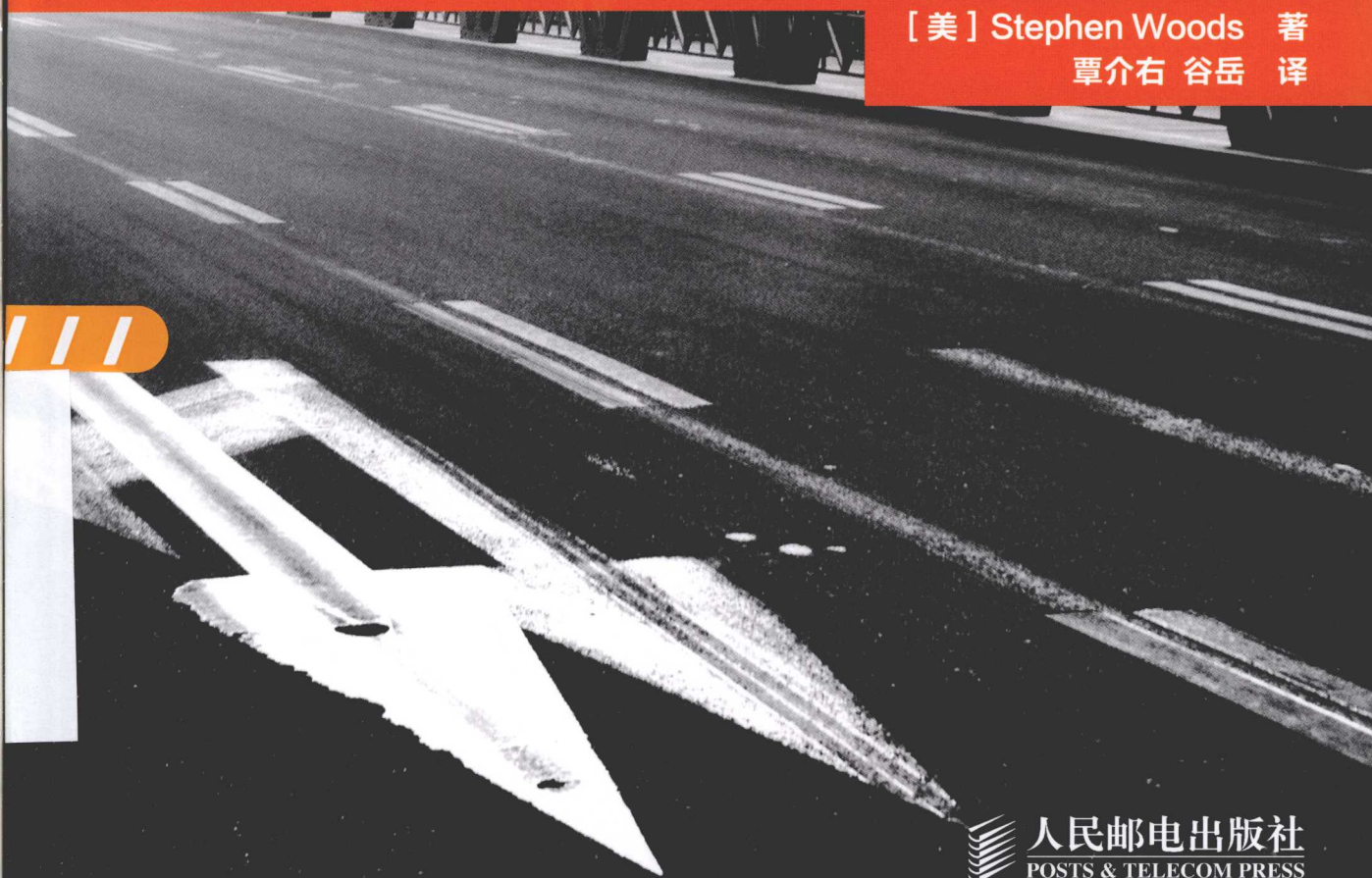




HTML5

触摸界面设计与开发

[美] Stephen Woods 著
覃介右 谷岳 译



人民邮电出版社
POSTS & TELECOM PRESS

HTML5

触摸界面设计与开发

[美] Stephen Woods 著
覃介右 谷岳 译

人民邮电出版社
北京

图书在版编目 (CIP) 数据

HTML5触摸界面设计与开发 / (美) 伍兹 (Woods, S.)
著; 覃介右, 谷岳译. — 北京: 人民邮电出版社,
2014.4

ISBN 978-7-115-34352-9

I. ①H… II. ①伍… ②覃… ③谷… III. ①超文本
标记语言—程序设计②触摸屏—界面—设计 IV.
①TP312②TP334.1

中国版本图书馆CIP数据核字(2014)第001814号

版权声明

Building Touch Interfaces with HTML5

ISBN: 9780321887658

Stephen Woods

Copyright © 2013

Authorized translation from the English language edition published by New Riders. All rights reserved.

本书中文简体版由美国 **Peachpit** 出版公司授权人民邮电出版社出版。未经出版者书面许可, 对本书任何部分不得以任何方式复制。版权所有, 侵权必究。

◆ 著 [美] Stephen Woods

译 覃介右 谷岳

责任编辑 赵轩

责任印制 王玮

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京天宇星印刷厂印刷

◆ 开本: 800×1000 1/16

印张: 13

字数: 262千字

2014年4月第1版

印数: 1—3500册

2014年4月北京第1次印刷

著作权合同登记号 图字: 01-2013-5702号

定价: 59.00元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第0021号

内容提要

