

学习计划

(突击版)



教研组
○
编著

测评结果分析

模块	知识点	掌握情况
前端基础知识	html 相关	一般
	浏览器原理	一般
	css 基础	一般
	Js 基础	一般
常用框架	Vue	一般
	React	薄弱
前端工程化	webpack 原理	一般
网络&web安全	HTTP 基础	一般
	XSS 攻击	薄弱
	死锁	一般
项目实践	多标签页通信	一般
	性能优化	一般
算法&手写题	基础算法（快排）	薄弱
	Promise	薄弱

整体学习建议

- 学员对于前端基础知识的掌握有待提高，部分问题没有掌握或是回答不完整理解不透彻。这方面建议学员以背题为主，多花时间去总结巩固自己的知识体系，可以多从课程中，或者别人总结的文章入手，梳理自己的知识结构图谱，尽可能提高自己的知识广度。
- 对于框架源码的掌握完整度还需要提高。后续可以将重心主要还是放在对于框架的理解和应用上，并且能够对框架源码的核心概念有一个基本的认识。之后这一块建议后续可以通过阅读源码解析文章或从他人总结的回答文章着手，对每个概念梳理出一个完整全面的回答，多多融合自己实践和理解。
- 工程化和计算机网络的回答有些许不足，还需加强。还需要对于 webpack、babel 等常见工具能有一定的实践与理论的掌握。
- 整体上建议学员跟随着课程模块来学习，优先掌握下述高优先级模块。

学习计划

模块	优先级	相关课程	学习/提升建议	学习时间
Js基础	高	《this指针/闭包/作用域》 《面向对象编程/原型及原型链》 《promise 规范及应用》 《JS 模块化详解》 《浏览器内置 JS 对象详解、浏览器事件模型详解、浏览器请求相关内容》 《js性能优化》 《JavaScript 高级内功修炼 函数式编程》 《ES6 Esnext 规范详解》 《ES6 项目实战》	作为前端的基座，此项为高优先级模块，建议学员先整理好知识结构（参考下文每一专题配图）。跟着课程学习，辅以面试题目做成果检验	30%
框架介绍与应用	中	《vue.js 基础》 《vue.js 高级用法》 《vue.js 状态管理及 ssr（路由及状态同构）》 《vue-cli详解》 《vue.js 核心模块源码解析（上） 》 《vue.js 核心模块源码解析（下） 》 《vue.js 前端路由及异步组件》	掌握基础 API 的用法，明白常见概念的原理和对比（例如组件传值、各生命周期的不同等），熟练在不同场景下使用高阶 API，理解组件设计与项目架构思想	30%
	中	《vue.js 项目实战》		
	高	《react.js 基础》 《react.js 高级用法》 《react.js 基础》 《react.js 高级用法》 《redux & mobx》 《react.js 核心源码解析（上） 》 《react.js 核心源码解析（下） 》 《react.js 路由及状态同构》	建议同学深入阅读 React源码的核心知识模块，了解框架内部工作机制，提升对 React的深刻理解。深入研究框架实现，例如fiber、hook、hoc、从数据驱动方面透彻说明白redux、antd组件化 设计思路等。同时可以参与开源社区，了解其他开发者的实践和经验，持续学习新的技术和最佳实践。	
	中	《react 实战-仿知乎页面框架》 《react实战-知乎细节与高级特性》 《首页鉴权登录与用户消息推送》		
工程化	高	《微前端（上） 》 《微前端（下） 》 《前端AST》 《模块化开发与webpack》 《自动化构建、测试、部署（上） 》 《自动化构建、测试、部署（下） 》	建议后面可以花时间去学习如何配置项目的构建流程，优化资源 管理和打包。并且了解在前端工程化中如何实现自动化构建、测试和部署流程，了解如何实现代码自动化集成和部署。	20%

模块	优先级	相关课程	学习/提升建议	学习时间
设计模式	中	《设计模式解析与实战（上）》 《设计模式解析与实战（下）》	优先级较低，有余力可补充学习	
Type-Script	中	《TS（上）》 《TS（下）》	优先级较低，有余力可补充学习	
计算机网络	高	《前端突击训练营—HTTP面试题》	了解前端优化技巧，如CDN加速、HTTP缓存策略等。 了解新兴技术，如HTTP3、QUIC等，以及其在提升性能方面的应用。	5%
算法	低	《前端训练营-数据结构与算法》 《前端面试训练营（突击班） 算法-数据结构篇》 《前端面试训练营（突击班） 算法-排序与双指针等》 《前端面试训练营（突击班） 算法-二分与回溯》 《前端面试训练营（突击班） 算法-贪心与动规》	【面试大厂需要重点关注】 学习常见的数据结构：线性结构、树、图等 去 LeetCode 平台刷题，难度以 medium 为主，hard 题掌握例题即可。 建议从最近开始，每天安排半小时左右专门用来做刷题训练。	
解决方案与应用	中	《性能优化》	优先级较低，有余力可补充学习	5%
前端可视化	中(根据自己实际掌握情况选择，作为加分项，不强求)	《canvas 与 WebGL》 《可视化领域 —— 针对可视化领域的深度挖掘》 《低代码领域 —— 切入最为火热的前端研究领域》 《低代码平台深度进阶 -进一步提升》	学有余力或者本身有一定基础的情况下，可以选择一个方向进行学习，作为自己简历上的加分项目	10%
React-Native		《RN入门与实战（上）》 《RN入门与实战（下）》		
小程序		《小程序开发基础》 《小程序开发框架解析》 《小程序项目实战》		
Flutter		《现代hybrid介绍与使用、flutter 与 dart 开发》 《flutter 实战》 《electron 入门与实战（上）》 《electron 入门与实战（下）》		
Node.js		《基于 Node 的后台技术》 《node.js 基础教学》 《node.js 原理详解》 《node.js 框架详解》 《node.js 网络 HTTP & 部署》 《node.js 实战》		

