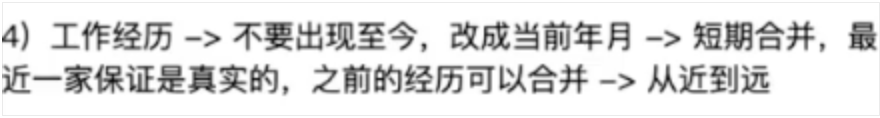
**0711会议总结——学习计划和简历修改**

## 教育经历

添加奖学金、证书荣誉

## 工作经历

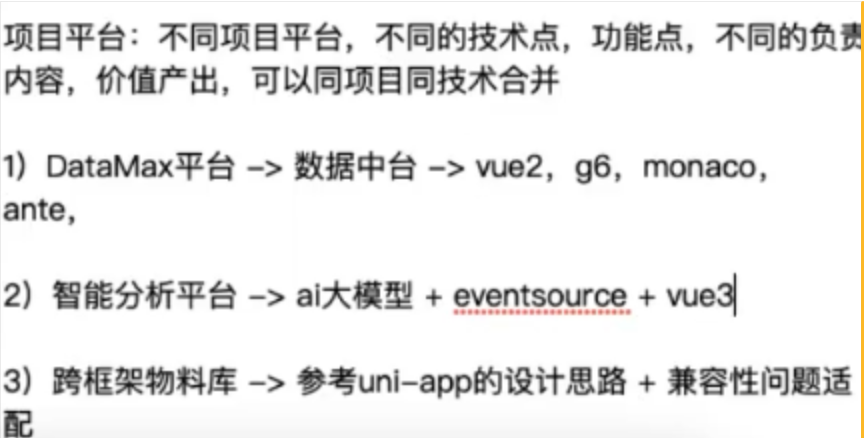
工作经历和项目经历拆分开



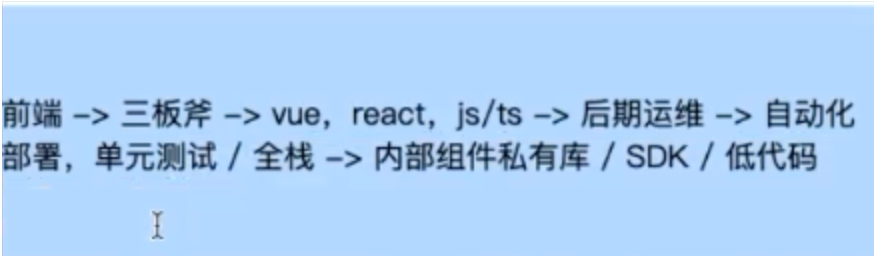
## 项目经历

按照班班给的模版修改简历；

* 跨语言组件库的梳理：梳理、完善自己对跨框架物料库架构思想的理解，参考uniapp的设计思路，吸取优势运用到跨框架物料库，沉淀出一篇博客；（有一个微前端和跨语言框架的思路是一个非常宝贵的经验）
* 梳理AI数据流通信是用的什么？
* 梳理不同项目不同技术不同内容不同价值产出

