

Amazon畅销书，被翻译成西班牙语等多种文字，广受好评，被誉为HTML5领域的经典著作
详尽讲解和分析了HTML5中的所有新特性和核心技术，能为迅速提升HTML5开发技能提供
绝佳指导

实战性和实用性强，包含297个精心组织的代码示例和16个快速参考索引

HTML5 for Masterminds

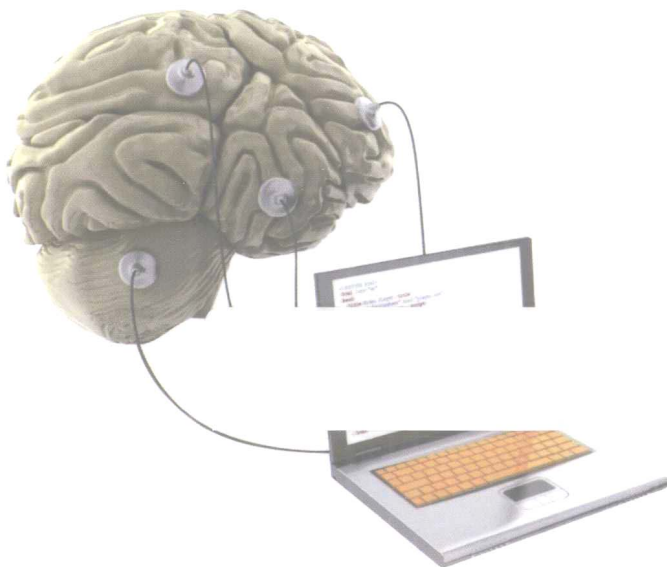
How to Take Advantage of HTML5 to Create Amazing Websites and Revolutionary Applications

HTML5精粹

利用HTML5开发令人惊奇的
Web站点和革命性应用

(美) J. D. Gauchat 著

曾少宁 张猛 赵俐 译



机械工业出版社
China Machine Press

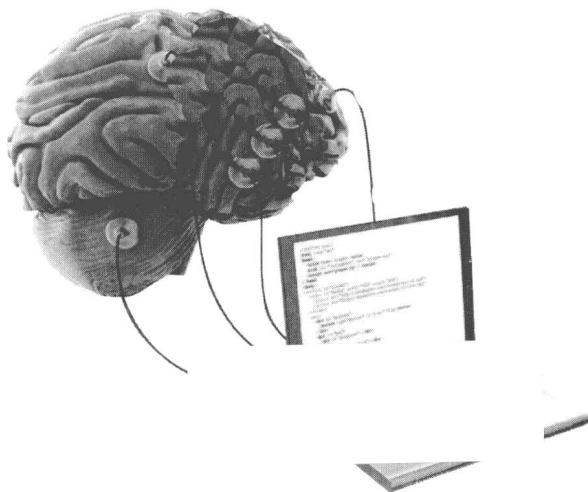
HTML5 for Masterminds
How to Take Advantage of HTML5 to Create Amazing Websites and Revolutionary Applications

HTML5精粹

利用HTML5开发令人惊奇的
Web站点和革命性应用

(美) J. D. Gauchat 著

曾少宁 张猛 赵俐 译



机械工业出版社
China Machine Press

Amazon 畅销书，被翻译成西班牙语等多种文字，广受好评，被誉为 HTML5 领域的经典著作。详尽地讲解和分析了 HTML5 中的所有新特性和核心技术，能为有一定 HTML 基础的读者迅速提升 HTML5 开发技能提供绝佳指导。全书包含 297 个代码示例和 16 个快速参考索引，极具实战性和实用性。

全书共 16 章：第 1 章介绍了 HTML5 文档的基本组成、全局结构、主体，以及新旧元素对比；第 2 章和第 3 章讲解了 CSS 的样式设置、框模型，以及 CSS3 的核心属性和新规则；第 4 章重点介绍了 JavaScript 的核心知识；第 5 章详细介绍了 HTML5 中的音频和视频特性，以及如何在 HTML5 中创建视频和音频；第 6 章讲解了 HTML5 中的各种表单（包含新增表达元素）及其新属性，以及表单 API；第 7 章介绍了 Canvas 的特性及其使用，以及 Canvas 的 API；第 8 章探讨了 HTML5 中的拖放操作及其 API；第 9 章介绍了 HTML5 中的地理位置信息及其 API；第 10 章讲解了 HTML5 中的各种 Web 存储技术，以及 Web 存储 API；第 11 章阐述了索引数据库的具体操作、实现原理，以及它的 API；第 12 章讲解了 HTML5 中的文件操作和文件系统，以及文件 API；第 13 章介绍了 HTML5 中的 Ajax、跨文档消息传递、Web 套接字等通信技术的原理，以及它们的 API；第 14 章讲解了 HTML5 中 Web Workers 的使用方法和实现原理，以及它的 API；第 15 章和第 16 章介绍了 HTML5 中的历史接口和脱机功能，以及它们的 API。

Authorized translation from the English language edition entitled HTML 5 for Masterminds: How to Take Advantage of HTML5 to Create Amazing Websites and Revolutionary Applications by J.D Gauchat, published by Marcombo, Inc, Copyright © 2011.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Marcombo, Inc.

