

# 如何进入大厂

@被删

# 别想了 没有这样的必胜之法

醒一醒，所谓“保证能进入大厂”，都是骗人(钱)的！！



# 如何进入大厂 如何胜任一份工作

@被删

1

首先要通过面试

# 一、完善自己的简历

## 1. 简单明了，重点突出

### 工作经历

- 2017.10 - 至今 XXXX 公司 前端开发工程师
- 负责 XXX 系统的技术选型、前端开发、自动化测试
- 参与需求讨论确定开发流程
- 开发组内工具应用:如使用 cpp 开发视频转码拼接工具
- 2016.07 - 2017.10 XXX 公司 前端开发工程师
- 参与 XXXXX 系统开发
- 开发浏览器插件播放视频:使用 firebreath 开发 NPAPI 插件

XXX

- 男 25 岁 广州 | XXXX 大学本科 2 年工作经验
- 156xxxxxxxx | xxxxxx@gmail.com | <https://github.com/xxxx>

# 一、完善自己的简历

## 2. 项目的亮点 (业务 + 开源)

### 项目经历

#### ■ 2017.10 - 至今 XXX 系统 前端开发工程师

项目简介:

- 研究人员可以上传图片 and 音视频文件到这个系统, 然后创建多种标注任务: 给图片、视频分类;
- 给图片上的每一个像素点指定标签(有笔刷、矩形、多边形等工具辅助);
- 把音频转换成波形图后, 拖拽创建分段, 指定标签或者任意文字;
- 渲染 obj/ply 三维模型文件, 标注上面点的位置。
- 主要涉及技术栈: React, TypeScript, RxJS, AntD, pixi.js, wavesurfer.js, babylon.js

#### ■ 2016.07 - 2017.10 XXX 项目 前端开发工程师

项目介绍:

- 一个人脸监控系统, 用户可以上传若干人脸, 然后链接到摄像头或者是离线视频, 系统发现视频里面有疑似人脸时会自动提醒用户。
- 主要涉及技术栈: Angularjs, TypeScript, Vue.js, React.js

- 项目的特色和特点
- 项目中的挑战
- 自己在项目中的角色
- 涉及的技术栈

# 一、完善自己的简历

## 3. 个人特色

### 自我评价

- 老 TypeScript 爱好者，三年前开始就使用 TypeScript，也为它贡献了一点微小的力量（此处为一个 Typescript 的 Pull Request 链接地址）
- 对使用到的开源项目(tslint, uglifyjs, vscodevim)也提了一些 pull request（此处贴上了相关的链接地址）
- 会用 C++，以及 Qt 框架，会灵活运用 C++11 smart pointer, right value referenc
- 熟悉 git commit merge rebase 等常用操作
- 英语优秀，乐于助人，曾活跃于 StackOverflow（此处贴上了 StackOverflow 的链接）
- 对函数式、响应式编程有所涉猎

- 开源贡献
- 开源项目的参与
- 技术栈特点
- 技术偏好
- 个人爱好

## 二、面试题目网上一大堆

1. 多刷题，找面经
2. 理解并让之成为自身的知识
3. 面试后的总结和复盘，将每个不懂的点完全搞明白
4. 失败乃成功之母



### 面试这件事 ▾

- 1.面试准备
- 2.面试流程
- 3.专业知识概述
- 4.项目经验概述
- 5.其他内容概述
- 6.Javascript相关



找到工作之后  
就可以放飞自我了吗？

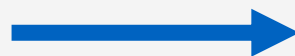
2

开发必备能力

# 一、解决问题的能力

加入一个团队，是为了替团队解决问题的

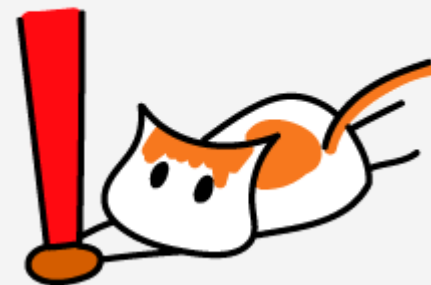
1. 发现问题
2. 找到解决方案
3. 解决问题



1. 搜索的能力
2. 项目经验
3. 思考方式


# 没有什么 是搜索引擎解决不了的


如果有，那就是搜索的方式不对!!!





