TURING 图灵原创

深入剖析Vue.js**源码**

刘博文◎著

深入浅出 Vue.js

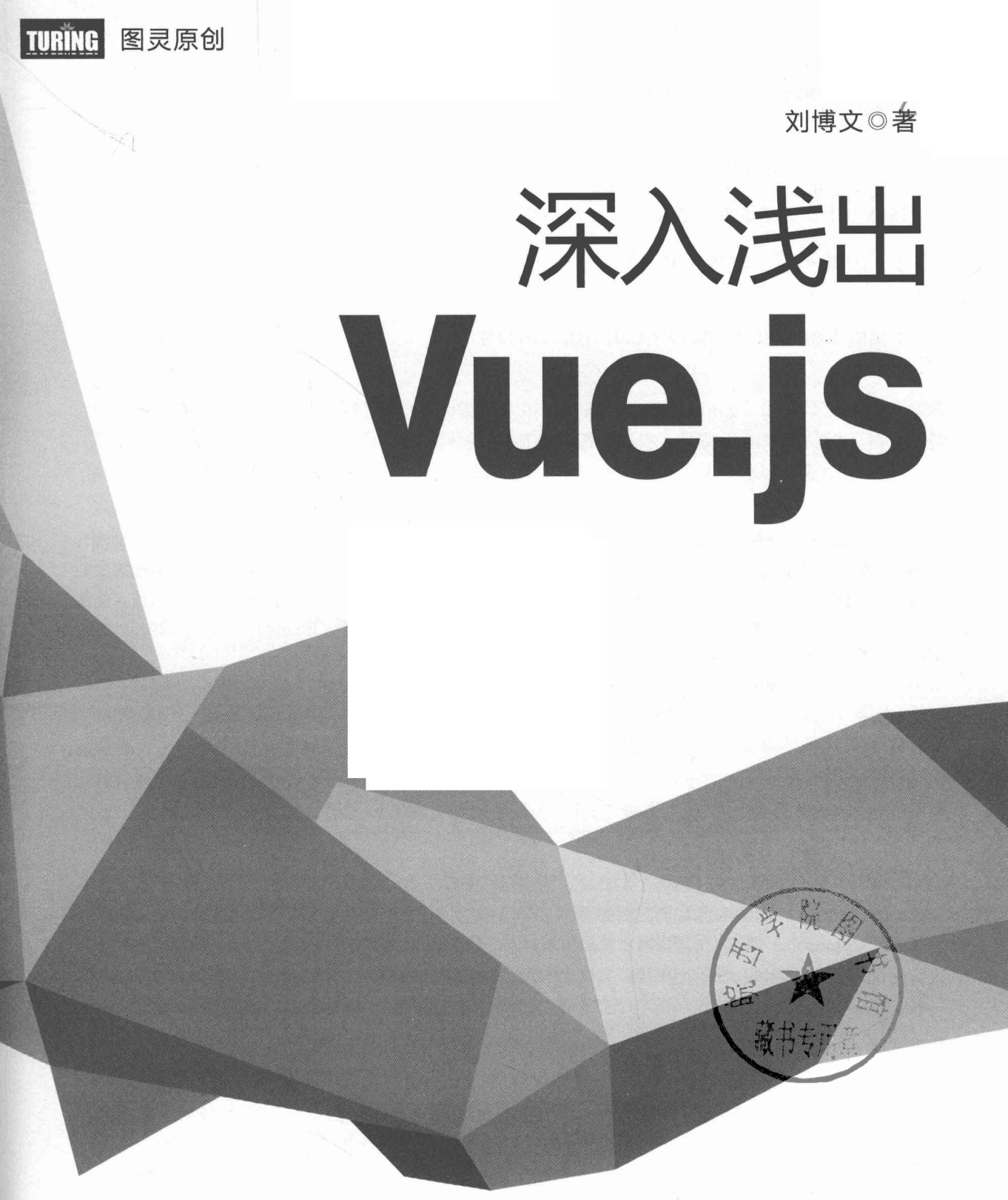
360奇舞团团长**月影**
《JavaScript高级程序设计》译者**李松峰** 作序推荐