学习资源

推荐学习资源

资源名称	链接
写给初中级前端的高级进阶指南	https://juejin.cn/post/6844904103504527374
一个合格(优秀)的前端都应该阅读这些文章	https://juejin.cn/post/6844903896637259784
历时8个月！ 10w字前端知识体系+大厂面试总结（基础知识篇）	https://juejin.cn/post/7146973901166215176#heading-107

第一专题：HTML + CSS

课程通常一般是从 JS 知识点讲起，不会包含这两块的知识点。这两块在面试时虽然被问的比例不大，但是是前端最根本需要掌握的知识点，务必花时间掌握。可以参考包含但不限于以下的方面：

- ◇ 多多练习页面的书写，多看看技术论坛大佬们开源的项目，模仿观察其页面 & 组件的 HTML 与 CSS 的写法，包含 HTML 标签的运用，页面结构的分配，CSS class 的命名等
- ◇ 掌握 CSS 处理器，例如 SASS、LESS 等，学会规范书写 CSS
- ◇ 掌握 HTML 与 CSS 常见的问题，例如常用的 CSS 布局、居中问题、BFC、FFC、GFC、响应式、移动端适配等，可以从别人发的面经中提炼总结

爪哇课程

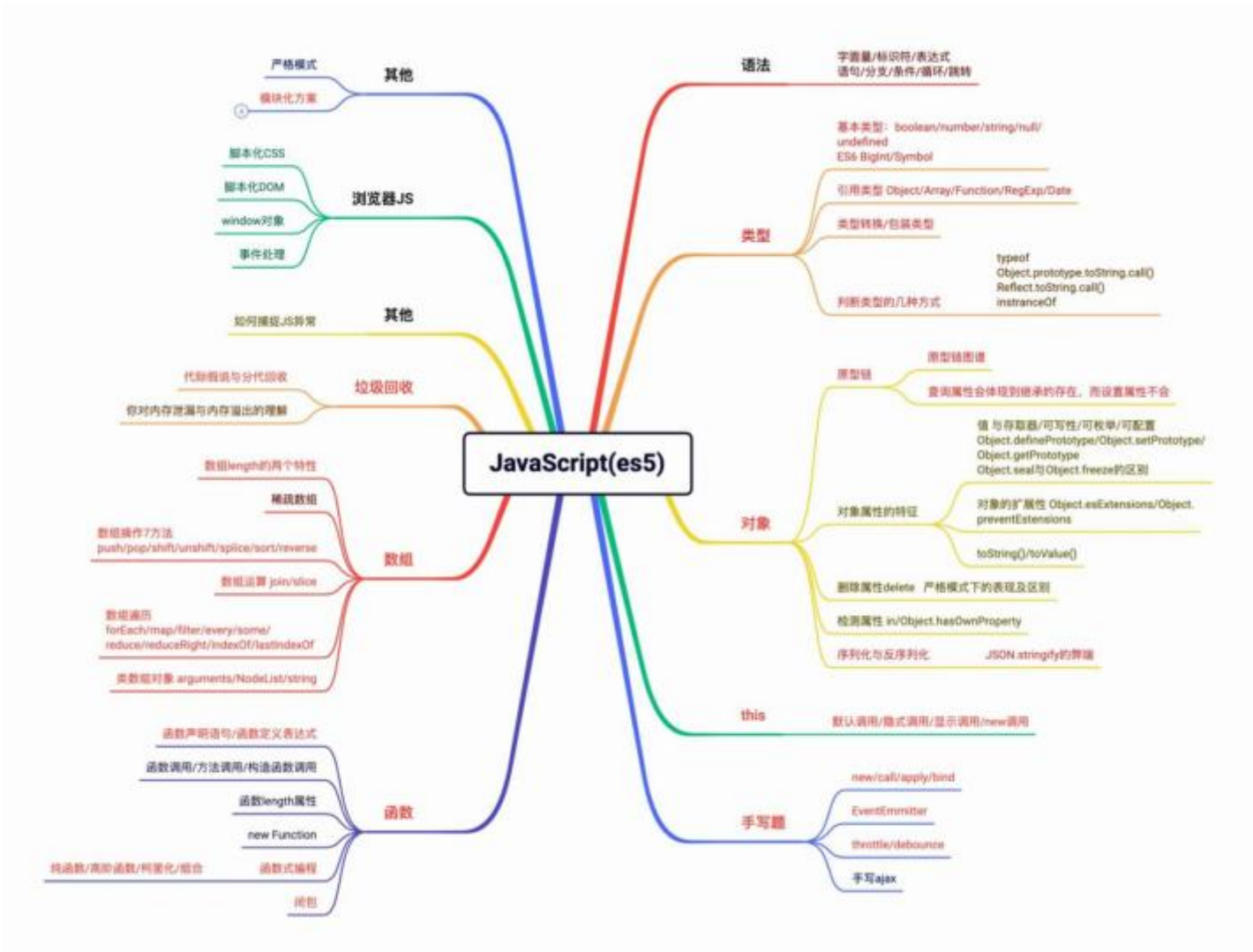
课程名称	链接
前端突击训练营—HTML面试题	https://xcdjj.h5.xeknow.com/sl/1xrxlu
前端突击训练营—CSS面试题	https://xcdjj.h5.xeknow.com/sl/3R2Tlv

