



# HTML5 应用开发实践指南

O'REILLY®



机械工业出版社  
China Machine Press

Zachary Kessin 著

陈升想 汪奋进 译

013024761

TP312HT  
91

---

# HTML 5应用开发实践指南

*Zachary Kessin* 著

陈升想 汪奋进 译



**O'REILLY®**

*Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo*

O'Reilly Media, Inc. 授权机械工业出版社出版

机械工业出版社



北航

C1632229

TP312HT  
91

137380810

## 图书在版编目 (CIP) 数据

HTML 5应用开发实践指南/ (美) 凯西恩 (Kessin, Z.) 著; 陈升想, 汪奋进译.  
—北京: 机械工业出版社, 2013.3

(O'Reilly精品图书系列)

书名原文: Programming HTML 5 Applications

ISBN 978-7-111-41451-3

I. H… II. ①凯… ②陈… ③汪… 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027166号

北京市版权局著作权合同登记

图字: 01-2012-6641号

©2012 Zachary Kessin

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Machine Press, 2013. Authorized translation of the English edition, 2012 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc. 出版2012。

简体中文版由机械工业出版社出版 2013。英文原版的翻译得到O'Reilly Media, Inc.的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc.的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式复制。

封底无防伪标均为盗版

本书法律顾问

北京市展达律师事务所

书 名/ HTML 5应用开发实践指南

书 号/ ISBN 978-7-111-41451-3

责任编辑/ 秦健

封面设计/ Karen Montgomery, 张健

出版发行/ 机械工业出版社

地 址/ 北京市西城区百万庄大街22号 (邮政编码 100037)

印 刷/ 三河市杨庄长鸣印刷装订厂

开 本/ 178毫米×233毫米 16开本 8印张

版 次/ 2013年4月第1版 2013年4月第1次印刷

定 价/ 49.00元 (册)

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: hzsj@hzbook.com

# O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

## 业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

# 目录

|                          |    |
|--------------------------|----|
| 前言 .....                 | 1  |
| 第1章 Web应用平台 .....        | 5  |
| 为Web应用增加力量 .....         | 5  |
| 开发网络应用程序 .....           | 6  |
| JavaScript的胜利 .....      | 7  |
| 第2章 JavaScript的力量 .....  | 10 |
| 非阻塞I/O和回调 .....          | 10 |
| 强大的Lambda函数 .....        | 12 |
| 闭包 .....                 | 13 |
| 函数式编程 .....              | 16 |
| 原型及如何扩展对象 .....          | 18 |
| 提取一个子串 .....             | 19 |
| 用原型扩展函数 .....            | 20 |
| 柯里化和对象参数 .....           | 23 |
| 数组迭代操作 .....             | 23 |
| 你也可以扩展对象 .....           | 27 |
| 第3章 测试JavaScript应用 ..... | 29 |
| QUnit .....              | 32 |
| Selenium .....           | 34 |

|                                     |           |
|-------------------------------------|-----------|
| <b>第4章 本地存储 .....</b>               | <b>49</b> |
| localStorage和sessionStorage对象 ..... | 50        |
| jQuery 插件 .....                     | 58        |
| <b>第5章 IndexedDB.....</b>           | <b>60</b> |
| 添加、更新记录 .....                       | 63        |
| 添加索引 .....                          | 64        |
| 检索数据 .....                          | 64        |
| 删除数据 .....                          | 65        |
| <b>第6章 文件 .....</b>                 | <b>66</b> |
| 二进制大对象 .....                        | 66        |
| 操作文件 .....                          | 68        |
| 上传文件 .....                          | 69        |
| 拖曳 .....                            | 70        |
| 全部整合到一起 .....                       | 70        |
| Filesystem文件系统 .....                | 72        |
| <b>第7章 离线处理 .....</b>               | <b>73</b> |
| 清单文件简介 .....                        | 73        |
| 事件 .....                            | 77        |
| 调试清单文件 .....                        | 78        |
| <b>第8章 把工作分割成Web Worker .....</b>   | <b>81</b> |
| Web Worker用例 .....                  | 83        |
| 使用Web Worker.....                   | 84        |
| Web Worker碎形示例 .....                | 86        |
| 测试和调试Web Worker.....                | 92        |
| 多线程复用模式 .....                       | 92        |
| Web Worker库.....                    | 96        |
| <b>第9章 Web Socket .....</b>         | <b>97</b> |
| Web Socket接口 .....                  | 99        |

