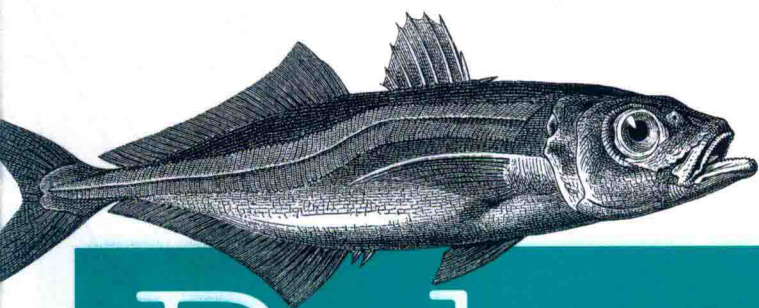
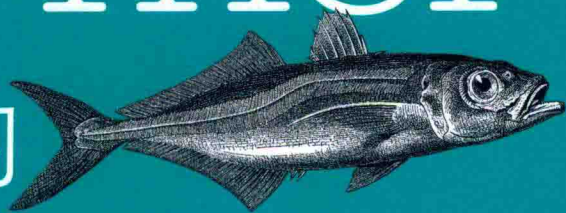


O'REILLY®

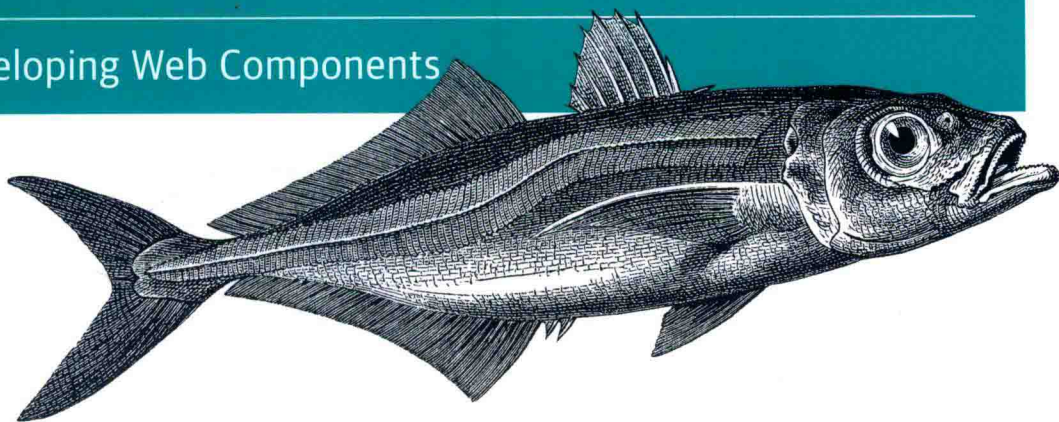


Polymer

面向未来的 Web组件开发



Developing Web Components



[美] Jarrod Overson Jason Strimpel 著

谢光磊 译



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Polymer

面向未来的Web组件开发

Developing Web Components

[美] Jarrod Overson 著
Jason Strimpel

谢光磊 译

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

Web 组件 (Web Component) 是一项新兴的 Web 技术。《Polymer: 面向未来的 Web 组件开发》将帮助你理解 W3C 关于 Web 组件的标准, 教你构建可重用、自定义的 HTML5 Web 组件。本书的主要内容包括: 文档流、定位、z-index 等核心概念, 以及定位、拖拽、改变尺寸的实现; UI 的内涵及从 Dojo、jQuery UI、Kendo UI 等中抽象出的典型模式; 深入理解 W3C 标准, 以及如何将自己的控件转化为一个功能齐备的 Web 组件; 利用 Google 的 Polymer 框架封装、打包、部署 Web 组件。

©2015 by Jarrod Overson and Jason Strimpel

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Publishing House of Electronics Industry, 2015. Authorized translation of the English edition, 2015 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.