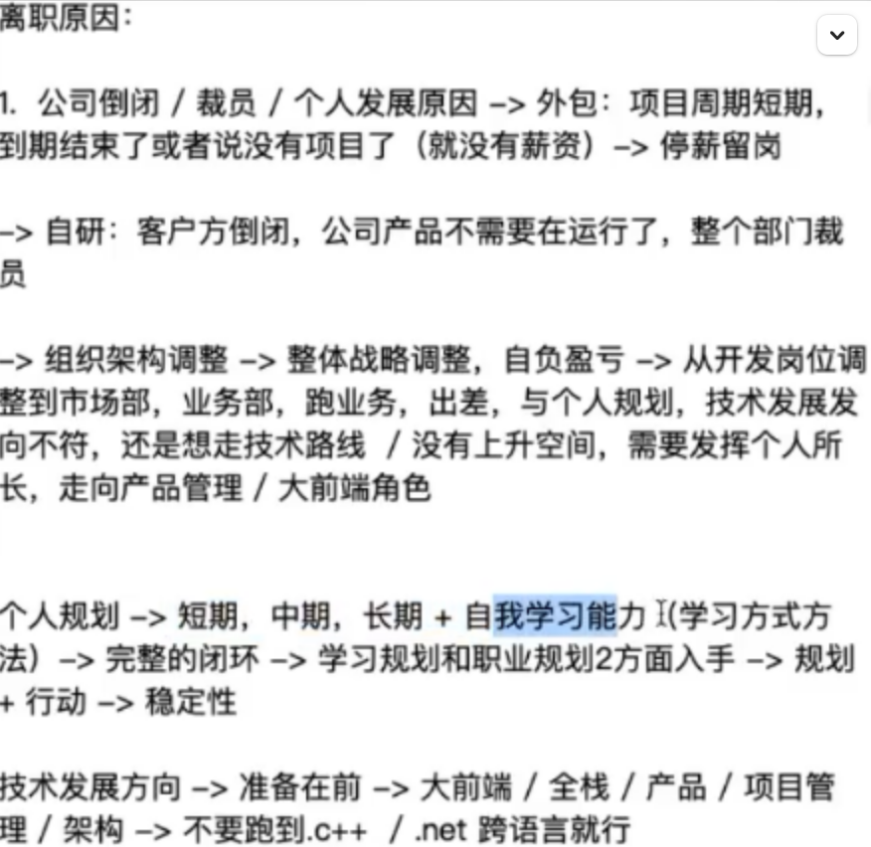


接下来学习重点



前端学习的核心和顺序如上所述。不管是微前端、antd、elemetui、uniapp 还是 taro，真正的前端三板斧就是 JS/TS、Vue 、React。

## 面试话术



### 自我介绍

有面试前一天，花五到十分钟去组织写一下。

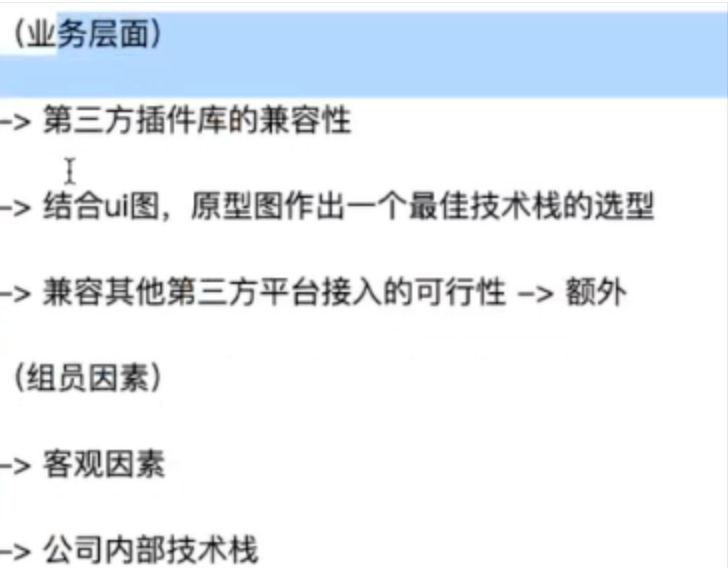
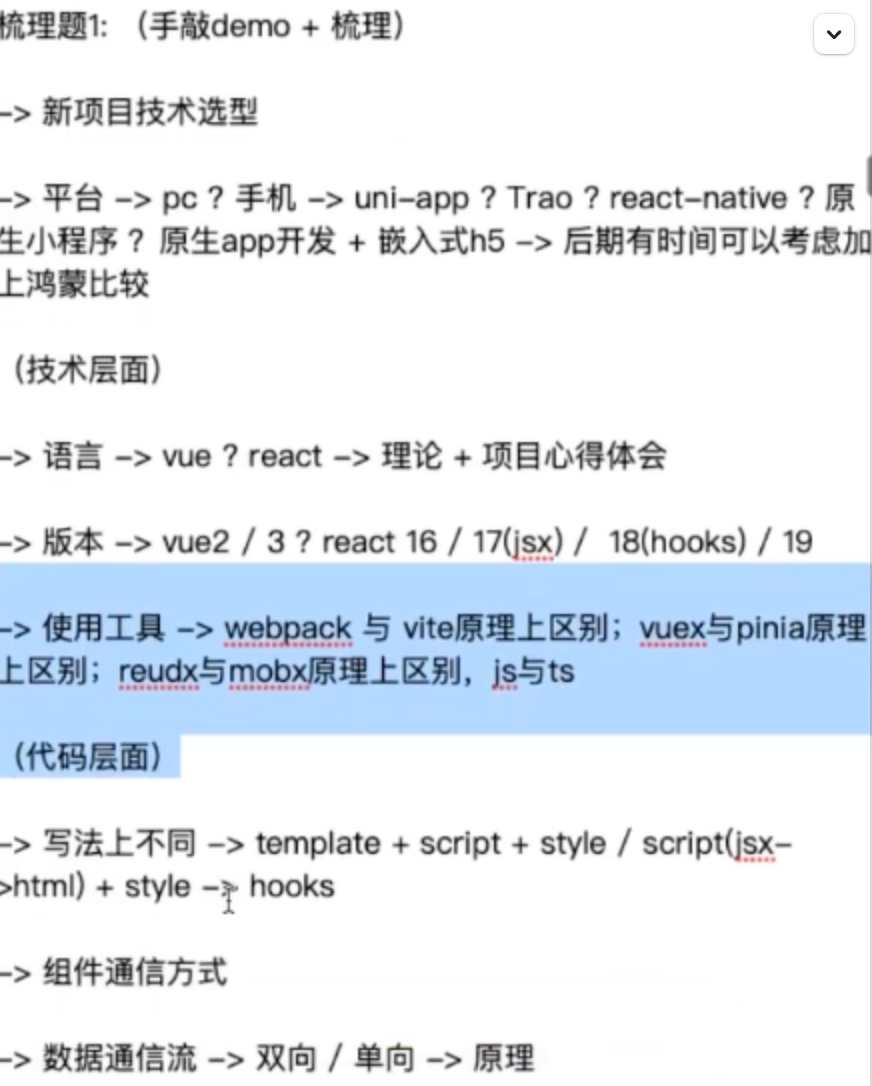
|  |
| --- |
| 简单模板：（技术栈不要介绍） |
| 我叫\*\*\*，从事了\*年前端开发，近\*年主要担任公司的（职位），负责公司（什么业务，其中获得的成就，亮点，收获，贡献，稍微 1-2 句，精简→有奖项的，2 句话带过，包含时间节点，获奖项目，参与其中的负责模块，不要太过于细化→没有就不要说了→起因 + 经过 + 结果 + 时间节点 + 对公司的收益）。  在工作中，我具备较强的产品意识，能够从业务价值角度思考技术实现；同时拥有清晰的风险预估与规避能力，能在项目推进中提前识别潜在问题并制定应对方案。 项目经验方面，我曾独立编写新项目的产品使用说明书，具备扎实的文档编写能力，能满足工作中各类文档输出需求；在跨团队协作中，能与产品、后端、测试等角色高效沟通配合，确保工作衔接顺畅。此外，我熟悉测试阶段的bug回滚机制，擅长敏捷开发模式，对项目上线发布流程中的风险点把控精准，能有效规避各类问题；在处理线上故障时，也具备快速定位根因并推动解决的能力，多次保障了项目的稳定运行。 这些经历不仅让我在技术落地与业务结合上积累了实践经验，也锻炼了我在复杂项目中统筹协作、风险控制的综合能力。  我对贵公司的前端开发岗位很有兴趣，希望今天面试顺利，有机会加入贵公司一起工作。 |
| 我叫Huey，在美的集团股份有限公司担任了近三年前端开发，主要负责公司大数据平台的业务从0到1的建设运维和前端横向项目的开发，同时负责前端规范化管理在大数据团队的推进。  在工作中我具有良好的产品意识、风险预估和规避意识、擅长敏捷开发，项目的上线发布流程避坑，以及线上问题的精准定位。我能够与其他项目成员进行沟通协作，完成工作的配合。  在美的三年，所在团队连续三年获得集团和事业部层面的技术突破奖和贡献奖，个人曾荣获优秀绩效。  我对贵公司的前端开发岗位很有兴趣，希望今天面试顺利，有机会加入贵公司一起工作。  