js基础----基本类型，怎么判断基本类型，原型链，作用域，继承，闭包==>

防抖节流的思路， 响应式原理

1.css选择器有哪些，和权重级别，es6常用到的，flex居中怎么实现

2.冒泡机制，冒泡和捕获的区别

3.静态资源性能优化可以做哪些存储（不是数据存储，是静态资源存储，比如说JS，css第一次加载缓存资源，提高性能）

4.后台接口json数据层级多数据量大的时候前端是怎么获取的，有没有提高性能缩小代码量的方法

5.数据获取性能优化方面能做哪些优化，webpack打包流程 webpack优化

6自己项目可以突出自己优势的点和技术，项目中遇到最难解决的是什么，解决方式是什么

还有flex布局的详细属性有哪些？

1、 盒模型及区别 2、去除浮动 3、浏览器的缓存机制

4、js事件的执行机制 5、react 的生命周期 6、vue 的生命周期

7、vue v-if和 v-show 的区别 8、定位 absolute 和 fixed的区别

v-for key的作用，vue中写scope的原理

vue2.0监听数组 滚动弹窗穿透问题解决

数组排序，bind，call，apply的区别，跨域三要素，怎么处理的跨域

写代码：缓存机制，去重代码，防抖节流 ，一个字符串里面出现字符最多的字符出现了几次

1. **js继承**
2. **防抖节流**
3. **重绘回流**
4. **js数据类型、深拷贝**
5. **Es新属性**
6. **输入URL并回车时发生了什么？**
7. **redux基本概念及重点记录**
8. **dispatch更改state是异步，为什么？如何处理？**
9. **react生命周期**
10. **js基础（需提问）**

* **js作用域**：js中的变量若在当前作用域中没有，会向父级寻找，直到最顶层。
* **块级作用域**：js ES5没有块级作用域。如果要形成，可以通过()()自执行函数实现函数中的作用域为块级作用域；ES6中每个{}都是一个块级作用域。
* **null和undefined的区别**：null是变量被定义但无值，属于空指针。undefined为从未声明。
* **讲讲闭包理解**：指的是一个函数可以访问另一个函数作用域中的变量，外层变量不会被垃圾回收机制回收。优点是：避免全局变量污染。缺点是：容易造成内存泄漏。
* **为什么闭包会导致内存泄漏**？答：闭包内部的匿名函数在调用后把外层的变量也包含在自己的作用域内了。此变量属于一直在引用中，所以垃圾回收机制不会将他回收。导致会比其他function占用更多内存。
* **如何解决闭包导致的内存泄漏**？答：1.调用时先用变量保存闭包函数，在执行完成后为此变量赋值为null，清除指针即可使垃圾回收机制正常回收。2.可以借助()()自执行函数，在闭包内部不使用return function而是自执行函数取代，因为自执行函数是私有块级作用域，即可在执行结束后将引用对象销毁。
* **其他场景的内存泄漏及解决方案**：
  + setTimeout及setInterval未在使用后清理，或循环多个定时任务，clear时有缺漏。解决方案：每个定时器应使用变量保存，且在使用后clearTimeOut/clearInterval。
  + addEventerListener、jquery的on、vue的$on等引用dom后需要做解绑处理
    - addEventerListener使用removeEventerlistener解绑
    - Jquery.on使用.off解绑
    - vue的$on使用$off解绑
  + Vue、react框架需要在生命周期的销毁周期中执行解绑操作，以及定时器清除。
* **实现instaceof**：判断实例的\_\_proto\_\_和类的prototype是否三等，并且返回结果。
* script异步加载：defer、async、动态创建dom（原生）
  + defer是script脚本下载好，在dom加载完成后顺序加载
  + async是script脚本异步下载完成后自动加载
* 跨域问题：
  + React、vue中使用proxy设置域名即可
  + ajax使用JSONP
* 事件循环：js是单线程的。会优先执行宏任务，执行完全部宏任务后检测是否有微任务，有的话将微任务一起执行
* 微任务包括什么？答：例如setTimeout、.then等

js消息队列：主线程会将执行过程中遇到的异步请求发送给这个消息队列，等到主线程空闲时再来执行消息队列中的任务。

数组借用filter来遍历元素并按照规则筛选，返回新的数组

Flex：flex包含了flex-grow、flex-shrink、flex-basic

flex-grow：还有剩余空间时，为此元素安排多少的空间

Flex-shrink：此元素宽度超出父级时，压缩多少

Flex-basic：指定元素宽度，优先级高于width

盒模型：margin、border、padding、content

ie盒模型和普通的区别：ie的将border和padding计算至宽高内，而普通盒模型的宽高只计算content的真实宽高。

Box-sizing属性会更改盒模型的标准，content-box是普通盒模型，ie标准的是border-box。

获取真实宽高使用window.getComputedStyle.width/height最为标准。

行内元素margin上下无效，左右有效。padding均有效。

BFC：指一个通过特定条件达成的独立容器，容器内不不会受外部样式影响

达成条件：

**1.overflow除了visiable都可以，有scroll、hidden、auto。**

1. **float设置除了none都可以。**
2. **position设置了除了static都可以。**
3. **display设置：inline-block、table-cell、table-caption。**

特性：

**1.内部元素会在垂直方向一个接一个的放置。**

1. **垂直方向元素距离靠上下文margin最大值来设置。**
2. **bfc内区域不会和float区域重叠。**
3. **计算bfc区域高度时会将float的高度也算进去**
4. **容器内部的子元素不会影响外部元素**

margin重叠：借助bfc的概念，使原本两个受影响的元素，将其中一个增加bfc父级。因为bfc的独立容器属性，不会影响外部元素。

清除浮动：1.使用伪类或空白块元素占位。必须是display:block且增加clear：both。clearboth是使元素的左右两侧都不允许浮动元素存在。

1. 将父级设置为bfc，因为bfc区域高度会将float的高度也算入，所以不会使父级无高度。

除了bfc还有ifc：ifc是行内格式化书写：按照水平方向放置，一行放不下折行。不接受上下边距，只接受左右。

达成条件：在一个块元素中只有行内元素。

rem原理：通过设置html（根元素）的font-size。内部元素的1rem = 1\*fontsize

http和https区别：http为名文传输，不安全。https多了ssl加密协议，为加密传输。

前端怎么修改？所有js和css、静态文件加载https地址。

proxy跨域原理：首先是http和https会引发跨域，不同域名、不同端口都会引发跨域。proxy原理是npm run dev时node服务会将请求转发至proxy设置好的域名地址，所以跨域这一步是在服务端中间件解决的。

axios拦截器原理：axios实例可以通过设置实例的interceptor.request/response.use方法，来设置拿到数据后的一些列操作。原理：给axios原型设置可共享方法，如果有拦截器，会添加到axios返回的promise对象中，请求拦截会在发起前调用，相应拦截会在拿到response后调用，并根据拦截规则返回数据。

小程序冷热启动：冷启动是小程序已经被销毁。热启动是未被销毁，再次打开。

冷启动会重新走app中的onLunch，热启动不会触发，只会触发onshow周期。

为什么会被销毁？小程序占用内存过高或小程序置入后台超过5分钟。