推荐学习资源

资源名称	链接
HTML梳理	https://blog.warmplace.cn/post/html
CSS基础入门	https://blog.warmplace.cn/post/css_base
CSS 盒模型与布局	https://blog.warmplace.cn/post/css_box_layout
CSS 布局方案与应用	https://blog.warmplace.cn/post/css_example
CSS 进阶	https://blog.warmplace.cn/post/14
CSS 预处理器	https://blog.warmplace.cn/post/15

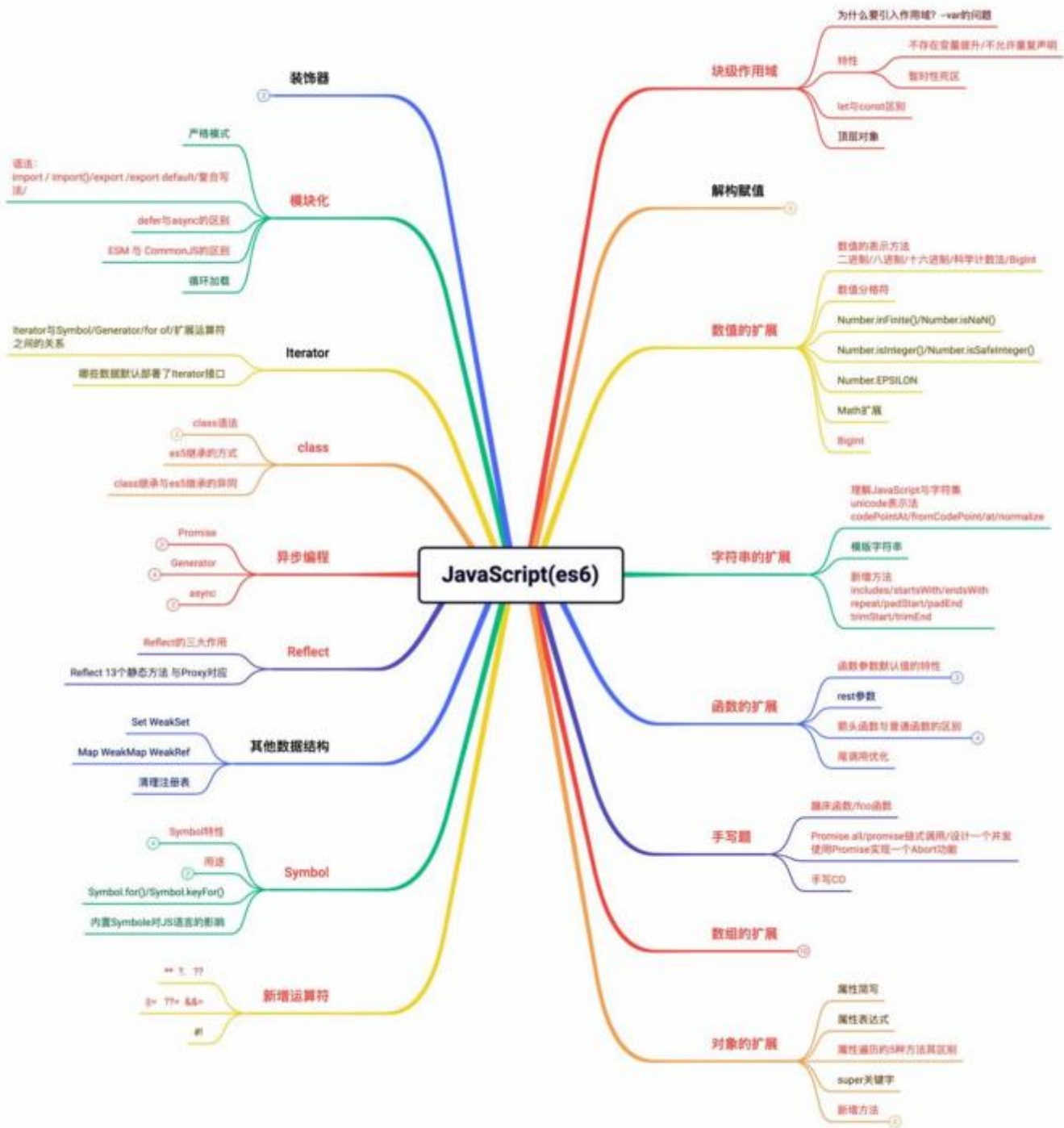
第二专题：JS基础

JS基础大致可以参照以下思维导图去入手学习，整体知识点会比较多也会比较散， 需要同学具有更多的耐心去慢慢掌握，其中如Promise、闭包、事件轮询机制等都是面试常见的考点， 需要重点掌握