本书简体中文版专有出版权由 O'Reilly Media, Inc. 授予电子工业出版社。未经许可, 不得以任何方式复制或抄袭本书的任何部分。专有出版权受法律保护。

版权贸易合同登记号 图字: 01-2015-3990

图书在版编目 (CIP) 数据

Polymer: 面向未来的 Web 组件开发 (美) 奥弗森 (Overson, J.), (美) 史特林贝尔 (Strimpel, J.) 著; 谢光磊译. —北京: 电子工业出版社, 2015.10

书名原文: Developing Web Componets

ISBN 978-7-121-27172-4

I. ① P…II. ① 奥… ② 史… ③ 谢…III. ① JAVA 语言—程序设计 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2015) 第 224154 号

责任编辑: 张春雨

印 刷: 北京中新伟业印刷有限公司

装 订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 14.75 字数: 306.8 千字

版 次: 2015 年 10 月第 1 版

印 次: 2015 年 10 月第 1 次印刷

定 价: 65.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

O'Reilly Media, Inc. 介绍

O'Reilly Media 通过图书、在线服务、杂志、调查研究和会议等方式传播创新者的知识。自 1978 年开始，O'Reilly 一直都是发展前沿的见证者和推动者。超级极客正在开创未来，我们关注着真正重要的技术趋势，通过放大那些“微弱的信号”来刺激社会对新科技的采用。作为技术社区中活跃的参与者，O'Reilly 的发展充满着对创新的倡导、创造和发扬光大。

作为出版商，O'Reilly 为软件开发人员带来革命性的“动物书”，创建了第一个商业网站 (GNN)；组织开放源代码峰会，以至于开源软件运动以此命名；通过创立 Make 杂志成为 DIY 革命的主要先锋；公司一如既往地用各种方式和渠道连接人们和他们所需要的信息。O'Reilly 的会议和峰会聚集了超级极客和高瞻远瞩的商业领袖，共同描绘出开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计算机用户。无论是通过印刷书籍、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信念——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 是位少有的商人，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

前言

为什么写这本书

13年前，我进入了 Web 开发领域。这 13 年来，Web 开发技术的进化一刻都未停止过，而且，其进化的速度正在逐渐加快。如果你了解一下，Google 在 Polymer 和 Web 组件技术中的投入，你多半会认为，在不久的将来，随着浏览器技术的进步，Web 组件将在 Web 开发领域占据一席之地。虽然目前 Web 组件还有一些需要解决的问题，比如搜索引擎优化（SEO）等，但我认为这不会阻碍 Web 组件的发展，我甚至认为这不是一件坏事。在 Web 开发领域，Web 组件技术（或具有类似关于继承、构建、封装、依赖的标准的技术）已经被期盼已久。即使你不太关心 Web 组件，那么看看这个领域的牛人们是如何实现自己的 UI 库（比如，使一个元素可拖拽之类），也没有坏处。

Glenn Vanderburg 说：“开发者应该理解自己日常工作之下的抽象层。”我赞同这句话，而且我认为，对于大部分前端工程师，这一层抽象层通常就是指控件库，以及 jQuery 插件甚至 jQuery 本身，因为大部分前端工程师的日常工作就是跟这些工具打交道。这一层抽象层中，包含控件库所进行的 DOM 操作、DOM 本身及其原生 API。理解这些库是怎么工作的，会帮助你写出高效的代码，帮助你提升。

有时候，这些常用的库并不能很好地适用你面临的具体情形（你的产品经理或信息技术官将此情形抛给你处理），这时，如果你不会“游泳”，就会“溺水而亡”。我想，提前学习一些“游泳”课程（也就是本书了），会让你在那一天到来之时不那么手忙脚乱。而我，就没你那么幸运了。

我曾经负责制作一个定宽的门户产品，其中每一个门户组件都是可以调整大小的。听上去很简单，是吗？首先，我被告知在我之前的 Web 开发者曾经尝试完成这个功能，但是都失败了，他们说这是不可完成的。好吧，听上去很挑战。当时 IE9 刚刚进入市场，因为之前的“技术债”，我们运行着三个不同版本的 jQuery 和两套不同版本的 jQuery UI，

没有一个组合能够很好地适配 IE9。于是，我工作的第一步，就是将站点使用的 jQuery 统一到一个版本，并适配 IE9。你也许认为，统一 jQuery 的版本非常简单，但刚刚接手，我就意识到为什么我们欠下了那么多的“技术债”。我们的站点上运行着大量新旧混杂、良莠不齐的 jQuery 插件。升级 jQuery 伴随着对这些插件的升级，以及打破了现有的代码体系。我对一些插件进行了升级或者修补，而更换了另外一些插件。这个过程需要对 DOM 的深入了解，当时我并不具备。我在控制台输出日志进行调试，在 Stack Overflow (<http://stackoverflow.com>) 寻求帮助，甚至进行逆向工程，最终解决了每一个问题。而这，仅仅是在开始开发新功能（门户组件可调整尺寸）之前所做的准备工作。