Hello, interviewer. My name is Huey, and I graduated from South China University of Technology. I worked as a front-end developer at Midea Group Co., Ltd. for nearly three years. Next, I will introduce my work content from four aspects.  Fisrtly, I mainly worked on the lifecycle development of big data product, such as full-chain construction of core modules of data gavermance, data modeling, and the productization of the platform.  Second, I devoted part of my time on fe infrastructure development.  Third, I lead to promote the development of the standardization of fe within the big data team. For example, I was responsible for integrating DataMax into the monitoring system and implementing front-end standardization solutions.  Lastly, i took on system operation and manitennance duties. this included user behavior anaysis and perform night-tien patrols. |

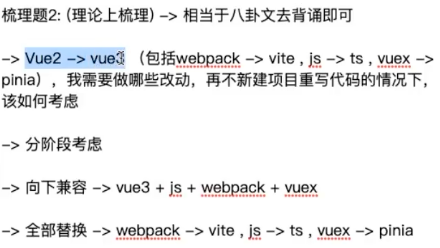
|  |
| --- |
| 高大上模板：（中，英） |
| 面试官你好，我叫\*\*，\*\*年毕业于\*\*大学\*\*专业（大学期间是否有获奖，可以讲一下 -> （时间+名称），精简），实习期间一直从事前端开发的工作。我的校园项目（实习项目）主要做的是（\*\*技术栈，\*\*项目平台，\*\*\*加上项目奖项）  在工作中我具有良好的产品意识,也具备风险预估和规避意识，（工作之后项目中的贡献，成就，收获 / 收益）此外我曾独立编写过新项目的产品使用说明书，能够满足工作中的文档编写需求，能够与其他项目组成员进行沟通协作，完成工作的配合。测试 bug 的回滚。擅长敏捷开发，项目的上线发布流程避坑，以及线上问题的精准定位。  我对贵公司的前端开发岗位很有兴趣，希望今天面试顺利，有机会加入贵公司一起工作。 |
|  |