爪哇课程

课程名称	链接
this指针/闭包/作用域	https://xcdjj.h5.xeknow.com/sl/2zYWKi
面向对象编程/原型及原型链	https://xcdjj.h5.xeknow.com/sl/9LQIz
promise 规范及应用	https://xcdjj.h5.xeknow.com/sl/3LjdRo
JS 模块化详解	https://xcdjj.h5.xeknow.com/sl/2NyOCR
浏览器内置 JS 对象详解、浏览器事件模型详解、浏览器请求相关内容	https://xcdjj.h5.xeknow.com/sl/2zIXLd
JS性能优化	https://xcdjj.h5.xeknow.com/sl/3RiEv4
JavaScript 高级内功修炼 函数式编程	https://xcdjj.h5.xeknow.com/sl/23Q74f

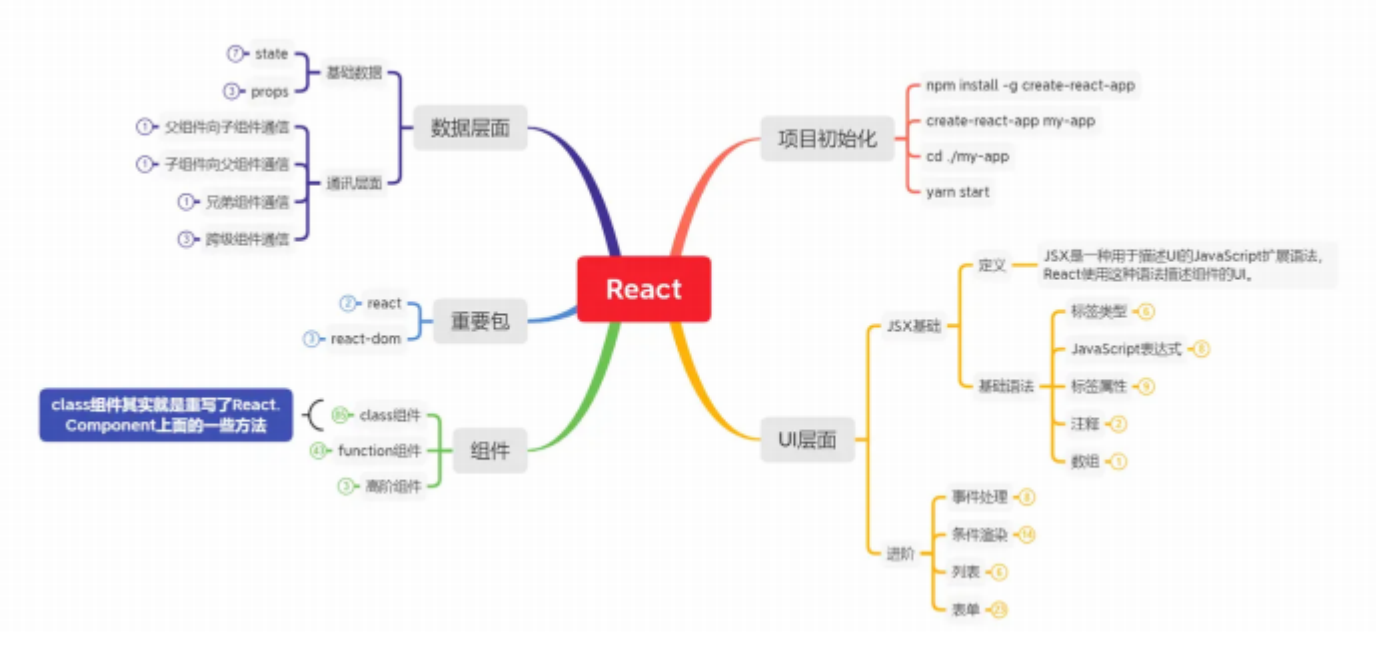


第三专题：React

首先也是需要把官方文档中的基本概念掌握， 需要熟练掌握官网中所提到的技巧， 就算没有用过， 也需要知道在什么场景下使用。

React 已经进入了 Hook 为主的阶段， 相关的社区库也逐步完善。

同时React 最好可以搭配和 TypeScript一同使用， 所以推荐在学习完React后也适当了解一些TypeScript的相关知识。



爪哇课程

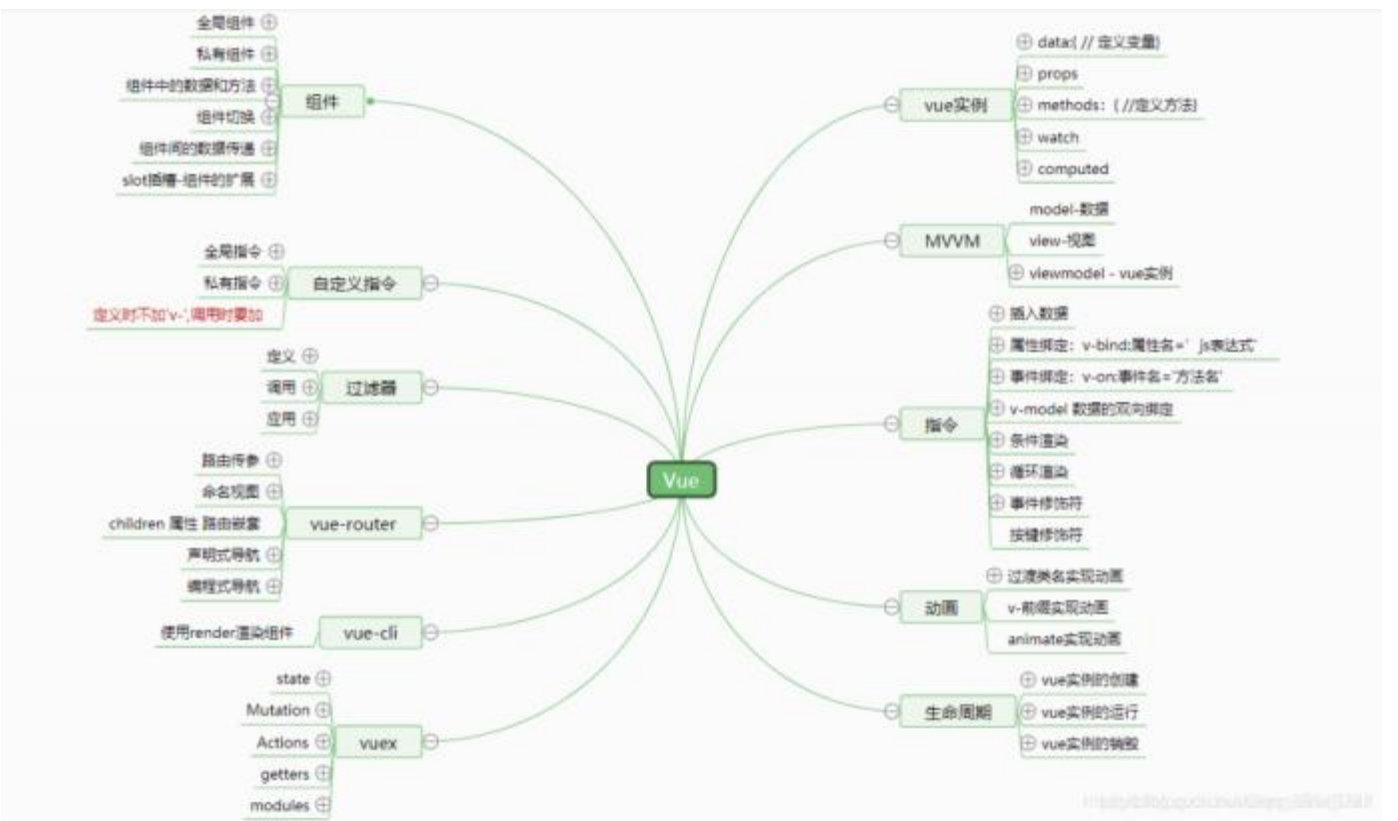
课程名称	链接
react.js 基础	https://xcdjj.h5.xeknow.com/sl/4dhP32
react.js 高级用法	https://xcdjj.h5.xeknow.com/sl/3Hgtmz
react.js 基础	https://xcdjj.h5.xeknow.com/sl/3OFnM8
react.js 高级用法	https://xcdjj.h5.xeknow.com/sl/2fjxwk
redux & mobx	https://xcdjj.h5.xeknow.com/sl/1X2AGO
react.js 核心源码解析（上）	https://xcdjj.h5.xeknow.com/sl/fBRDu
react.js 核心源码解析（下）	https://xcdjj.h5.xeknow.com/sl/3ktWDM
react.js 路由及状态同构	https://xcdjj.h5.xeknow.com/sl/4Fux5j
react 实战-仿知乎页面框架	https://xcdjj.h5.xeknow.com/sl/Ch97p
react实战-知乎细节与高级特性	https://xcdjj.h5.xeknow.com/sl/lizLW