Chinese simplified language edition published by China Machine Press.

Copyright © 2012 by China Machine Press.

本书中文简体字版由 Marcombo 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。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2012-2647

图书在版编目（CIP）数据

HTML5 精粹：利用 HTML5 开发令人惊奇的 Web 站点和革命性应用 /（美）高奇特（Gauchat, J. D.）著；曾少宁，张猛，赵俐译．—北京：机械工业出版社，2012.9

（华章专业开发者丛书）

书名原文：HTML5 for Masterminds: How to Take Advantage of HTML5 to Create Amazing Websites and Revolutionary Applications

ISBN 978-7-111-39771-7

I . H… II . ① 高… ② 曾… ③ 张… ④ 赵… III . 超文本标记语言—程序设计 IV . TP312

中国版本图书馆 CIP 数据核字（2012）第 218889 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：谢晓芳

薰城市京瑞印刷有限公司印刷

2012 年 9 月第 1 版第 1 次印刷

186mm×240mm·19 印张

标准书号：ISBN 978-7-111-39771-7

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

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

译者序

HTML——由来已久

说起我与 HTML 的缘分，由来已久。1997 年，刚开始使用浏览器，就开始制作小网页，制作简单的网站。但那个时候，学习编程的我，对于标签语言（尤其是定义标签的 <> 符号）有那么一点点不习惯。因此主要的工作还是通过编程，也就是利用单机或 C/S 架构的程序来解决问题。

1999 年，受 ZDNET 网站之约，我为 ZDNET 网站翻译了完整的 HTML4.0 规范。翻译的好处是，可以强迫自己细读全部标签、属性，从此，我打下了坚实的 HTML 基础。虽然我的主要工作是管理开发、提供解决方案，但在协调开发人员和美工进行 B/S 架构的开发时，依靠纯粹手写 HTML 的经验，让我减少了许多沟通和管理障碍。

直到深入理解了 HTML，我才明白，一切界面都是拼接、画出来的，不论是多么复杂的网站、多么精美的应用、多么漂亮的设计，都源自那些“朴实”的标记。

时隔 8 年之后，2008 年年底，我又翻译了《HTML、XHTML 和 CSS 宝典（第 4 版）》，这使我对 HTML、CSS、JavaScript 再次温故而知新。

HTML——不变应万变之技术老枪

IT 技术领域有一个现象，就是新技术层出不穷，IT 人员疲于追逐新技术，却很少精通哪项技术。十几年间，Windows 等操作系统、.NET、Java 等编程技术经历了几代的演进，Web 也从 1.0 发展到 2.0。但作为互联网基石的 HTML 始终保持了相对的稳定。如果说有什么技术在学习了之后，可以保持长久的生命力，HTML、CSS、JavaScript 无疑是其中的典范。每次遇到年轻的项目组开发人员只知道关注那些后台框架，却对 HTML 一知半解的时候，我这杆“老枪”不得不亲自上阵，用我最喜欢的写字板来解决他们的 HTML 代码问题。HTML 的魅力，即使在多年之后，仍然给我带来了无限的冲击力。

在这个过程中，我也面临选择。2009 年接手一个公司的项目，受微软新技术的吸引，一定要使用 SilverLight。当我用 HTML+Web 2.0+SNS 设计出全新架构之后，SilverLight 已经弃用了。

关于 Flex、Flash 的计算机外版图书，我其实也翻译过不少。但当有一家电信公司跟我说脱离了 Flex，就做不到百万级数据报表在浏览器中实时显示的时候，我又用 HTML+Web 2.0 给予了回击。

HTML5——揭开崭新篇章

但是，随着 B/S 架构、普适计算和移动运算的普及，PC、平板电脑、手机、家电各种计算

设备的整合,各种专有技术,包括 ANDROID、MS、APPLE、ADOBE FLASH 等,都暴露出自己的瓶颈,虽然程序员精心开发了一个应用,但是大多数情况下只能适用于一个平台,为了保持兼容性,又要投入巨大的精力。这种人为的壁垒束缚了程序员的创造力,也限制了用户的选择。

终于,巨头们达成一致,齐心协力共同开发 HTML5。HTML5 展示了美好的前景:良好的跨浏览器标准支持、本地存储、脱机运行、GPS 地理位置支持、视频、绘图、多媒体……可以说,对传统的 B/S 架构的应用和最新的移动应用提供全面支持。

