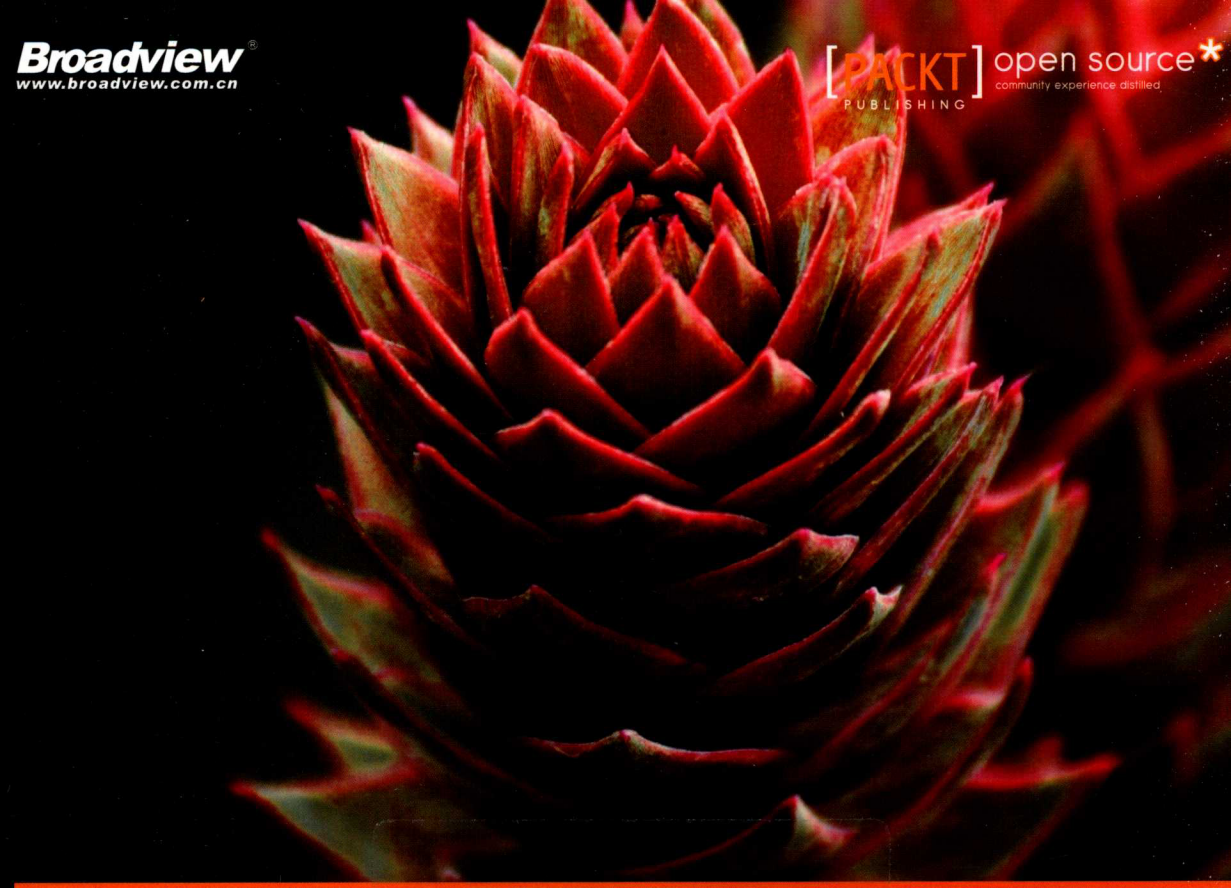
Broadview
www.broadview.com.cn

[PACKT] PUBLISHING open source*
community experience distilled



Mastering PostCS[redacted]ign

深入PostCSS Web设计

用强大的PostCSS编写高性能、模块化的现代CSS网页代码

[英] Alex Libby 著
大漠 孙崇升 姬忠静 肖少彦 译

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Mastering PostCSS for Web Design