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创

刘博文◎著

深入浅出 Vue.js



人民邮电出版社
北 京

图书在版编目（CIP）数据

深入浅出Vue.js / 刘博文著. -- 北京 : 人民邮电出版社, 2019.3（2019.5重印）
（图灵原创）
ISBN 978-7-115-50905-5

Ⅰ. ①深… Ⅱ. ①刘… Ⅲ. ①网页制作工具－程序设计 Ⅳ. ①TP393.092.2

中国版本图书馆CIP数据核字(2019)第037890号

内 容 提 要

本书从源码层面分析了Vue.js。首先，简要介绍了Vue.js；接着详细讲解了其内部核心技术“变化侦测”，并带领大家从0到1实现一个简单的“变化侦测”系统；然后详细介绍了虚拟DOM技术，其中包括虚拟DOM的原理及其patching算法；再后详细讨论了模板编译技术，其中包括模板解析器的实现原理、优化器的原理以及代码生成器的原理；最后详细介绍了其整体架构以及提供给我们使用的各种API的内部原理，同时还介绍了生命周期、错误处理、指令系统与模板过滤器等功能的原理。

本书适合前端开发人员阅读。

◆ 著　　刘博文
责任编辑　王军花
责任印制　周昇亮

◆ 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
涿州市京南印刷厂印刷

◆ 开本：800×1000　1/16
印张：18.5
字数：437千字　　2019年3月第1版
印数：5 501－7 000册　　2019年5月河北第3次印刷

定价：79.00元

读者服务热线：(010)51095183转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字20170147号

站在巨人的肩上

Standing on Shoulders of Giants

iTuring.cn

序　一

近几年，JavaScript 的流行库和框架带有元编程（metaprogramming）的特征。所谓元编程，简单来说，是指框架的作者使用一种编程语言固有的语言特性，创造出相对新的语言特性，使得最终使用者能够以新的语法和语义来构建他们的应用程序，从而在某些领域开发中获得更好的开发体验。

早期的 jQuery 库之所以获得开发者们的认可，很大程度上是因为它独创的链式语法和隐式迭代语义。尽管 jQuery 仅仅通过巧妙设计 API 就能支持上述特性，并不依赖于编程语言赋予的元编程能力，但是毫无疑问，它以一种精巧的设计理念和思路，为 JavaScript 库和框架的设计者打开了一扇创新的大门。

今天的 Web 产品对构建用户界面的要求越来越高，jQuery 的方式不能满足构建复杂用户界面的需要，新的 UI 框架快速发展，其中一个最流行的框架就是 Vue.js。与 jQuery 相比，Vue.js 更强大，也具有更加明显的元编程特征。动态绑定属性和变化侦测、内置模板和依赖于模板语法的声明式渲染、可扩展的指令、支持嵌套的组件，这些原生 JavaScript 并不具备的特征和能力被一一融入，框架的使用者在使用 Vue.js 开发 Web 应用时，事实上获得了超越 JavaScript 原生语言特性的能力。

尽管 Vue.js 框架赋予开发者众多特性和能力，但它仍然是使用原生 JavaScript 实现的应用框架。JavaScript 自身提供了许多元编程特性，比如从 ES5 就开始支持的属性访问器（property accessor），ES6 支持的代理（proxy），还有标准提案已经处于 Stage 3 阶段的装饰器（decorator）。基于这些语言特性，我们能够比较方便地扩展新的语言特性，将这些特性融入应用框架，从而使得应用开发者能够更加得心应手地使用框架开发出优雅、简洁的应用程序模块。

如何设计 API 和如何使用元编程思想将新特性融入到框架中，是现代 JavaScript 框架设计的两个核心，Vue.js 更侧重于后者。理解元编程思想有助于深刻理解 Vue.js 的本质。而理解元编程思想本身最好的方法又是通过深入研究 Vue.js 的源码，因为元编程思想一旦涉及具体实现，不仅仅是使用 JavaScirpt 提供的特性来扩展能力那么简单，这其中有许多细节需要考虑，比如要做到向下兼容，那么就要对一些特性的实现方式做出取舍，一些语言能力可以通过书写向下兼容代码来弥补，而另一些则需要通过编译机制来做到，还有一些则必须舍弃；同样，基于性能考虑，一些特性也可能需要做出一定的修改或妥协。这些问题不仅在框架设计和实现的过程中会遇到，而

