

# 前方的路，让我们结伴同行

## 总结

首先感谢各位购买这本小册。这是我的第一本小册，内容可能会存在瑕疵，感谢大家选择这本小册，选择相信我这个作者。

相信大家都看过一句话：面试造火箭工作拧螺丝。诚然，现在大公司的门槛确实高，但这也是因为僧多粥少，供大于求的问题，因此大公司需要择优而录。那么为了进入大公司（我相信大部分人都有这个想法），我们势必需要驱动自己去不断学习，探索更深入的领域。而不是故步自封，原地踏步，认为会使用框架 API 能干活就行了。

面试时的信心源自于我们面试前的充分准备和平时的积累。有了信心，才能在面试中披荆斩棘，无往不胜。而不是毫无依据的自我感觉良好，面试失败就怪考官刁难、考题太难，责怪于外部因素而不是寻找个人的不足。

我也不说互联网寒不寒冬，毕竟物价、房价摆在那里，想做的好，赚的钱多，只能好好学，好好干，无论春夏秋冬。很感谢看到最后一章节的各位，相信这本小册能给大家带来一份不小的收获。

## 展望未来

小册不是我们学习的终点，为了减少大家找寻学习资料的时间，接下来我会提供一些我认为不错的学习资料供大家参考。

当然了资料是一部分，其实我更推荐在工作中学习。深入学习工作中用到的技术栈并且储备一些未来可能用到的技能，这样对个人职业发展是很有帮助的。

## JS

- [You-Dont-Know-JS \(https://github.com/getify/You-Dont-Know-JS\)](https://github.com/getify/You-Dont-Know-JS), 这套书深入的讲解很多 JS 的内容, 英文版是开源免费阅读的, 如果你英文不好的话, 国内这套书已经有出版了, 可以选择购买。
- [Functional-Light-JS \(https://github.com/getify/Functional-Light-JS\)](https://github.com/getify/Functional-Light-JS), 这本书是讲解函数式编程的, 函数式编程也是一种编程范式, 轻量级的函数式可以很方便的解决很多问题, 有兴趣的可以一读。
- [33-js-concepts \(https://github.com/leonardomso/33-js-concepts\)](https://github.com/leonardomso/33-js-concepts), 这份资料讲解了 33 个前端开发必须知道的 JS 概念, 内容是英文的, 如果你英文不好的话, 可以寻找这份资料的中文版。
- [前端精读周刊 \(https://github.com/dt-fe/weekly\)](https://github.com/dt-fe/weekly), 这是一份前端好文集合, 每周都会更新, 目前已经更新了 84 篇文章。
- [前端性能清单 \(https://github.com/JohnsenZhou/Front-End-Performance-Checklist\)](https://github.com/JohnsenZhou/Front-End-Performance-Checklist), 这是一份前端性能清单, 如果你需要优化一个项目的话, 可以根据这份清单一个个来检查优化项。
- [30-seconds-of-code \(https://github.com/30-seconds-of-code/30-seconds-of-code\)](https://github.com/30-seconds-of-code/30-seconds-of-code), 30 秒系列, 很短的代码片段让你了解一个知识点。
- [must-watch-javascript \(https://github.com/AllThingsSmitty/must-watch-javascript\)](https://github.com/AllThingsSmitty/must-watch-javascript), 这份资料包含了很多高质量的前端相关视频, 值得一看。

## CSS

- [css-protips \(https://github.com/AllThingsSmitty/css-protips\)](https://github.com/AllThingsSmitty/css-protips), 通过这份资料你可以学习到很多 tips 来提高你的

CSS 技能。

- [30-seconds-of-css \(https://github.com/30-seconds/30-seconds-of-css\)](https://github.com/30-seconds/30-seconds-of-css), 30 秒系列, 很短的代码片段让你了解一个知识点
- [CSS 世界 \(https://item.jd.com/12262251.html\)](https://item.jd.com/12262251.html), 张鑫旭出版的书籍, 没什么好说的了, 看就是了。
- [一些有趣的 CSS 话题 \(https://github.com/chokcoco/iCSS\)](https://github.com/chokcoco/iCSS), CSS 奇技淫巧, 在这里, 都有。

## 框架

框架这里其实我不想推荐任何的资料, 如果你单纯想学习一个框架的话, 我只推荐阅读官方文档学习, 没有任何的必要去学习其他的入门资料, 因为基本上都是照搬文档的。

如果你想进一步学习框架的内容的话, 我推荐去阅读框架核心团队成员的博客。比如 React 核心团队成员 Dan Abramov 的 [blog \(https://overreacted.io/\)](https://overreacted.io/)。

## Node

Node.js 几乎是资深前端工程师躲不过去的一道坎, 也是一个团队的通用底层能力, 学习 Node 可以更好的使用工具, 建立起一套数据中间层服务于整个团队。

- [Node.js 调试指南 \(https://github.com/nswbmw/node-in-debugging\)](https://github.com/nswbmw/node-in-debugging), 这是一本专注于讲解 Node 调试的书籍, 已经出版了, 但是可以开源免费阅读。
- [Node.js: 来一打 C++ 扩展 \(https://item.jd.com/12380404.html\)](https://item.jd.com/12380404.html), 死月出版的书籍, 没什么好说的, 看就是了。
- [Node.js 最佳实践](#)