本书专注于触摸界面的开发，内容的结构和优化网站的思路大概一致。上半部分涵盖了能使各类网站，特别是移动网站变快的基本概念。第2章和第3章告诉你如何创建一个简单的网站，并能快速加载。第4章可以帮助你使用缓存来提高用户再次访问时的速度。第5章是关于要摒弃一次性的页面加载方式和重构应用程序以使实际上和感觉到的性能都达到最优的。书的后半部分是专门讲触摸界面的，特别是尽可能地让它们更平稳和快速。本书适合具有一定经验的 Web 开发者阅读参考。

北京出版集团

北京

献给“生鱼片”——最棒的猫

致谢

感谢 Jeff Riley、Nancy Peterson、Michael Nolan，还有 Peachpit 的工作人员，你们推动了本书的诞生，并让文字变得通顺。感谢 Nicholas Zakas 对本书非常细致、深思熟虑的批评，还有他在 Yahoo! 的指导。

还要感谢 Stoyan Stefanov 的终审和他在技术文章写作领域上的宝贵经验。同样感谢 Guy Podjarny 为本书付出的时间和做过的研究。

如果没有我的经理 Ross Harmes 以及 Flickr 前端团队其他同事们的支持，绝不会有这本书的诞生。

感谢 Benjamin 为我展示了移动设备的众多用途。

最后，感谢 Elise 容忍我每天晚上一连几个小时地盯着电脑和一堆手机。

前言

在我写作时，11.42% 的网页流量来自移动设备（根据 StatCounter.com 的统计数据）。这个数字一年前是 7%，三年之前是 1.77%。虽然桌面设备还会陪伴我们一段时间，但是 Web 的未来是在移动设备上的。

对于 Web 开发者，支持移动设备是自 21 世纪初的 Web 标准革命以来最大的变革。移动设备都搭载了支持 HTML5 的现代浏览器。它们还装有容量较小的内存和缓慢的 CPU，经常通过高延迟的网络来连接。而且它们几乎全都有触摸界面。

为移动设备开发就是为触摸开发。许多桌面 Web 开发中的技巧对于移动 Web 依然适用，不过有些却是完全不同——弄清楚这些不同会有一定难度。我写这本书来帮助你去弄清楚它们。

谁适合读这本书

这本书是为两类读者写的：

- 经验丰富的 Web 开发者，但从未开发过移动或触摸界面的网页，想要学习
- 一直致力于让他们的移动版网站更优秀的开发者

这本书不适合绝对的初学者。你需要有 Web 前端的背景知识：HTML、CSS 和 JavaScript。对 HTML5、CSS3 的新接口和功能有一定经验会更好。

