1. 基础知识

<https://juejin.im/post/5e7af0685188255dcf4a497e>

1. 技术进阶

<https://juejin.im/post/5e7c08bde51d455c4c66ddad>

1. js面试点
2. js的数据类型：null，undefined，string，number，boolean，symbol；object，function
3. 闭包：一个可以访问其他函数体内的变量的函数。可以访问函数外变量的现象。
4. 原型，原型链：每一个对象都有\_proto\_属性，其原型指向的是构造函数的prototype。当访问对象内某一属性或方法时，自身找不到的时候就会去找他的构造函数的prototype是否含有这个属性或方法。当构造函数也找不到时就继续寻找object的prototype，一直往下寻找，从而形成一个链。最终指向null。
5. 继承方式：原型链继承（父类的实例作为子类的原型），构造函数继承（父类的构造函数来增强子类的实例），组合继承（原型链+构造函数），寄生组合继承（）
6. 防抖：限制一个动作在规定时间内的实现次数。若超过次数，重新计算时间。
7. 全局变量：在整个程序可以调用，随程序销毁而销毁；局部变量：在当前函数或块作用域内可访问，函数执行结束后就不在了。
8. Es5和Es6的区别：系统库的引入/出方式，require/model.export，import/export；组件的定义，“React.createClass({ render(){ //dom }});”，React.Component{ // 生命周期，钩子函数 }；
9. Es6新增的：let,const，箭头函数，解构赋值，扩展符，set函数，模板字符串，for…of，class，promise
10. Promise：处理异步操作的对象，
11. Class类：类声明，类表达式，constructor方法创建一个对象，getter，method
12. Let、const：没有变量提升，仅在块作用域有效使用。
13. 数组的方法：forEach（）、map（）、reduce（）、find（）、indexOf（）、替换数组：filter（）、concat（）、slice（）、变更数组：push（）、pop（）、splice（）、shift（）、unshift（）、reverse（）、…..
14. 性能优化
15. 减少请求次数
16. 交互体验
17. Seo：
18. React
19. 生命周期：

挂载阶段：constructor() --- getDerivedStateFromProps() ---- render() --- componentDidMount()

更新阶段：getDerivedStateFromProps() ----shouldComponentUpdate()---render()---getSnapShotBeforeUpdate()----componentDidUpdate()

卸载阶段：componentWillUnmount()

1. 组件传参

父->子：prop，可以是参数，也可以是方法（用以子组件调用）

子->父：调用父组件的方法，eg：this.props.functionName(newVal)

1. 修改This指向：

在constructor中，this.funName = this.funName.bind(this)，推荐这个方法；

使用箭头函数；

在调用处使用bind或者箭头函数。

1. SetState：在事件处理中是异步的；
2. Constructor(props): 创建构造函数。主要涉及了两个方面，初始化内部state和绑定this指向（事件函数绑定实例）
3. Super（props）：使子类的constructor中能使用this.props
4. Vue
5. 生命周期：

Created（）-- beforeMount（）---- mounted（）--- beforeDestrory（）

1. 组件传参

父—子：prop，

子—父：$emit，

1. 双向绑定的原理：Object.defineProperty（）劫持各个属性的setter、getter。
2. V-if与v-show的区别：前者为条件渲染，true才渲染节点，具有更高的切换开销；后者都会渲染且留在dom中了，但以display决定是否显示，具有更高的初始渲染开销。若频繁切换，建议v-show。
3. Computed和watch：

前者是计算属性，具有getter（默认）、setter方法；后者是侦听器，更多的是用以处理异步或者开销较大的操作；

1. Vue-router：

this.$router访问路由器，返回的是路由实例对象，包括了跳转方法、钩子函数；this.$route访问当前路由，返回的是路由信息，包括了path、params、hash、query、fullpath、matched、name等参数

定义过程：

router = new VueRouter({routes}) 创建实例；

routes定义路由配置path和component；

new Vue({router})创建和挂载根实例

调用（展示）过程：

<router-link to=”链接”>

<router-view>路由出口

动态路由：”/:param”

声明式导航：router-link

编程式导航：$router.push()

1. 循环引用组件

调用自身：通过唯一的name

互相调用：。。。

1. 修饰符（.trim、.number、.lazy）、事件修饰符、按键修饰符、系统修饰符
2. 动态组件： :is = “activeComponent” activeComponent利用函数创建，内部import引入变量字符串
3. Vue-router：
4. param传参：类似于post，参数不会出现在浏览器导航栏，但刷新后就会消失；
5. query传参：类似于get，直接将参数暴露给了用户，刷新不会消失。
6. 命名视图：
7. Vuex：model（state、getter、mutation、action）

辅助函数

1. 浏览器的那些事
2. 兼容问题（css+js+hack）：

Promise 在IE9和低版本的常用浏览器内，并不支持。需要安装一个babel-polyfill ，然后在webpack中配置 entry: ["babel-polyfill", "./app/js"]