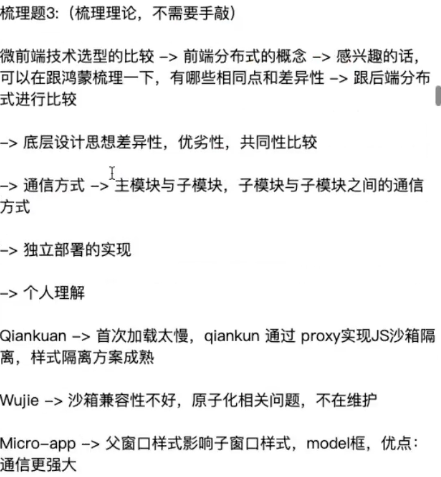
忌讳：不要长篇大论去介绍简历的项目经历，不要去读简历，突出个人能力和为公司带来的收益/个人收获/续约率 + 客户留存率

目前我的水平主要突出个人能力和个人收获。

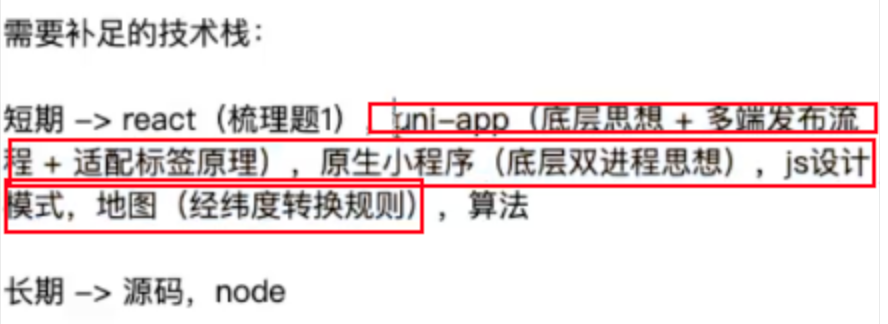
### 面试题梳理思路





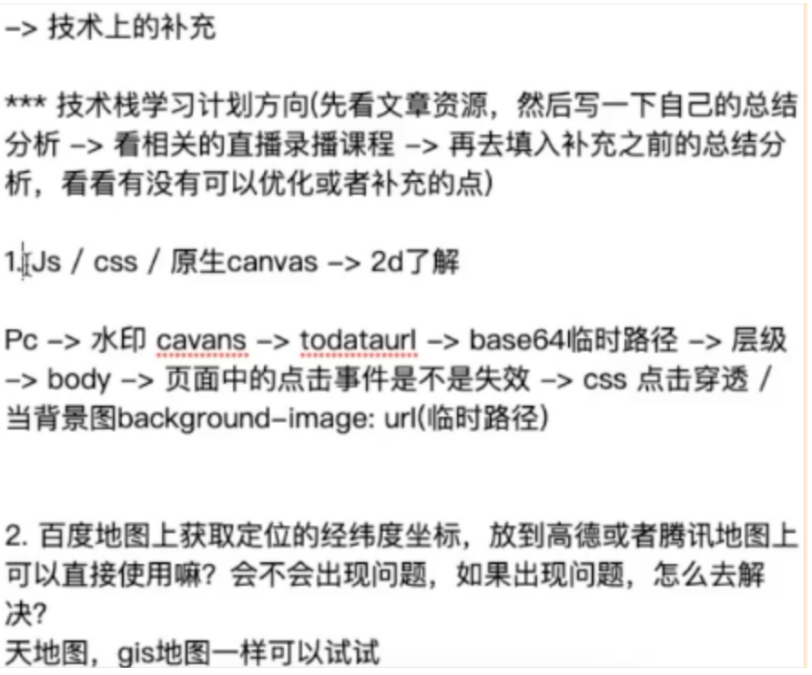


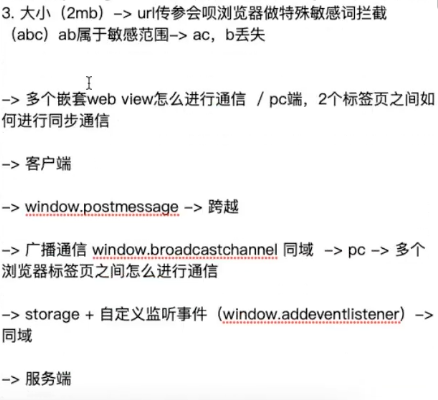
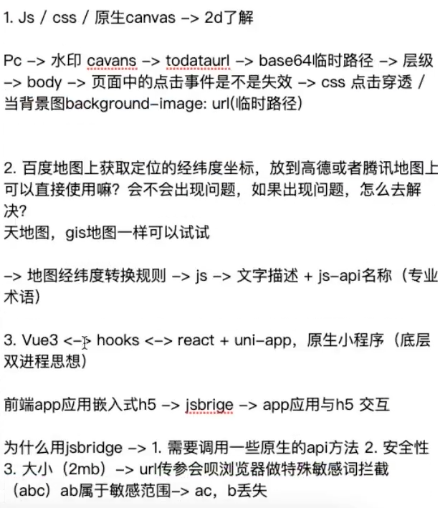
需要补足的技术栈



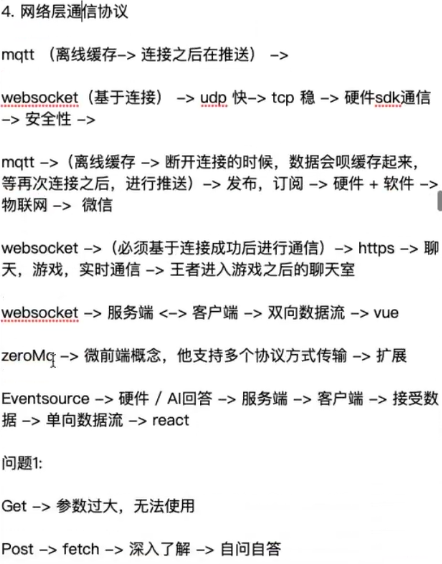
圈住的地方只需要了解单一的核心点就可以了，至于其具体使用过程中，有项目做到的时候再去查相关文档。

