

TURING

图灵程序设计丛书

Web开发系列

Apress®

Pro HTML5 Programming

Powerful APIs for Richer Internet Application Development

HTML5

高级程序设计

[荷] Peter Lubbers

[美] Brian Albers 著

Frank Salim

李杰 柳靖 刘淼 译

- 引领下一代Web开发潮流
- 全面涵盖开创性HTML5 API
- 真实示例，轻松上手



人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (C I P) 数据

HTML5高级程序设计 / (荷) 柳伯斯 (Lubbers, P.),
(美) 阿伯斯 (Albers, B.), (美) 萨利姆 (Salim, F.)
著; 李杰, 柳靖, 刘淼译. — 北京: 人民邮电出版社,
2011. 1

(图灵程序设计丛书)

书名原文: Pro HTML5 Programming: Powerful
APIs for Richer Internet Application Development
ISBN 978-7-115-24487-1

I. ①H… II. ①柳… ②阿… ③萨… ④李… ⑤柳…
⑥刘… III. ①超文本标记语言, HTML—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2010)第236844号

内 容 提 要

本书首先介绍了 HTML5 的历史背景、新的语义标签及与以往 HTML 版本相比的根本变化, 同时揭示了 HTML5 背后的设计原理。从第 2 章起, 分别围绕构建令人神往的富 Web 应用, 逐一讨论了 HTML5 的 Canvas、Geolocation、Communication、WebSocket、Forms、Web Workers、Storage 等 API 的使用, 辅以直观明了的客户端和服务端示例代码, 让开发人员能够迅速理解和掌握新一代 Web 标准所涵盖的核心技术。本书最后探索了离线 Web 应用并展望了 HTML5 未来的发展前景。

本书面向有一定经验的 Web 应用开发人员, 对 HTML5 及未来 Web 应用技术发展抱有浓厚兴趣的读者也可以学习参考。

图灵程序设计丛书 HTML5高级程序设计

◆ 著 [荷] Peter Lubbers

[美] Frank Salim

译 李杰 柳靖 刘淼

责任编辑 朱 巍

执行编辑 毛倩倩

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京艺辉印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 14.75

字数: 357千字 2011年1月第1版

印数: 1-4 000册 2011年1月北京第1次印刷

著作权合同登记号 图字: 01-2010-6674号

ISBN 978-7-115-24487-1

定价: 45.00元

读者服务热线: (010)51095186 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition, entitled *Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development* by Peter Lubbers, Brian Albers, Frank Salim, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2010 by Peter Lubbers, Brian Albers, Frank Salim. Simplified Chinese-language edition copyright © 2011 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Apress L.P. 授权人民邮电出版社独家出版。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。

献给我漂亮的妻子 Vittoria,
也献给我的儿子——Sean 和 Rocky。我为你们感到无比骄傲!

——Peter Lubbers

献给 John，是你让这一切都有了意义。

——Brian Albers

献给热爱读书的朋友们。

——Frank Salim

译者序

尽管 HTML5 尚未正式发布，但人们对它的关注却日益升温。各大技术资讯网站充斥着 HTML5 的新闻，似乎不谈 HTML5 就已经落伍了。如果把这一切当成一种炒作的话，那可就大错特错了！Google、Opera、Apple、Mozilla 等浏览器厂商频繁更新旗下浏览器产品，添加对 HTML5 新特性的支持。连一向“偏执”的微软也积极推出了 IE9 beta 版开放下载。得益于各家浏览器的实践性反馈，HTML5 进入了快速良性发展的循环，真可谓博观实践经验而约取，厚积反馈优化而薄发。

说起本书，不能不提本书的三位作者 Peter、Brian 和 Frank，他们都是 Web 领域从业 10 年以上的资深技术专家，其中 Peter 还曾经是一名荷兰皇家特种部队的战士！他们没有照搬 HTML5 规范，流水账式地介绍 API，而是挑选最实用的 HTML5 技术，结合示例，引领读者循序渐进、由浅入深地畅游迷人的 HTML5 殿堂。

本书由山西移动的李杰（猴子）、FreeWheel 的柳靖（松鼠）和百度的刘森（小猫）协作翻译，并由柳靖负责最终统稿、润色及审校。翻译过程亦是学习实践与总结提高的过程，我们深感 HTML5 的博大精深，并为作者深厚的功力所折服。