首页鉴权登录与用户消息推送	https://xcdjj.h5.xeknow.com/sl/4yRtWq

推荐学习资源

资源名称	链接
React 知识结构	https://juejin.cn/post/7020781173043363877
useEffect完整指南	https://overreacted.io/zh-hans/a-complete-guide-to-useeffect/
函数式组件与类组件有何不同	https://overreacted.io/zh-hans/how-are-function-components-different-from-classes/
React + Typescript 工程化治理实践	https://juejin.cn/post/6844903996826583048
React 源码学习笔记	https://juejin.cn/user/2831946593863688/posts
React 常见知识点总结	https://juejin.cn/post/6844904183934484494
React 常见面试题	https://juejin.cn/post/6941546135827775525

第四专题：Vue

在熟练掌握框架基本使用的基础上，掌握高阶方法例如 nextTick、slot 等等。这些高阶方法在 处理一些较为复杂的场景和排查问题的时候很有用，平时需要多累计，多看看文档，同样也可以多看一些好的项目来学习写法。官网的 api 你基本上要全部过一遍。并且你要利用一些高级的 api 去实现巧妙的封装。还需要掌握组件设计原则，可借鉴 Element-UI(例如通过 extend 实现一个 messageBox)，合理设计业务组件与基础组件。



爪哇课程

课程名称	链接
vue.js 基础	https://xcdjj.h5.xeknow.com/sl/495R2o
vue.js 高级用法	https://xcdjj.h5.xeknow.com/sl/PHDiE
vue-cli详解	https://xcdjj.h5.xeknow.com/sl/22KBHF
vue.js 状态管理及 ssr（路由及状态同构）	https://xcdjj.h5.xeknow.com/sl/ct9AK
vue.js 核心模块源码解析（上）	https://xcdjj.h5.xeknow.com/sl/4D6xYc
vue.js 核心模块源码解析（下）	https://xcdjj.h5.xeknow.com/sl/3ocV2x
vue.js 前端路由及异步组件	https://xcdjj.h5.xeknow.com/sl/FrTyU
vue.js 项目实战	https://xcdjj.h5.xeknow.com/sl/Y4R0J