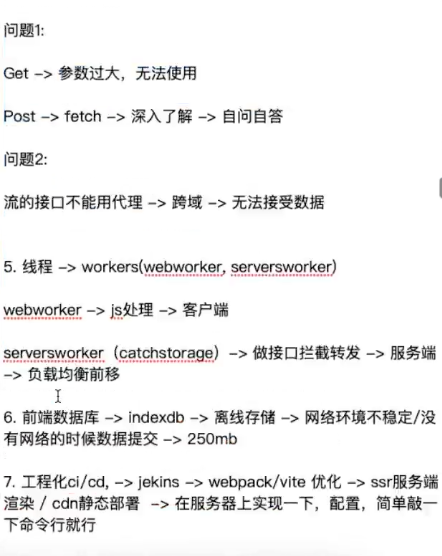
### 技术上的补充

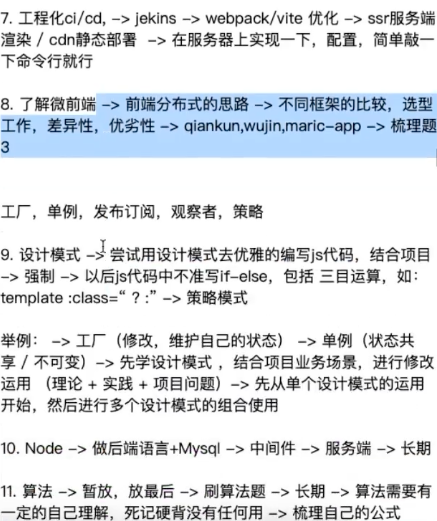


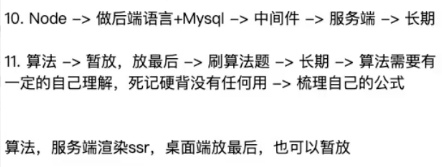


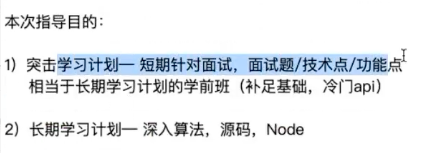




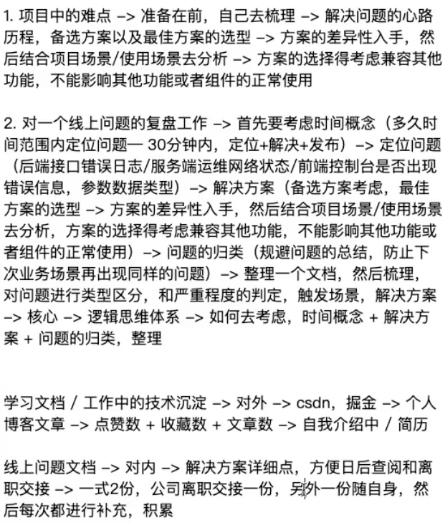


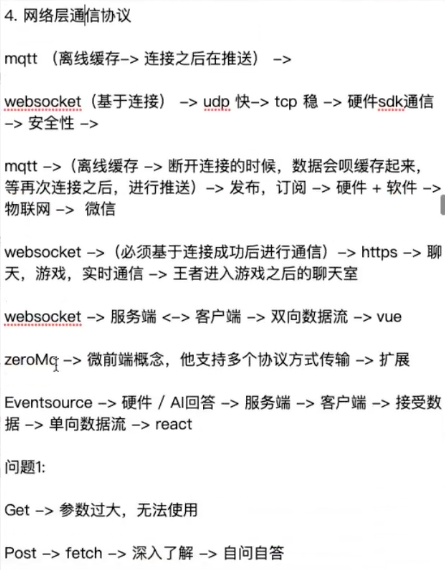






### 开放性面试题梳理





### 学习方式方法

缺乏的是一个逻辑思维体系 + 专业术语的表达（理论知识 + JS - api描述）

\*\*\* 建议是五休二，五天学习新的知识点 + 直播录播课程 + 总结分析，周末2天回滚之前学习的总结分析 + 扩展性学习

开发中 -> 总结分析 -> 下班前后花10 - 20分钟梳理当天的难点，或者问题的pass方案，第二天解决问题之后在整理解决方案与Pass方案之间的差异性，卡壳点在哪 -> 产品需求 + 业务场景 + 解决方案的心路历程 -> 扩展性学习（放到周末时候去进行回滚工作日五天的总结）-> 备选方案与最佳方案的比较选型 -> 梳理自己的逻辑思维体系 -> 总结概括性语句写进简历中 + 结果导向即可 -> 5 + 2模式 + 月回滚 -> 5天整理总结，2天回滚扩展，一个月做一次大回滚

学习过程 -> 学一个文章资料 -> 写一个心得体会 -> 结合录播，直播课程 -> 看看有木有自己理解遗漏忽视的地方 -> 在补充进之前的心得体会 -> 之后做一次扩展性学习 -> 了解其运用的业务场景以及相似的api功能之间的差异性；优劣性做一个比较 -> 最后在实际工作中运用遇到的问题 -> 补充进之前的心得体会 -> 技术方案 -> 理论 + 业务场景 / 产品需求 + 项目问题 + 备选方案 + 最佳方案选项 + 收益 -> 闭环 -> csdn / 掘金 -> 生成个人的博客文章