<https://github.com/i0natan/nodebestpractices/blob/master/README.md>  
这是对 Node.js 最佳实践中排名最高的内容的总结和分享

## 安全

- [the-book-of-secret-knowledge](https://github.com/trimstray/the-book-of-secret-knowledge)  
(<https://github.com/trimstray/the-book-of-secret-knowledge>), 这是一份安全领域的资料, 如果你对安全感兴趣的话, 可以阅读一下内容。

## 周报

- [奇舞周刊 \(https://weekly.75team.com/\)](https://weekly.75team.com/), 每周都会整理一份不错的中文文章合集。
- [TechBridge Weekly \(https://weekly.techbridge.cc/\)](https://weekly.techbridge.cc/), 这是一份台湾地区整理的一份多个技术领域的周报。
- [JavaScript Weekly \(https://javascriptweekly.com/\)](https://javascriptweekly.com/), 这是一份相当有名气的英文周报, 整理的文章质量都很高, 如果你只想订阅一份周报, 那就是它了。
- [Pony Foo Weekly \(https://ponyfoo.com/weekly\)](https://ponyfoo.com/weekly), 这也是一份不错的英文周报, 文章质量也很高, 并且和上一份周报重叠的内容不多。

## Medium

Medium 上我并没有怎么固定阅读, 更多的是订阅它的日报或者从别的周报上看到的 Medium 的文章, 但是如果一定要推荐两个组织的话, 我只推荐这两个, 毕竟他们的文章质量都很高。

- [freecodecamp \(https://medium.freecodecamp.org/\)](https://medium.freecodecamp.org/)
- [hackernoon \(https://hackernoon.com/\)](https://hackernoon.com/)

## Youtube

Youtube 有很多高质量的视频，但是门槛大家都知道，这里我推荐一些值得订阅的频道。

- [JSConf](https://www.youtube.com/channel/UCzoVCacndDCfGDz0iA)  
(<https://www.youtube.com/channel/UCzoVCacndDCfGDz0iA>), 很多会议的视频你都可以在这里找到。
- [Google Chrome Developers](https://www.youtube.com/channel/UCnUYZLuoy1rq1aV)  
(<https://www.youtube.com/channel/UCnUYZLuoy1rq1aV>) Google 金字招牌，没啥好说的。
- [Computerphile](https://www.youtube.com/channel/UC9-y-6csu5WGm29I7JiwpnA)  
(<https://www.youtube.com/channel/UC9-y-6csu5WGm29I7JiwpnA>), 内容偏向于计算机领域。
- [Coding Tech](https://www.youtube.com/channel/UCtxCXg-UvSnTKPOzLH4wJaQ/videos)  
(<https://www.youtube.com/channel/UCtxCXg-UvSnTKPOzLH4wJaQ/videos>), 内容偏向于入门。
- [Fun Fun Function](https://www.youtube.com/channel/UCO1cgjhGzsSYb1r)  
(<https://www.youtube.com/channel/UCO1cgjhGzsSYb1r>) 如果你想学习函数式编程的一些内容，这是一个值得订阅的频道。
- [DevTips](https://www.youtube.com/user/DevTipsForDesigners/v)  
(<https://www.youtube.com/user/DevTipsForDesigners/v>) 每周更新一个视频，能够学习到不少开发中的 Tips。

## 其他

- [互联网公司技术架构](https://github.com/davideuler/architecture.of.internet-product)  
(<https://github.com/davideuler/architecture.of.internet-product>), 这份资料介绍了当下互联网公司的一个技术架构。
- [javascript-algorithms](https://github.com/trekhleb/javascript-algorithms)  
(<https://github.com/trekhleb/javascript-algorithms>), 这份资料作者使用了 JS 来实现了大部分的数据结构和算法。
- [小型编译器](https://github.com/jamiebuilds/the-super-tiny-compiler) (<https://github.com/jamiebuilds/the-super-tiny-compiler>), 这份资料告诉了我们该如何去实现一个小型

的编译器，很适合前端开发者阅读。

- [every-programmer-should-know](https://github.com/mtdvio/every-programmer-should-know) (<https://github.com/mtdvio/every-programmer-should-know>)，这份资料列举了很多每个开发者都应该知道的知识点。

## 最后

你可能发现我推荐的很多内容都是英文的或者你并不能打开，这里我只能说一声抱歉。因为我获取学习资料更多的来源于国外，可能不能很好照顾到英文不好的同学。但是说一句肺腑之言吧，技术需求的英文真的要求不高，花点时间静下心来去阅读英文资料，坚持个几个月，从此技术的大门就完完整整的打开了。

如果大家也有不错的资料想要分享，欢迎在留言区留言。

在小册的最后一章，打一个自己公众号的广告，如果你想了解到一些前端的热点、新知识、我的学习感悟等等，你可以关注我的公众号「前端真好玩」。



最后的最后，真的很感谢购买我小册的朋友，同时也感谢一些业内的大佬花时间阅读我的小册，并给我提出了修改意见，是你们促使我一直写作下去的。我们，下本小册见！