# **360资深前端刘博文：代码无终点，坚持跑，就不会输 // 图灵访谈**

[](https://www.zhihu.com/people/turingbooks)

**[图灵君](https://www.zhihu.com/people/turingbooks" \t "https://zhuanlan.zhihu.com/p/_blank)**

计算机/科技新知/Python/数据分析 好书推荐

已关注

25 人赞同了该文章

2012年，他17岁，中专毕业。  
2015年，他20岁，加入国内知名互联网公司360，成为360最大前端团队奇舞团的一员。现任360导航事业部资深前端工程师。  
2019年，他24岁，出版首部技术书籍《深入浅出Vue.js》。  
目前负责360导航首页及二级页创新项目等亿级PV站点的设计与优化，推动Vue.js成为部门内广泛使用的核心技术栈，独立研发相关开发工具与技术解决方案并使之成功落地。  
从中专毕业的门外汉，到360前端工程师并出版技术书籍，他完成了职业生涯的巨大跨越。他是怎样自主学习，快速成长的？  
**本期图灵访谈对话刘博文，一起来了解他的前端之路。**

## **北京的起点，从8000块钱开始**

「虽然也很努力，但我觉得更多的是靠运气。」

17岁那年我中专毕业，学的计算机。但当时的中专就是告诉我什么是计算机，让我知道有个职业可以用计算机去工作。

刚毕业的时候，我到沈阳的一家企业去上班，不会JS，CSS也只是懂一点。面试的时候，他们可能不知道我水平，觉得我还能干活，就让我去了。结果入职后发现，我啥也不会啊，又不好意思直接让我走。就给了我两个选择，一个就是接着干，当学徒，学徒就没有工资；还有一个就是别在这干了。我想了想觉得自己确实挺菜，那就跟着学吧。同学都说我傻，不给钱还给人干活。现在回头看，是我运气好。如果没有这个当学徒的机会，我就没办法踏入这个行业。

2013年，我18岁，想来北京，那时家里人有一点担心。他们是不支持我来的，但还是给了我5000块钱，他们原本是想等我把钱花完了就自己回来了，总不能在那饿死。但他们肯定想不到，我就用这5000块钱，加上自己攒的，一共8000块钱，一直坚持到现在。

当时我都算好了这8000块钱怎么花，首先留出要在北京租房子的钱，押一付三，一个月房租按1500算，一共需要6000。我拎包来北京找工作，刚开始住一个小旅店，每天按200块钱算。也就是说我要在10天内找到工作并租个房子。

当时也是命好，真的找到一个特别小的公司，让我去上班。我知道自己水平，能找到工作还挺高兴的，先能养活自己就行。而且公司最好的一点就是供住，员工可以每个月花200块钱住在宿舍里面。这就直接解决了我的生存问题，也是我在北京迈开的第一步。



工作了一段时间后，我还是只会写页面，切图，而且忙到没有时间学习。这样下去肯定不行，我就换了一个比较轻松的公司，这样就有了时间去学 JS。后来学到一定阶段的时候，我遇到了瓶颈，因为学的东西没办法在工作中用上。

我就又换了一个公司，这家公司只用一种语言就是 JS，服务端是用 Node.js 写的，比较符合我的需要。我可以充分去实践和提升技术，而且工作任务很繁重，那是我成长最快的一个阶段。

再后来，就到了现在的公司360。360给我的感觉像学校，整体工作氛围是比较轻松和自由的，任务不会把人压到一点时间都没有，我们有充分的时间自己去学东西。而且像月影、成银、李松峰老师和屈屈这种大牛会经常在公司内部讲课，有什么不懂的还可以去请教。

## **与Vue.js 的渊源！**

初识Vue.js时，它还未被众人认可。想不到5年后，为它写了一本书。

我接触 Vue.js 比较早，大概是2014年。因为工作中接手了一个同事的项目，是用 Vue.js 写的。当时它是零点几的版本，还没有正式的一点零版本。我简单了解了一下，发现它和 Angular 1 很像，挺轻，挺优雅的。需求都能满足，学习成本还不高。

但当时用 Vue.js 的人比较少，大家都没怎么听说过。它不火到什么程度呢？我们组新来了一个人，跟我一起写项目，我说项目是用 Vue.js 写的，就给他看了一下代码，讲了一下项目。然后，第二天他就离职了。

有半年的时间吧，我都在想是不是因为我们组的这个项目，用 Vue.js 他觉得太 low 了，所以不想干了？直到后来 Vue.js 被大家广泛认可，我才打消了这个想法。