|                                    |            |
|------------------------------------|------------|
| 建立Web Socket连接 .....               | 99         |
| Web Socket示例 .....                 | 100        |
| Web Socket协议 .....                 | 101        |
| <b>第10章 新标记 .....</b>              | <b>105</b> |
| 应用标记 .....                         | 105        |
| 通过WAI-ARIA无障碍访问 .....              | 106        |
| microdata .....                    | 107        |
| 新的表单类型 .....                       | 108        |
| audio和video .....                  | 109        |
| Canvas和SVG .....                   | 109        |
| 地理位置 .....                         | 110        |
| 新的CSS .....                        | 110        |
| <b>附录A 需要了解的JavaScript工具 .....</b> | <b>111</b> |

---

# 前言

这本书反映了Web技术的革命。之前，人们普遍认为Web无须编程，只要把脚本硬塞入网页之中就行了。现在，HTML和JavaScript在实现良好用户体验的过程中产生了重要作用。通过阅读本书，你将会掌握Web发展进程中最前沿的技术。

## 本书主要内容

本书覆盖的内容如下：

### 第1章 Web应用平台

介绍在新的HTML 5平台编程的原因，以及新平台为JavaScript编程人员提供的便利。

### 第2章 JavaScript的力量

介绍JavaScript中很强大但是你可能并不知道的功能，以及为什么你需要使用这些功能去探索本书涵盖的HTML 5的新特性以及相关库。

### 第3章 测试JavaScript应用

展示由JavaScript和浏览器提供的特定环境中测试程序的方法。

### 第4章 本地存储

描述使用localStorage和sessionStorage对象缓存在浏览器中的简单数据。

### 第5章 IndexedDB

展示支持本地存储的更强大的NoSQL数据库。

### 第6章 文件

描述从用户系统中读取和上传文件的方法。

## 第7章 离线处理

描述让用户在设备不能联网的情况下使用你开发的应用程序所要做的操作。

## 第8章 把工作分割成Web Worker

展示HTML 5和JavaScript的多线程能力。

## 第9章 Web Socket

演示如何通过使用Web Socket提高浏览器端和服务端的数据传输效率。

## 第10章 新标记

总结HTML 5引入的对Web开发人员最为有用的新标记。

## 附录A 需要了解的JavaScript工具

描述本书中用到的以及其他可以让编程更快更准确的工具。

# 排版约定

本书采用了以下排版约定：

### 斜体 (*Italic*)

表示新术语、网址、电子邮件地址、文件名和文件扩展名。

### 等宽字体 (`Constant width`)

用于表示代码清单以及正文中引用的代码元素，如变量或函数名称、声明和关键词。

### 等宽斜体 (*`Constant width italic`*)

表示需要由用户提供或由上下文决定的值来替换的文本。

---

**注意：** 表示提示、建议或者一般性注释。

---

---

**警告：** 表示警告或注意事项。

---

# 代码示例的使用

本书旨在帮助你完成你的工作。一般来说，可以在程序和文档中使用本书的代码。如果你复制了代码的关键部分，那么你需要联系我们获得许可。例如，利用本书的几段代码编写程序是不需要许可的。售卖或出版O'Reilly书中示例的CD-ROM需要我们的许可。引用本书回答问题以及引用示例代码不需要我们的许可。将本书的大量示例代码用于你的产品文档中需要许可。

如果你在参考文献中提到我们，我们会非常感激，但并不强求。参考文献通常包括标题、作者、出版社和ISBN。例如：“*Programming HTML 5 Applications* by Zachary Kessin (O’Reilly). Copyright 2012 Zachary Kessin, 978-1-4493-39908-5”。

如果你认为对代码示例的使用已经超出以上的许可范围，我们很欢迎你通过 [permissions@oreilly.com](mailto:permissions@oreilly.com) 联系我们。

## 联系我们

有关本书的任何建议和疑问，可以通过下列方式与我们取得联系：

美国：

O’Reilly Media, Inc.  
1005 Gravenstein Highway North  
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街2号成铭大厦C座807室（100035）  
奥莱利技术咨询（北京）有限公司