最重要的是，这本书适合那些并不满足于他们的移动网站的现状的人。如果你想要建设一个迅速而流畅的网站，这本书再适合你不过。

你将能学到什么

本书专注于触摸界面的开发，内容结构和优化网站思路大概一致。上半部分涵盖了我认为能使各类网站，特别是移动网站变快的基本概念。第 2 章和第 3 章告诉你如何创建一个简单的网站，并能快速加载。第 4 章可以帮助你使用缓存来提高用户再次访问时的速度。第 5 章是关于要摒弃一次性的页面加载方式和重构应用程序以使实际上和感觉到的性能都达到最优的。书的后半部分是专门讲触摸界面的，特别是尽可能地让它们更平稳和快速。随着层层深入，本书也逐渐复杂和深奥。如果你觉得后面的章节太难了，先尝试在工作中运用你所学到的东西，再回头来看我在本书末尾介绍的一些想法，如一个好用的移动网站不需要双指缩放的功能。

你需要用到什么

如果你想要在本书中学到更多东西，除了你的电脑外，你至少还需要一台触摸设备。如果只能有一台，我推荐 iOS6 或 Android 4 的设备。如果条件允许，上述两台都是有最理想的了。

开发移动网页时，设法利用尽可能多的设备。iOS 和 Android 模拟器无法取代真机。在写这本书时，我使用了一台搭载 Android 4.0.4 的 Samsung Galaxy S III、一台 iPhone 4、一台 iPhone 5、一台 iPad 1，还有一台 HTC 8X (Windows 8)。再用模拟器作为补充。

在 Flickr，我们拥有的设备和上述类似，不过还有一些 Android 平板和一台 Kindle Fire。

框架

本书没有用 jQuery 或是其他任何 JavaScript 框架。你会了解到一些专用的库，不过我们还是尽可能多的关注原生的 DOM API。这并不是说你应该避免使用框架或者远离框架！只是我想要确保你能理解这些东西最根本的原理。这样，当你决定使用 jQuery mobile、Backbone.js、Zepto.js 或是其他框架时会更轻松。

理解原生 DOM API 的另一个巨大的好处是，当你在某个库中找到一个 bug 的时候，你可以自行修复并创建一个包含你修复代码的 pull request，从而造福整个社区。

配套网站

本书上的所有示例代码和最新的改动可以在配套网站：touch-interfaces.com 上找到。这些代码也在 GitHub 上做了镜像，在这里你可以找到有关示例代码的 issue，还可以提交 pull request 到：<https://github.com/saw/touch-interfaces>。

欢迎进入移动网页

网页是用 HTML、CSS 和 JavaScript 构建的。移动网页也是一样。只需要一个网页浏览器和一个文本编辑器就可以开始，但要更高效的话，我建议再多几个工具。

工具

最简单的过程，是用一个文本编辑器和桌面浏览器开发，然后在旁边准备一个触摸设备用来测试。



一个文本编辑器 & 一个 WebKit 内核的浏览器

我用适用于 Mac OS X 的 TextMate 2 (github.com/textmate/textmate)，不过任何其他编辑器都没问题。

因为绝大多数的移动设备运行的是 WebKit 的浏览器，所以，你会发现 Chrome 或 Safari 是能提高效率的必备工具。虽然与真实设备上的测试会有些不同，但它简单易用，必不可少。



一个 Web 服务器

为了在真实的设备上测试网站，你需要通过无线局域网提供页面。在 Mac 上，我发现 MAMP (www.mamp.info) 是一个非常方便的工具，不过使用内置的 Apache Web 服务器同样可以。



一台触摸设备

没有什么可以替代一台真实的设备。如果你能负担得起的话，我建议型号较新的 Android 手机和 iOS 设备各有一部。如果你只买得起一部，找到愿意借手机给你测试一段时间的人，会有很帮助。

跨设备平台测试

不能想当然地认为所有的 Webkit 的浏览器都是一样的。你应该在 iOS 5、iOS 6、Android 2.3、Android 4.0、Android 4.1 (Chrome) 和 IE 10 里面测试你的 App。如果你不能直接使用这些设备，这里是如何对它们进行测试的指南。