且在具体实现应用程序的过程中也会遇到。因此，通过学习 Vue.js，我们不仅能够掌握设计应用程序框架的一般性技巧，还可以在实现应用程序时运用其中的具体设计思想和方法论。

本书的作者刘博文是我的同事，也是奇舞团的一员，后来由于业务变动，博文所在的团队从奇舞团独立了出去，但是同为 360 的前端团队，我们也始终保持着项目合作和技术交流。很早就听到博文要写这样一本书，当时我很高兴，我一直鼓励大家写书，因为这种创作既能使自己成长，又能使读者获益。我自己也写过技术类的书，深知技术创作的不易，要把 Vue.js 这样的流行框架讲透也着实需要下一番苦功。有时候，作为朋友，我会和博文开玩笑，说他的书再不出版，Vue.js 3.0 版本就要发布了，但这仅仅是玩笑，我不愿意博文因为要赶出版时间而草草了事，那样就无法真正做到“深入浅出”，毕竟这不是一本 Vue.js 的使用手册，而是真正能够透过 Vue.js 的设计思路去学习元编程思想，并将这种思想运用于程序开发中的书。只有这样，读者才能真正从这本书中获益。我想，在这一点上，博文没有让我失望，我也希望这本书没有让你们失望。

月影

360 奇舞团团长

2019 年 2 月 1 日

序　二

“奇舞团”办公地点在“南瓜屋”7层，导航前端在“南瓜屋”8层。2017年某一天，我去8层的时候路过导航前端工位，梁超看到我，高兴地说：“李老师，博文正在写书呢。”我脱口而出：“谁是博文，给哪个出版社写？”由此我便认识了博文，也知道了他是王军花（本书策划编辑）发掘的作者。当时听到这个消息我也很兴奋，知道是在给图灵写书，而我又在图灵待过几年，熟悉图灵的“套路”，就忍不住当场给博文分享了一些选题和写作思路。听着我滔滔不绝地讲“写书经”，博文频频点头，好像很受启发的样子。

2018年年初，360 W3C工作组成立，博文加入了Web性能工作组。于是几乎每周的例会上，我都会问问博文新书写作和出版的进度。时值年末，这本书终于要出版了。而这时候，我因为支持智能音箱项目临时搬到了11层，开发、联调非常繁忙。11月16日下午，博文突然在微信上问我能不能帮他写个序。我说：“你能不能先给我看看书稿？”然后博文把我加到了他GitHub的私有仓库。

两周来，我利用空暇时间大致浏览了一遍书稿。无奈时间紧迫，大部分章节来不及细读。一是因为公司项目开发进度必须保证，二是自己还有一个字体服务的项目在并行迭代。虽然大部分内容未曾细读，但仅就仔细读过的几章而言，着实让我受益匪浅。我想，等到手头的项目开发告一段落之后，一定要抽时间重新研读两遍。没错，这本书至少要读两遍以上。

浏览书稿的时候，我也在回忆第一次跟博文分享“写书经”的情景。当时我说，要想让技术书畅销，一是读者定位必须是新手，因为新手人数众多；二是要注重实用，书中的例子最好能立即照搬到项目上。然而，这本书的读者定位显然不是新手，而且书中的源码分析似乎也不能直接套用到项目上。其实这也是没办法的事，因为博文写这本书的初衷就是把自己研究Vue.js源码的心得分享出来。就Vue.js源码分析而言，这本书确确实实是非常棒的。反正我是爱不释手。

这本书取名“深入浅出”是名副其实的。因为它确实有相当的深度，而且语言真的浅显易懂。最重要的是，与其他源代码分析类的技术书连篇累牍地堆砌、照搬项目源代码的做法截然不同，这本书里很少看到超过一页的代码片段。所有代码片段明显都被作者精心筛选、编排过，而且层层递进，加上了“新增”“修改”之类的注释。再辅以明白浅显的文字和配图，原本隐晦、抽象、艰深的代码逻辑，瞬间变得明白易懂，让人不时有“原来如此”之叹，继而“拍手称快”！