接下来，我收到了一个可笑的需求，就是门户组件列的调整，只能精确地调整为某些预设好的尺寸，而不能随意调整。当时并没有一个插件能够满足这个要求，于是我编写了自己的插件。

当我把（意大利面条似的）插件代码写完之后，我就需要应用插件使每一个门户组件可被调整大小。各种鼠标事件必须协同起来，保证拖拽的时候不会从外层容器中溢出。从此我爱上了 `setTimeout`，直至今天。

如果你对我的故事感同身受，那么你自己一定也有类似的故事。就像我一样，你把各种 Stack Overflow 上获得的代码片段、各式 jQuery 插件，用自己的“胶水代码”粘在一起，你需要 100 个不同的分支，处理各种各样的异常，这些异常均因为你对内在机制缺乏了解而造成的。我也曾经处于那种情境下，那一点都不好玩。所以，我用自己的经验写下了这本书。

这本书除了能让你避免“溺水而亡”，还能帮助你了解 UI 组件功能的底层细节。

- 本书允许你定义对自己有意义的 API，因为 jQuery 插件并不适用每一个人。
- 让你拥有整个技术栈的控制，方便你调试，允许你自己决定某些变化发生在哪个层面。
- 允许你自己决定哪些特性是重要的，哪些是不必要的，以控制代码的体积。这在对代码体积比较关心的场合（如手机应用）很有用。
- 允许你针对自己的应用和用户场景做额外的优化。

最后，我认为每个前端工程师都应该理解原生 DOM 和 JavaScript。这份理解，和设计方案的能力，是优秀的前端工程师所必须的。

本书包含什么

本书编写的最初目的，是向读者传授基础知识，用以开发和部署 Web 组件。本书更像是入门教程，而不是详尽、彻底的教科书。本书中的知识被分为四个部分，每个部分为一

篇，每一篇又分几个章节，全部是围绕一份项目代码编写的。

第 I 部分 UI 核心概念

第 I 部分的内容包含了诸如克隆节点、渲染图层、堆叠上下文，以及 `z-index` 的相关知识，理解这些知识能帮助我们z对页面上的元素进行定位、拖拽和尺寸调整。这些概念经常被开发者误解（或不完全理解），进而导致不佳的软件设计和实现，影响应用的性能与代码的可维护性。如果你对这些概念已经了如指掌，那么也可以跳过第 I 部分。



第 I 部分，我们将引入一个组件基类（它会派生出对话框控件类）。组件基类和对话框将贯穿本书的其余部分。如果你跳过了第 I 部分，但又想了解一下组件基类的细节，可以快速回顾一下第 2 章。

第 II 部分 构建 UI

本书第 II 部分介绍了当下前端类库，如 Dojo (<http://dojotoolkit.org>)、jQuery UI (<http://jqueryui.com>)、Kendo UI (<http://www.kendoui.com>)、Twitter Bootstrap (<http://getbootstrap.com>) 遵循的概念和模式。你日常工作中用到的 UI 组件，可能就是遵循这些概念和模式实现出来的。如果不深入其中，这些抽象和 UI 组件提供的好处是很难理解的。我强烈建议前端工程师使用一个提供了良好生命周期、继承模式、工具函数和工具类的基础库，比如 jQuery UI。如果你想追寻一个更轻量级的库，那么你可以了解一下 jQuery++ (<http://jquerypp.com>)，这个库提供了拖拽、调整尺寸等功能，代码体积仅为 162KB。它也允许你挑选其中的部分特性生成一份代码，以进一步减少库的代码体积。

本书将实现一个对话框组件，实现该组件的过程就是实践上述概念和模式的过程。

如果你对这些模式都非常清楚，想更深入地研究 Web 组件，那么也可以从第 III 部分开始。

第 III 部分 构建 HTML5 Web 组件

本书的第 III 部分，介绍了由 W3C 定义、浏览器实现的 Web 组件技术。这一部分使用了前两部分编写的对话框控件，将其转化为一个功能全面的 Web 组件。

第 IV 部分 使用 Polymer 测试，构建，部署 Web 组件

本书的第 IV 部分涵盖了如何使用 Google 的 Polymer (<http://www.polymer-project.org>) 桥接 Web 组件技术与当前浏览器环境。Polymer 提供了方便的 API，便于你创建可扩展的 Web 组件。这一部分，还将指导你将第 III 部分编写的对话框 Web 组件转化为一个 Polymer 组件。最后，本书还会介绍如何使用构建工具和包管理工具打包和部署 Web 组件（包括 Polymer 化版本的组件）。