推荐学习资源

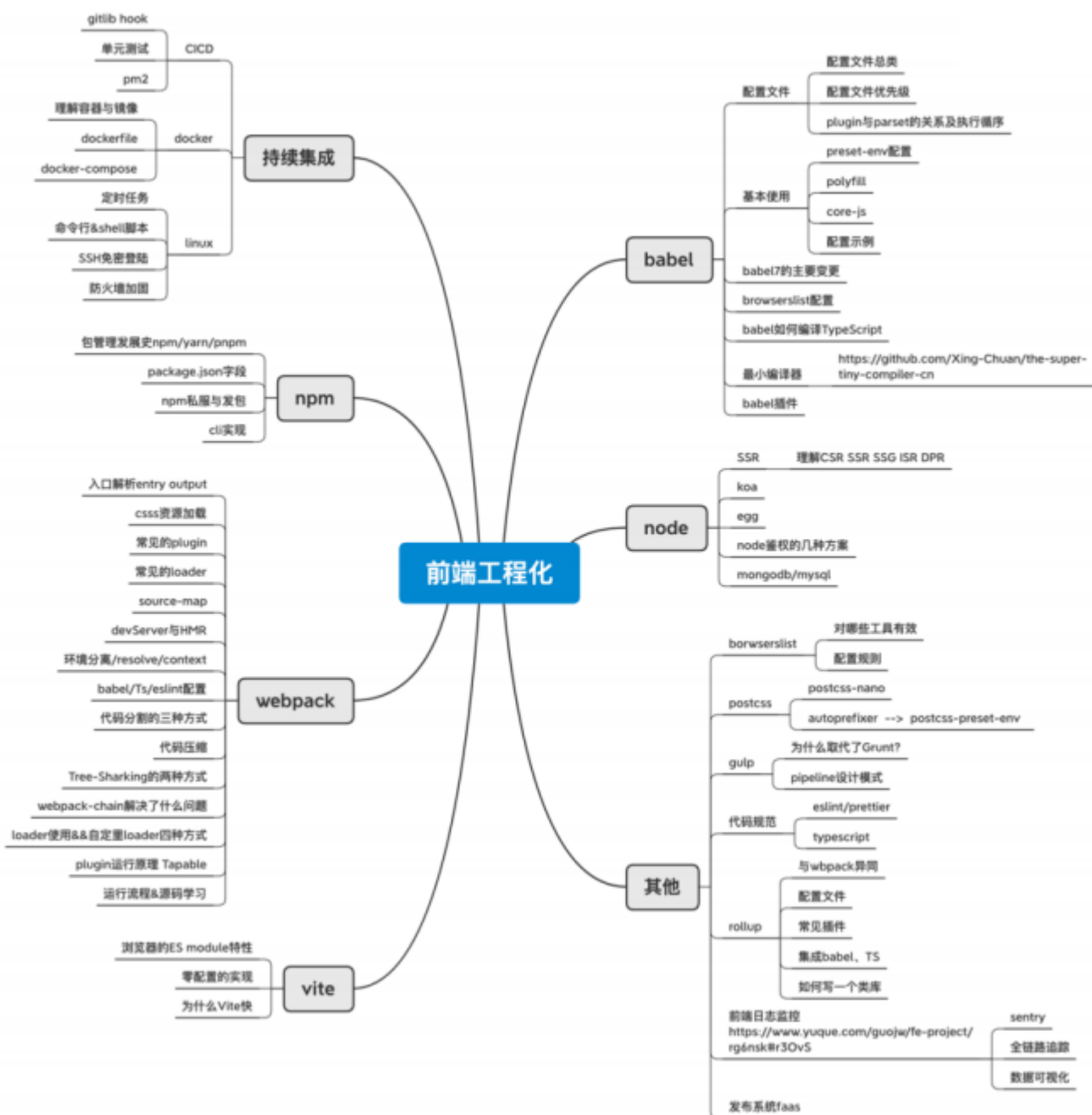
资源名称	链接
Vue基础知识结构	https://cloud.tencent.com/developer/article/1692379
Vue-admin(常见的组件封装、路由设计)	https://panjiachen.github.io/vue-element-admin-site/guide/
组件的实现(组件设计)	https://github.com/sl1673495/vue-netease-music/blob/master/src/base/confirm.vue
长列表渲染优化(性能优化实践)	https://zhuanlan.zhihu.com/p/66779396
vue.js技术揭秘	https://ustbhuangyi.github.io/vue-analysis/v2/prepare/
vue.js技术内幕	http://caibaojian.com/vue-design/art/

第五专题：前端工程化

掌握 webpack、babel 的用法，包括常用配置、优化手段等等。大厂还需掌握其实现原理，例如 webpack 热更新、自己写 loader、plugin 等。否则掌握其常用优化配置手段即可。除此之外还应该了解目前主流的打包工具如Vite，以及最新出的Turbopack。

爪哇课程

课程名称	链接
微前端（上）	https://xcdjj.h5.xeknow.com/sl/21HG4B
微前端（下）	https://xcdjj.h5.xeknow.com/sl/T9EQ9
前端AST	https://xcdjj.h5.xeknow.com/sl/2RPscz
模块化开发与webpack	https://xcdjj.h5.xeknow.com/sl/2v8lfq
自动化构建、测试、部署（上）	https://xcdjj.h5.xeknow.com/sl/2ACKMK
自动化构建、测试、部署（下）	https://xcdjj.h5.xeknow.com/sl/2Tz6Ls



