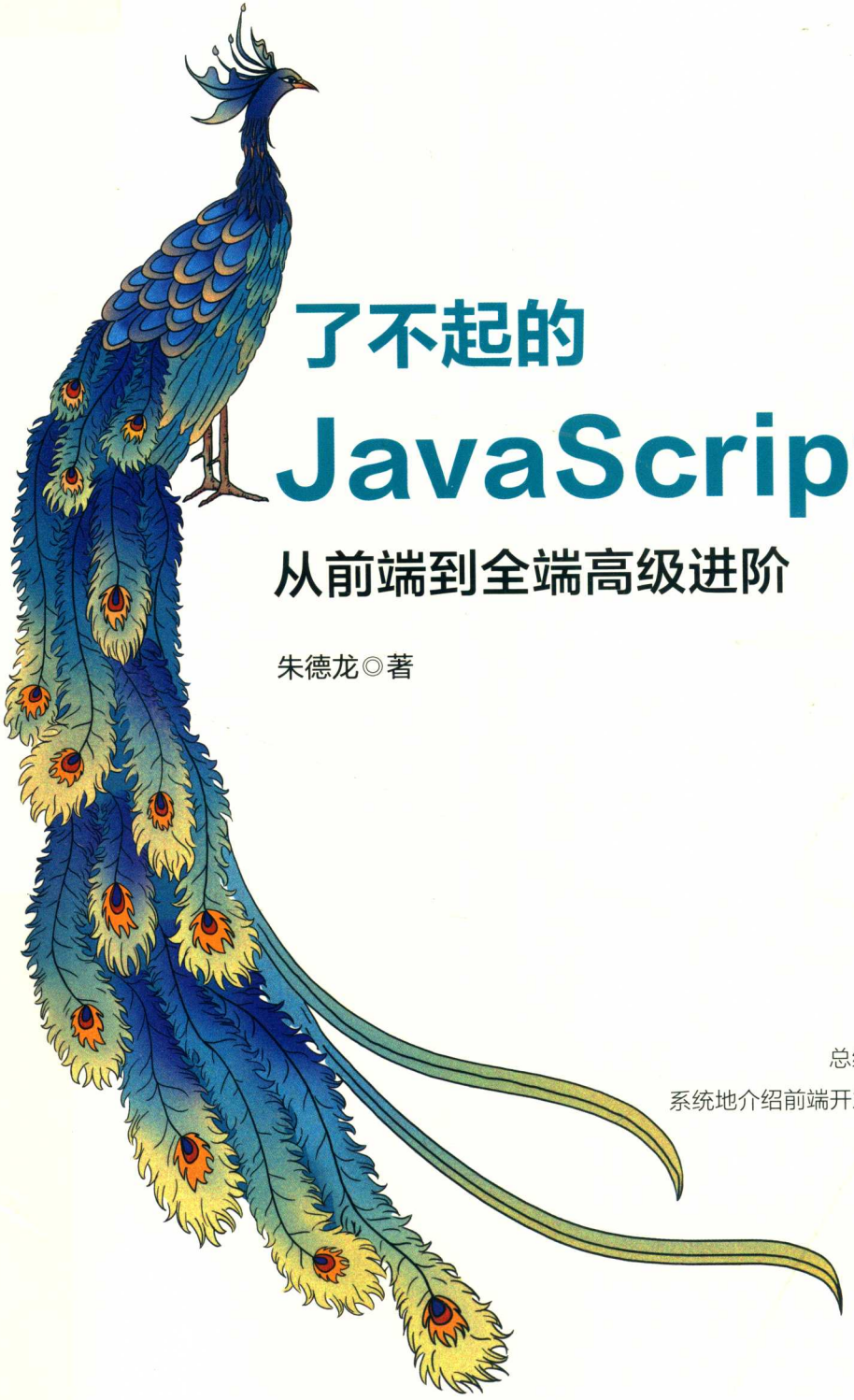
Broadview®
www.broadview.com.cn



了不起的 JavaScript 工程师

从前端到全端高级进阶

朱德龙◎著

总结多年丰富的前端开发经验
系统地介绍前端开发工作中需要的技能及工具
快速进入“全端”的境界

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

了不起的 JavaScript 工程师

从前端到全端高级进阶

朱德龙◎著

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

内 容 简 介

本书讲述了开发者使用 JavaScript 在各种 Web 开发场景下所需要掌握的重点知识和概念。从最基础的开发工具讲起，再到开发思维方式和前端页面开发，然后扩展到小程序开发和开发工具的混合应用，再讲解前后端交互最常用的网络协议及 API 设计，最后讲解了使用 Node.js 开发服务器端应用程序所需要掌握的核心概念。

