

本文由 [简悦 SimpRead](#) 转码，原文地址 [kaiwu.lagou.com](#)

你好，我是黄轶（常用 ID：[ustbhuangyi](#)），现任 Zoom 前端架构师，曾先后于百度、滴滴从事前端研发工作。我平时喜欢钻研新技术、新框架，关注前端自动化、工程化、前端架构。和很多常年打磨自身编程能力的开发者一样，我对代码有洁癖，一直在努力追求高质量的代码。

## 为什么你要学习 Vue.js 源码？

前端技术日新月异的今天，前端应用的复杂度也在日益提升，熟练掌握一门 MVVM 前端开发框架已经成为必然要求，因为它能够很大程度上帮助前端开发者提高生产力。Vue.js、React 和 Angular 是目前国内最流行的三个前端 MVVM 框架，其中 Vue.js 凭借轻量、易上手的优势收获了大批粉丝。

百度、阿里、腾讯、滴滴、头条、美团等大厂已经在大面积使用 Vue.js 开发 Web 前端项目，很多中小型企业也因为易上手，开发效率高而选用 Vue.js。此外，很多小程序的跨端方案，例如 uni-app、Mpx、chameleon、WePY 等框架也选择了类 Vue.js 的语法。总体而言，**市场对于 Vue.js 人才的需求非常旺盛。**

但也正因为 Vue.js 上手门槛低，市场需求与人才现状间存在不少现实矛盾：

- 很多初学者通过简单的培训后便入行，但所学大多是 Demo 级别的项目知识，到了真实的工作环境中往往水土不服；
- 工作中只会简单地调用 API，而复杂的组件非常依赖开源的实现，如果找不到相关组件甚至难以完成开发需求；
- 没有深入研究过，或者根本不懂 Vue.js 底层实现原理，开发中遇到 Bug 后不懂得如何分析问题，也不懂如何调试；
- 工作中往往需要通过阅读源码去了解当前项目和一些第三方依赖库的实现方式和原理，但是简单的知识填充式的培训并不能教会这些，初学者也很难自己形成这样的能力。

初级开发人员已经很难满足当前市场需求，而高阶开发人员却显得供不应求。面试早已不只是考察你应用层面的掌握情况，面试官还喜欢考察技术背后的实现原理来判断你对技术的掌握程度，以及是否有对技术的钻研精神。如果你对于 Vue.js 的使用只是浮于表面，技术能力不过关，那你将很难在行业中立足。

以我多年的从业经历来看：

**了解技术实现原理是前端工作的必然要求，而看源码是了解技术实现原理的最直接手法，是高效提升个人技术能力的有效途径。**

此外，学习 Vue.js 源码还能够从更多层面提升你的技术实力：

**首先，有助于提升你的 JavaScript 功底。**Vue.js 源码底层是用纯原生 JavaScript 写的，你可以在阅读 Vue.js 源码的过程中学习很多 JavaScript 编程技巧。这种贴合实战的学习方式，比你天天抱着编程书看，效率要高得多。

**其次，提升工作效率，形成学习与成长的良性循环。**了解技术的底层实现原理，会让你在工作中对它的应用更加游刃有余，在遇到问题后可以快速定位并分析解决。这样你的工作效率就会大大提升，帮你省出更多的时间来学习和提升。

**再次，借鉴优秀源码的经验，学习高手思路。**你可以通过阅读优秀的项目源码，了解高手是如何组织代码的，了解一些算法思想和设计模式的应用，甚至培养“造轮子”的能力。实际上，Vue.js 3.0 的设计实现中就参考了很多优秀的开源 JavaScript 库。

**最后，提升自己解读源码的能力。**读源码本身是很好的学习方式，一旦你掌握了看源码的技巧，未来学习其他框架也会容易得多。而且，工作中也可以通过阅读项目已有代码快速熟悉项目，提高业务逻辑分析能力和重构代码的能力。

## 道理我都懂，就是做不到？

学习源码有这么多好处，很多人也明白这个道理，为什么却很少有人愿意去读源码呢？

- 因为学习源码很枯燥，不像开发项目那样能够快速得到反馈、看到立竿见影的效果；
- 学习源码相对于开发项目来说更抽象，理解起来也更难，很多人学着学着就放弃了；
- 还有很多人想要更深入地学习 Vue.js，希望能够再进阶一个高度，却不得法门。

这正是我设计这个课程的原因之一。**我希望结合自己多年研究源码和 Vue.js 实践经验，并结合一些在实际项目中的使用场景，来带你一起阅读源码，深入浅出地帮助你了解其技术实现原理。**

我曾经使用 Vue.js 重构整个滴滴出行的 WebApp，负责其中的架构设计和组件库开发，也主导过 Vue.js 开源组件库 cube-ui 的开发。我也在 Zoom 工作期间为了配合安全组 CSP 安全策略需求，通过直接魔改 Vue.js 源码的方式，开发了 Vue.js 2.x 的 CSP 兼容版本，该版本目前在 Zoom 内部运行稳定，服务于几十个用 Vue.js 做增强开发的页面。

此外，我平时喜欢写作和分享，曾经帮助很多人入门和进阶 Vue.js。在以往分享经验和对用户的答疑解惑过程中，我更加直观地感受到了 Vue.js 学习者的困惑之处，也懂得了如何才能帮助你更好地学习源码。

所以只要你能认真跟随我学习源码，你会发现原本枯燥的事情也许会变得有趣起来，随着你不断深入地理解 Vue.js 的实现，你也会越来越有成就感，学习的动力也就越来越强了。