感谢我们的家人和同伴，感谢你们一如既往的支持，包容我们在生活中的种种任性。

感谢人民邮电出版社图灵公司的杨海玲、李松峰、杨爽和毛倩倩老师，没有你们的信任与指导，我们不可能在这么短的时间内完成本书的翻译。

感谢中国移动通信集团山西有限公司 IT 规划建设中心的大力支持。赵文婧、李彬、王雪、王仲妮、高莹、赵华杰、李琦、阎丽娜、杨琳等同学为本书的出版倾注了大量的智慧和汗水。

感谢 FreeWheel 北京研发中心为本书的付出，一切技术难题在这里都会迎刃而解，而良好的技术氛围和宽松的工作环境是译文完稿最可靠的技术保证。

感谢百度商务搜索测试部，你们对新技术的跟踪能力着实让人吃惊。

由于时间仓促，而且译者水平有限，译文中难免会有错漏，望广大同仁批评指正。

仅以此译作献给在前端海洋中畅游的广大开发人员！

猴子 松鼠 小猫

2010 年 10 月 27 日于北京

序

2004 年 6 月，语义网（Semantic Web）联盟、各大浏览器厂商和 W3C 的代表们在美国加州圣何塞齐聚一堂，共同讨论如何发展 Web 标准以适应 Web 应用的快速增长。会议第二天的最后举行了一次投票，表决 W3C 是否应该扩展 HTML 和 DOM，以满足 Web 应用中的新需求。几分钟后，匿名投票结果出来了，结果匪夷所思：“8 票同意，14 票反对。”

分歧导致了组织的分裂：两天后，主流浏览器厂商成立了 WHATWG，目的是解决 Web 应用带来的新问题。同时，W3C 继续向前推进 XHTML 2 规范，但仅过 5 年就放弃了，转而与 WHATWG 结盟集中发展 HTML5。

现在，6 年过去了，我们看到了曾经充满激情的 HTML5 设计所带来的现实意义。这些功能不仅被当作事实标准使用了多年，还为下一代 Web 应用打下了坚实的基础。使用这些功能意味着带给用户更具吸引力和交互性的 Web 体验，并且通常情况下，代码量会更小。

本书中，你会发现一条经过良好设计的学习曲线，以助你快速了解 HTML5 和相关规范的特性。从中你将学习到最实用的 HTML5 支持情况检测、恰当的示例代码以及很多在标准规范中学习不到的知识。精心设计的示例代码，并非只是对各个 API 进行呆板烦琐地介绍，还将带领读者逐步建立实际的 Web 应用。希望本书能为你带来帮助，也希望你会像我一样对下一代 Web 兴奋不已。

Paul Irish

Google 和 jQuery 开发团队成员、Modernizr 的首席开发者

前言

HTML5 是全新的。事实上，它甚至还没有完全成熟。如果你听一些“坏脾气”专家的介绍，他们会告诉你 HTML5 在未来十年甚至更久的时间里都不会完全成熟！

那么，为什么会有人认为现在是时候编写一本讨论 HTML5 编程的书呢？原因很简单。对于希望自己的 Web 应用程序能够卓而不群的人，HTML5 正是众望所归。本书作者致力于研究开发和讲授 HTML5 技术已有两年多，现在可以肯定地说，在实际 Web 应用中新标准的采纳程度正在以令人目眩的速度不断加快。即使在编写本书的过程中，我们都被迫不断更新书中的浏览器支持表格，重新评估哪些技术又具备了使用条件。

面对自己正在使用的浏览器，大多数用户并不真正了解其具备的功能有多强大。当然，他们在浏览器自动更新后可能会发现一些细微的界面改变。但他们可能不知道，新版本的浏览器刚刚引入了可自由绘图的 canvas、实时网络通信以及其他一些潜在的功能升级。

本书的目标是帮助开发者释放 HTML5 的潜力。

本书读者对象

本书针对熟悉 JavaScript 编程且有经验的 Web 应用程序开发者。也就是说，本书将不涉及 Web 开发的基础知识。如果了解 Web 编程的基础知识，目前的资源已经够多了。如果读者遇到了下面的情况，那么本书可以为你提供有用的见解和信息，这些见解和信息可能正是你在努力寻找的。