如果你因为苹果公司封闭的开发体系和 Object-C 不好掌握而错过了开发《愤怒的小鸟》等热门游戏的大好机会;如果你因为安卓平台版本众多、兼容困难、投入大、见效少而痛苦;如果你因为微软 .NET 平台对平板、手机的支持欠缺而错失移动互联网的机会……那么,现在一个新的机会正呈现在你面前,那就是 HTML5。从理论上讲,用 HTML5 开发的应用程序,在台式机、平板设备、安卓手机、苹果手机、其他智能设备、各个主流浏览器上,都能完美运行。

重大的技术更新,必然带来一次重大的应用更新,催生新的市场,而这种技术是跨越所有主流厂商私有技术并得到普遍支持的技术,这个市场会有多大,有多少机会,我不予猜测。

如本书所述,本书介绍的许多技术,在当前主流浏览器中的实现,目前仍然处于“实验”阶段。而据最新报道,HTML5 的最新标准大约要到 2014 年才能推出。另外,微软目前推出的 Windows 8 和 Internet Explorer 10 据说全面支持 HTML5 标准,其他厂商也不甘落后。那么,这当中会有多少机会,相信已经不言而喻。

但是,无论如何,有一点是可以肯定的,HTML 是这些年来最稳定、最持久、最成功的技术,或者谦虚一点,加上两个字:之一。

关于本书——快速升级书

本书适合于有一定 HTML、CSS、JavaScript 基础的读者阅读。首先,作者是一个资深程序员,行文作风很像是在研讨会上对听众介绍新技术,言简意赅,重点突出,但显然不会顾及全部细节。另外,本书只有前 4 章对 HTML、CSS、JavaScript 做了简要介绍,其余 12 章,全部放在 HTML5 的新增功能方面。这样做,有助于通过这本薄薄的书就能掌握这些技术的精华,让你的技术一夜之间实现大幅飞跃。

最后,衷心感谢机械工业出版社的编辑在本书翻译过程中给予的精心指导和宝贵意见,为本书付出的大量心血和耐心细致的审阅。由于译者水平有限,难免有不当之处,恳请各位读者不吝指正。

前言

HTML5 并不是旧标记语言的新版本——甚至也不是对已“过时”技术的改进，而是移动设备、云计算和网络时代的一种网站和应用程序开发新概念。

很久之前，使用简单版本的 HTML，就可以创建基本网页结构、组织内容和共享信息。最初，这种语言和 Web 主要是用于实现基于文本的通信方式。

由于 HTML 的使用范围有限，因此许多企业都开发了新的语言和软件，以便在 Web 中添加新的特性。这些最初的开发逐渐成为强大和流行的插件。简单游戏和动画效果很快转变成复杂的应用程序，给人们带来全新的体验，并从此改变了 Web 的概念。

在所有插件中，Java 和 Flash 是最成功的。它们应用广泛，并且被认为是互联网的未来。但是，随着用户的增长，而且互联网从最初的计算机爱好者之间的互联工具转变为以商业和社交互动为主的场所，这两种技术的局限性最终使它们越来越不受欢迎。

Java 和 Flash 最主要的问题在于缺乏集成性。两者从一开始便以插件形式出现，有时候还需要插入文档结构中，但是实际上只是与该结构共享屏幕的一部分空间。在应用程序和文档之间不存在通信能力和集成性。

缺乏集成性的问题越来越严重，也为语言的演变做好了准备，使之共享 HTML 文档的空间，成为不受插件限制影响的组件。JavaScript 是嵌入在浏览器中的一种解释性语言，同时也是一种改善用户体验和实现 Web 功能的方法。然而，在过去几年里，由于推广及滥用问题，市场并没有完全接受这种语言，而且其流行性也有逐渐下降。批评者有着很好的理由反对这种语言。在那段时间里，JavaScript 无法替代 Flash 和 Java 的功能。甚至，有一个很明显的现实情况是，Java 和 Flash 限制了 Web 应用程序的范围，并且隔离了 Web 内容，而一些流行特性（如流式视频）正成为 Web 的重要组成部分，并且只有通过这些技术才能有效地实现。

尽管取得了很大的成功，但是 Java 也存在一些缺陷。这种语言具有非常复杂的特性，发展缓慢，缺乏集成性，这些问题都直接限制 Java 在目前主流 Web 应用程序中的应用。如果不使用 Java，人们只能使用 Flash。然而，Flash 实际上与其竞争对手具有相同的 Web 特征，因此注定会退出市场。

同时，访问 Web 的软件在不断发展。除了增加新特性和提高互联网访问速度，浏览器还不断改进其 JavaScript 引擎。增强的功能带来了更多的机会，而这种脚本语言也已经做好了准备。