我刚入职360的时候，我们组的项目都运行了很长时间，很稳定。一次偶然的机会，我们打算新开发一个后台管理系统，大家开会讨论技术选型。就业务来讲，我认为使用框架和对应的组件库会极大降低开发成本，就强烈推荐使用 Vue.js 技术栈，因为考虑到学习成本比较低，而且我对 Vue.js 比较熟。

现在大家已经习惯了使用框架开发，但在当时，我的提议遭到了非常强烈的反对。大家不停地提出各种问题，我也不停地给出解决方案，会议室现场变成了辩论会。

最终，我的 leader 给了我一个机会，如果想使用 Vue.js，就要在短时间解决两个最重要的问题，登录和部署流程。因为公司的统一登陆中心是结合后端来实现的，纯单页静态项目就意味着之前的登陆完全不能用了。部署流程也需要全新的解决方案。

当时我还有其他任务，所以就只能利用下班和周末的时间去做，好在最终问题解决了。这就是我们组正式使用 Vue.js 技术栈的时间点，也算是在后台项目中的一次试水。后来，我们组开发一个新产品，是面向C端的项目。在技术选型时，我又一次强烈推荐 Vue.js。因为上一个项目，有一些同学已经熟悉了 Vue.js 的开发模式，这次我也解决了一些遇到的问题（由于产品是图文内容类，存在 SEO 问题，等等）。就是这个项目真正推动 Vue.js 成为了我们的核心技术栈。

对 Vue.js 越来越熟悉，我在博客上陆续发布了一些梳理它内部原理的文章，作为总结和记录。2018年，王军花老师看到了我的博客，找到了我，问我有没有兴趣写本书。当时我感觉很突然，这东西我写不了啊。但心里边是很想写的，只是担心写不好，内心很挣扎。一天后，我和军花老师说可以试一下。

可能跟性格有关系，我从来都不是等把一切都准备好了，再去做一件事。一般都是机会来了，先干着再说。中间有问题再去解决问题。

当时我给自己定了的目标是6月份交稿。我列了一个大纲，然后倒推，以一个月为一个节点。写作过程中，每个节点的进度可能会比预期的快或慢，但总体都在可控的范围内。

写作时间就是午休和下了班之后，一天差不多要写两三个小时。其实后期也会觉得枯燥，没有灵感。那时候就会开始怀疑到底值不值得，这件事真的这么重要吗？我是不是用这些时间做其他事更划算？但没有真正想过放弃，就这样坚持下来了。



其实在写作过程中，我也有很多意外收获：我对一些 API 原理的细节理解得更深入了。

举个例子，我发现 Vue.js 对函数报错这方面做了很严谨的处理。当我们使用 Vue.js 开发项目时，编写的所有代码都是 Vue.js 调用并执行的，所以它在执行用户的代码时，做了错误的捕获处理。

还有就是计算属性。一个函数，可以返回计算后的结果。它要实现一个很重要的功能，就是当计算属性所依赖的某个状态发生变化时，计算属性的返回结果也需要做相应的变化，这个我之前确实没想过 Vue.js 是怎么做到的。

**工程师赛道上决定输赢的，只有两个字叫「坚持」**

「我自己特别在意的事，多苦多累都要把它干完。」

一直以来，驱动我做事情的都是「我想」，而不是外界的期望。

这种性格有优点，也有缺点。**比如说上学的时候，我学习不好，倒不是因为笨，而是因为我当时觉得，学习的结果就是分高分低一点，而这个分数什么用都没有。优点就是我自己特别在意的事，多苦多累都要把它干完。包括来北京弄这个计算机。**

我觉得我的职业生涯，更像是一条没有终点的赛道。而且这条赛道是不公平的，大家不是在同一条起跑线上开枪往前跑。当我刚开始跑的时候，可能别人已经跑了五年了。别人跑了好几万米了，我才刚开始第一米。

好多人想问我怎么才能跑得更快，把这场比赛跑赢。其实没有任何方法和经验可以让谁跑得更快。即使在短期内快一些，但在这条没有终点的赛道上，没有任何意义。大部分人跑到中途就主动放弃了，这就是为什么大牛那么少。唯一能决定这场比赛输赢的，只有两个字叫坚持。在这条赛道上跑赢的，不是那些跑得快的人，而是为数不多坚持跑的人。他们能跑赢，只是因为他们还在跑。



读代码其实是一种能力，可以锻炼。你一开始可能读不懂比较复杂的源码，可以读像 Underscore 那种简单一点的工具函数，重要的是训练你的大脑。经常阅读代码的人，理解力会逐渐上升。如果你不经常看代码，一段就研究老半天。尤其是框架，不是一段代码，是一坨，你直接就蒙了。