- 你有时会发现自己在想：“如果我的浏览器可以……”
- 你发现自己通过页面的源代码和开发工具来分析一个令人印象深刻的网站。
- 你喜欢查看最新浏览器的版本发布信息，了解其更新了什么功能。
- 你在寻找优化或简化应用程序的方法。
- 你想针对使用最新浏览器的用户定制网站，以便尽可能提供最佳用户体验。

如果上述任何一项跟你的情况吻合，那么这本书可能就很适合你。

虽然我们在适当情况下特意指出了浏览器支持的局限性，但目的并非要给出一个兼容旧浏览器且可无缝运行的解决方案。经验表明，浏览器更新换代的速度一日千里，如果要获取浏览器兼容解决方案方面的相关信息，本书不是最好的渠道。相反，我们专注于 HTML5 规范及其使用方法。兼容的解决方案可以在因特网上找到，而随着时间的推移，这些解决方案也会渐渐被人遗忘。

本书内容

本书的 11 章内容涵盖了从 HTML5 API 中挑选出来的适用面广、功能强大的 API。在某些情况下,为了更好地演示程序,我们需要用到前面章节已经介绍过的功能。

第 1 章“HTML5 概述”,从 HTML 版本的发展历程说起,介绍了 HTML 规范过去和现在的版本情况,然后介绍了新的高级语义标签,以及一些根本性的改进,同时还分析了 HTML5 背后的设计理念。了解这些对读者是有益的。

第 2 章“Canvas API”和第 3 章“音频和视频”,讨论了新的可视化元素和媒体元素。在这两章中,集中讨论如何在无插件和无服务器交互的情况下优化用户界面。

第 4 章“Geolocation API”介绍的是一个全新的功能。在此之前,它很难通过模拟方式实现,它赋予应用程序确定用户当前位置的能力,并可以用来定制用户体验。这里对隐私的保护也很重要,所以我们会介绍隐私保护的相关内容。

第 5 章“Communication API”和第 6 章“WebSockets API”展示了 HTML5 提供的日益强大的通信能力。有了这两个 API,Web 应用不仅可以同其他网站进行通信,而且还能以最简单的代码和最小的网络开销进行实时数据流的传递。这两章中的技术将有助于开发人员简化目前网络上部署的过于复杂的架构。

第 7 章“Forms API”,参照这章介绍的内容,开发人员通过细小的调整即可增加桌面 Web 应用程序和移动 Web 应用程序的可用性。利用这一章介绍的其他新特性,则可以检测大多数常见场景中的页面输入错误。

第 8 章“Web Workers API”、第 9 章“Web Storage API”和第 10 章“构建离线 Web 应用”,解决了应用程序的内部数据管道问题。在这三章中,开发人员会学到如何优化现有系统来获得更好的性能和更好的数据管理功能。

最后,第 11 章“HTML5 未来展望”讨论了一些可能会在 HTML5 中陆续出现的功能,这些功能可能让大家垂涎已久了。

示例代码和配套网站

本书中的示例代码都可从 Apress 网站的 Source Code 部分找到。访问 www.apress.com,单击 Source Code,然后查找这本书的标题即可。读者可以从本书主页上下载源代码。此外,我们还建立了一个配套网站地址为 www.prohtml5.com,读者也可以在此下载示例代码和一些实用附加功能源码。^①

联系作者

感谢购买此书,我们希望你喜欢它,并把它当做一个宝贵的资源。尽管已经尽了最大努力,但我们知道一时疏忽就可能引发错误,在此,我们为可能的疏忽表示歉意。我们欢迎你对此书的内容和源代码发表意见和评论。你可以发送邮件至 prohtml5@gmail.com 来与我们取得联系。

^① 相关代码也可在图灵网站下载。——编者注

致 谢

我想感谢我的妻子 Vittoria，感谢她为我付出的爱和忍耐。感谢我的天才儿子 Sean 和 Rocky。孩子们，为你们的理想努力吧！

感谢我的父母 Herman 和 Elisabeth，感谢我的妹妹 Alice 和我的兄弟 David，感谢他们一直这么信任我。感谢我已故的祖母 Gebbechien，她在纳粹占领荷兰时的勇敢行为给我们全家上了伟大的一课。