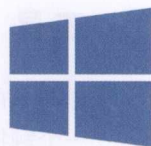
IOS SAFARI

Apple 提供了一个相当好的模拟器 XCode。它可以模拟搭载 iOS 5 或 6 的平板和手机。它还支持结合 Safari 浏览器的远程调试。这真的是一个伟大的工具，假如你有一台 Mac，这是这堆工具中关键的一部分。XCode 在 Mac App 商店免费提供。



ANDROID

谷歌提供了几乎所有版本的 Android 的模拟器。这些都可以 在 Android SDK (developer.android.com/sdk) 中找到。只要有了 Android SDK，就可以分别下载各种 Android 版本的镜像。请记住，这些都是谷歌官方的版本，Android 版本在实际设备上可能有很大不同。



WINDOWS 8

微软确实提供了一台针对 Windows Phone8 模拟器，它在 SDK (dev.windowsphone.com/en-us/downloadsdk) 中。模拟器只能在 Windows 上运行。IE10 的移动版与桌面版是相同的，所以大部分 Debug 工作可以在桌面上进行，而不是在模拟器里面。



DEBUGGING

调试手机网页是一个苦差，但也有很多工具使其变得更容易。我提供了一些工具的列表，在配套网站上的附录 A 中。

作者简介

Stephen Woods 是 Flickr 团队的一名资深前端工程师。从 20 世纪 90 年代起，一直致力于 Web 用户界面的研发。2006 年 Stephen 入职 Yahoo!。在加入 Flickr 团队之前，他开发支持 Yahoo! 主页的 JavaScript 平台，还曾在 Yahoo! Personals 的 UI 团队工作，并且是一位精通 Web 的全栈工程师，但主要兴趣是使用 Web 技术构建响应式的用户界面。Stephen 曾在 SXSW 和 HTML5DevConf 就触摸交互进行技术分享，并发表在《.net》杂志上。

目录

第 1 章 移动设备概述 0

- 1.1 触摸设备和桌面设备之间的差异 2
- 1.2 广泛使用的设备 3
- 1.3 HTML 56
- 1.4 神秘谷，是什么让触摸界面反应灵敏？ 7
- 1.5 总结 9

第 2 章 创建一个简单的内容型网站 10

- 2.1 选择一个观念：移动优先或置后 12
- 2.2 创建标记 13
- 2.3 奠定基础的 <head> 标签 15
- 2.4 理解 Viewport 15
- 2.5 响应式的 CSS 19
- 2.6 总结 24

第 3 章 提高第一次加载的速度 25

- 3.1 浏览器是如何加载页面的 27
- 3.2 为什么页面加载缓慢？ 27
- 3.3 用 YSlow 和 PageSpeed 提升速度 30
- 3.4 解决常见的问题 33
- 3.5 将它们全部放在一起 41
- 3.6 总结 41

第 4 章 加快下一次访问的速度 42

- 4.1 在 HTTP 中缓存 44
- 4.2 为移动优化 45
- 4.3 使用网络存储（Web Storage） 46
- 4.4 应用缓存（Application Cache） 53
- 4.5 总结 57