深入PostCSS Web设计

[英] Alex Libby 著
大漠 孙崇升 姬忠静 肖少彦 译

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

PostCSS 是目前 CSS 处理器中最流行的一个处理器。PostCSS 依托其强大的插件生态系统，为 CSS 处理器增加了无穷的可能性。

本书共十四章内容，包括介绍 PostCSS，创建变量和混合宏，嵌套规则，创建媒体查询，管理颜色、图片和字体，创建网格，动画元素，PostCSS 插件开发，简写型插件、降级插件和包型插件，定制处理器，管理自定义语法，混合处理器，排除、解决 PostCSS 的相关问题，为未来做准备。

以上内容将带你深入了解 PostCSS 以及如何使用 PostCSS。如果你还没有准备好去了解 PostCSS 能做什么，那么，请跟着这本书的步骤进行系统而深入的学习，你将进入到 CSS 的全新世界。

版权贸易合同登记号　图字：01-2016-9179

图书在版编目（CIP）数据

深入 PostCSS Web 设计 /（英）亚历克斯·利比（Alex Libby）著；大漠等译. — 北京：电子工业出版社，2017.7

书名原文: Mastering PostCSS for Web Design

ISBN 978-7-121-31817-7

Ⅰ. ①深… Ⅱ. ①亚… ②大… Ⅲ. ①网页制作工具 Ⅳ. ①TP393.092.2

中国版本图书馆 CIP 数据核字（2017）第 129802 号

策划编辑：张春雨

责任编辑：徐津平

印　　刷：北京天宇星印刷厂

装　　订：北京天宇星印刷厂

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：22.5　　字数：432 千字

版　　次：2017 年 7 月第 1 版

印　　次：2017 年 7 月第 1 次印刷

定　　价：99.00 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）88254888，88258888。

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

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

作者

Alex Libby 从事 IT 支持工作，有近 20 年的多领域终端用户支持经验，目前在英国的一家全球经销商担任 MVT 测试开发者。虽然 Alex 的日常工作是和各种技术打交道，但他最关注的还是开源社区的动向，尤其是 CSS/CSS3、jQuery 和 HTML5 等技术。到目前为止，Alex 已经通过 Packt 出版了 10 本技术书籍并参与了多本书籍的审校工作，涉及 jQuery、HTML5 视频、Sass 和 CSS 等技术，《深入 PostCSS Web 设计》是 Alex 通过 Packt 出版的第 11 本书。

作者致谢

感谢家人和朋友支持我写完了本书，感谢审校者提出的宝贵建议，如果没有他们就不会有这本书！特别要感谢 Andrey Sitnik 开发了 PostCSS，以及他在本书编写过程中对 PostCSS 核心技术的耐心解答和支持。此外，还要感谢来自 NPM 的 Ernie Salazar，在他的帮助下我成功发布了本书所创建的插件；感谢 David Clark 通过 postcss-bem-linter 插件帮我纠正了插件开发的方式，本书也使用到了这一插件。最后，感谢参与本书编写的所有人，没有你们就不会有这本书的存在！

审校者

Michael Ebbage 是一位软件架构师，专注于电子商务领域和面向 Web 应用的 Java 技术。从他开发第一个 Web 页面到现在已经过了近 20 年的时间，从那时开始他为英国最大的几家公司开发了数百个网站，在这段时间里，工具、技术发生了持续的更新和变革。

他本身是一位软件开发者，拥有计算和信息系统学士学位，还是 Stack Overflow 的顶级答案贡献者之一。你会发现他的回答广泛涉及了 Web 开发的各种语言和技术。

审校者致谢

感谢本书作者 Alex 给我参与本书的机会，这是一次很棒的学习经历，我非常敬佩 Alex 为囊括如此多的知识到本书中所付出的努力。还要感谢我挚爱的妻子和儿子，感谢他们在我参与本书审校过程中给予的耐心和支持。

译者

常用昵称“大漠”，W3CPlus 创始人，目前就职于淘宝。对 HTML5、CSS 和 CSS 处理器等前端脚本语言有非常深入的认识和丰富的实践经验，专注于 CSS、CSS 处理器和 Web Animation 技术的研究，是国内最早研究和使用 CSS3 和 CSS 处理器技术的一批人。CSS3、Sass 和 Drupal 的中国布道者。2014 年出版著作《图解 CSS3：核心技术与案例实战》。