## 课程设计

**我会对 Vue.js 3.0 的源码进行透彻分析，但不会一味地去解释源码，而是更加注重解读 Vue.js 在实现某个 feature 的时候，它的设计思想是什么以及为什么会这么做。**相比单纯解释源码这种“翻译”的工作，我更喜欢做“阅读理解”，把每部分源码的前因后果分析清楚。

课程共分三大模块，合计 22 篇文章。我会结合实际用例，循序渐进地带你深入 Vue.js 的内核实现。

- **核心模块**，我会带你分析 **Vue.js 3.0 组件的实现原理、响应式原理**，以及 **Vue.js 3.0 新特性 Composition API 的实现原理**。因为组件化一直都是 Vue.js 最核心的实现内容，Composition API 也是 Vue.js 3.0 非常亮眼的 API 设计，所以我会优先讲这两块内容。经过学习，你会对组件如何渲染和更新有深刻的理解，并且掌握 Composition API 背后的实现原理和应用场景。
- **进阶模块**，我会带你分析 **Vue.js 3.0 模板的编译和优化过程**。Vue.js 3.0 运行时的性能之所以有很大的提升，主要得益于其编译层面的优化，所以这部分内容是非常值得学习的，但由于它的难度较大，所以我把它设置成了进阶阶段。学完之后，你能够知道 Vue.js 是如何编译模板并生成代码的，以及编译过程背后的性能优化思想。
- **扩展模块**，前面你已经了解了 Vue.js 的核心实现和编译原理，那么接下来我会带你分析 **Vue.js 3.0 的内置组件的实现原理、Vue.js 3.0 一些实用特性的实现原理，以及 Vue.js 3.0 官方生态实现原理**，这些内容非常贴合实际开发工作。学完之后，你会更加了解这些功能的实现原理和职责边界，在平时工作中应用起来更加得心应手。

# 《Vue.js 3.0 核心源码解析》课程大纲

开篇词 | 解析 Vue.js 源码，提升编码能力

导读 | 一文看懂 Vue.js 3.0 的优化

## 模块一：直击 Vue.js 核心组件的实现

1 组件渲染：vnode 到真实 DOM 是如何转变的？

2 组件更新：完整的 DOM diff 流程是怎样的？（上）

3 组件更新：完整的 DOM diff 流程是怎样的？（下）

## 模块二：学会新设计 Composition API

4 Setup: 组件渲染前的初始化过程是怎样的？

5 响应式：响应式内部的实现原理是怎样的？（上）

6 响应式：响应式内部的实现原理是怎样的？（下）

7 计算属性：计算属性比普通函数好在哪里？

8 侦听器：侦听器的实现原理和使用场景是什么？（上）

9 侦听器：侦听器的实现原理和使用场景是什么？（下）

10 生命周期：各个生命周期的执行时机和应用场景是怎样的？

11 依赖注入：子孙组件如何共享数据？

### 模块三：编译过程和背后的优化思想

12 模板解析：构造 AST 的完整流程是怎样的？

13 AST 转换：AST 节点内部做了哪些转换？

14 生成代码：AST 如何生成可运行的代码？

### 模块四：探索更多实用特性背后的实现原理

15 Props：Props 的初始化和更新流程是怎样的？

16 插槽：如何实现内容分发？

17 指令：指令完整的生命周期是怎样的？

18 v-model：双向绑定到底是怎么实现的？

### 模块五：学习 Vue 内置组件的实现原理

19 Teleport 组件：如何脱离当前组件渲染子组件？

20 Suspence 组件：如何优雅地实现组件异步处理流程？

21 KeepAlive 组件：如何让组件在内存中缓存和调度？

22 Transition 组件：过渡动画的实现原理是怎样的？

## 特别放送：研究 Vue 官方生态的实现原理

### 13 Vue Router：如何实现一个前端路由？

### 24 Vuex：如何实现前端的状态管理？

@拉勾教育

当然，你的其他一些担忧，我也提前为你想到了：

Vue.js 源码是一直在更新维护的，课程中的一些代码片段可能会更新，但代码容易过时，思想并不会，所以相较于代码，我会更注重思想的解读，让你知其然也知其所以然；Vue.js 版本更新也会引入一些实用的新功能，届时我也会紧随其后对新功能做解读，并且更新我们这个线上课程，以便你能够学习到新的知识点；为了便于没有 TypeScript 经验的同学理解，我会尽量将编译后的 JavaScript 代码展示出来，并且通过注释说明代码的主要功能；我还会尽量精简代码的分支逻辑，方便你理解核心流程；结合图例帮助你理解一些晦涩难懂的代码功能；结合实际用例，让你可以更加直观地明白源码背后想要解决的实际场景问题。

## 总结

我在百度工作的时候需要写编译打包工具，于是期间我阅读了 FIS 和 Gulp 的源码；到了滴滴以后，我使用了 Vue.js 开发项目，就开始阅读 Vue.js 的源码；开源库 better-scroll，也是在我充分阅读 iScroll 源码的基础上重构并一点点优化出来的。通过不断学习源码，我逐渐搞懂了这些工具框架背后的设计思想，学习到很多优秀的编程技巧，大幅提升了我的学习效率和技术能力，让我受益匪浅。

因此，**这门课我不仅希望帮你深入理解 Vue.js，更希望带你提升读源码的能力，提升技术实力。**

学习源码的过程就像在翻越一座座大山，但我会一直陪伴在你身边，做你坚强的后盾。学习的过程中，你可能会遇到一些问题，但是不要担心，你可以随时在评论区留言和提问，我会尽量抽出时间来认真解答你的提问。

准备好，让我们一起来感受 Vue.js 3.0 的美吧。