毋庸置疑，Vue.js是一个优秀的前端框架。一个优秀的前端框架如果没有一本优秀的解读著

作，确实是一大缺憾。应该说，本书正是一本优秀的 Vue.js 源码解读专著。全书从一个新颖的“入口点”——“变化侦测”切入，逐步过渡到“虚拟 DOM”和“模板编译”，最后展开分析 Vue.js 的整体架构。如果想读懂这本书，读者不仅要有一些 Vue.js 的实际使用经验，而且还要有一些编译原理（比如 AST）相关的知识储备，这样才能更轻松地理解模板解析、优化与代码生成的原理。本书最后几章对 Vue.js 的实例方法和全局 API，以及生命周期、指令和过滤器的解读，虽然借鉴了 Vue.js 官方文档，但作者更注重实现原理的分析，弥补了文档的不足。

虽然本书不是写给新手看的，但鉴于 Vue.js 在国内的用户基数巨大，我对它的销量还是很乐观的。这些年来，前端行业一直在飞速发展。行业的进步，导致对从业人员的要求也不断攀升。放眼未来，虽然仅仅会用某些框架还可以找到工作，但仅仅满足于会用一定无法走得更远。随着越来越多“聪明又勤奋”的人加入前端行列，能否洞悉前沿框架的设计和实现将会成为高级人才与普通人才的“分水岭”。

“欲穷千里目，更上一层楼。”我衷心希望博文这本用心之作，能够帮助千千万万的 Vue.js 用户从“知其然”跃进到“知其所以然”的境界。最后想说一句，有心购买本书的读者大可不必纠结于 Vue.js 的版本问题。因为优秀源代码背后的思想是永恒的、普适的，跟版本没有任何关系。早一天读到，早一天受益，仅此而已。

李松峰
360 奇舞团高级前端开发工程师
前端 TC 委员、W3C AC 代表
《JavaScript 高级程序设计》译者
2018 年 12 月 2 日

前　言

时至今日，Vue.js 就像曾经的 jQuery，已经成为前端工程师必备的技能。不可否认，它可以极大地提高我们的开发效率，并且很容易学习。

这就造成了一个很普遍的现象，大部分前端工程师对框架以及第三方周边插件的关注程度越来越高，甚至把自己全部的关注点都放在了框架上。

在我看来，这多少有点亚健康，不是很利于前端工程师的技术成长。因为我发现大家关注框架时，更多的是关注其用法（包括框架自身、第三方插件和 UI 组件库等）、奇淫技巧和最佳实践等。

而我希望大家拿出一部分精力去关注框架所解决的问题以及它是如何解决这些问题的。这有助于我们提升自己的技术和解决问题的能力。

大家在使用 Vue.js 开发项目时，不免总会遇到一些奇奇怪怪的问题，而我们是否能很快解决这些问题以及理解这些问题为什么会发生，主要取决于对 Vue.js 原理的理解是否足够深入。

本书目的

所有技术解决方案的终极目标都是在解决问题，都是先有问题，然后有解决方案。解决方案可能并不完美，也可能有很多种。

Vue.js也是如此，它解决了什么问题？如何解决的？解决问题的同时都做了哪些权衡和取舍？

本书将带领大家透过现象看到 Vue.js 的本质，通过本书，我们将学会：

- Vue.js 的响应式原理，理解为什么修改数据视图会自动更新；
- 虚拟 DOM（Virtual DOM）的概念和原理；
- 模板编译原理，理解 Vue.js 的模板是如何生效的；
- Vue.js 整体架构设计与项目结构；
- 深入理解 Vue.js 的生命周期，不同的生命周期钩子之间有什么区别，不同的生命周期之间 Vue.js 内部到底发生了什么；
- Vue.js 提供的各种 API 的内部实现原理；
- 指令的实现原理；