从某种程度上讲，那些不使用 Java 或者 Flash 的开发人员在这个过程中可以为越来越多的用户提供创建应用程序所需要的工具。这些开发人员开始在他们的应用程序中以全新的方式编写 JavaScript 代码。这种创新及其令人吃惊的结果引起了越来越多程序员的注意。很快，所谓的

“Web 2.0”开始出现，开发者社区对于 JavaScript 的认识也发生了根本性改变。

显然，JavaScript 是一种允许开发者在网页上创新和实现特殊效果的语言。近几年来，全世界的程序员和网页设计人员使用了很多方法，希望克服这种技术的局限性及一直存在的可移植性问题。JavaScript、HTML 和 CSS 显然是引领 Web 变革的最佳组合。

事实上，HTML5 正是对这个组合的改进，是将整合这一切的黏合剂。HTML5 标准涉及 Web 的各个方面，也清晰定义每一种技术的用途。从现在开始，HTML 负责设置文档结构（结构元素），CSS 则关注于如何将结构转换为可视化效果和可用性，而 JavaScript 则负责实现功能和开发完整的 Web 应用程序。

网站与应用程序之间的界限已经完全消失。所需要的技术也一应俱全。网页的未来是光明的，而将这三种技术（HTML、CSS 和 JavaScript）演变和整合为一个强大的规范便是将 Internet 转化成一个主导的开发平台。HTML5 显然是朝着这个方向发展的。

重要提示：目前，并非所有的浏览器都支持 HTML5 特性，而且大多特性仍处于设计阶段。我们建议您阅读各个章节，并在最新版本的 Google Chrome、Safari、Firefox 或 Internet Explorer 上执行示例代码。Google Chrome 基于 WebKit，这是一个开源浏览器引擎，几乎支持所有的 HTML5 特性，因此 Google Chrome 是一个很好的测试平台。Firefox 是一个很适合开发者使用的优秀浏览器，它采用 Gecko 引擎，同样也完全支持 HTML5。最后，新版本的 Internet Explorer（IE9）已经支持 HTML5 及大部分新特性。

不管使用哪种浏览器，一定要注意：优秀的开发者必须在市面上所有浏览器上安装和测试所编写的代码。要在每一个浏览器上测试本书所提供的示例代码。

通过以下链接可以下载最新版本的浏览器：

- www.google.com/chrome
- www.apple.com/safari/download
- www.mozilla.com
- windows.microsoft.com
- www.opera.com

本书最后提供了多种兼容旧版浏览器的设计方法，使未支持 HTML5 的浏览器也能够正常访问网站与应用程序。

目 录

译者序

前言

第 1 章 HTML5 文档	1
1.1 基本组成	1
1.2 全局结构	1
1.2.1 Doctype	2
1.2.2 <html>	2
1.2.3 <head>	3
1.2.4 <body>	3
1.2.5 <meta>	4
1.2.6 <title>	5
1.2.7 <link>	5
1.3 正文结构	6
1.3.1 组织	7
1.3.2 <header>	9
1.3.3 <nav>	9
1.3.4 <section>	10
1.3.5 <aside>	11
1.3.6 <footer>	12
1.4 深入正文	13
1.4.1 <article>	14
1.4.2 <hgroup>	16
1.4.3 <figure> 和 <figcaption>	18
1.5 新旧元素	20
1.5.1 <mark>	20
1.5.2 <small>	20
1.5.3 <cite>	21
1.5.4 <address>	21

1.5.5 <time>	21
1.6 快速参考——HTML5 文档	22
第 2 章 CSS 样式设置与框模型	24
2.1 CSS 与 HTML	24
2.2 样式与结构	24
2.2.1 块级元素	25
2.2.2 框模型	25
2.3 样式设置基础	26
2.3.1 内联样式	26
2.3.2 嵌入样式	27
2.3.3 外部文件	27
2.3.4 引用	28
2.3.5 通过关键字引用	28
2.3.6 通过 id 属性引用	29
2.3.7 通过 class 属性引用	29
2.3.8 通过任意属性引用	30
2.3.9 通过伪类引用	31
2.3.10 新选择器	34
2.4 在模板上应用 CSS	35
2.5 传统框模型	35
2.5.1 模板	35
2.5.2 全局选择器 *	37
2.5.3 新的标题层次结构	38
2.5.4 声明新的 HTML5 元素	38
2.5.5 居中显示正文	39
2.5.6 创建主框	39
2.5.7 标头	40
2.5.8 导航栏	40
2.5.9 节与边栏	41