我们会在本书的网页中列出勘误表、示例和其他信息。可以访问：

<http://shop.oreilly.com/product/0636920015116.do>

技术问题或评论本书，请发送邮件至：

[bookquestions@oreilly.com](mailto:bookquestions@oreilly.com)

获取有关本书的更多信息、会议、资源中心和O’Reilly网站的相关信息，请查看我们的网站：

<http://www.oreilly.com>

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

在Facebook上联系我们：<http://facebook.com/oreilly>

在Twitter上联系我们：<http://twitter.com/oreillymedia>

在Youtube上联系我们：<http://youtube.com/oreillymedia>

## 致谢

每本书都是团队贡献的结果，如果没有一个强大的团队帮助我，我不可能完成这本书。首先，我必须感谢Simon St. Laurent给我写这本书的机会，并且在我创作的过程中不断地支持我。同样，我必须感谢Andy Oram，是他的编辑才能让这本书更优秀。还有，谢谢我的技术顾问Shelley Powers和Doinysios Synodinos，你们的反馈起了很大作用。

我还必须感谢Israeli开发者社区，比如我在Mytopia的前同事一年多来一直在项目过程中支持我，还有在Sayeret Lambda的伙伴们，现在Sayeret Lambda已经成为Tel Aviv交流编程心得的地方。

最后，谢谢我的妻子Devora一直以来对我的支持。没有她我不可能完成这本书。

# Web应用平台

HTML 5可以将Web打造成创建真正应用程序的一流环境。HTML 5提供了对浏览器API的一系列关键扩展，以此加强了JavaScript现有的工具集，这些扩展使开发人员更简便地创建自我完善的应用，而非现在的这种仅仅显示远程服务器进程的界面。

Web最初是一种分享存储在Web服务器上文件的方法，这些文件只是偶尔变化。开发人员很快想出了如何实时生成那些文件，从而向构建应用程序迈出了第一大步。下一大步是向浏览器客户端中添加交互性。随着“浏览器大战”的肆虐和之后突然停止，JavaScript 和文档对象模型（DOM）开始让开发人员能够创建动态HTML。几年后，Ajax将这些技术用于样式中。Ajax增加了一些工具，使页面与服务器以更小的单元进行通信。

HTML 5建立在这20年的发展之上，填补了一些重要的空白。从表面上看，许多HTML 5的变化是增加了对以前需要用插件实现的功能（尤其是多媒体和图像）的支持，实际上，它为JavaScript程序员提供了一些独立（或者至少更松散合作）创建应用程序的工具，有了这些工具他们可以用HTML搭建架构、用CSS制作界面、用JavaScript表达逻辑和行为。

## 为Web应用增加力量

HTML 5提高了Web应用的标准。尽管它仍然需要工作在安全约束条件下，但最终会提供桌面开发人员已经期盼多年的工具：

### 本地存储

引用键—值系统多达5MB的数据。

## 数据库

起初是一个基于SQLite的API，形势似乎已经转向IndexedDB，IndexedDB是一种JavaScript 原生的NoSQL系统。

## 文件

尽管出于安全的原因Web应用依旧不能自由地访问文件系统，但是已经能够使用用户指定的文件，并开始创建文件了。

## 离线操作

当笔记本电脑或手机在“飞行模式”时，Web应用无法与服务器通信。清单（Manifest）文件通过缓存文件到本地帮助开发人员实现离线应用。

## Web Worker

线程和子进程一直是个问题，但是JavaScript根本不提供这些。Web Worker提供了一种方法，将应用进程放到独立空间中，在那里它们可以互不干扰地工作。

## Web Socket

尽管超文本传输协议（HTTP）随着时间的推移有一些更新，但它一直是Web的基础。Web Socket将“请求/响应”的通信方式转化成创建更灵活的通信系统。

当然，还有更多工具，从地理信息、音频和视频、画布图形到各种小的新标记，这些工具为在HTML 5上构建工业级的应用提供了基础。

# 开发网络应用程序

过去，复杂的Web应用程序只不过是一个目录、一个从数据库派生的静态页面或者一个JavaScript生成的计算器，没有人梦想用JavaScript做复杂的应用程序。复杂的应用程序需要用Java、C或C++写的专门的客户端/服务器应用程序。事实上，在DOM和Ajax出现前想要用JavaScript这样做几乎是不可能的。然而，Ajax引入了不需要重新加载页面即可与服务器通信的功能，而且DOM允许程序员即时改变HTML。