本书不包含什么

本书的目的不是成为一份开发 Web 组件的“权威指南”，教会你在任何能够访问互联网的设备上开发 Web 组件。因为以下两个原因，这几乎是不可能的。

首先，这几年浏览器性能正在飞速的提升，而 Internet 也从一个“文档传递”网络逐渐变成了一个准应用平台。在这个转变的过程中，浏览器变得更加稳定，具有了更多丰富的特性。在编写表格布局的页面和使用 Netscape 4 进行测试的年代里，这是不敢想象的事情。很多原本需要使用 JavaScript 实现的功能，现在只需要 CSS 就能够完成了。然而问题是，虽然 Web 的标准在进化，但浏览器并没能完全跟上，或者说不同浏览器对新特性的实现进度并不一致。浏览器的引擎，是浏览器厂商自己基于 W3C 标准而实现的，不同厂商的实现细节自然不一样。这些新的特性，对于 Web 组件开发来说尤为重要，但是浏览器世界是一个新的主题，一本书也不一定说得完，更不可能在本书中讨论清楚浏览器问题的细节了。而且当你开发 Web 组件的时候，哪些浏览器支持哪些不支持，这些也不是最重要的。

其次，这五年来，能够访问互联网的设备，其数量和种类（手机，平板电脑等）也在飞速增长。新设备的进化和增长，以及设备厂商的努力，使得之前的诸如有限的处理器性能、有限的存储空间、较小的屏幕、较差的网络连接等问题得到了解决。新设备给 Web 开发领域的影响是有趣的，也带来了一些新概念和新模式，如响应式布局设计、触摸和手势事件，等等。但是，每个设备都有自己的“怪癖”，不断增长的移动设备市场，也会给 Web 开发领域带来新的课题。但是，这个课题本书也并不会涉及。

试图在一本书中同时解释清楚这些复杂的问题，几乎是不可能的。本书的内容比《计算机编程艺术》还要宽泛，但是在第一章写完之前，也许就过时了。



除了关于市场上设备数量和功能的问题，你还需要知道的是，W3C 制定的 Web 组件标准仍然在发展中。因此，本书中包含的内容，也许很快就会过时。等 Web 组件标准制定好之后，会再出一版新版。我将尽最大的可能紧紧跟随标准。

本书使用的排版约定

本书的排版约定如下：

斜体

用于显示新术语、URL、邮件地址、文件名、文件扩展名。

等宽字体

用于显示程序源代码。在正文中引用程序中的元素，如变量名、函数名、数据库、数据类型、环境变量、表达式、关键字。

加粗等宽字体

用于显示命令或者需要用户逐字符输入的内容。

斜体等宽字体

占位符，显示需要根据用户或上下文才能确定的值。



这个图标表示提示或建议。



这个图标表示笔记。



这个图标表示警告或注意。

如何使用代码范例

你可以在 <https://github.com/webcomponentsbook>(<https://github.com/webcomponentsbook>) 上下载到本书相关的学习材料（代码示例、练习等）。

本书可以帮助你完成工作。一般来说，可以在自己的程序和文档中使用本书的代码。除非将重新编译代码中的重要部分，你不需要联系我们获取授权。比如，使用本书中的若干程序来编写自己的代码是无须授权的，但是销售或发布 O'Reilly 出版书籍配套光盘中的代码是需要授权的。通过本书中的示例程序回答问题是无须授权的，而将本书中重要部分的示例代码整合到你的产品的文档中是需要授权的。

我们感谢你在使用我们的代码时给出引用说明，但这不是必需的。一个引用说明通常包括标题、作者、出版社和 ISBN。比如，本书的引用说明：“*Developing Web Components* by Jarrod Overson and Jason Strimpel (O’Reilly). Copyright 2015 Jarrod Overson and Jason Strimpel, 978-1-491-94902-3。”。