对于我的合著者——永不疲倦的 Brian 和代码超人 Frank，能同你们一起合作是我的荣幸。

还要感谢 Apress 出版社 Clay 的支持。最后，感谢 Kaazing 的 Jonas 和 John，是他们促使我们编写完成了一本“真正”的书，我相信如果没有他们，哪怕是“非官方电子书”也肯定只是空想！

——Peter Lubbers

我的父母 Ken 和 Patty Albers，我深爱你们，感谢你们为给我创造这么多机会而付出的牺牲。没有你们的鼓励和灌输给我的价值观，不要说是这次，所有生命中的成就我都将无法达成。你们一直在指引着我人生道路上的每一步。

对 John 致以最深切的感谢。每次加班时间延长到两小时、三小时，甚至更长的时候，John 都显示出了极大的耐性，这深深地震撼并激励了我。

Pitch、Bonnie 和 Penelope，我由衷地承诺晚餐不会再这么晚了。感谢曾经来过的那只猫，你的呼噜声还回响在我耳边。

非常感激 Kaazing 的合作者，让我能与最棒、最聪明的人们一起共事。

特别感谢 Apress 出版社的工作人员，感谢你们认定了 HTML5 书籍出版的契机，同时感谢在我们尝试追随 HTML5 这个迅速发展的事物期间，你们给予的极大耐心。

——Brian Albers

我想感谢我的父母 Mary 和 Sabri，有了他们才有了我的存在，不夸张地说，如果没有他们就没有这本书。

——Frank Salim

目 录

第 1 章 HTML5 概述.....	1	2.1.2 canvas 是什么	23
1.1 HTML5 发展史	1	2.1.3 canvas 坐标	23
1.2 关于 2022 年的那个神话	2	2.1.4 什么情况下不用 canvas	24
1.3 谁在开发 HTML5	3	2.1.5 替代内容	24
1.4 新的认识	3	2.1.6 CSS 和 canvas	24
1.4.1 兼容性和存在即合理	3	2.1.7 浏览器对 HTML5 Canvas 的 支持	25
1.4.2 效率和用户优先	4	2.2 使用 HTML5 Canvas API	25
1.4.3 化繁为简	4	2.2.1 检测浏览器支持情况	25
1.4.4 通用访问	5	2.2.2 在页面中加入 canvas	26
1.5 无插件范式	5	2.2.3 变换	28
1.6 HTML5 的新功能	8	2.2.4 路径	30
1.6.1 新的 DOCTYPE 和字符集	8	2.2.5 描边样式	32
1.6.2 新元素和旧元素	9	2.2.6 填充样式	33
1.6.3 语义化标记	9	2.2.7 填充矩形区域	34
1.6.4 使用 Selectors API 简化选取 操作	15	2.2.8 绘制曲线	35
1.6.5 JavaScript 日志和调试	18	2.2.9 在 canvas 中插入图片	36
1.6.6 window.JSON	19	2.2.10 渐变	37
1.6.7 DOM Level 3	19	2.2.11 背景图	39
1.6.8 Monkeys、Squirrelish 和其他 JavaScript 引擎	19	2.2.12 缩放 canvas 对象	41
1.7 小结	20	2.2.13 Canvas 变换	42
第 2 章 Canvas API	22	2.2.14 Canvas 文本	43
2.1 HTML5 Canvas 概述	22	2.2.15 应用阴影	45
2.1.1 历史	22	2.2.16 像素数据	46
		2.2.17 Canvas 的安全机制	48
		2.3 使用 HTML5 Canvas 创建应用	48