第 5 章 使用 PJAX 提升触摸体验 58

- 5.1 页面加载的代价 60
- 5.2 浏览器历史 API 62
- 5.3 添加 PJAX 69
- 5.4 总结 77

第 6 章 轻触 VS 单击：基本的事件处理 78

- 6.1 是什么让轻触不同？ 80
- 6.2 触摸事件简介 80

6.3 处理轻触 83

6.4 总结 89

6.5 项目 89

第 7 章 CSS 过渡、动画和变换 90

7.1 动画元素 92

7.2 CSS 变换 105

7.3 总结 112

7.4 项目 112

第 8 章 JavaScript 性能最优化 113

8.1 性能测试和调试 115

8.2 只写入 DOM 116

8.3 给用户反馈的优先级是最高的 117

8.4 将它们一起使用：无限滚动 118

8.5 总结 127

8.6 项目 127

第 9 章 手势的基本内容 128

9.1 为什么用手势呢？ 130

9.2 创建一个循序渐进增强的触摸控制 131

9.3 创建一个触摸控制的灯箱（lightbox） 136

9.4 总结 150

9.5 项目 150

第 10 章 滚动与滑动 151

10.1 滚动 153

10.2 布局失效 159

10.3 让鸟类浏览工具可滑动 161

10.4 总结 176

10.5 项目 177

第 11 章 双指缩放和其他复杂的手势 178

11.1 了解多点触摸的限制和支持情况 180

11.2 处理多点触摸 180

11.3 处理双指缩放 185

11.4 总结 195

11.5 项目 195

1.1 移动设备设计面临的挑战

一切都因 iPhone 开始。在这之前，手机网页就是其桌面网页的简化版。极少数浏览器能够支持 JavaScript，大多数浏览器只是将网页转化成适合小屏幕而已。iPhone 上的 Safari 是一个真正的 Web 浏览器，它支持 JavaScript 和 CSS。不仅如此，它还是一个非常前沿的浏览器。为 iPhone 设计的网站可以使用一些新生的工具套件，即现在的 HTML5。如今，市面上有数百种的移动设备在售。它们千差万别，各有千秋，但它们有一个共同的特点：先进的 Web 浏览器和触摸界面。

1.2 移动设备设计面临的挑战

1.1 触摸设备和桌面设备之间的差异

大多数触摸设备是移动设备。为桌面设备开发网站的开发者最担心的是它们的网页能否兼容各种浏览器。所以，开发者很大一部分的工作是理解各种浏览器之间的差异。

移动领域问题不在于各种浏览器，而在于不同的移动设备。当然，有几种不同的移动浏览器，并且它们之间的确存在一些差异，但大体上移动设备版本与其相应的桌面版本差异不大。移动设备和桌面设备真正差异在于以下 4 个因素：

- 规格
- 计算能力不足
- 人们如何使用它们
- 触摸界面

1.1.1 规格

移动设备和台式电脑（或笔记本电脑）之间最明显的差异是尺寸。移动设备的尺寸比较小，所以搭载较小屏幕，且没有鼠标，大多数情况下也没有键盘。

触摸设备目前主要有三种规格：手机、小型平板和大型平板。大型平板包括 iPad（不包括 iPad mini）和各种 10 英寸的 Android 平板，如 Nexus10。小平板屏幕一般约为 7 英寸，如 Nexus7、Galaxy Note 和 iPad mini。

所有这些设备上的浏览器始终都是全屏的。用户“调整”浏览器大小的唯一途径是通过旋转移动设备在横向和纵向模式之间进行切换。只为一种尺寸的设备优化并不困难，困难的是要确保你的网站适用于所有的设备。

1.1.2 移动设备的计算能力不足

写本书时，最强大的触摸屏设备是第四代 iPad。它是性能强大的机器。根据相同的标准，它的性能与 2004 年发布的 Power Mac G5（最后一代摩托罗拉版）相同。

可以肯定地说，大多数用户并没有使用最新和最强大的移动设备。大多数运营商提供“合约”手机，如 iPhone 4 和三星 Galaxy Exhilarate，这些手机的运行速度要慢得多。它们的 CPU 性能和 20 世纪末、21 世纪初最快的台式电脑差不多。iPhone 3GS 仍然被广泛使用，它拥有单核处理器、256 MB 的内存，CPU 性能大致相当于最后一代与 CRT 显示器搭配销售的 iMac。虽然移动设备的性能仍在继续增长，但与一般桌面电脑的性能相比，它们还是太慢了。值得高兴的是，大部分近期发布的设备都拥有独立的 GPU。一个 iPhone 4 在性能上可以被认为是一台装有强大图形处理器，但性能稍逊的电脑。

1.1.3 人们使用触摸设备的方式大相径庭

人们使用手机和桌面电脑的习惯不同。人们使用桌面电脑一般是为了完成特定的工作，所以可能不会做很多其他的事情。至于移动设备，不论是手机还是平板电脑，通常人们都是在做别的事情的间隙来使用它们。他们可能正在乘坐公交车、排队等候或只是在参加一个无聊的会议。

例如，你搭建了一个财经新闻网站。在桌面电脑上，用户可能会花费一段时间单击几次链接，然后用几分钟的时间阅读一篇文章。而移动用户可能正在做一件别的事情，只会在空闲时的几秒时间来完成上面的操作。桌面网站的速度一般不慢，且桌面用户的容忍时间更长。如果你的网站载入时间有一点长，用户就会切换到别的标签去看其他的东西，然后再回来。但在手机上，用户永远不会给你这样的机会。