推荐学习资源

资源名称	链接
常见 Webpack 面试题	https://juejin.cn/post/6844904094281236487
Webpack 打包产物分析(一)(结合前端模块化理解)	https://zhuanlan.zhihu.com/p/52826586
Webpack 打包产物分析(二)	https://zhuanlan.zhihu.com/p/53044886
读完这篇你还不懂 Babel 我给你寄口罩	https://juejin.cn/post/6844904065223098381
AST 入门	https://juejin.cn/post/6844904035271573511
AST 实践: try-catch	https://juejin.cn/post/6844903886898069511

第六专题：计算机网络与安全

这部分内容作为知识面的补充， 也是在面试中的常见题型，由于在平时的业务开发中很少会关注浏览器网络本身的底层逻辑，就需要同学对这部分内容进行专门的学习。常见考察的知识点包括但不限于以下：

- HTTP版本之间的区别
- CRRF & XSS
- 缓存
- HTTPS
- 常见状态码
- DNS
- TCP/UDP

以上也是列举了常见考察的知识点， 其中尤其如http1、2、3的区别以及CRRF和XSS常见的攻击方式和防御手段更是重中之重。除了以上知识点之外， 通常会结合实际场景题目来考察，例如比较典型的「从输入 URL 到回车网页渲染完成的整个过程发生了什么」这类问题，还可能结合网络请求优化、代码部署等问题。



爪哇课程

课程名称	链接
前端训练营-HTTP面试题	https://xcdjj.h5.xeknow.com/sl/3qnfDP

推荐学习资源

资源名称	链接
前端代码部署方案	https://www.zhihu.com/question/20790576/answer/32602154
细说浏览器输入 URL 后发生了什么	https://juejin.cn/post/6844904054074654728

第七专题：Node

Node作为JS服务端开发的基本， 虽然不如前几个知识点那么重要， 但偶尔也会在面试中遇见， 尤其Node事件循环机制与浏览器的区别， 更是常见问题。

建议先掌握 Node 的入门知识后， 选择一个常见的框架， 例如 Koa2 或 Express， 通过其文档了解基本 使用后， 尝试搭建一个简单的服务端项目， 并通过自己的前端项目进行联调。熟悉了框架的基本使用后 可以尝试去搭建一个博客系统， 这里会涉及到后端的架构设计、数据库连接等知识， 可以把重点放在登 陆模块、联表查询模块等重点核心模块的实现上。Node 的生态已经较为成熟， 我们常用的 CLI 脚手架 工具都是基于此。因此在学习 Node.js 过程中可以围绕脚手架、后端服务来展开。

爪哇课程

课程名称	链接
基于 Node 的后台技术	https://xcdjj.h5.xeknow.com/sl/21xSk1
node. js 基础教学	https://xcdjj.h5.xeknow.com/sl/2bnmqr
node. js 原理详解	https://xcdjj.h5.xeknow.com/sl/1E4Q5U
node. js 框架详解	https://xcdjj.h5.xeknow.com/sl/2LY5B4
node. js 网络 HTTP & 部署	https://xcdjj.h5.xeknow.com/sl/1ZyV9u
node. js 实战	https://xcdjj.h5.xeknow.com/sl/3hFI3u

推荐学习资源

资源名称	链接
Node. js 入门指南和实战	https://juejin.cn/post/6844904029219192839
推荐 9 个 Node. js 学习、进阶、 debugging 分析、实战 的重磅开源项目	https://juejin.cn/post/6961101653709684772
基于 Express 搭建博客项目	https://juejin.cn/post/6844903718345768973
Node. js 相关面试题	https://juejin.cn/post/6844904071501971469

第八专题：TypeScript

课程名称	链接
TS （上）	https://xcdjj.h5.xeknow.com/sl/Ollyq
TS （下）	https://xcdjj.h5.xeknow.com/sl/JgjGz

第九专题：设计模式

课程名称	链接
设计模式解析与实战（上）	https://xcdjj.h5.xeknow.com/sl/HBWXQ
设计模式解析与实战（下）	https://xcdjj.h5.xeknow.com/sl/3YYrgD

第十专题：解决方案与应用

课程名称	链接
性能优化	https://xcdjj.h5.xeknow.com/sl/1J8YGs

第十一专题：前端可视化

课程名称	链接
canvas 与 WebGL	https://xcdjj.h5.xeknow.com/sl/Fzb1p
可视化领域 —— 针对可视化领域的深度挖掘	https://xcdjj.h5.xeknow.com/sl/3AplLf
低代码领域 —— 切入最为火热的前端研究领域	https://xcdjj.h5.xeknow.com/sl/1oyCAU
低代码平台深度进阶 -进一步提升	https://xcdjj.h5.xeknow.com/sl/xUSHE

第十二专题：React-Native

课程名称	链接
RN入门与实战（上）	https://xcdjj.h5.xeknow.com/sl/3Uh5a3
RN入门与实战（下）	https://xcdjj.h5.xeknow.com/sl/1yj6MY

第十三专题：小程序

课程名称	链接
小程序开发基础	https://xcdjj.h5.xeknow.com/sl/2udoao
小程序开发框架解析	https://xcdjj.h5.xeknow.com/sl/4r7m9Z
小程序项目实战	https://xcdjj.h5.xeknow.com/sl/1JTO1K