## 二、提升个人能力

技术能力的提升，在于不断地发现和解决项目中的问题

- 性能优化：

```
graph LR; A[找到性能瓶颈  
(耗时/资源占用)] --> B[分析/解决]; B --> C[优化]
```
- 监控搭建：

```
graph LR; A[埋点] --> B[上报]; B --> C[监控]; C --> D[告警]; D --> E[排查]
```
- 多人协作：

```
graph LR; A[代码规范] --- B[+]; B --- C[流程规范]; C --- D[+]; D --- E[代码质量]; E --- F[+]; F --- G[自动化工具]
```
- 效能提升：

```
graph LR; A[持续集成] --- B[+]; B --- C[持续部署]; C --- D[+]; D --- E[定时发布]
```
- 架构设计

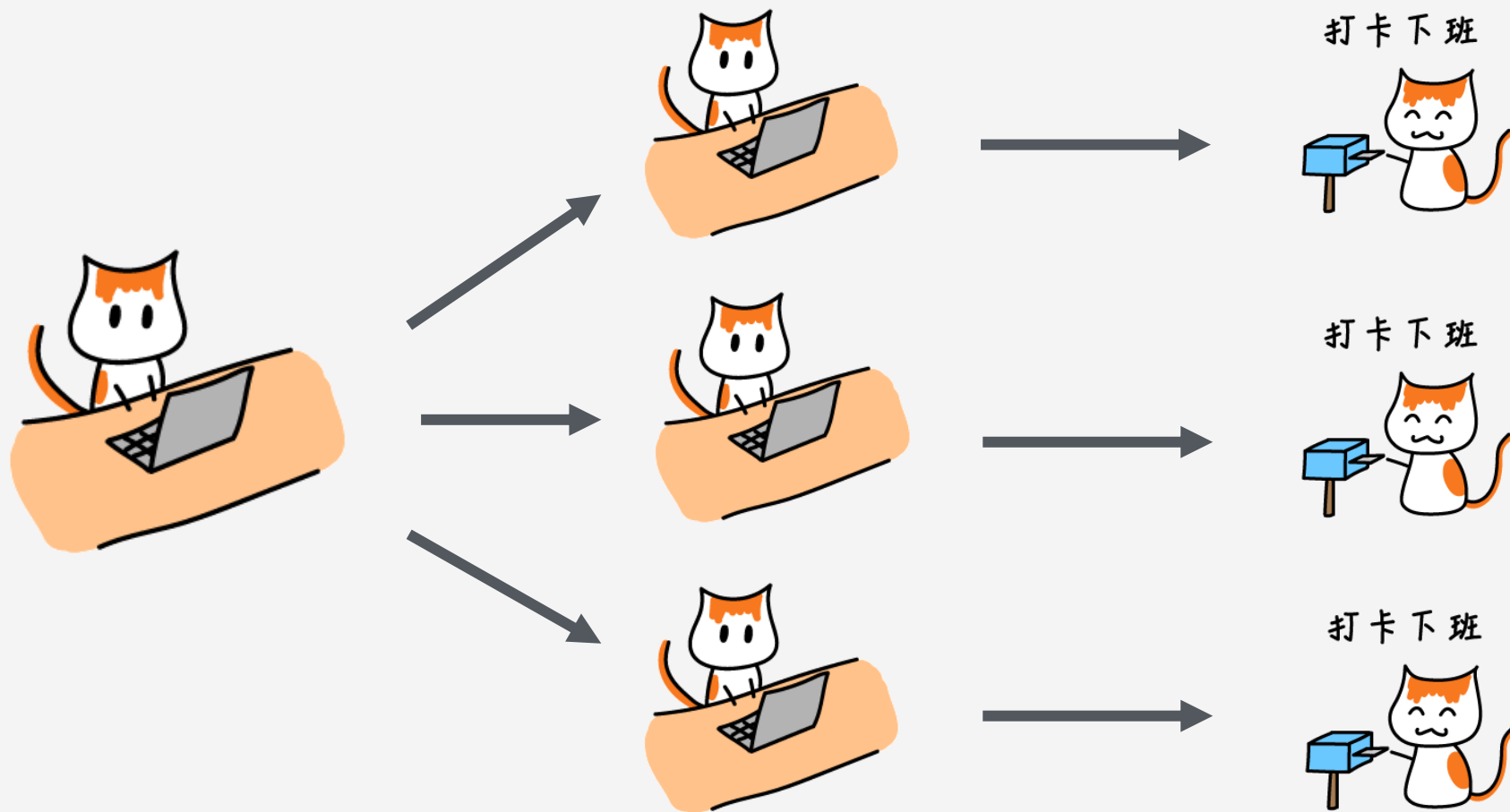
# 多思考 发现项目瓶颈并解决

工作中的项目实践，是提升能力最快的方式！！



### 三、帮助团队提升的能力

经验分享和传授，可以快速提升一个团队的技术能力

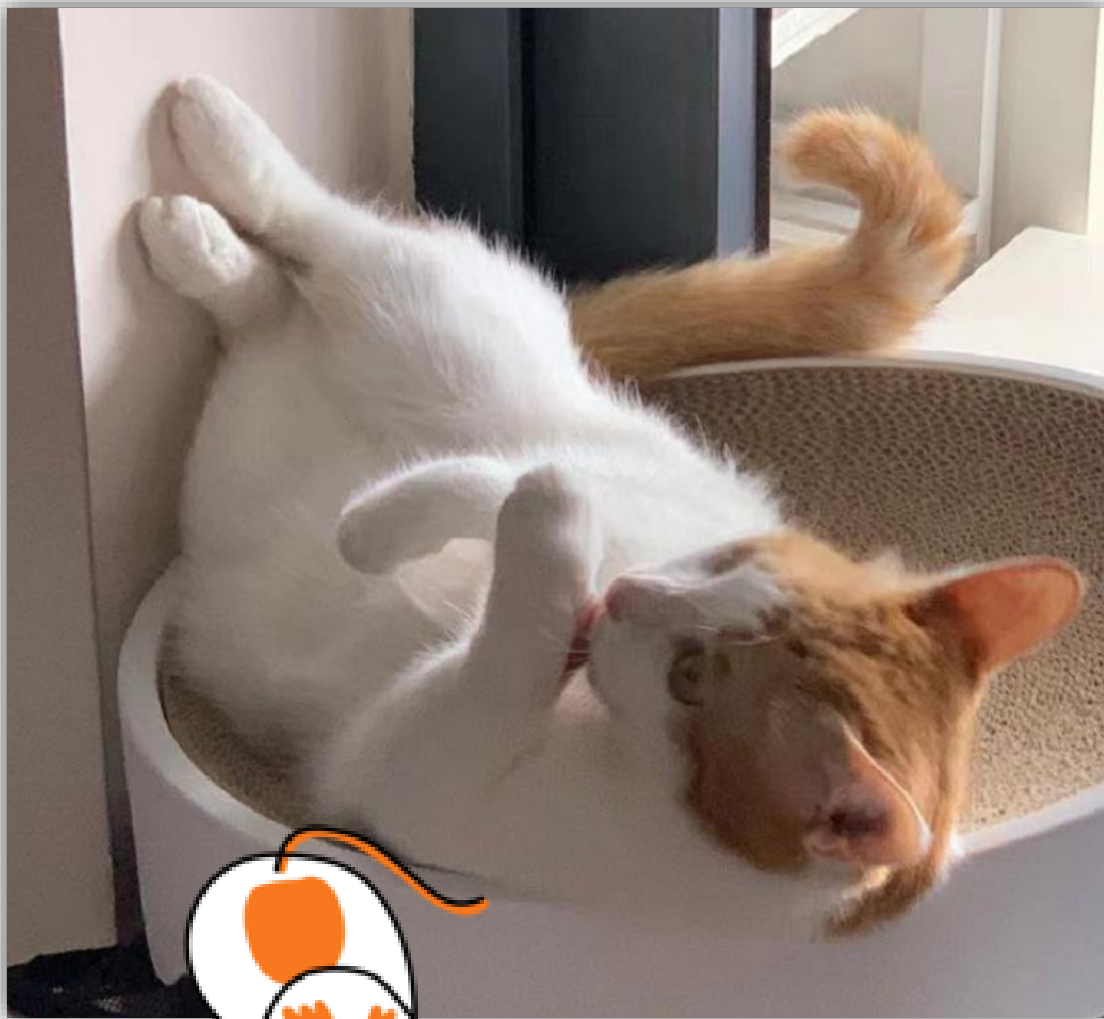


# 不要害怕被替代 分享也一样可以提升自己

[与更多的人交流和讨论，才能不断拓宽自己的认知！]







谢谢



Github: godbasin  
@被删