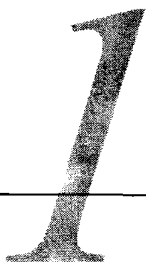
2.4 小结	52	4.5.2 处理 Geolocation 数据	85
第3章 音频和视频	53	4.5.3 最终代码	88
3.1 HTML5 Audio 和 Video 概述	53	4.6 进阶功能	90
3.1.1 视频容器	53	4.6.1 现在的状态是什么	91
3.1.2 音频和视频编解码器	54	4.6.2 在 Google Map 上显示“我在 这里”	92
3.1.3 HTML5 Audio 和 Video 的限制	55	4.7 小结	93
3.1.4 audio 元素和 video 元素的 浏览器支持情况	56	第5章 Communication API	94
3.2 使用 HTML5 Audio 和 Video API	56	5.1 跨文档消息通信	94
3.2.1 浏览器支持性检测	57	5.1.1 理解源安全	96
3.2.2 理解媒体元素	58	5.1.2 跨文档消息通信的浏览器支持 情况	97
3.2.3 使用 audio 元素	62	5.1.3 使用 postMessage API	97
3.2.4 使用 video 元素	63	5.1.4 使用 postMessage API 创建 应用	98
3.2.5 进阶功能	68	5.2 XMLHttpRequest Level 2	103
3.3 小结	70	5.2.1 跨源 XMLHttpRequest	104
第4章 Geolocation API	72	5.2.2 进度事件	105
4.1 位置信息	72	5.2.3 XMLHttpRequest Level 2 的 浏览器支持情况	106
4.1.1 纬度和经度坐标	73	5.2.4 使用 XMLHttpRequest API	106
4.1.2 位置信息从何而来	73	5.2.5 创建 XMLHttpRequest 应用	107
4.1.3 IP 地址地理定位数据	74	5.3 进阶功能	111
4.1.4 GPS 地理定位数据	74	5.3.1 结构化的数据	111
4.1.5 Wi-Fi 地理定位数据	74	5.3.2 Framebusting	111
4.1.6 手机地理定位数据	75	5.4 小结	112
4.1.7 用户自定义的地理定位数据	75	第6章 WebSockets API	113
4.2 HTML5 Geolocation 的浏览器支持 情况	75	6.1 HTML5 WebSockets 概述	113
4.3 隐私	76	6.1.1 实时和 HTTP	113
4.3.1 触发隐私保护机制	77	6.1.2 解读 HTML5 WebSockets	115
4.3.2 处理位置信息	78	6.2 HTML5 WebSockets 的浏览器支持 情况	120
4.4 使用 HTML5 Geolocation API	78	6.3 编写简单的 Echo WebSocket 服务器	120
4.4.1 浏览器支持性检查	78	6.4 使用 HTML5 WebSockets API	126
4.4.2 位置请求	79	6.4.1 浏览器支持情况检测	126
4.5 使用 HTML5 Geolocation 构建实时 应用	83		
4.5.1 编写 HTML 显示代码	84		

6.4.2 API 的基本用法	127	8.3.5 示例代码	163
6.5 创建 HTML5 WebSockets 应用程序	130	8.4 构建 HTML5 Web Workers 应用	164
6.5.1 编写 HTML 文件	131	8.4.1 编写 blur.js 辅助脚本	165
6.5.2 添加 WebSocket 代码	133	8.4.2 编写 blur.html 应用页面	166
6.5.3 添加 Geolocation 代码	133	8.4.3 编写 blurWorker.js	167
6.5.4 合并所有内容	134	8.4.4 与 Web Worker 通信	168
6.5.5 最终代码	136	8.4.5 运行程序	169
6.6 小结	138	8.4.6 示例代码	170
第 7 章 Forms API	139	8.5 小结	174
7.1 HTML5 Forms 概述	139	第 9 章 Web Storage API	175
7.1.1 HTML Forms 与 XForms	139	9.1 HTML5 Web Storage 概述	175
7.1.2 功能性表单	140	9.2 HTML5 Web Storage 的浏览器支持 情况	176
7.1.3 HTML5 Forms 的浏览器支持 情况	140	9.3 使用 HTML5 Web Storage API	176
7.1.4 输入型控件目录	141	9.3.1 检查浏览器的支持性	176
7.2 使用 HTML5 Forms API	144	9.3.2 设置和获取数据	178
7.2.1 新的表单特性和函数	144	9.3.3 封堵数据泄漏	179
7.2.2 表单验证	147	9.3.4 localStorage 与 session- Storage	180
7.2.3 验证反馈	151	9.3.5 Web Storage API 的其他特性 和函数	180
7.3 构建 HTML5 Forms 应用	152	9.3.6 更新 Web Storage 后的通信	182
7.4 小结	157	9.3.7 探索 Web Storage	183
第 8 章 Web Workers API	158	9.4 构建 HTML5 Web Storage 应用	184
8.1 HTML5 Web Workers 的浏览器支持 情况	158	9.5 浏览器数据库存储展望	195
8.2 使用 HTML5 Web Workers API	159	9.6 进阶功能	197
8.2.1 浏览器支持性检查	159	9.6.1 JSON 对象的存储	198
8.2.2 创建 HTML5 Web Workers	160	9.6.2 共享窗口	198
8.2.3 多个 JavaScript 文件的加载与 执行	160	9.7 小结	200
8.2.4 与 HTML5 Web Workers 通信	160	第 10 章 构建离线 Web 应用	201
8.3 编写主页	161	10.1 HTML5 离线 Web 应用概述	201
8.3.1 处理错误	161	10.2 使用 HTML5 离线 Web 应用 API	203
8.3.2 HTML5 Web Workers	162	10.2.1 检查浏览器的支持情况	203
8.3.3 HTML5 Web Workers 的嵌套 使用	162	10.2.2 搭建简单的离线应用程序	203
8.3.4 使用定时器	163	10.2.3 支持离线行为	204