PM -> 需求分析 -> 技术方案 -> 资源评估 -> 风险预案

->执行 ->复盘 ->循环

->概括性语句 ->起因 + 经过 + 结果导向 / 性能优化 ->20 - 30（上下百分之10的浮动）

->梳理自己的逻辑思维导（谱雀思维导图）->指导只有一次，珍惜每一次的相遇，我能做的是从旁点播，主要还是要靠自己，养成良好的学习方法（5 + 2 + 月回滚）和总结分析。

开发过程：需求来了->百度查->cv复制 ->测试 ->发布 ->优化（备选方案）->总结分析(扩展性的学习)->刷手机

开发中 ->技术能力（敲代码+百度）->风险把控 ->没有逻辑思维体系的组织能力

逻辑思维体系的培养 ->学习/总结分析

->理论 + 实践 + 业务场景 / 产品需求 ->心得体会 / 总结分析

-> + 实际工作中遇到的相关技术难点 -> 核心描述的概念

项目开发中/学习中，总结分析，写下来，用自己的精简语言去描述，去组织 => 就是为了简历中的负责内容，项目亮点，做铺垫

第一次150字，语句不通畅，逻辑混乱

第二次50字，用作跟面试官的对答

第三次10 - 20字，用作简历中的负责内容，项目亮点

面试-> 枯草，就是考察语言组织能力，理论知识

面试问题类型: 1) 自我介绍 2) 理论 (面试题+自己的理解) 3) 项目 (日积月累的总结分析) 4) 个人规划 5) 技术发展 —大厂必问

文章资料和录播，直播课程，相辅相成，结合看，自己总结分析，写下来，用自己的语言去组织自己的理解，最后网上看看别人的理解跟你的是不是有相同点 (词能达意即可，没必要完全照搬别人的)

-> 真正解决的问题其实还是原生的js

-> 解决问题不要过度依赖框架和第三方库

个人建议 -> 学一段时间之后拿外包当面经，不一定要去，但是别反感面试，当作验证自己一段时间学习的成果和自己的学习方式方法的检验，同时看看外包面试过程中有哪些自己的知识盲点，面完之后梳理总结，同步添加到自己的学习规划中进行补充学习，等有自研时候，争取一次性通过！！