译者序

众所周之，任何一个 Web 页面或者 Web 应用程序都离不开 CSS。CSS 规范从最初的 CSS1 到现在的 CSS3，再到 CSS 规范的下一个版本，规范本身一直处于不断发展的演化之中。这给开发人员带来了效率上的提高。不过与其他 Web 领域的规范处境相似，CSS 规范在浏览器兼容性方面一直存在各种各样的问题。不同浏览器在 CSS 规范实现方面的进度也存在很大差异。另外 CSS 规范本身的发展速度与社区的期待还有一定的差距，这也是为什么 Sass、LESS 和 Stylus 等 CSS 处理器可以流行的重要原因。Sass、LESS 和 Stylus 等 CSS 处理器提供了很多更实用的功能，也体现了开发人员对 CSS 语言的需求。而本书介绍的 PostCSS 是目前 CSS 处理器中最流行的一个处理器。PostCSS 依托其强大的插件生态系统，为 CSS 处理器增加了无穷的可能性。

PostCSS 本身是一个功能比较单一的工具。它提供了一种使用 JavaScript 代码处理 CSS 的方式。它负责把 CSS 代码解析成抽象语法树结构，再交给插件进行处理。插件基于 CSS 代码的 AST 所能进行的操作是多种多样的，比如可以支持变量、混合宏、嵌套，增加浏览器相关的私有前缀，或是把符合未来的 CSS 规范的样式规则转译成当前 CSS 规范支持的格式。从这个角度来说，PostCSS 的强大之处在于其不断发展的插件体系或者说其插件生态。目前 PostCSS 已经有 200 多个功能各异的插件。开发人员也可以根据项目的需求，开发出自己的 PostCSS 插件。最为庆幸的是，来自全球各地的 PostCSS 插件开发人员根据自己项目的需求开发出不同的功能插件，并且将这些插件开源贡献给其他有需要的开发人员使用。

PostCSS 从诞生之时就给社区带来了对其类别进行划分的争议。这主要是由于其名称中的 Post，Post 很容易让人联想到 PostCSS 是用来做 CSS 后处理的，从而将其与已有的 CSS 处理（以前我常称之为 CSS 预处理器）语言，比如 Sass、Less 和 Stylus 等进行对比。实际

上，PostCSS 的主要功能只有两个：第一个功能是前面提到的把 CSS 解析成 JavaScript 可以操作的 AST，第二个功能是调用插件来处理 AST 并得到结果。因此，不能把 PostCSS 简单地归类成 CSS 预处理器或后处理器。PostCSS 所能执行的任务非常多，其同时涵盖了传统意义上的预处理和后处理。PostCSS 是一个全新的工具，给开发人员带来了不一样的处理 CSS 的方式。而且这种方式提高了 Web 开发人员编写 CSS 的效率，更降底了个人或者团队管理和维护 CSS 的成本，特别是针对于一个大型的项目，这种优势体现得更为明显。

如果你想真正了解 PostCSS 的话，你应该尽快学会 PostCSS 是什么以及如何使用它。

本书通过十四章的内容带你深入了解 PostCSS 以及如何使用 PostCSS。如果你还没有准备好去了解 PostCSS 能做什么，那么，请跟着这本书的步骤进行系统而深入的学习，你将进入到 CSS 的全新世界。

我们能够使用这么优秀的 CSS 处理器来编写、维护和管理 CSS，需要特别感谢@Andrey Sitnik，是他给我们创造了这么强大的处理器，并且让 PostCSS 以一个惊人的速度发展，越来越多的人开始了解它、使用它。

我们也要特别感谢@Alex Libby，是他花费了大量时间和精力为我们编写了一本深入浅出，带我们一步一步了解 PostCSS 的所有内容的优秀图书。如果你认真阅读完本书，你将能熟练地使用 PostCSS，借助 JavaScript 编写出自己想要的 PostCSS 插件，甚至还可以定制一个属于自己或团队的 CSS 处理器。