2.5.10 页脚	42	3.2 快速参考——CSS3 属性	79
2.5.11 最后一步	44	第 4 章 JavaScript	81
2.5.12 框尺寸	44	4.1 JavaScript 相关性	81
2.6 弹性框模型	45	4.2 引入 JavaScript	81
2.7 理解弹性框模型	52	4.2.1 内联脚本	81
2.7.1 display	53	4.2.2 嵌入脚本	82
2.7.2 box-orient	53	4.2.3 外部文件	84
2.7.3 box-direction	54	4.3 新选择器	84
2.7.4 box-ordinal-group	54	4.3.1 querySelector()	85
2.7.5 box-pack	56	4.3.2 querySelectorAll()	85
2.7.6 box-flex	57	4.4 事件处理程序	86
2.7.7 box-align	60	4.4.1 内联事件处理程序	87
2.8 快速参考——CSS 样式设置与框模型	61	4.4.2 在属性中添加事件处理程序	87
2.8.1 弹性框模型	61	4.4.3 addEventListener() 方法	87
2.8.2 伪类与选择器	62	4.5 API	88
第 3 章 CSS3 属性	63	4.5.1 Canvas API	89
3.1 新规则	63	4.5.2 拖放 API	89
3.1.1 强大的 CSS3	63	4.5.3 地理位置 API	89
3.1.2 模板	63	4.5.4 存储 API	89
3.1.3 圆角	65	4.5.5 文件 API	90
3.1.4 阴影效果	66	4.5.6 通信 API	90
3.1.5 文字阴影	68	4.5.7 Web Workers API	90
3.1.6 @font-face	69	4.5.8 历史 API	90
3.1.7 线性渐变	70	4.5.9 离线 API	91
3.1.8 放射渐变	71	4.6 外部库	91
3.1.9 RGBA	71	4.6.1 jQuery	91
3.1.10 HSLA	72	4.6.2 Google Maps	91
3.1.11 轮廓	73	4.7 快速参考——JavaScript	92
3.1.12 边框图像	73	4.7.1 元素	92
3.1.13 转换与转变	74	4.7.2 选择器	92
3.1.14 一次实现全部转换	77	4.7.3 事件	92
3.1.15 动态转换	78	4.7.4 API	92
3.1.16 过渡	78	第 5 章 视频与音频	94
		5.1 在 HTML5 中播放视频	94

5.1.1	<video> 元素	95	6.1.14	本地日期与时间类型	112
5.1.2	<video> 属性	95	6.1.15	颜色类型	112
5.2	编程实现视频播放器	96	6.2	新属性	112
5.2.1	设计	96	6.2.1	placeholder 属性	113
5.2.2	编码	98	6.2.2	required 属性	113
5.2.3	事件	98	6.2.3	multiple 属性	113
5.2.4	方法	99	6.2.4	autofocus 属性	113
5.2.5	属性	100	6.2.5	pattern 属性	114
5.2.6	实际代码	100	6.2.6	form 属性	114
5.3	视频格式	103	6.3	新的表单元素	114
5.4	在 HTML5 中播放音频	103	6.3.1	<datalist> 元素	115
5.5	编程实现音频播放器	105	6.3.2	<progress> 元素	115
5.6	快速参考——视频与音频	105	6.3.3	<meter> 元素	115
5.6.1	元素	105	6.3.4	<output> 元素	115
5.6.2	内嵌属性	106	6.4	表单 API	115
5.6.3	视频属性	106	6.4.1	setCustomValidity()	116
5.6.4	事件	106	6.4.2	无效事件	117
5.6.5	方法	106	6.4.3	实时验证	119
5.6.6	属性	107	6.4.4	有效性约束	121
第 6 章	表单与表单 API	108	6.4.5	willValidate	122
6.1	HTML 网页表单	108	6.5	快速参考——表单与表单 API	122
6.1.1	<form> 元素	108	6.5.1	类型	122
6.1.2	<input> 元素	109	6.5.2	属性	122
6.1.3	电子邮件类型	109	6.5.3	元素	123
6.1.4	搜索类型	109	6.5.4	方法	123
6.1.5	URL 类型	110	6.5.5	事件	124
6.1.6	电话号码类型	110	6.5.6	状态	124
6.1.7	数字类型	110	第 7 章	Canvas API	125
6.1.8	范围类型	110	7.1	准备 Canvas	125
6.1.9	日期类型	111	7.1.1	<canvas> 元素	125
6.1.10	周类型	111	7.1.2	getContext()	126
6.1.11	月份类型	111	7.2	在 Canvas 上绘图	126
6.1.12	时间类型	111	7.2.1	绘制矩形	126
6.1.13	日期与时间类型	112	7.2.2	颜色	127