如果你不确定使用的示例代码是否超出了上述约定，可以随时通过电子邮件联系我们。我们的电子邮件地址是 permissions@oreilly.com。

Safari® 在线图书



Safari Books Online (www.safaribooksonline.com) 是一个发布来自全球技术和商业领域的顶尖作者写的书和视频等优质内容的按需数字化图书馆。

大量的技术软件、软件开发、Web 设计师，以及商业和创意设计都在使用 Safari Books Online，并将其作为研究、解决日常问题，学习及认证培训的首要资源。

Safari Books Online 为各种组织、政府机构和个人提供了一组产品包和计费系统。订阅者可以访问成千上万种数据、培训视频、正式出版的手稿。这些内容来自以下出版社的全文检索数据库：O’Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 等。关于 Safari Books Online 的更多信息，请访问我们的在线站点。

联系我们

关于本书的建议，可以与下面的出版社联系。

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

我们将关于本书的勘误表、示例以及其他信息列在本书的网页上，地址是：<http://bit.ly/developing-web-components>。

如要评论本书或咨询关于本书的技术问题，请发邮件到 bookquestions@oreilly.com。

了解更多关于本书、课程、会议和新闻的信息，请访问我们的网站：<http://www.oreilly.com>。

我们的 Facebook：<http://facebook.com/oreilly>

我们的 Twitter：<http://twitter.com/oreillymedia>

我们的 YouTube 视频：<http://www.youtube.com/oreillymedia>

致谢

Jason Strimpel

首先，我要感谢我的妻子 Lasca 对我的鼓励和理解。我无法想象比她更好的人生伴侣，也没有见过另一个像他这样聪慧、美丽和特别的人。我爱你，你是我的一切。

我还要感谢我的老板 Claude Jones，他帮助和指导我完成了本书。没有他，就没有这本书的存在。

我还要感谢本书的另一位作者 Jarrod Overson，他答应编写本书的第 IV 部分。如果本书有什么不恰当的地方，那一定是我的责任。跟我相比，你是一位更有天赋的作者、演说家和工程师。

此外，我还要感谢 Simon St.Laurent，他倾听了我的不成熟的想法，并给出了很好的建议，这些建议帮助我把这些想法最终转化为了本书。同样，本书的编辑 Brian Anderson，他是我见过的最耐心的人。如果没有他，本书只会是一堆没有上下文，无法被其他人理解的手稿。谢谢你，Brian。

最后，我还要感谢本书的“隐秘合伙人”：社区中的那些编程牛人。他们的编程水平远胜于我，教会了我无数的知识和经验。他们从公众视野中为我们选择出了最精华的部分。如果你足够幸运，能够与他们共事，那么请相信我，你一定会马上发现（他们是真正的牛人），并听从他们的经验。正是因为他们，我才能够从一个资质平平的开发者成长为一名工程师，才能够分享我拥有的这些知识。

Jarrold Overson

如果没有我耐心的妻子，Kate Lane，我在本书中的工作根本无法完成。这些年，她已经习惯于听我唠唠叨叨地讲述那些关于 JavaScript 体系和物理模拟的话题，默默支持我，还能保持好奇心。是你的支持，使我能够安心出国演讲、写代码到深夜、花费很多时间组织社区事务。没有你的支持，就没有我所拥有的一切，包括我们的两个可爱的孩子。

谢谢你 Finn，你为我的生活带来了太多的乐趣。作为你的父亲，我常常被你的天才想法所惊讶和激动。同样谢谢你 Norah，你总能够让我抛却烦恼。你们两个是我的骄傲和前进的动力。

最后，谢谢你 Jason。你在编写本书的时候想到了我，你是 San Diego 最优秀的开发者之一，与你共事是我的荣耀。希望在将来，能够与你有更多的合作。

目录

前言	xiii
----------	------

第 1 章 概述	1
----------------	---

什么是 Web 组件	2
HTML 模板	3
HTML 引用	3
自定义元素	3
Shadow DOM	3
为什么使用 Web 组件	4

I UI 核心概念

第 2 章 基础知识	8
------------------	---

DOM 抽象层的重要性	8
控件的 API 和生命周期	9
继承模式	11
依赖	12
优化	13
Web 组件不仅仅是 JavaScript	14
示例控件	14
Voltron 控件基类	14
对话框类	17
对话框的 CSS 和 HTML	18
总结	18