Css hack：解决低版本浏览器样式问题

1. 存储问题：

Cookie：一般大小限制在4kb，每次请求都会带上（浪费带宽流量），

Webstorage:一般大小限制在5M，节省带宽

1. url请求过程：
2. 调试工具：

element（HTML结构+css样式），

console（控制台输入信息），

source（当前站点所引用的js资源列表），

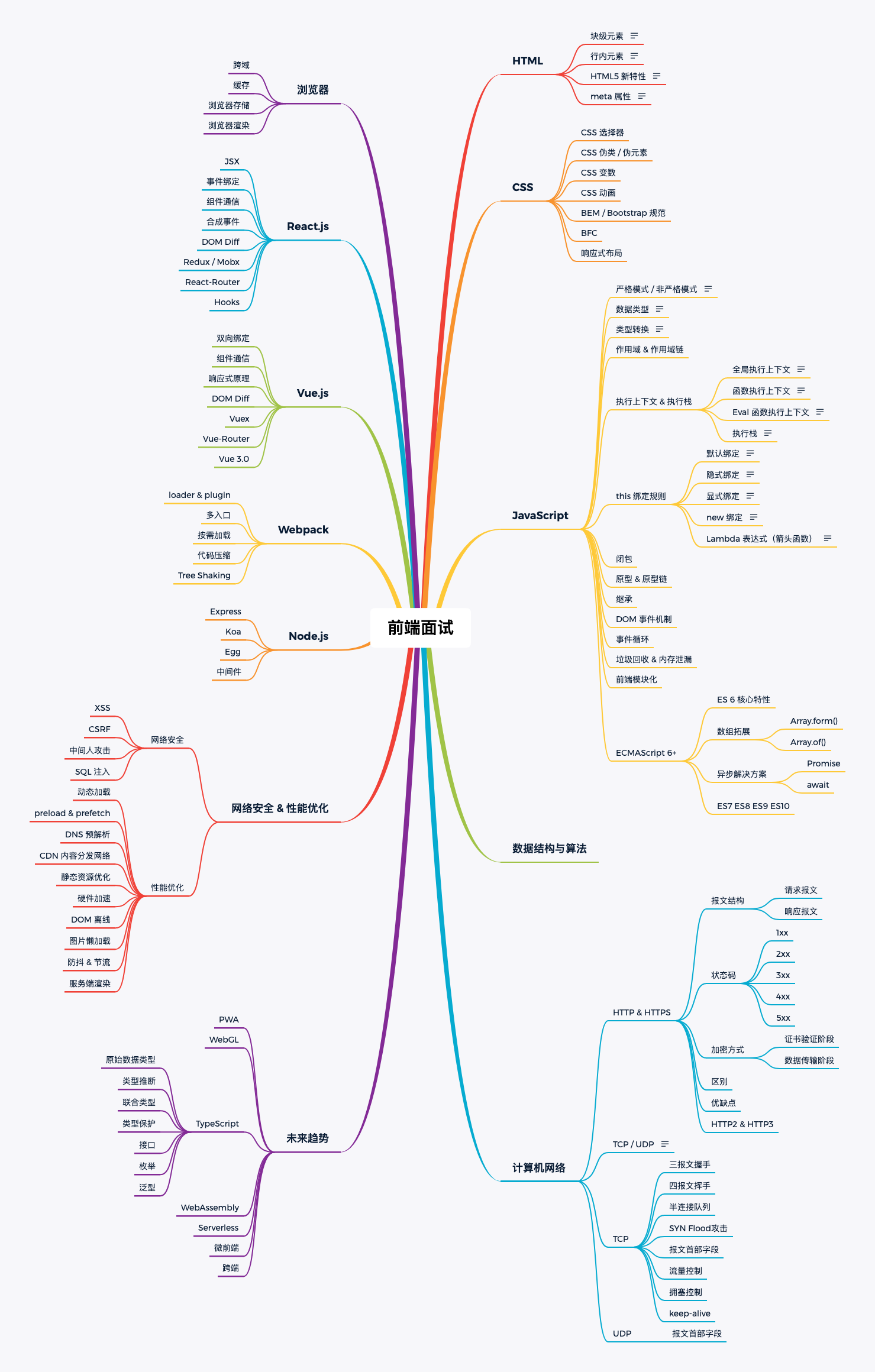
network（网站请求的网络资源，js,css,img,xhr），

application（本地存储数据），

memory，

performance，

audits

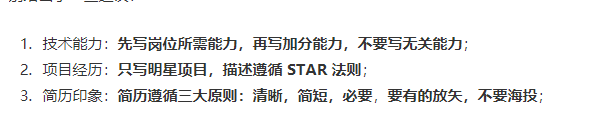


1. 性能优化

（1）

1. 微信小程序
2. Git版本管理

（2）简历要点



1. 个人面试反馈
2. 对于原生js要求较高；
3. React的生命周期、传参、类定义；
4. Vue的传参，状态管理，路由的传参；
5. 域名解析