我在本书的翻译过程中得到了电子工业出版社的张春雨、田志远以及其他工作人员的帮助，在此一并表示由衷的感谢。

本书主要由我和南北、彦子和静子几位译者共同翻译。虽然我们经常参与社区前端技术文档的翻译，但翻译图书还是初次，因此全书难免存在一些错误或者不当之处，敬请广大读者批评指正。译者非常愿意通过微博（http://weibo.com/w3cplus）或电子邮件（w3cplus@hotmail.com）与各位同行探讨有关 PostCSS 或 CSS 处理器的相关技术问题。

大　漠

2017 年 6 月于杭州

前言

作为一名开发人员，我猜你有一个完美的工作流程。前端工程师可以使用传统的 CSS 书写样式，也可以使用当前某个处理器（如 Sass 或 Less）来实现它们，还可以使用类似 Autotoprefixer 的插件，手动或使用工具（如 Grunt 或 Gulp）来添加浏览器前缀。

听起来很熟悉？如果它适合你，为什么还要打破这些呢，对吧？

麻烦的是，某个朋友或同事已经开始谈论一个新的处理器 PostCSS。他们已经引起了你的兴趣，你想要获得更多关于 PostCSS 是什么、PostCSS 如何工作的信息。

欢迎来到快速增长的 PostCSS 生态系统！PostCSS 本身不做任何事情，但是它可以与数百个可用的插件配合，从而有潜力成为一个真正强大的处理器。开发者不得不依靠 Sass 这样的单一库的日子就要过去，取而代之的是，开发者可以根据项目的要求准确选择使用哪些插件。PostCSS 是一个非常快速的处理器，你准备好了吗？

相信答案是肯定的，让我们开始吧。

这本书涵盖了哪些内容

第 1 章，PostCSS 简介。本章将开启我们的 PostCSS 世界之旅，探索 PostCSS 的功能以及如何使用 PostCSS 生态系统将基本代码转换为可在项目中使用的、有效的 CSS 样式。你会发现使用这个生态系统的好处，它的架构和模块化的方法使我们能够组合一个定制的处理器专门针对我们的需求。

第 2 章，创建变量和混合宏。我们在本章可以看到一些现有处理器技术中常见的基本概念，如变量和混合宏。本章将介绍如何将其转换为 PostCSS，并了解从这些技术转换到

使用 PostCSS 可以获得的优势。

第 3 章，嵌套规则。探讨了现有处理器如 Sass 或 Less 如何利用嵌套等概念来减少我们需要编写的代码量，以及如何在 PostCSS 处理器中获得相同的功能。

第 4 章，创建媒体查询。本章介绍了使用 PostCSS 和媒体查询向网站添加响应式支持的基础知识。我们将学习如何改进对旧版网站和浏览器的支持，并探讨如何在 CSS4 媒体查询已经出现的情况下做更进一步的工作，并在 PostCSS 中提供支持。

第 5 章，管理颜色、图像和字体。本章介绍了可用于处理和操作 PostCSS 中的图像、颜色和字体的插件。我们将通过一些示例来说明如何在 PostCSS 中操纵图像和颜色，例如使用系统中的调色板创建图像精灵或更改颜色。

第 6 章，创建网格。通过使用网格构建网站的骨架，我们将探索使用网格的基本概念，并发现一些可用于创建网格的 PostCSS 插件。本章将介绍一些使用 Bourbon Neat 网格系统的例子，并在随后利用等效的 PostCSS 插件进行相同的实现，并为结果代码添加响应能力。

第 7 章，动画元素。首先，简要介绍如何使用 JavaScript 来添加动画，然后，介绍如何切换到使用 CSS 创建动画；最后，转换到使用 PostCSS。在使用 PostCSS 创建快速演示，并学习如何使用 PostCSS 优化动画之前，我们将探索使用一些更知名的库，如 Animate.css。

第 8 章，PostCSS 插件开发。本章我们将学习如何使用插件来扩展 PostCSS，并探索这种插件的典型架构。然后，我们将要看一些示例插件，并使用可用的样板代码创建自己的插件，接着测试并让互联网用户可以下载插件。

