局的模式，构成了 HTML5 文档的基本结构。

代码清单 1-1：语义化标记元素的网页整体布局的实例

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>HTML5</title>
<link rel="stylesheet" href="15-1.css">
</head>
<body>
<header>
  <h1>Header</h1>
</header>
<div id="container">
  <nav>
    <h3>Nav</h3>
    <a href="">新闻中心</a>
    <a href="">产品展示</a>
    <a href="">人才招聘</a>
    <a href="">关于我们</a>
  </nav>
  <section>
    <h2>section</h2>
    <article>
      <header> <h1>Article Header</h1> </header>
      <p>An article should make sense on its own and it should be possible to distribute it
independently from the rest of the site. </p>
      <footer> <h2>Article Footer</h2> </footer>
    </article>
  </section>
</div>

```

```

<header> <h1>Article Header</h1> </header>
<p>The article tag is new in HTML5.The article tag specifies independent, self-contained content. </p>
<p>An article should make sense on its own and it should be possible to distribute it
independently from the rest of the site. </p>
<footer> <h2>Article Footer</h2> </footer>
</article>
</section>
<aside>
<h3>Aside</h3>
<p>The Aside tag is new in HTML5.The aside tag defines some content aside from the content
it is placed in. The aside content should be related to the surrounding content. </p>
</aside>
<footer> <h2>Footer</h2> </footer>
</div>
</body>
</html>

```

构建完程序后，下面为代码 1-1 应用样式表。本实例中使用了 CSS3 的一些新的特性，例如圆角处理。CSS3 同 HTML5 一样仍处于开发阶段，并且逐渐被大多数浏览器所支持。下面是代码 1-1 对应的 CSS 文件。

```

body {
    background-color:white;
    font-family:Tahoma, Geneva, sans-serif;
    margin: 0px auto;
    border:solid;
    border-color:#006; }
header {
    background-color: #00F;
    display:block;
    color:#FFF;
    text-align:center; }
h1 {
    font-size: 72px;
    margin: 0px; }
h2 {
    font-size: 24px;
    margin: 0px;
    text-align:center;
    color: #FFF; }
h3 {
    font-size: 18px;
    margin: 0px;
    text-align:center;
    color: #FF0; }
nav {
    display:block;

```

```
width:20%;
float:left; }

nav a:link, nav a:visited {
display: block;
border-bottom: 3px solid #fff;
padding: 10px;
text-decoration: none;
font-weight: bold;
margin: 5px; }

nav a:hover {
color: #F00;
background-color: #FF0; }

nav h3 {
margin: 15px;
color: #FFF; }

section h3 {
margin: 15px;
color: #FFF; }

#container { background-color: #888; }

section {
display:block;
width:60%;
float:left;
background-color: #0F0; }

article {
background-color: #eee;
display:block;
margin: 10px;
padding: 10px;
-webkit-border-radius: 10px;
-moz-border-radius: 10px;
border-radius: 10px; }

article header {
-webkit-border-radius: 10px;
-moz-border-radius: 10px;
border-radius: 10px;
padding: 5px; }

article footer {
-webkit-border-radius: 10px;
-moz-border-radius: 10px;
border-radius: 10px;
padding: 5px; }

article h1 { font-size: 18px; }

aside {
display:block;
width:20%;
```

```
float:left; }
aside h3 {
  margin: 15px;
  color: white; }
aside p {
  margin: 15px;
  color: #0F0;
  font-weight: bold;
  font-style: italic; }
footer {
  clear: both;
  display: block;
  background-color: #00F;
  color: #FFFFFF;
  text-align:center;
  padding: 15px; }
footer h2 {
  font-size: 14px;
  color: white; }
a { color: #00F; }
a:hover {text-decoration: underline; }
```

图 1-1 是代码 1-1 应用了 CSS3 后，在页面中的实现效果。从图中不难看出各个语义化标记元素在构建页面布局中的作用。

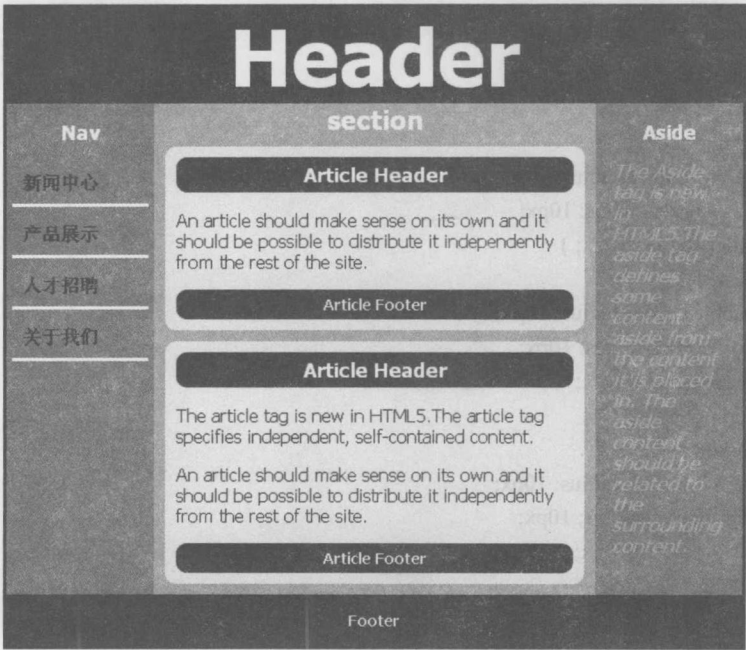


图 1-1 语义化标记元素构建布局效果

1.5 检测技术

由于 HTML5 的新规范还在完善之中,一些老的浏览器对 HTML5 的特性还不支持或只有部分支持,所以对浏览器的支持性进行检测是很有必要的。以下提供了四种可以用于检测浏览器是否支持某种 HTML5 特性的方法:

- (1) 使用 Modernizr 检测库。
- (2) 检测全局对象是否拥有特性属性。
- (3) 创建一个元素,然后检测该元素的 DOM 对象是否拥有特定的属性。
- (4) 创建一个元素,给这个元素的 DOM 对象设定特定的属性值,然后检查浏览器是否保留了该属性值。

下面以使用 Modernizr 检测库检测为例进行说明,其他检测方法将在后面章节中进一步介绍。

Modernizr 是一个开源的 JavaScript 库,用于检测浏览器是否支持 HTML5 和 CSS3 的特性。可以根据浏览器对 HTML5 和 CSS3 的支持程度提供更加便捷的前端优化方案。

可以从 www.modernizr.com 下载 Modernizr 的最新稳定版。Modernizr 的使用非常简单,只需在<script>标签中加载 Modernizr 库,并把该标签放在<head>标签下。Modernizr 在运行时创建一个 Modernizr 的全局对象,该对象会为每个需要检测的特性创建一个布尔型的属性。

下面以检测当前浏览器是否支持 Canvas 为例,说明如何使用 Modernizr 特性检测库。代码如下:

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Modernizr 特性检测</title>
<script src="modernizr.custom.js" ></script>
<script type="text/javascript">
    if(Modernizr.canvas){
        alert("Your browser supports canvas!!");
    }else{ alert("Sorry, Your browser does not support canvas!!"); }
</script>
</head>
<body>
.....
</body>
</html>
```