7.2.3 渐变	128	9.1.3 getCurrentPosition(location, error, configuration)	166
7.2.4 创建路径	129	9.1.4 watchPosition(location, error, configuration)	167
7.2.5 线型	134	9.1.5 Google Maps 实战	168
7.2.6 文字	135	9.2 快速参考——地理位置 API	169
7.2.7 阴影	136	9.2.1 方法	169
7.2.8 转换	137	9.2.2 对象	170
7.2.9 恢复状态	139	第 10 章 Web 存储 API	171
7.2.10 globalCompositeOperation	139	10.1 两个存储系统	171
7.3 处理图像	141	10.2 sessionStorage	172
7.3.1 drawImage()	141	10.2.1 数据存储的实现	172
7.3.2 图像数据	142	10.2.2 创建数据	173
7.3.3 图案	144	10.2.3 读取数据	174
7.4 在 Canvas 上实现动画	145	10.2.4 删除数据	175
7.5 在 Canvas 上处理视频	146	10.3 localStorage	176
7.6 快速参考——Canvas API	148	10.3.1 storage 事件	177
7.6.1 方法	149	10.3.2 存储空间	178
7.6.2 属性	151	10.4 快速参考——Web 存储 API	178
第 8 章 拖放 API	152	10.4.1 存储类型	179
8.1 Web 拖放	152	10.4.2 方法	179
8.1.1 新的事件	152	10.4.3 属性	179
8.1.2 dataTransfer	155	10.4.4 事件	179
8.1.3 dragenter、dragleave 与 dragend	156	第 11 章 IndexedDB API	180
8.1.4 选择有效的源	157	11.1 底层 API	180
8.1.5 setDragImage()	158	11.1.1 数据库	180
8.1.6 文件	160	11.1.2 对象和对象库	181
8.2 快速参考——拖放 API	161	11.1.3 索引	181
8.2.1 事件	161	11.1.4 事务	182
8.2.2 方法	162	11.1.5 对象库方法	182
8.2.3 属性	162	11.2 实现 IndexedDB	183
第 9 章 地理位置 API	163	11.2.1 模板	183
9.1 定位	163	11.2.2 打开数据库	184
9.1.1 getCurrentPosition(location)	163	11.2.3 数据库版本	185
9.1.2 getCurrentPosition (location, error)	165		

11.2.4 对象库和索引	186	12.3.6 处理文件	213
11.2.5 添加对象	186	12.3.7 移动	213
11.2.6 获取对象	187	12.3.8 复制	215
11.2.7 完成代码并测试	188	12.3.9 删除	216
11.3 列出数据	188	12.4 文件内容	217
11.3.1 游标	188	12.4.1 写入内容	217
11.3.2 修改顺序	190	12.4.2 添加内容	219
11.4 删除数据	192	12.4.3 读取内容	220
11.5 搜索数据	192	12.5 真实的文件系统	221
11.6 快速参考——IndexedDB API	195	12.6 快速参考——文件 API	225
11.6.1 环境接口 (IDBEnvironment 和 IDBFactory)	195	12.6.1 Blob 接口 (文件 API)	225
11.6.2 数据库接口 (IDBDatabase)	195	12.6.2 File 接口 (文件 API)	225
11.6.3 对象库接口 (IDBObjectStore)	196	12.6.3 FileReader 接口 (文件 API)	225
11.6.4 游标接口 (IDBCursor)	196	12.6.4 LocalFileSystem 接口 (文件 API : 目录和系统)	226
11.6.5 事务接口 (IDBTransaction)	197	12.6.5 FileSystem 接口 (文件 API : 目录和系统)	226
11.6.6 范围接口 (IDBKeyRange-Constructors)	197	12.6.6 Entry 接口 (文件 API : 目录和系统)	226
11.6.7 错误接口 (IDBDatabase-Exception)	197	12.6.7 DirectoryEntry 接口 (文件 API : 目录和系统)	226
第 12 章 文件	199	12.6.8 DirectoryReader 接口 (文件 API : 目录和系统)	227
12.1 文件存储	199	12.6.9 FileEntry 接口 (文件 API : 目录和系统)	227
12.2 处理用户文件	199	12.6.10 BlobBuilder 接口 (文件 API : 写入器)	227
12.2.1 模板	200	12.6.11 FileWriter 接口 (文件 API : 写入器)	227
12.2.2 读取文件	201	12.6.12 FileError 接口 (文件 API 和扩展)	228
12.2.3 文件属性	202	第 13 章 通信 API	229
12.2.4 blob	203	13.1 Ajax Level 2	229
12.2.5 事件	205	13.1.1 获取数据	229
12.3 创建文件	206		
12.3.1 模板	206		
12.3.2 硬盘	207		
12.3.3 创建文件	208		
12.3.4 创建目录	209		
12.3.5 列出文件	210		