2007年Google引入了Gears，这是一个浏览器扩展，它带给开发人员前所未有的力量。Gears允许浏览器离线工作，可以让用户在浏览器中存储更多数据，并设计了一个工作池（Worker pool）用于处理长时间运行的任务。Gears已经停用，但是它的大多数功能经过修改后已移植到HTML 5中。

现代Web看起来像一个全方面的网站，包含各种内容，从维基百科那样依旧有效的旧式文档集合，到Facebook、YouTube和eBay这种提供与他人交互的网站，再到可以称为替代桌面应用程序的东西（例如Gmail 和Google Docs）。许多以前独立的应用程序，例如

邮件客户端，已经变成了Web体验中不可分割的一部分。在现代Web中，应用程序和页面间的界线已经非常模糊，只在网站的用途上有所区别。

在浏览器中运行应用程序对于用户和开发人员都有很大的优势。对用户来说，使用Web应用没有负担：用户可以试用应用程序，如果不喜欢就换个页面而不会在硬盘上留下任何东西。尝试新的应用也相当安全，因为它们运行在一个沙箱环境中。当开发人员更新代码时，新版本的应用程序会自动下载到浏览器。Web应用程序很少有版本号，至少已公开的应用程序是这样的。

对开发人员来说，在浏览器中运行应用程序优势更大。首先，对用户来说的优势对开发人员也是优势。其次，开发人员不需要写安装程序，而且新版本可以自动发送给用户，使得小量的更新切实可行。另外，还有其他好处。

Web是跨平台的，开发人员都希望开发的应用能运行在多个系统上，例如Windows XP、Windows Vista、Windows 7、Mac OS X、Linux、iPhone/iPad和Android，这个愿景如果用传统的开发工具来实现将是一项非常艰巨的任务，但是用Web和一些具有前瞻性的技术来实现，就会变得十分简单。一个用标准类库（例如jQuery）建立的网络应用能运行在上述所有这些和其他几个平台的多数浏览器上。Sun公司一度希望Java applets将Web定义为一个平台，现在JavaScript已成为默认的Web平台。

你甚至可以在移动设备上运行Web应用，至少现在可以在智能手机上运行。你可以用一个包装（例如PhoneGap）来创建一个HTML 5应用，并把它打包，在App Store、Android Market和其他网站上出售。你可以创建一个与Web服务器大量交互或者完全独立的应用，两者皆可。

HTML 5之前的Web真正的不足之处在于，一个运行在计算机上的Web应用要占用上千兆字节内存和磁盘空间，运行起来像老式vt320终端上一样慢。所有数据存储必须在服务器上完成，所有文件必须从服务器加载，每一个交互必须完成一次与服务器间的往返。这会使用户感觉较慢，特别是当用户距离服务器很远时。如果用户查看页面每次要耐心等待至少400ms，用户就会感觉应用程序运行很慢。从我在特拉维夫的办公室到加利福尼亚的一个服务器，ICMP ping的一次往返时间约为250ms。到服务器上的任何操作都将耗费额外的时间，使应用运行慢下来。当然，移动设备通信更慢。

## JavaScript的胜利

尽管JavaScript自1995年出现以来已经成为Web开发的关键组成部分，但是这十几年来它一直背负着坏名声：语法古怪、性能低下、产生奇怪的错误，以及对DOM（Document Object Model，文档对象模型）的依赖。尽管浏览器已经将其封闭在“沙箱”中，简化了

用户的安全问题，但是要为JavaScript开发人员提供一些在传统桌面应用开发中显得微不足道的功能却很难。

脚本文化造就了它自身的问题。提供非常低的进入门槛是件好事，但是需要付出代价。代价之一就是，这样一种语言往往允许没有经验的程序员做一些非常不明智的事情。起初，程序员可以很容易地从Web上找到JavaScript例子，剪切、粘贴或做一些修改，就能得到大多数可用的东西。遗憾的是，随着时间的推移对这些代码的维护变得越来越困难。

随着Ajax的流行，开发人员对JavaScript有了新的看法。一些开发者致力于对解释和运行JavaScript代码的引擎进行改进，致使运行速度大幅度提升。还有一些开发者则致力于语言本身，他们意识到JavaScript有一些很不错的功能，能够开发最好的应用，正如Douglas Crockford的《JavaScript 语言精粹》（JavaScript: The Good Parts, O'Reilly 2008出版）中所讲述的一样。