- 过滤器的实现原理；
- 使用 Vue.js 开发项目的最佳实践。

组织结构

本书共分四篇，全方位讲解了 Vue.js 的内部原理。

- 第一篇：共 3 章，详细讲解了 Vue.js 内部核心技术“变化侦测”，并一步一步带领大家从 0 到 1 实现一个简单的“变化侦测”系统。
- 第二篇：共 3 章，详细介绍了虚拟 DOM 技术，其中包括虚拟 DOM 的原理及其 patching 算法。
- 第三篇：共 4 章，详细介绍了模板编译技术，其中包括模板解析器的实现原理、优化器的原理以及代码生成器的原理。
- 第四篇：这是本书占比最大的一部分，详细介绍了 Vue.js 的整体架构以及提供给我们使用的各种 API 的内部原理。同时还对 Vue.js 的生命周期、错误处理、指令系统与模板过滤器等功能的原理进行了介绍。在本书最后一章，我们为大家提供了一些使用 Vue.js 开发项目的最佳实践，这些内容中一大部分是 Vue.js 官网提供的，还有一小部分是我自己总结的。

在撰写本书时，Vue.js 的最新版本是 2.5.2，所以本书中的代码参考该版本进行撰写。如果你想对照源码来阅读本书，可以在 GitHub 上找出该版本的源码。此外，关于本书的任何意见和建议，都可以在这里讨论：https://github.com/berwin/Blog/issues/34。关于本书的微信群，也请参见这个页面。

致谢

这本书的诞生我要感谢很多人。我曾幻想过如果有一天自己能出版一本技术书，那该有多好，但从来没有想到这一天来得这么快，我更想象不到这一天会在我 23 岁时发生。在我看来，这件事不可能发生在我的身上，但它确确实实发生了。

这一切都要感谢王军花老师，是她给了我这个机会。最初她找到我，问我有没有兴趣写一本深入介绍 Vue.js 的书时，我的内心很挣扎。因为这可以实现我的一个梦想，但我又担心自己写不好，觉得自己不够资格出版一本书。最终经过激烈的思想斗争后，我决定接受这个挑战，做一些让自己佩服自己的事。

不止是感谢军花老师给我这个机会，我还非常感谢她前前后后跟进这本书，包括书的进度以及与我一起审校和修改这本书等很多事情，非常感谢！

其次我要感谢我的领导 LC（梁超）和肆爷（何烁），当他们听说我要写一本书时，给了我很大的帮助和支持。本书没有拖稿，我按时写完了所有章节，这一切都是源于他们对我的大力支

持。如果没有他们，我想我也没有办法按时交稿，非常感谢！

同时我也非常感谢李松峰老师，在开始写这本书时，李老师给我讲了很多写作方面的技巧，并且教我怎样写一本好书，怎么写出阅读体验良好的书。并且在这本书写完之后，李老师还答应给这本书写序，真的非常感谢！

我更要感谢我的父母，感谢你们对我多年的养育之恩，辛辛苦苦把我养大。如今，我虽有了一份稳定的工作，但回家的次数却越来越少。我很愧疚不能在你们身边工作，不能经常陪在你们身边。现在，我出版了一本书，不知道你们会不会为我感到骄傲。

我还要感谢堂姐王砚天，在写作期间给了我很多精神鼓励与支持，并且给我买了很多好吃的。

除此之外，我要感谢第一批内测读者（刘冰晶、姚向阳、周延博、王建兵、陈凤），感谢你们的阅读以及给我提供的宝贵修改意见，非常感谢！

最后，我要感谢正在阅读这部分的你，感谢你阅读本书，感谢你对我的支持，谢谢！

目　录

第 1 章 Vue.js 简介

在过去的 10 年时间里，网页变得更加动态化和强大了。通过 JavaScript，我们已经可以把很多传统的服务端代码放到浏览器中。身为一名前端工程师，我们所面临的需求变得越来越复杂。