13.1.2 响应属性	231	14.1.7 共享 worker	260
13.1.3 事件	231	14.2 快速参考——Web Workers API	263
13.1.4 发送数据	233	14.2.1 worker	263
13.1.5 跨源请求	235	14.2.2 专用 worker	264
13.1.6 上传文件	235	14.2.3 共享 worker	264
13.1.7 真实应用程序	237	第 15 章 历史 API	265
13.2 跨文档消息传递	240	15.1 History 接口	265
13.2.1 构造函数	240	15.1.1 Web 导航	265
13.2.2 消息事件和属性	240	15.1.2 新增方法	265
13.2.3 发送消息	241	15.1.3 伪造 URL	266
13.2.4 筛选器和跨源	243	15.1.4 跟踪	268
13.3 Web 套接字	245	15.1.5 实例	269
13.3.1 WS 服务器的配置	245	15.2 快速参考——历史 API	272
13.3.2 构造函数	246	第 16 章 脱机 API	273
13.3.3 方法	246	16.1 缓存代码清单	273
13.3.4 属性	246	16.1.1 代码清单文件	273
13.3.5 事件	247	16.1.2 分类	274
13.3.6 模板	247	16.1.3 注释	274
13.3.7 开始通信	248	16.1.4 使用代码清单文件	275
13.3.8 完整应用程序	249	16.2 脱机 API	276
13.4 快速参考——通信 API	250	16.2.1 错误	276
13.4.1 XMLHttpRequest Level 2	250	16.2.2 联机 and 脱机	277
13.4.2 Web 消息传递 API	252	16.2.3 缓存过程	278
13.4.3 WebSocket API	252	16.2.4 进度	280
第 14 章 Web Workers API	253	16.2.5 更新缓存	281
14.1 需要做的艰巨工作	253	16.3 快速参考——脱机 API	283
14.1.1 创建 worker	253	16.3.1 代码清单文件	283
14.1.2 发送和接收消息	253	16.3.2 属性	283
14.1.3 检测错误	256	16.3.3 事件	284
14.1.4 终止 worker	257	16.3.4 方法	284
14.1.5 同步 API	258	结束语	285
14.1.6 导入脚本	259		

第 ① 章

HTML5 文档

1.1 基本组成

HTML5 有三个基本特色：结构、样式和功能。虽未正式发布 HTML5，甚至其中也不包含一些 API 和整个 CSS3 规范，但 HTML5 仍被认为是 HTML、CSS 和 JavaScript 结合的产物。三者都是极为可靠的技术，并且在 HTML5 规范下组织为一个整体。HTML 负责结构，CSS 负责在屏幕上呈现该结构及其内容，而 JavaScript 则负责其余一些仍极为重要的工作（我们会在本书后面看到）。

虽说集三种不同技术于一身，但结构仍然是 HTML5 文档的核心部分。它提供了分配静态或动态内容所必需的元素，也是应用程序的基本平台。随着各种类型的设备可以访问 Internet，以及用于网络交互的界面日趋多样化，结构作为一个基本方面，俨然成为文档至关重要的部分。现在，结构必须提供外观、组织和灵活性，同时还必须如房屋地基一般坚固。

在使用 HTML5 创建网站和应用程序之前，首先需要扎实地了解它的结构，以便将来充分利用 HTML5 带来的新机会。

因此，让我们一步步从基础开始学起。在第 1 章中将学习如何使用 HTML5 引入的新的 HTML 元素来构建一个模板，以供未来的项目使用。

动手实践：用你喜欢的文本编辑器创建一个新的空文档，以便在浏览器中测试本章中的所有代码，从而帮助你记住新的标签并熟悉这个新的标记。

基础知识回顾：HTML 文档是一个文本文件。如果没有任何开发人员软件，可以使用 Windows 中的记事本或任何其他文本编辑器。文件必须保存为 .html 扩展名，文件名可以任取（例如：mycode.html）。

重要提示：为了了解更多信息和代码清单示例，请访问本书的网站：www.minkbooks.com。

1.2 全局结构

HTML 文档是严格组织的。文档的每个部分都放在特定的标签中进行声明并加以区分。在本章的这一部分中，我们将了解如何创建 HTML 文档的全局结构和 HTML5 中包含的新的语义元素。

1.2.1 Doctype

首先，需要指定要创建的文档类型（见代码清单 1-1）。在 HTML5 中，这是非常简单的。

代码清单 1-1: 使用 <doctype> 元素

```
<!DOCTYPE html>
```

重要提示：这行代码必须是文件的第一行，前面不能有任何空格或空行。这是激活标准模式的一种方式，如果浏览器支持 HTML5，它会强制要求浏览器解释 HTML5，如果浏览器不支持 HTML5，则忽略它。