第 9 章，简写型插件、降级插件和包型插件。本章首先介绍了一些可用的简写型插件和包型插件，然后探索了如何使用自己的快捷键插件来补充它们。我们还将了解如何使用 PostCSS 提供的插件包来审查和优化代码，并学习如何为 PostCSS 代码提供候选方案以帮助维护对旧版浏览器的支持。

第 10 章，定制处理器。本章汇集了在前几章介绍过的一些技术，以生成一个定制处理器，我们可以将其用作我们项目中转换代码的基础。在添加源映射和浏览器前缀支持之前，你还将研究如何优化输出，然后在网站上进行测试。最后，我们将学习如何使用 CSStyle 框架扩展处理器，以编写适用于 Sass 和 PostCSS 的代码。

第 11 章，管理自定义语法。本章介绍了如何使用 API 编写自定义语法，并探讨了一些可用于对 Sass 或 LESS 语法编写的代码进行解析的方法。在将输出转换成可以在屏幕上显示或保存到文件中的内容之前，我们介绍了一些使用 PostCSS 解析代码的示例。我们还添加了使用 midas 库高亮显示代码的支持。

第 12 章，混合处理器。本章介绍了如何从混合处理器开始过渡到使用 PostCSS。在安

装并使用 PostCSS 的功能之前，先介绍了 Pleeease 库。然后，我们将在使用它之前设置一个编译过程，以便对标准的 WordPress 主题进行更改。

第 13 章，排除、解决 PostCSS 的相关问题。本章介绍了使用 PostCSS 时可能遇到的一些常见问题，例如“taskname not in our gulpfile”错误，同时介绍了遇到其他失败时应该如何处理。本章还将介绍如何在 PostCSS 核心系统或其插件中获取帮助或错误记录的详细信息。

第 14 章，为未来做准备。本章涵盖了从人们所知的 CSS4 中支持未来风格标准的一些可能的选择。我们还将探索其中包含的风险，以及如何使用现有的插件来获得同样的支持，或扩展它们以增加对新的 CSS4 选择器的支持。

阅读这本书需要的准备工作

想要运行本书中的大部分示例，你只需要做如下准备：安装一个简单的文本或代码编辑器，安装一个浏览器，安装适用于你的操作系统的 Node.js 环境，保持网络连接。本书建议安装 Sublime Text 3，它与 Node 和 Gulp 配合良好，我们将在整本书中使用它。

一些示例中使用的大多数的插件可以直接在 Node.js 中安装，详细信息包含在每章相应的部分，我们可以在对应的部分查看插件源代码和文档的链接。

这本书为谁而写

本书面向熟悉 HTML5 和 CSS3，同时希望掌握 PostCSS 作为简化开发流程、摆脱对现有处理器(如 Sass 或 Stylus)的依赖的前端开发人员。为了充分利用本书，你应该对 HTML，CSS 和 JavaScript 有很好的了解，如果有一些使用 Sass、LESS 或 Stylus 等预处理器的经验会更加理想。

本书约定

在这本书中，你会发现一些区分不同信息的文本样式。以下是这些样式的一些示例和其含义的解释。

文本、数据库表名、文件夹名称、文件名、文件扩展名、路径名、虚拟 URL、用户输入和 Twitter 句柄中的代码的字体样式如下。

“我们首先安装此演示所需的相关插件：我们需要使用 PostCSS 嵌套，自动完成

postcss-scss 插件。”

代码块的字体样式如下。

```
gulp.task('rename', ['styles'], function () {
  return gulp.src('dest/example.css')
    .pipe(postcss([ cssnano ]))
    .pipe(rename('example.min.css'))
    .pipe(gulp.dest("dest/"));
});
```

警告或重要提示出现在这样的框中。

提示和技巧如此出现。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- 下载资源：本书如提供示例代码及资源文件，均可在 下载资源 处下载。
- 提交勘误：您对书中内容的修改意见可在 提交勘误 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- 交流互动：在页面下方 读者评论 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：http://www.broadview.com.cn/31817

目录