10.2.4	manifest 文件	204	10.3.8	添加离线事件处理程序	212
10.2.5	applicationCache API	205	10.4	小结	213
10.3	使用 HTML5 离线 Web 应用构建应用	206	第 11 章	HTML5 未来展望	214
10.3.1	创建记录资源的 manifest 文件	208	11.1	HTML5 的浏览器支持情况	214
10.3.2	创建构成界面的 HTML 和 CSS	208	11.2	HTML 未来的发展	214
10.3.3	创建离线 JavaScript	209	11.2.1	WebGL	215
10.3.4	检查 applicationCache 的支持情况	210	11.2.2	设备	217
10.3.5	为 Update 按钮添加处理函数	211	11.2.3	音频数据 API	217
10.3.6	添加 Geolocation 跟踪代码	211	11.2.4	视频元素改进	218
10.3.7	添加 Storage 功能代码	212	11.2.5	触摸屏设备事件	218
			11.2.6	P2P 网络	220
			11.2.7	最终方向	221
			11.3	小结	221

第 1 章

HTML5 概述



这是一本关于 HTML5 编程的书。不过在学习之前，有必要先了解一下背景知识，什么是 HTML5？它经历了怎样的发展历程？HTML4 和 HTML5 有什么区别？

本章中，我们会集中讨论大家关注的一些实际问题。为什么是 HTML5？为什么它能掀起风潮？是什么设计理念使得 HTML5 真正具有革命性的进步？HTML5 如何在大幅改动的同时保持高度兼容？无插件范式意味着什么？HTML5 包含什么，不包含什么？HTML5 新增加了哪些特性，为什么能揭开整个 Web 开发新时代的序幕？下面我们一起来了解一下。

1.1 HTML5 发展史

HTML 的历史可以追溯到很久以前。1993 年 HTML 首次以因特网草案的形式发布。20 世纪 90 年代的人见证了 HTML 的大幅发展，从 2.0 版，到 3.2 版和 4.0 版（一年出了两个版本！），再到 1999 年的 4.01 版。随着 HTML 的发展，W3C（万维网联盟）掌握了对 HTML 规范的控制权。

然而，在快速发布了这四个版本之后，业界普遍认为 HTML 已经到了穷途末路，对 Web 标准的焦点也开始转移到了 XML 和 XHTML 上，HTML 被放在了次要位置。不过在此期间，HTML 体现了顽强的生命力，主要的网站内容还是基于 HTML 的。为能支持新的 Web 应用，同时克服现有的缺点，HTML 迫切需要添加新功能，制定新规范。

致力于将 Web 平台提升到一个新的高度，一小组人在 2004 年成立了 WHATWG（Web Hypertext Application Technology Working Group，Web 超文本应用技术工作组）。他们创立了 HTML5 规范，同时开始专门针对 Web 应用开发新功能——这被 WHATWG 认为是 HTML 中最薄弱的环节。Web 2.0 这个新词也就正是在那个时候被发明的。Web 2.0 实至名归，开创了 Web 的第二个时代。旧的静态网站逐渐让位于需要更多特性的动态网站和社交网站——这其中的新功能真的是数不胜数。

2006 年，W3C 又重新介入 HTML，并于 2008 年发布了 HTML5 的工作草案。2009 年，XHTML 2 工作组停止工作。又过了一年，也就到了现在。因为 HTML5 能解决非常实际的问题（随后可以看到），所以在规范还未定稿的情况下，各大浏览器厂家就已经按耐不住了，开始对旗下产品进行升级以支持 HTML5 的新功能。这样，得益于浏览器的实验性反馈，HTML5 规范也得到了持续地完善，HTML5 以这种方式迅速融入到了对 Web 平台的实质性改进中。