全书以数据链为线索，对重要概念进行精练的分析和对比，从而帮助读者更好地理解和记忆。本书既包括知识技能，又包括设计思想，“道”与“术”并重，让读者“知行合一”，既能“坐而论道”，也能“起而行之”。

本书适合初级和中级前端开发者用来提升技术水平与视野，建立较为完备的开发技能和可迁移的学习能力，帮助读者拥有全端的技术和全局的视野，成为“了不起的 JavaScript 工程师”。

图书在版编目（CIP）数据

了不起的 JavaScript 工程师：从前端到全端高级进阶 / 朱德龙著. —北京：电子工业出版社，2019.8

ISBN 978-7-121-37129-5

Ⅰ. ①了… Ⅱ. ①朱… Ⅲ. ①JAVA 语言－程序设计 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字（2019）第 150110 号

责任编辑：董　英　　特约编辑：田学清
印　　刷：三河市双峰印刷装订有限公司
装　　订：三河市双峰印刷装订有限公司
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：21　　字数：504 千字
版　　次：2019 年 8 月第 1 版
印　　次：2019 年 8 月第 1 次印刷
印　　数：3000 册　　定价：79.00 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。

前　言

先给出下面三个问题：

- 如何快速成为职场老手，通过提升工作效率加快自己的成长速度？
- 前端技术框架层出不穷，该保持什么样的学习心态？
- 掌握 JavaScript 之后可以做哪些事情，有什么样的晋级路线？

术与道

对于第一个问题，要提升工作效率，大体上可分为两种途径：

- 改变开发习惯，选择好的软硬件环境；
- 积累经验，提升思维能力。

第一种途径是从“术”的层面，在微观、客观上进行提升，非常直接也非常有效，但这种途径也有缺陷，就是效率提升的空间有限。

本书的第 1 章便是教读者一些容易掌握的开发软硬件的知识及如何培养好的开发习惯，因为这些是能够在短时间内提升效率的方法。

第二种途径是从“道”的层面，宏观、主观地进行提升，提升速度缓慢，提升效果也很难快速直接地反映在工作中，但是提升空间可以说是无限的。

本书的第 3 章“高效编写/组织代码的心法”及第 4 章“模块”都属于此类，掌握这些基本原则和思路对前端知识甚至其他语言的学习都会有很大帮助。

JavaScript 是语言工具，是“术”；但是如果能在熟悉 JavaScript 之后进行总结归纳、建立自己的学习模型、通过模型来快速掌握其他高级语言，那么这就是“道”。

关于“术”与“道”的理解，大家可能会有以下两个误区。

1．“道”太难了，把“术”研究透了也能成为高手

一部分读者可能认为“量变引起质变”，只要“术”用得多了，自然能提升“道”。

“量变引起质变”其实是个半真命题，因为通常情况下引起质变是有条件的，这些条件既可能是主观创造的，也可能是客观存在的。

举个例子，比如一个电商网站，每天只有 1000 人使用，但是突然业务暴增到每天一亿人。如果不对系统架构进行调整可能就无法正常运行，这就是客观要求引起的质变。

再举个反例，比如一个开发者做了很多前端项目，“开发经验很丰富”，但是面试的时候很多问题却答不上来。网上的“你是具有 3 年开发经验，还是把 1 年开发经验重复了 3 遍”说的就是这种现象。

对于这种情况虽然有量变，但如果缺乏主动的归纳总结提升，很难引起质变。

有的读者可能会疑惑，如果这位开发者不断地使用新框架，比如从 jQuery 切换到 Vue 算不算是质变呢？这就好比一个人之前骑摩托车出行，现在改开汽车出行了，这算是生活的质变吗？

