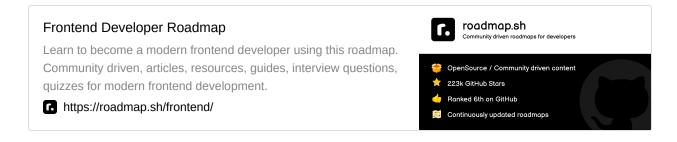


# 前端开发者路径

参见 Roadmap ,每年会有一定程度的更新。



# 推荐的学习网站

1. Youtube 偏向于视频学习的话,Youtube 上能够找到大量的学习教程,很多也都且纳为一些系列,且档到一个视频列表中,十分方便。Youtube 上种类很多,前后端框架、技术点、工具等等,都能够找到相应的一些学习视频,当前也有很多是中文的教程,不过一般情况下,中文的教程都会比英文稍微慢一步。所以,学习编程的话,建议有良好的英文读写能力,会让你事半功倍的。

### YouTube

Enjoy the videos and music you love, upload original content, and share it all with friends, family, and the world on YouTube.



https://www.youtube.com/

2. <u>Freecodecamp</u> 国外免费的职业生涯学习站点,按照不同的技术分类,Step by step 的进行技术学习,所见即所得的学习方式,学习通过后还可以申请认证,获取证书。

# freeCodeCamp.org Learn to Code - For Free freeCodeCamp(A) https://www.freecodecamp.org/

3. <u>Medium</u> 喜欢文章式学习话,一定不要错过,medium 根据账号的偏好设置,推送给你一些新鲜的教程,建议学习者,在上面找一些连载类型的教程,跟着教程实现一遍编码,帮助理解的同时,也能够提升自己的编码能力。

### Medium - Where good ideas find you.

Medium is an open platform where readers find dynamic thinking, and where expert and undiscovered voices can share their writing on any topic.

https://medium.com/



4. <u>Dev.to</u> 另一个博客类的站点是 Dev.to,相比 medium,优点是没有每月两篇的阅读限制,缺点是相关搜索做的比较差,有时候所搜的内容不尽如人意。



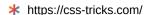
nttps://dev.to/



5. <u>CSSTricks</u> 如果你是CSS的爱好者,那你一定不要错过这个网站,最为前沿的最为精妙的 CSS 技巧,帮助你编写绚丽且易用的样式,为网站页面增加一抹色彩。

### **CSS-Tricks**

You know, this is the time of year where Chris normally publishes a big ol' reflection of the past year. The first one was published in 2007, the same year CSS-Tricks began, and it continued all the way





6. <u>Canluse</u> 当你工作中遇到了浏览器兼容性问题时,不妨来这个网站上检查下,你使用的属性的支持情况,然后针对性的使用一些 shim 包、polyfill 包来解决你的问题。

### Can I use... Support tables for HTML5, CSS3, etc

Test a feature Our partnership with BrowserStack now lets you test your website for compatibility across 2,000+ real browsers and devices. Test on: Did you know? Next If a feature you're looking for is not available on the site, you can vote to have it included.

https://caniuse.com/

7. <u>Product Hunt</u> 如果你成长到了更高的阶段,你不仅要有业务视角,也要格局打开,多 多关注一些最新出的产品,也许其中一个就是你所在公司的对手产品,多了解取长补 短,也能为公司产品做出更大的贡献。

### Product Hunt - The best new products in tech.

Looking for the perfect gift for that special code wrangler in your life? Unfortunately, we can't provide an Iron Man-esque workspace, but we do have some more slightly less exciting but still great gift



https://www.producthunt.com/

# 多点实践

新人总要以模仿和造轮子居多。先学,才能创造。有一些可以做的项目:

- 设计系统
- 组件库
- CLI 提效工具
- 封装属于自己公司的脚手架
- .....

我更偏向于基础设施的建设,因为太多人都会关注在使用工具的层面,如果我们没有创造工具的能力,请优先去理解工具,定不可养成拿来就用的习惯。在万物互通的网络时代,咨询和知识的获取是十分廉价的。只学用法的习惯,会渐渐的使自己失去竞争力。提升自己的不可替代性,永远是一个好的观念。

## 一些建议

我常常和我的同事聊起我写代码的习惯时,我说,如果我要写的页面模块,可以直接 CV 然后稍微改改就可以使用,那我也绝不会真的 CV,我还是会重新敲一遍代码,哪怕开双 屏,左右对照着抄一遍,你可不要小看抄一遍,抄的过程,不仅是理清思绪的过程,也能够学习前人代码中使用的技巧,掌握构建一个前端页面的思路。

对于前端新人,在学习到一定阶段时候,尤其当学习完一些框架如 React 之后,这个时候在公司里写业务太多的话,能够提升的也只能是写业务的能力和熟练度,这相当于从一个青涩工,转变为熟练工,但是当你离开了这家公司之后,你到了一个新的领域,面对新的业务,你又会变成一个新人了。

所以,最好是先想明白,自己的职业之路要怎么走。不管在哪里,你都应该经历入门、初级前端、中级前端、高级前端、项目负责人、研发负责人、技术总监等等。如果你在第一家公司,从初级成长到了中级,然后由于个人心境导致你不再愿意待在公司,当你出现这种"任性"想法的时候,首先要想的你,下一家公司能够继续从中级开始,还是说仍然要从初级起步。想明白这个问题,你才能不断地进步,而不是待在舒适区里划水。