-> 面试也需要经验的积累

第一次外包面试 -> 追求，自我介绍的流畅性与回答问题的逻辑顺序 -> 理论知识 + 个人理解 + 专业术语的表达

第二次外包面试 -> 追求回答问题的完整逻辑思路 -> 技术方案 -> 理论 + 产品需求/业务场景 + 项目问题 + 备选方案 + 最佳方案选型 + 结果导向 / 收益（扩大上下20%）

第三次外包面试 -> 知识盲点，哪些知识点是课程和学习资料中没有涉及到的，结束之后梳理加同步添加到学习规划中，找相关资料作一定的学习梳理

面试回答技巧：（语言组织能力）

回答问题的逻辑顺序

逻辑思维体系 -> js 原型链（往上查找）-> 逻辑思维链 (往下深挖)

\*\*\*\*\*\*广度 -> 范围大的讲 -> 不停的缩小范围 -> 以某个具体的技术为切入点 -> 结合项目进行深度的解析 -> 符合什么业务场景来进行选型

-> 广度（备选方案）-> 深度（备选方案的差异性，优略性，做比较）-> 业务场景（选择最佳方案）-> 三要素 -> 项目难点 / 项目选型的回答三要素

-> 梳理总结分析 -> 多个技术点，知识点，通过业务场景 / 产品需求 -> 进行串联 -> 多个业务场景 / 产品需求 -> 就是一个完整的项目 -> 逻辑思维体系中主干是业务场景 / 产品需求，分支应该是技术点，功能点 -> 类似蜘蛛网，丝是产品需求业务场景 -> 连接处是技术点，功能点 -> 越织越大就是自己的逻辑思维体系网

-> 点成线 -> 线成面

-> 面试官想听的 -> 体现你的逻辑思维体系

-> 项目中的问题 -> 解决问题不能太过片面化，需要去做多个/其他备选方案的考虑 -> 2 - 3 解决方案 -> 结合自己实际项目的业务场景需求，进行最佳方案的选型 -> 当面试题自己去梳理下怎么回答

八股文面试题回答逻辑：

1) 官方答案（百度刷题）— 60分

带一点自己的思路/理解

2) 专业js-api（功能的核心api名称）— 80分

Js, object / array 或者原生Js的，

防抖节流 -> let timer = settimeout(\*\*\* 闭包 return) -> cleartimeout

3) 结合自己的项目场景/使用场景，用自己的语言去描述（举一反三）— 100分

\*\*\*忌讳：不要长篇大论去描述功能的代码实现（跟面试官说怎么敲代码的）

面试官想听的是自己的思路和理解 -> 体系

\*\*\*\* 中英结合 -> 自己的语言组织 + 核心代码api名称 + 简单的举例 + 精简语句 + 结果导向

-> 写一下，梳理一下，按照面试回答三技巧来组织

开放性的面试题梳理：

1. 项目中的难点 -> 准备在前，自己去梳理 -> 解决问题的心路历程，备选方案以及最佳方案的选型 -> 方案的差异性入手，然后结合项目场景/使用场景去分析 -> 方案的选择得考虑兼容其他功能，不能影响其他功能或者组件的正常使用

2. 对一个线上问题的复盘工作 -> 首先要考虑时间概念（多久时间范围内定位问题—30分钟内，定位+解决+发布）-> 定位问题（后端接口错误日志/服务端运维网络状态/前端控制台是否出现错误信息，参数数据类型）-> 解决方案（备选方案考虑，最佳方案的选型 -> 方案的差异性入手，然后结合项目场景/使用场景去分析，方案的选择得考虑兼容其他功能，不能影响其他功能或者组件的正常使用）-> 问题的归类（规避问题的总结，防止下次业务场景再出现同样的问题）-> 整理一个文档，然后梳理，对问题进行类型区分，和严重程度的判定，触发场景，解决方案 -> 核心 -> 逻辑思维体系 -> 如何去考虑，时间概念 + 解决方案 + 问题的归类，整理）

学习文档 / 工作中的技术沉淀 -> 对外 -> csdn，掘金 -> 个人博客文章 -> 点赞数 + 收藏数 + 文章数 -> 自我介绍中 / 简历

线上问题文档 -> 对内 -> 解决方案详细点，方便日后查阅和离职交接 -> 一式2份，公司离职交接一份，另外一份随自身，然后每次都进行补充，积累

【最后的心语】

不要浪费自己的时间去改变别人。世上最难的，就是把自己的思想装进别人的脑袋。我们都应该学会克制自己改造他人的欲望，别把自己的手，在别人的世界里伸得太长。

【公平】

每个人的时间是公平的，每个人都有24小时，怎么过是可以选择的，其他的都可能不公平，人比人气死人，做自己，改变不了命运，但是我可以让每一天都很充实，是金子总会发光，往事皆可序，未来皆可盼。