“大家好，我是 Brian^①，HTML 的铁杆老粉丝。

1995 年，我创建了第一个属于自己的个人主页。那时候的‘主页’就是用来介绍自己的。上面的照片通常不清晰，代码中用了很多<blink>标签，页面上会告诉大家我住在哪儿、读过什么书、正在做什么跟计算机相关的工作。我和我的那些所谓‘万维网开发者’不是在大学里读书就是在大学里工作。

那时候的 HTML 非常初级，没有任何工具可用。Web 应用几乎没有，顶多有少量的文本处理脚本。页面代码都是用大家各自喜欢的文本编辑器写出来的。页面的更新频率基本上是数周或者数月。

不知不觉，我们已经走过了漫长的 15 个年头。

今天，用户对其在线资料一天更新很多次已经是很平常的事了。当然，如果没有在线工具持续稳定的更新换代，也不会有今天这样的交互方式。

提醒各位读者，大家在看这本书的时候心里要明白，我们的示例虽然现在看起来非常简单，但潜力是巨大的。就像 20 世纪 90 年代中期那些率先使用标签的人一样，他们又怎么会知道在十年以后，很多人都已经在线编辑和储存照片了；而我们要有一种前瞻性。

我们希望书中示例的基本思路能够激发读者无穷的创意，从而为 Web 的下个十年奠定新的基础。”

——Brian

1.2 关于 2022 年的那个神话

今天，我们看到的 HTML5 规范已经以工作草案的形式发布了——还不是最终版。那什么时候 HTML5 规范才能尘埃落定呢？现在就来了解一下几个关键时间点。第一个时间点是 2012 年，目标是发布候选推荐版。第二个时间点是 2022 年，目标是发布计划推荐版。哦！那等着吧，还早着呢！可能大家会这么想，然后就把书合上，扔到一边，等十年后再说。那就大错特错了，在明白这两个时间点的真正意义之前，可别急着下结论。

第一个，也就是最近的 2012 年，可以说是最重要的时间点，因为这个时间点一到就意味着 HTML5 规范编写完成了。想象一下，这并不久远，也就两年后的事情。计划推荐版（普遍认为距今还有点远）的重要性在于届时将会有两个对 HTML5 的互通实现，意味着将有两个浏览器会完全支持整个 HTML5 规范的所有功能——这个远大的目标让 2022 年这个时间点看起来又似乎太近了。毕竟，现在连 HTML 4 都还没有实现这个目标呢^②。

关键是现在浏览器厂家已经着手支持 HTML5 中很多优秀的新功能了。只要用户有需求，现

^① Brian，本书作者之一。——译者注

^② HTML 4 最早于 1997 年成为 W3C 推荐标准，到现在 10 多年早已经过去了，仍然不存在两个完全支持这一规范的浏览器。——编者注

在就可以利用这些新功能进行 Web 应用的开发。虽然一些细节方面的改造还会持续进行, 相应的 Web 应用可能需要改动, 不过, 相对于使用 HTML5 为用户带来的体验来讲, 这点付出不算什么。当然, 如果用户的浏览器是 IE6 的话, 很多新功能是不支持的, 需要模拟——不过这也不能成为抛弃 HTML5 的理由, 毕竟这些用户最终都会升级浏览器版本, 很多可能会直接选用 IE9, 而且微软承诺在 IE9 中持续增加对 HTML5 的支持。实际上, 通过使用新的浏览器和改进的模拟技术意味着用户现在和不久的将来便可以使用很多 HTML5 功能了。

1.3 谁在开发 HTML5

我们都知道开发 HTML5 需要成立相应的组织, 并且肯定需要有人来负责。这正是下面这三个组织的工作。

- WHATWG: 由来自 Apple、Mozilla、Google、Opera 等浏览器厂商的人组成, 成立于 2004 年。WHATWG 开发 HTML 和 Web 应用 API, 同时为各浏览器厂商以及其他有意向的组织提供开放式合作。
- W3C: W3C 下辖的 HTML 工作组目前负责发布 HTML5 规范。
- IETF (Internet Engineering Task Force, 因特网工程任务组): 这个任务组下辖 HTTP 等负责 Internet 协议的团队。HTML5 定义的一种新 API (WebSocket API) 依赖于新的 WebSocket 协议, IETF 工作组正在开发这个协议。