第十四专题：Flutter

课程名称	链接
现代hybrid介绍与使用 、flutter 与 dart 开发	https://xcdjj.h5.xeknow.com/sl/CVji3
flutter 实战	https://xcdjj.h5.xeknow.com/sl/1bgibM
electron 入门与实战（上）	https://xcdjj.h5.xeknow.com/sl/3ugXzp
electron 入门与实战（下）	https://xcdjj.h5.xeknow.com/sl/muZb7

第十五专题：算法

先去找相关的入门文章，了解算法大致的分类，每个分类的典型例题需要掌握。对算法有了大概的概念后，去 LeetCode 平台刷题，难度以 medium 为主，hard 题掌握例题即可。刷题频率无需过高，保持手感即可，切记盲目疯狂刷。因为很多题目解法是相同的，万变不离其宗，需要自己多总结，多记录。建议按分类做题，但是要自己总结每个题型的解题模板和套路。比如滑动窗口、递归、动态规划等等这些都是有一些基本固定的套路的。按目录刷题的时候，做完一定要去看题解，这里不局限于 js 的题解，可以看看热门的几个解答。里面通常会给出解题的思路，这里就需要你总结出来，然后可以去找同类的题型去套用看看。一个类型基本上做两三道就有感觉了。重点在于总结。刷题不要过分追求数量，找到解题方式之后适当做一些变种题型即可。在把每种题型都过一遍之后，再去进行综合随机刷题即可。

爪哇课程

课程名称	链接
前端训练营-数据结构与算法	https://xcdjj.h5.xeknow.com/sl/31Ukxg
前端面试训练营（突击班）算法-数据结构篇	https://xcdjj.h5.xeknow.com/sl/3DEjkC
前端面试训练营（突击班）算法-排序与双指针等	https://xcdjj.h5.xeknow.com/sl/1kOtJZ
前端面试训练营（突击班）算法-二分与回溯	https://xcdjj.h5.xeknow.com/sl/2jxgD9
前端面试训练营（突击班）算法-贪心与动规	https://xcdjj.h5.xeknow.com/sl/3hDjTc

推荐学习资源

资源名称	链接
写给前端的算法进阶指南，我是如何两个月零基础刷 200 题	https://juejin.cn/post/6847009772500156429
算法随想录	https://programmercrl.com/

第十六专题：手写

在面试中常见的一种情况会需要面试者实现一些常见基础方法实现，如Promise、new、call、bind等，对于这些问题，只需要同学在平时闲暇自己实现一遍即可，同时需要注意下其中的一些细节逻辑。

推荐学习资源

资源名称	链接
前端手写题	https://www.yuque.com/guojw/yp2lh5/ftgy6q
手写 Promise 20 行	https://juejin.cn/post/6844904094079926286

简历准备

简历中的重点主要集中在项目经历部分。如果是要去好一些的公司，这一块的准备需要突出自己的亮点。面试中不可避免会被问到“项目中有遇到什么问题(或难点)”。一般的书写思路是 **star** 法则，即问题的背景+你的职责是什么+你做了什么样的事+产生了什么样的结果。这里主要围绕你做的工作以及一个可量化的结果。这里的可量化指的是最好有具体数据支撑，例如提升了 **xx** 秒，提高了 **xx%**，而不是空泛的一句“极大提升了页面的效率”。

如果在这方面没有过多的亮点，在书写简历的时候避免堆砌过多同质化的内容，例如两个项目之间使用的技术栈与做的工作基本类似。这种情况尽可能地多展示自己会的技术栈，即展示自己的广度。

总体来说还是希望学员在工作中能够找到可以提效或优化的地方作为自己的亮点。这一块值得单独花时间整理，最终呈现在简历上。

亮点

简历上是否有亮点是面试中很重要的一环，而这也是大多数人遇到的问题。大部分的亮点大致可以总结为三个方面：效率、优化、解决方案设计。


建议可以学习前端性能优化方面的相关知识作为准备，因为这部分的内容可以作为万金油在面试的时候展开叙述，并能作为自身工作的亮点。如FCP、TTI、CLS、LCP等相关前端性能指标的计算方式以及优化方式等。

查漏补缺

时间较为紧凑的情况下，不要盲目去盯着不会的新知识点，而是应该专注在巩固自己掌握的部分

必须会的知识点还是需要的，比如想进大厂的话需要刷算法题目，时间紧的话就是刷每个题型的典型例题，时间宽裕的情况可以有计划地对每个题型进入深入的研究，掌握题型的延伸变种等等。

在整个复习的过程中，要总结出自己的“杀手锏”，即自己最有信心的领域。例如可以是对源码的掌握度非常高，抑或是对于JS语言的理解很深入，项目打包优化有很多实践等等。以此在面试过程中可以给面试官留下印象

面试官不会因为你某一题不会就把你 **pass**，换句话说，你可以对某一块的知识点不那么熟悉，但是你不能对任何一块都不熟悉，这才是你被 **pass** 的原因 

岗位分析

看职位的时候，比较重要的是看职责描述。这里面一般来说会描述你之后会负责的产品和项目。通常来说，描述中会直接体现出工作负责的类型，例如小程序、数据可视化、低代码等等。因此在做面试准备的时候需要有相关的侧重点。比如你有小程序开发的相关经验和岗位职责温和，那么投递的时候简历中最好可以将其作为一个突出点来描述。