移动设备就是触摸设备

所有在售的新型智能手机和平板电脑有着一个非常重要的共同特点：它们都具有触摸界面——在大多数情况下，还可以多点触控。合适的大小、美观的屏幕以及其他的先进功能，这些功能相对于使用触摸屏这样的革命性革新，都只是微小的改进而已。

1.2 广泛使用的设备

在 2007 年，iPhone 刚出现时，没有任何一部其他的手机拥有真正能触摸感知的 Web 浏览器。也没有任何支持多点触摸的设备。其他的智能手机大多数使用手写笔进行交互，当时市场上的几种平板电脑也是一样。今天，市面上流通的有数百种触摸设备。我们不可能了解每一种新设备，不过它们可以根据操作系统、规格、搭载的 Web 浏览器将它们分为几类。

1.2.1 操作系统

如今市场上有四种主要的操作系统适用于触摸设备：

- iOS
- Blackberry OS
- Android
- Windows 8

Android 是目前最流行的手机操作系统。根据 International Data Corporation (IDC) 的数据，2012 年第三季度智能手机出货量的 75% 运行的是 Android 系统，14.9% 运行的是 iOS 系统。其他操作系统的市场份额只有个位数（这不包括的 Windows Phone 8，因为它在 2012 年 10 月才发布）。

IOS

iOS 是所有苹果手机和平板运行的操作系统。因为苹果控制着硬件，所以所有的 iPad 和 iPhone 能以相同的方式运行，只有大小和处理能力上的差异。苹果推出软件更新时无需依赖运营商，这意味着，苹果设备更容易运行最新版本的操作系统。iOS 开发者，David Smith，发现他开发的应用程序的用户中，有 79.2% 正在使用的 iOS 6 或更高版本（相对 2012 年 12 月）的操作系统，有 94% 的人正在使用 5.1 或更高版本的操作系统。



注：David Smith 会在他的网站（david-smith.org/iosversionstats/）上更新这些统计数据。

苹果随着操作系统的升级同步升级浏览器，它不允许用户安装不同内核的浏览器。至少现在，可以认为 iOS 用户主要安装着上一个较新版本的 iOS 和 iOS Safari。

ANDROID

Android 的情况很不一样。Android 是开源的，手机厂商有权自由定制成他们认为合适的样子，所以在手机之间实际的用户界面差异很大。

Android 有三个主要的版本：2、3 和 4。Android 3.0（蜂巢）是平板专用操作系统。大约 50% 的 Android 手机运行着 2.3.x（姜饼）。最新版本的 Android 4.1（果冻豆）并没有被广泛使用，在 2013 年 1 月，只有不到 10% 的设备在运行它，不过这一数字正在迅速增长。时下的“旗舰”手机，如 Samsung Galaxy SIII，仍然搭载着 Android 4.0.x（冰淇淋三明治）系统出售。Android 4.1 同时支持平板电脑和手机。



注：谷歌在 developer.android.com/about/dashboards/ 提供了各版本的统计

Android 包括一个默认的浏览器，而且允许让用户安装不同的浏览器。4.1 版之后，Android 上默认的浏览器改为了 Chrome。

Kindle Fire

Amazon Kindle Fire 运行着高度定制版的 Android，这个定制后的版本不包括任何原生的 Android 应用程序，并拥有自行研发的浏览器，称为 Silk。值得注意的是，Silk 因其在客户端设备和 Amazon 云上的运行能力而知名。这样可以显著地提高性能，尤其在高延迟的网络链接下。这对开发者有一些影响，我们将在第 4 章“提高下一轮的访问速度”中阐述。

BLACKBERRY OS

Blackberry 手机是最早真正获得成功的智能手机。但到了 2012 年 12 月，其市场份额已经迅速减少至大约 4.3%。Blackberry 也出售平板，称为 Playbook。这两种设备使用相同的基于 WebKit 的网页浏览器。尚存的非触摸屏式的黑莓设备占据着智能手机市场的长尾。