当应用程序开始变复杂后，我们需要频繁操作 DOM。由于缺乏正规的组织形式，我们的代码变得非常难以维护。

这本质上是命令式操作 DOM 的问题，我们曾经用 jQuery 操作 DOM 写需求，但是当应用程序变复杂后，代码就像一坨意大利面一样，有点难以维护。我们无法继续使用命令式操作 DOM，所以 Vue.js 提供了声明式操作 DOM 的能力来解决这个问题。

通过描述状态和 DOM 之间的映射关系，就可以将状态渲染成 DOM 呈现在用户界面中，也就是渲染到网页上。

1.1 什么是 Vue.js

Vue.js，通常简称为 Vue，是一款友好的、多用途且高性能的 JavaScript 框架，能够帮助我们创建可维护性和可测试性更强的代码。它是目前所有主流框架中学习曲线最平缓的框架，非常容易上手，其官方文档也写得非常清晰、易懂。

它是一款渐进式的 JavaScript 框架。关于什么是渐进式，其实一开始我琢磨了好久，后来才弄懂，就是说如果你已经有一个现成的服务端应用，也就是非单页应用，可以将 Vue.js 作为该应用的一部分嵌入其中，带来更加丰富的交互体验。

如果希望将更多业务逻辑放到前端来实现，那么 Vue.js 的核心库及其生态系统也可以满足你的各种需求。和其他前端框架一样，Vue.js 允许你将一个网页分割成可复用的组件，每个组件都有自己的 HTML、CSS 和 JavaScript 来渲染网页中一个对应的位置。

如果要构建一个大型应用，就需要先搭建项目，配置一些开发环境等。Vue.js 提供了一个命令行工具，它让快速初始化一个真实的项目工程变得非常简单。

我们甚至可以使用 Vue.js 的单文件组件，它包含各自的 HTML、JavaScript 以及带作用域的 CSS 或 SCSS。我本人在使用 Vue.js 开发项目时，通常都会使用单文件组件。单文件组件真的是

一个非常棒的特性，它可以使项目架构变得非常清晰、可维护。

1.2　Vue.js 简史

2013 年 7 月 28 日，有一位名叫尤雨溪，英文名叫 Evan You 的人在 GitHub 上第一次为 Vue.js 提交代码。这是 Vue.js 的第一个提交（commit），但这时还不叫 Vue.js。从仓库的 package.json 文件可以看出，这时的名字叫作 Element，后来被更名为 Seed.js。

2013 年 12 月 7 日，尤雨溪在 GitHub 上发布了新版本 0.6.0，将项目正式改名为 Vue.js，并且把默认的指令前缀变成 v-。这一版本的发布，代表 Vue.js 正式问世。

2014 年 2 月 1 日，尤雨溪将 Vue.js 0.8 发布在了国外的 Hacker News 网站，这代表它首次公开发布。听尤雨溪说，当时被顶到了 Hacker News 的首页，在一周的时间内拿到了 615 个 GitHub 的 star，他特别兴奋。

从这之后，经过近两年的孵化，直到 2015 年 10 月 26 日这天，Vue.js 终于迎来了 1.0.0 版本的发布。我不知道当时尤雨溪的心情是什么样的，但从他发布版本时所带的格言可以看出，他心里一定很复杂。

那句话是：

> “The fate of destruction is also the joy of rebirth.”

翻译成中文是：

> 毁灭的命运，也是重生的喜悦。

并且为 1.0.0 这个版本配备了一个代号，叫新世纪福音战士（Evangelion），这是一部动画片的名字。事实上，Vue.js 每一次比较大的版本发布，都会配一个动画片的名称作为代号。

2016 年 10 月 1 日，这一天是祖国的生日，但同时也是 Vue.js 2.0 发布的日子。Vue.js 2.0 的代号叫攻壳机动队（Ghost in the Shell）。

同时，这一次尤雨溪发布这个版本时所带的格言是：

> “Your effort to remain what you are is what limits you.”

翻译成中文是：

> 保持本色的努力，也在限制你的发展。

在开发 Vue.js 的整个过程中，它的定位发生了变化，一开始的定位是：

> “Just a view layer library”

就是说，最早的 Vue.js 只做视图层，没有路由，没有状态管理，也没有官方的构建工具，只有一