【未来可盼】

如果你想干一件事，那就先去干，哪怕很烂，哪怕一点都不完美。一个粗糙的开始就是最好的开始，别等自己准备的万无一失才行动，因为根本没有完美的准备，只有在行动中不断调整和进步。踏出第一步和原地踏步，你选哪一个？

【努力的意义】

我希望你们可以从今晚得到的是，那些你早起努力工作的时光，那些你熬夜努力的时光，那些你很累但不想努力，但却坚持下来的那段时光，那就是追寻梦想的意义，那才是梦想的力量，但是真正的梦想不是到达终点，而是享受旅途，如果你们能够理解这个道理，你会发现，在努力的过程中，你的梦想或许会实现，也或许不会，但总会有一些更伟大的事情随之而来～

【改变从来不晚】

减少内耗和抑郁，做好自己手上能做的事情，每一天的积累，总会有收获的时刻，不要错过机会，目前趋势我们不仅要跟同行竞争，还要跟刚毕业大学生竞争，还要跟ai竞争，一句话，可以替换5个政务部门的编制岗位，我们还不努力，如何苟活，当前主流环境回暖的期望遥遥无期，一定要坚持努力搭建自己的逻辑思维体系，改变自己的学习方式方法，就从现在开始，一切归零，从头来过。

【永不言弃 <-> Never give up】

不想学的时候 -> 拿出手机 -> 看余额 -> 看账单

接下来的学习步骤：

1) 梳理项目中的功能点，技术点（罗列出来） + 梳理总结分析（给到班班）

2) 梳理跟客户对接需求，发布上线，部署过程中或者项目开发中遇到的问题/难点（同时考虑备选方案和最佳方案的选型，给到班班）

3) 罗列学习时间计划表（5 + 2模式 + 月回滚）

4) 自己搭建项目脚手架 / 分支 + 补充广度（学习计划） + 设计模式 + 总结分析 -> 理论 + 实践（思考项目中符合的直接强制替换） + 业务场景 -> 总结分析

5) 填充简历 -> 对总结分析的一个概括性归纳（用一些高大上的语句措辞）

6) 投简历 / 找到工作之后考虑长期计划

7) 梳理离职原因+自我介绍，个人规划 + 技术发展发向（有约面之后进行梳理）

### 建议方案

* 先梳理跨框架物流库的兼容性和自己开发负责熟悉的模块+梳理现有项目中的功能点，技术点，开发中的问题，解决方案，以及备选，最佳方案的考虑
* 考虑语雀中的记录，以PM项目的角度再去复盘回看一遍，能否看的更广更深 + 理论知识的填充 -> 发布到csdn/掘金/ github
* 梳理自己的时间计划表（举例：工作日，上午1小时，下午1小时 + 梳理总结，周末3 - 4小时 + 回滚扩展）-> 羽雀 / 飞书 做一些记录，周末回滚笔记，然后考虑项目中是否有遇到过，然后加上理论 + 业务场景 + 项目问题 + 备选方案 + 最佳方案选型 -> csdn / 掘金
* 从基础开始学习js,css,canvas，地图 -> 梳理总结分析，-> csdn，掘金 -> 理论+业务场景/产品需求 -> 项目中有使用过的 -> 梳理项目问题 + 备选方案 + 最佳方案选项 资料 + 直播 / 录播结合看 -> 填充之前的总结分析
* 学到vue / react -> 资料 + 直播 / 录播结合看 -> 工作日/周末抽空增加1 ~ 2小时敲代码，同时运用新学到的知识和js - api -> 梳理工作日的总结分析，敲代码过程中的问题，心得体会（期间记得运用设计模式，或者先用js/ts实现逻辑，然后学到设计模式时候在进行重构）
* 建议：敲完demo之后，后期有时间补补上自动化测试ci/cd + 前端后端线上自动部署/python爬虫 + node 中间件 -> 完善你自我介绍中的全栈/大前端/架构角色介绍
* 中期或者长期目标 -> 自动化部署，单元测试 -> 性能监控体系 -> 先从业务纬度提升到架构/项目层面 -> 还是在开发范畴内
* 运维层面 -> 部署，ci/cd，单元测试
* 注意：先从配置跨域到封装组件开始，再找2个简单的功能模块进行开发，然后在找2个最难的模块进行开发，最后找一个中等可以有深度挖掘的模块进行开发（期间记得运用设计模式，或者先用js/ts实现逻辑，然后学到设计模式时候在进行重构）
* 3 - 4周 / 1 - 2个月，约外包面试 -> 当面经 -> 验证自己的方式方法和效果 -> 有约面时候梳理自我介绍，个人规划，技术发展方向 -> 面试过程中记得面完后梳理自己的知识盲点，同步到个人学习规划中