1.4 新的认识

HTML5 是基于各种各样的理念 (在 WHATWG 规范中有详述) 进行设计的, 这些设计理念体现了对可能性和可行性的新认识。

- 兼容性
- 实用性
- 互通性
- 通用访问性

1.4.1 兼容性和存在即合理

别担心, HTML5 并不是颠覆性的革新。相反, 实际上 HTML5 的一个核心理念就是保持一切新特性平滑过渡。一旦浏览器不支持 HTML5 的某项功能, 针对功能的备选行为就会悄悄进行。再说, 互联网上有些 HTML 文档已经存在 20 多年了, 因此, 支持所有现存 HTML 文档是非常重要的。

HTML5 的研究者们还花费了大量的精力来研究通用行为。比如, Google 分析了上百万的页面, 从中分析出了 DIV 标签的通用 ID 名称, 并且发现其重复量很大。例如, 很多开发人员使用 `DIV id="header"` 来标记页眉区域。HTML5 不就是要解决实际问题吗? 那何不直接添加一个 `<header>` 标签呢?

尽管 HTML5 标准的一些特性非常具有革命性, 但是 HTML5 旨在进化而非革命。毕竟没有从头再来的必要。(就算有必要的话, 也不应该是 HTML5, 起码也要发明一个更好的!)

1.4.2 效率和用户优先

HTML5 规范是基于用户优先准则编写的，其宗旨是“用户即上帝”，这意味着在遇到无法解决的冲突时，规范会把用户放到第一位，其次是页面作者，再次是实现者（或浏览器），接着是规范制定者（W3C/WHATWG），最后才考虑理论的纯粹性。因此，HTML5 的绝大部分是实用的，只是有些情况下还不够完美。

看看这个示例，下面的几种代码写法在 HTML5 中都能被识别。

```
id="prohtml5"  
id=prohtml5  
ID="prohtml5"
```

当然，肯定会有人反对这种不严格的语法，我们不去辩论对错，只去关心一个底线，那就是最终用户其实并不在乎代码怎么写。当然，我们并不提倡入门者一开始写代码就这么不严谨，毕竟归根结底，受害者还是最终用户，因为一旦由于开发人员的原因造成页面错误导致不能正常显示，那么被折磨的肯定是最终用户。

HTML5 也衍生出了 XHTML5（可通过 XML 工具生成有效的 HTML5 代码）。HTML 和 XHTML 两种版本的代码经过序列化应该可以生成近乎一样的 DOM 树。显然 XHTML 的验证规则严格得多，刚才示例中后两行代码是无法通过验证的。

1. 安全机制的设计

为保证 HTML5 足够安全，HTML5 在设计时就做了大量的工作。规范中的各个部分都有专门针对安全的章节，并且安全是被优先考虑的。HTML5 引入了一种新的基于来源的安全模型，该模型不仅易用，而且对各种不同的 API 都通用。这个安全模型可以让我们做一些以前做不到的事情，不需要借助于任何所谓聪明、有创意却不安全的 hack 就能跨域进行安全对话。在这方面，我们肯定不会怀念过去的“好”时光了。

2. 表现和内容分离

在清晰分离表现和内容方面，HTML5 迈出了巨大的步伐。HTML5 在所有可能的地方都努力进行了分离，也包括 CSS。实际上，HTML5 规范已经不支持老版本 HTML 的大部分表现功能了，但得益于先前提到的 HTML5 在兼容性方面的设计理念，那些功能仍然能用。表现和内容分离的概念也不是全新的，在 HTML 4 Transitional 和 XHTML 1.1 中就已经开始用了。Web 设计者把这个概念当做最佳实践使用了很久，不过现在清晰地分开表现和内容显得更为重要，否则会有如下弊端：

- 可访问性差；
- 不必要的复杂度（所有样式代码都放在页面中，代码可读性很差）；
- 文件变大（样式内容越多，文件越大），带来的后果就是页面载入变慢。

1.4.3 化繁为简

HTML5 要的就是简单、避免不必要的复杂性。HTML5 的口号是“简单至上，尽可能简化”。因此，HTML5 做了以下这些改进：