



前端开发核心知识进阶：50 讲从夯实基础到突破瓶颈

来自 Lucas ... · 盐选专栏

[查看详情 >](#)

## 结束语：大话社区和一个程序员的自我修养

在西班牙语中，有一个很特别的词语叫做「Sobremesa」，它专指「吃完饭后，大家在饭桌上意犹未尽交谈的那段短暂而美好时光」。我们的专栏也已经全部更新完毕，历经「枯燥的程序知识」，我希望在最后部分，让我们放松心情，我将会从两方面谈一谈「软素质」，作为全部专栏内容的结束语。

### 融入社区

我们每一个人作为全球开发者的一份子，如果能够参与到社区，这对于个人成长会很有帮助的。这其中一个非常重要的话题就是「Github 使用礼仪」。大家可能比较熟悉了 Github 中使用 star、fork、watch 等基本功能。如果想深度参与社区并成为开源库的贡献者，一般有两种方式：

提 issue

提 pull request

首先，提 issue 一般有两种情况：报 bug 和提需求。如果是报 bug，最重要的是确认 bug 和表达清楚复现方式。对于一些较为复杂的复现场景，我们可以写一个 demo 帮助维护者发现问题；如果是提需求，我们就要尽量把控需求的合理性，这一点对于天天和产品「撕逼」的我们来说，应该不是太大问题。

提 pull request，就是申请往主库中合并代码。这其中涉及到一些 Git 的基本操作，主要流程是先 fork 目标仓库，进行修改之后进行推送，最后进入目标项目页面，发起 PR。

当然，我们不仅仅可以给开源库贡献代码，也可以作为创始者，向社区贡献内容。这个环节中，「如何写好一个现代库」，这个问题涉及到了方方面面的知

识。比如你要思考：

证书如何设置

文档如何设计，让使用者快速上手

TODO 和 CHANGELOG 需要遵循哪些规范

如何设计构建流程

如何设计编译范围和流程

如何设计模块化方案和打包流程

如何设计自动规范化链路

如何保证版本规范和 commit 规范

如何进行测试设计

如何引入可持续集成

其他最佳实践

为此，我给大家推荐我作为核心开发者的一个开源项目：[Jslib-base](#)，这是一个「为了写库而写的库」，它可以帮助开发者通过简单的命令就能创建一个库的脚手架和基础代码。整个项目实现非常简单，但尽可能结合「最佳实践」。如果你想写一个库，那我建议你考虑使用它来开启第一步；如果你想了解如何从零设计一个项目，也许可以通过它收获启发。给大家推荐相关文章：

### 8102 年如何写一个现代的 JavaScript 库

如果有机会，我会专门介绍一下这个库的编写和设计思想，虽然这不复杂，但是我们从能够延伸出来的问题都很有趣。

## 自我修养

论「程序员」的修养，这个话题非常开放。我们能想到很多关键词，比如：

保持热情

谦虚谨慎

学会阅读

学会提问

善用搜索

学会写作（文档/博客等）

「科学上网」

时间管理

知识管理

英语学习

我个人很不喜欢所谓的「成功学」和「方法论」，更讨厌「制造焦虑」，「兜售」鸡汤。免入俗套，我打算从两种动物来说一下「废话」。

不管是学习进阶之路，还是工作中的项目，我们能够遇到的真正问题只有两类：第一种是**看不见的**，我把它比作为黑天鹅，总会在你意想不到的时间和地点出现，并彻底颠覆一切；第二种是被我们**视而不见的**，我把它比喻成灰犀牛：你知道且习惯于它的存在，但是它会在某个时刻突然爆发，一旦爆发就会席卷一切，无从抵抗。

项目开发和个人成长都有黑天鹅和灰犀牛的危机。

黑天鹅

「新技术的爆发，技术的更新换代」就是职业生涯的黑天鹅。但我们需要辩证地来认识他：对于菜鸟来说，新技术和未知领域让年轻人有机会弯道超车，减少因为欠缺经验和阅历而带来的劣势；对于有一定工作经验和阅历的程序员来说，「颠覆」和「变革」这样的词语似乎不那么友好。

但是新技术说到底也只是工具，而真正资深程序员的核心价值在于：逻辑、分析、数据、算法等抽象能力。技术工具只是这些抽象能力的表述形式。从汇编语言转到 C 语言，其实更能发挥 C 的强大控制能力；从 C 转到 Java，只需要理解面向对象和虚拟机就能很快适应并脱颖而出；从 Java 转到 Python 的程序员，甚至都会感叹写代码「太简单了」！

总之，黑天鹅即是危机，也是机会。新技术作为新工具，总能带来新的价值蓝海。如果能把黑天鹅当做机会，保持敏感、好奇和进取的心态，扩展技能树，就能驯服来势汹汹的新技术。我们所有人一起共勉。

## 灰犀牛

社会中，很多职业是越老越值钱：老警察、老医生、老艺术家，说起来就让人觉得技术高超，令人信赖。

职业进阶就是一只灰犀牛。在悄悄溜走的时间中，我们可能习惯了日复一日的重复劳动。程序员怕的不是变老，而是变老的同时没有变强。如何击退这只灰犀牛，这需要我们从天天接触的工作代码入手，从熟悉的事物出发，找到突破。

比如，在这个专栏的工程化章节中，我重点突出了：如何增强程序的健硕性、如何让我们的开发效率提升、如何持续不断地完善项目、如何从零开始打磨基础构建体系。仔细思考，里面的内容也许就能接入你的项目当中。

从机械的工作抽象出更完美的工程化流程，这样的话题似乎永远说不完。我也总有新的心得和体会想和大家一起分享、交流。专栏已完结，但是衷心希望我们的技术探险之旅，仅仅是拉开帷幕。

## 最后

站在跑道的起点，你不知道跑到哪里肌肉会开始发痛，呼吸急促，想要停下来休息；在二三十岁的年纪，我们无从得知学习了一份课程，能对自己的水平提高和职业发展起多大作用。但我记得，在开篇词中，提到了村上春树的《当我谈跑步

时，我谈些什么》这本书：也许不论是跑步还是写代码，都是在探索生命的种种可能。

—— 不去跑，永远不知道能跑多远；不去做，永远不知道能做好。