除了核心语言之外，开发人员还创建了一些工具使JavaScript更易于调试。尽管Venkman（一款早期的调试器）在1998年就出现了，但是2006年发布的Firebug成了JavaScript调试器的黄金标准。它允许开发人员对Ajax调用进行跟踪，查看DOM和CSS的状态，并单步执行代码等。基于WebKit的浏览器提供了类似的内置功能，尤其是苹果公司的Safari和Google Chrome，Opera Dragonfly则对Opera提供支持。甚至，在移动设备封闭空间中工作的开发人员现在也能用Weinre（Web Inspector Remote：远程Web调试器）像Firebug一样进行调试。

类库是JavaScript最近大规模投入的最终关键组成部分。开发人员可能依旧不能理解他们使用的全部代码，但是能够将代码组织得更易于升级，有时甚至可以用可互换的类库简化代码管理。

### jQuery

如果有一个类库可以称为JavaScript类库的“黄金标准”的话，那一定是John Resig的jQuery类库。jQuery围绕DOM和其他JavaScript对象（如XMLHttpRequest对象）形成了一个包装器，使得用JavaScript实现各种功能更轻松、更有趣。从许多方面讲，jQuery都是每个JavaScript程序员应该熟知、必不可少的类库。学习jQuery请访问网站<http://jquery.org>，或者阅读由O'Reilly出版的大量优秀图书，例如《深入浅出jQuery》（Head First jQuery）或者《jQuery锦囊妙计》（jQuery Cookbook）。本书中的许多示例都是用jQuery编写的。

### ExtJS

鉴于jQuery形成了一个DOM包装器ExtJS，因此人们从Sencha（<http://sencha.com>）开始设法尽可能地把它从中剥离。ExtJS以丰富的widget控件集为特点，可以在任

意Web页面上执行，并提供了许多桌面开发人员所熟悉的widget控件，如树、网格、标单、按钮等。整个系统经过深思熟虑并良好地组织在一起，使许多应用的开发成为一种享受。虽然ExtJS的库往往都很大，但是它非常适合一些应用开发。ExtJS有一个不错的功能，即它的许多对象都知道如何保存自身的状态。因此，如果用户有一个表格，并当再次浏览该表格时对它的列进行了重新整理，那么可以设置它来保存这个状态。第4章中的“在ExtJS中使用localStorage”讲述了如何利用本地存储加强这一功能。

#### *Google Web套件及其他*

一些工具如GWT允许程序员编写Java代码，然后编译成JavaScript，这样就可以在浏览器中运行了。

# JavaScript的力量

JavaScript语言编程并不难，但是要达到真正的JavaScript专家级水平非常有挑战性。成为熟练的JavaScript程序员有几个关键因素。本章中的技术将反复出现在本书剩余部分所介绍的类库和编程实践中，因此在继续介绍其他章节前需要先熟悉这些技术。

附录中列出了一些优秀的JavaScript编程工具。这些工具可以提供大量的帮助。例如，JSLint（见附录：JSLint）能捕获一些程序员可能会漏掉的错误。一些网站是这类编程工具的很好来源，如StackOverflow和O'Reilly Answers。

本章并非要完整地介绍JavaScript的力量。有关内容可参阅下列JavaScript书籍：

- 《JavaScript语言精粹》（JavaScript, The Good Parts），Douglas Crockford
- 《JavaScript权威指南》（JavaScript: The Definitive Guide），David Flanagan
- 《高性能JavaScript编程》（High Performance JavaScript），Nicholas C. Zakas
- 《JavaScript模式》（JavaScript Patterns），Stoyan Stefanov

## 非阻塞I/O和回调

撇开语言本身不谈，学习JavaScript的首要关键是理解事件驱动编程。JavaScript运行环境中的操作往往是异步操作，这就是说先在某个地方创建操作，当一些外部事件发生后再执行。

这样可以说明传统语言中I/O方面发生的主要变化。例2-1是一个典型的传统语言I/O示例，本例使用PHP语言。`$db->getAll($query)`；这一行要求数据库访问硬盘，因此它花费的运行时间将比函数中的其他部分多很多。当程序等待服务器执行时，查询语句被阻