2.“术”不重要，只要潜心向“道”便能成为高手

这种理解也相当理想化。“术”与“道”的关系就好像实践与理论的关系，两者相辅相成。精通“术”之后可以明白“道”，懂了“道”之后又可以用于“术”。

那些理解了“道”的高手很多时候都是从“术”开始学习的。比如《神雕侠侣》中的独孤求败，就是不断地提升剑术来减少对剑本身的依赖的，最后掌握剑道，以至于“不滞于物，草木竹石均可为剑”。

本书一方面会讲解技术重点，比如 HTTP 协议、浏览器，同时也更加注重关于“道”的知识讲解，比如从单页应用开发讲到小程序开发、从 JavaScript 前端讲到其在 Node.js 后端的应用。通过不同知识点重要概念的抽象与对比，最终让读者形成可迁移的、适应性强的学习能力。

危与机

其实开始提出的第二个问题暗示了前端工程师的两个危机。

1. 职业天花板较低

前端工程师相对于算法工程师、数据库管理员等其他开发岗位来说，并不是一个有较高天花板的职位（不一定随着工作年限增加而不断提升个人能力和薪酬待遇）。相反，由于前端技术框架层出不穷，一般情况下，先入行的工程师和后入行的工程师相比，难以有较大的优势。用了几年 jQuery 的开发者和入职一年的初级开发者相比，学习使用 Vue 和 React 并不具有太多的优势。技术总监和 CTO 很少有前端工程师出身的。

2. 职业门槛较低

学会 HTML、CSS 这种静态语言就可以实现一个网页效果，JavaScript 上手难度也不算大，同时容错率较高，也就是说，如果代码出现 bug 通常表现在界面和交互上，一般不会对公司造成较大影响，很多控制台错误不容易被客户察觉到。所以，很多 IT 培训机构必开前端培训课程，除了市场需求，门槛低也是一个重要的因素。

那么在这种情况下前端工程师该怎么通过学习和积累来建立自己的优势呢?

至少可以从以下两个方面入手。

1. 提升学习能力

如前面提到的“术”与“道”，就能有效地提升学习能力与学习效率。有了强大的学习能力之后，很多新技术都能快速理解并上手。这样的开发者的能力可以说是无限的，自然也不存在天花板。

2. 多维度学习

本书以前端为起点，紧贴常用的 Web 技术知识（Vue、Angular、React、Node.js、HTTP 等），帮助大家成为能独立完成开发任务的“全端工程师”。全端提供的是另一种可能性，带你了解不同的开发技能点，通过多种技能的组合使你成为真正强大的开发者。

树与叶

我曾经参加过一场线下技术分享沙龙，有一场关于架构师的讨论，一位同学说到自己想成为前端架构师，然后罗列了一大堆技术名词及要学习的点，如看一看 Vue 的源代码、熟练掌握 Webpack 的使用等。这些点看上去都没错，也紧贴热点，但是这位同学以目前的状态可能很难达到目标。因为他没有形成自己的知识框架!

前端的知识点其实很多，也很细碎，对于开发者而言，并不需要把它们都记住，而是要形成自己的知识框架，然后不断地去丰富它。对于新手而言可能比较难，不一定能做到，但是心中一定要有这个意识。

以本书的写作线索为例。一条明线是围绕 JavaScript 这门语言进行编写的，讲述了 JavaScript 的主要使用场景及核心知识，一条暗线便是根据数据流来进行编写的，即从用户端到服务器端；除了这两条纵向的线索，还有一条横向的线索就是从前端到多端（小程序）扩展。这些就是知识框架的体现。

只有先形成框架才能有效地把繁杂的知识点内化成为自己的经验。用罗振宇的话来说就是“把新知识缝合到自己原有的知识体系中”。学无涯而生有涯，知识框架还能帮助我们对知识进行过滤，避免在一些无关的知识点上消耗过多时间。

致谢

个人能力再强，也毕竟是有限的。一个人的成就，往往离不开其他人的帮助与支持。所以本书能够出版要感谢很多人。

感谢家人对我写书的支持。

感谢董英编辑的信任和指点，以及其他编辑的细心校对。

感谢一直支持我的读者，这其中包括本书的读者及博客文章的读者。

感谢学习我慕课网课程学生的支持。

感谢对于我写书提出意见的朋友们。

感谢参考文章和资料的作者。

感谢所有曾经帮助我成长的人。

……

反侵权盗版声明

目 录