**全栈工程师可以这样养成**

对于习惯计划与记录的人，时间的脉络变得清晰和可控。

我每年都会给自己定目标，应该在哪个技术方向上深入一些，然后把相关的经典的书买下来，看一看。平时也会读一些所谓没有用的书来调节一下，比如哲学类和心理学的书。

几年前，我发现一个人很难把所有东西都学会。如果漫无目的地去学，很多东西看完之后就忘了。我会挑比较感兴趣的领域去研究，这个领域中的所有问题都看一下，但是对于其他领域的比较深入的知识，可能就先放一放，以后再去研究。

如果自己有目标的话，哪些东西是没看的，哪些东西是应该看的，其实心里都有数。

平时做事我有一个小技巧，是使用番茄工作法。不是用作秒表，到点了就停。对我来说，它是统计的工具。比如说以一周，一个月为周期，记录我每天有效的专注时间是多长。据我统计，我每天专注的时间很短，也就两三个小时，差不多四五个番茄钟。

一旦得到了这个信息，我就可以规划，每天把专注的时间用于哪些重要的事。不重要的，或者一些不太需要脑力的工作，都可以往后放。

除了这种大目标，一年中我还会定几次小目标。比如说一个月或两个月，坚持做一件事。每天我都会为它分配一部分专注的时间，持续下去，直到把这件事做完。

**可能是我的性格原因，没有办法同时做很多事。我更适合一次只做一件事。**



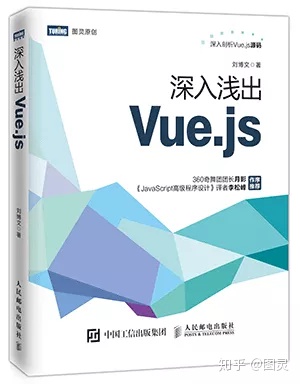
不管是写本书，还是平时自己学习，我觉得做一件大事要比无数小事要好。前几年，我就是学得很杂，看什么火学什么，没事就看论坛的各种文章。过了一年，都不知道自己看了啥，完全记不住。就像一个漏斗，我细碎的时间全部漏下去，什么都留不住。我应该把我的时块变大，才能卡在这。

未来我想成为一名真正的工程师，而不只是前端工程师，打算涉猎计算机其他领域。现在前端一些颠覆式的工具和创新，比如 Webpack，Babel，都不是一个纯粹的前端工程师能创造的。好多超大型的项目，都需要前端后端综合的解决方案。如果只做前端，只能写个 JS 的工具函数，仅此而已，解决不了真正的复杂场景下的问题。

生活方面的话，我比较喜欢旅游，每年都会计划出去玩两趟，未来还是会出去多玩一玩。我喜欢去自己没去过的地方，看一看，接受一些新鲜的东西。要是有缘分的话，找个女朋友。

世上并没有偶然，如果一个人务必要得到什么，并最终得到了，这就不是偶然，而是他自己的功劳，他的意愿将他领向了那里。——赫尔曼黑塞

图片：刘博文友情提供  
采访：乐馨，李冰  
撰文：李冰



深入讲解Vue.js实现原理

[《深入浅出Vue.js》](https://link.zhihu.com/?target=https://item.jd.com/12573168.html?dist=jd" \t "https://zhuanlan.zhihu.com/p/_blank)

作者：刘博文

360前端工程师精心打造

从源码层面深入讲解Vue.js实现原理和思想

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

作者首先简要介绍了Vue.js；然后详细讲解了其内部核心技术“变化侦测”，这里会带领大家从0到1实现一个简单的“变化侦测”系统；接着详细介绍虚拟DOM技术，其中包括虚拟DOM的原理及其patching算法；紧接着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。

推荐阅读[《不夸张，这真的是本Vue.js宝藏书！360前端工程师Vue.js源码解析》](https://zhuanlan.zhihu.com/p/61324025)

试读：[《解析器》](https://link.zhihu.com/?target=http://www.ituring.com.cn/book/tupubarticle/25715" \t "https://zhuanlan.zhihu.com/p/_blank)

目录

第1章　Vue.js简介

第一篇　变化侦测

第2章　Object的变化侦测

第3章　Array的变化侦测

第4章　变化侦测相关的API实现原理

第二篇　虚拟DOM

第5章　虚拟DOM简介

第6章　VNode

第7章　patch

第三篇　模板编译原理

第8章　模板编译

[第9章　解析器](https://link.zhihu.com/?target=http://www.ituring.com.cn/book/tupubarticle/25715" \t "https://zhuanlan.zhihu.com/p/_blank)

第10章　优化器

第11章　代码生成器

第四篇　整体流程

第12章　架构设计与项目结构

第13章　实例方法与全局API的实现原理

第14章　生命周期

第15章　指令的奥秘

第16章　过滤器的奥秘