第 3 章 文档流与定位 19

- 文档流 19
- 定位元素 20
 - offsetParent 20
 - 定位 22
- 计算元素的位置 24
 - 相对于视窗 24
 - 相对于文档 25
- 定位对话框控件 26
- 总结 27

第 4 章 理解和操作 z-index 28

- z-index 究竟是什么? 28
- 渲染层、堆叠次序与 z-index 29
 - 默认堆叠次序 29
 - 重写默认堆叠次序 31
- 堆叠上下文 31
 - 堆叠上下文是怎么创建的? 31
 - 事情越来越复杂了 32
- 管理 z-index 32
 - z-index 管理者对象 33
- 转化为 jQuery 插件 40
- 向对话框控件类添加 z-index 管理功能 41
- 总结 42

II 构建 UI

第 5 章 克隆节点 44

- 使用 cloneNode 方法 45
- 使用 jQuery.clone 46
- 对话框控件 48
- 总结 48

第 6 章 创建浮层 49

- 定义 API 49
- 工具 51
 - 检测滚动栏的宽度 51

计算容器宽度时考虑滚动条	52
获取元素尺寸和坐标	52
监听尺寸改变事件和滚动事件	54
更新选项	54
销毁	54
定位	55
相对视窗或相对另一个元素定位	56
相对另一个元素定位元素	58
向对话框控件添加浮层功能	60
总结	61
第 7 章 拖动元素	62
鼠标事件	62
\$.mousemove	62
\$.mousedown	63
\$.mouseup	63
鼠标事件的最佳实践	63
1. 在 \$.mousedown 事件中绑定 \$.mousemove 事件	63
2. 在 \$.mouseup 事件中解除 \$.mousemove 的绑定	64
3. 将 \$.mouseup 事件响应函数绑定到 <body> 上	64
4. 命名所有事件	64
定义 API	64
创建拖拽柄	65
拖动起来	66
\$.mousedown 响应函数	66
\$.mousemove 响应函数	67
\$.mouseup 事件响应函数	68
销毁可拖拽实例	69
使对话框控件可拖拽	70
总结	71
第 8 章 调整元素尺寸	72
鼠标事件和最佳实践	72
事件	72
最佳实践	73
调整元素尺寸	73
编写调整尺寸的 API	74

拖拽/尺寸调整柄区	75
绑定鼠标事件	76
\$.mousedown 事件响应函数	76
\$.mousemove 事件响应函数	78
\$.mouseup 事件响应函数	79
销毁工作	79
完成调整尺寸功能	79
使对话框控件可调整大小	84
总结	85
第 9 章 完成对话框控件	86
添加样式	86
添加 CSS	86
合并 JavaScript 文件	87
总结	88
III 构建 HTML5 Web 组件	
第 10 章 模板	90
理解模板的重要性	91
延迟加载资源	91
延迟渲染内容	92
从 DOM 中隐藏内容	92
创建和使用模板	92
检测浏览器支持	92
将模板放到标签中	92
将模板内容插入到 DOM 中	93
使用模板编写对话框组件	93
创建和包装对话框模板 API	94
实例化对话框组件	95
抽象对话框模板包装	95
总结	96
第 11 章 Shadow DOM	97
什么是 Shadow DOM?	97
Shadow DOM 基础概念	98
Shadow 宿主	98

Shadow 根元素	98
在 Shadow DOM 中使用模板	99
Shadow DOM 的样式	100
样式封装	100
为宿主元素添加样式	101
在文档中设置 shadow 子树根节点样式	103
内容映射	105
通过 content 标签映射	105
通过选择器映射内容	106
节点分发和接入点	107
Shadow 接入点	108
事件与 Shadow DOM	110
使用 Shadow DOM 更新对话框模板	110
对话框标签	110
对话框的 API	111
更新对话框 show 方法	112
初始化对话框实例	112
总结	113
第 12 章 自定义元素	114
自定义元素简介	115
注册自定义元素	115
扩展元素	116
扩展自定义元素	116
扩展自原生元素	117
定义属性和方法	117
解析自定义元素	118
参与自定义元素的生命周期	119
createdCallback	119
attachedCallback	119
detachedCallback	120
attributeChangedCallback	120
为自定义元素添加样式	121
在自定义元素中使用模板和 Shadow DOM	121
将对话框组件实现为一个自定义元素	123
创建对话框自定义元素	123
实现对话框自定义元素的回调函数	124
实现对话框自定义元素的 API	125