动手实践：你现在就可以在 HTML 文件中写入代码并添加后面将要学习的各种新元素。

1.2.2 <html>

声明文档类型之后，必须建立 HTML 树结构。像以往一样，树的根元素是 <html> 元素（见代码清单 1-2）。所有的 HTML 代码都要放在这个元素中。

代码清单 1-2: 使用 <html> 元素

```
<!DOCTYPE html>
<html lang="en">

</html>
```

开始标签 <html> 中的 lang 属性只有在 HTML5 中才需要指定。此属性定义了创建的文档内容的人类语言，在这个例子中，en 是英语的意思。

基础知识回顾：HTML 使用标记语言创建网页。HTML 标签由关键字和属性构成，其中关键字和属性放在尖括号内，例如 <html lang="en">。在这个例子中，html 是关键字，lang 是属性，en 是属性的值。大多数 HTML 标签都是成对出现的，一个是开始标签，另一个是结束标签，内容放在这两个标签之间。在这个例子中，<html lang="es"> 表示 HTML 代码的开始，而 </html> 则表示结束。对比开始和结束标签，可以发现结束标签的区别是在关键字前面加一个斜杠（例如 </html>）。所有代码都需要插入 <html> 和 </html> 这两个标签之间。

重要提示：HTML5 在用来创建文档的结构和元素方面是相当灵活的。<html> 元素可以不包含任何属性，甚至可以完全省略。但出于兼容性以及其他一些原因（这些原因并不重要，这里不再列出）的考虑，我们建议遵守一些基本规则。下面将介绍如何根据一些最佳实践来创建 HTML 文档。

关于 lang 属性的其他语言选项，请参见此链接：www.w3schools.com/tags/ref_language_codes.asp。

1.2.3 <head>

继续完成模板。<html> 标签之间的 HTML 代码主要分成两部分。这里与之前版本的 HTML 相同，第一部分是文档头，而第二部分是文档正文。因此，下一步是在代码中使用已知元素 <head> 和 <body> 创建创建这两个部分。

当然，首先是 <head>，与其他结构元素相似，它包含开始标签和结束标签（见代码清单 1-3）。

代码清单 1-3：使用 <head> 元素

```
<!DOCTYPE html>
<html lang="en">
<head>

</head>

</html>
```

这个标签本身与之前的版本并没有任何区别，其作用也保持不变。在 <head> 标签内，可以定义网页的标题，声明字符编码方式，提供一些关于文档的信息，以及加入一些外部文件，包括用于显示页面的样式表、脚本或者图像。

除了标题与图像之外，文档中位于 <head> 标记之间的其他信息通常都是不可见的。

1.2.4 <body>

接下来的部分是 HTML 文档的主要组成部分，即正文。正文是文档的可见部分，位于 <body> 标签之间（见代码清单 1-4）。这个标签与之前版本相同：

代码清单 1-4：使用 <body> 元素

```
<!DOCTYPE html>
<html lang="en">
<head>

</head>
<body>

</body>
</html>
```

基础知识回顾：到目前为止，代码虽然不多，但是其拥有非常复杂的结构。这是因为 HTML 代码并不是连续指令集。HTML 是标记语言，由一组标签或元素构成，它们通常成对出现，并且可以嵌套（完全包含在其他元素中）。在代码清单 1-4 的第一行代码中，首先是单个文档定义标签，紧接着是开始标签 <html lang="en">。这个标签与末尾的结束标签 </html> 表示 HTML 的开始与结束。位于 <html> 标签之间的是另外两个重要的基本结构：<head> 和 <body>。这两个标签也是成对出现的。本章后面的内容将介绍更多可能出现在 <head> 和 <body> 之间的标签。这种结构是一种类树型结构，其中 <html> 标签是树的根节点。

1.2.5 <meta>

现在创建文档头。文档头有一些变化和创新，其中一点发生在定义文档字符编码的标签中。即 meta 标签（见代码清单 1-5），它规定文字在屏幕上的显示方式。

代码清单 1-5：使用 <meta> 元素

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">

</head>
<body>

</body>
</html>
```

与其他情况相似，在 HTML5 中，这个标签的主要创新就是简单化。新的字符编码标签 meta 更短并且更简洁。当然，编码方式 utf-8 是可以根据需要修改的，而这里还可能有 description 或 keywords 等其他的 meta 标签，如代码清单 1-6 所示：

代码清单 1-6：添加更多的 <meta> 元素

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="description" content="This is an HTML5 example">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">

</head>
<body>

</body>
</html>
```

基础知识回顾：文档可能会用到几种用于声明一般信息的 meta 标签，但是这些信息不会显示在浏览器的可见窗口中，它仅供搜索引擎和设备使用，它们可能需要预览或汇总文档的相关数据。如前所述，除了标题或图像，用户是无法看到 <head> 标签之间的大多数信息的。在代码清单 1-6 中，<meta> 标签的 name 属性规定了元数据类型，而 content 则声明它的值，但是这些值都不会显示在屏幕上。想要了解更多关于 <meta> 标签的信息，请访问我们的网站，进入本章的链接。

在 HTML5 中，自动闭合标签不一定需要在末尾添加斜线，但是考虑到兼容性，推荐添加自动闭合符号。上面的代码可以修改为代码清单 1-7：