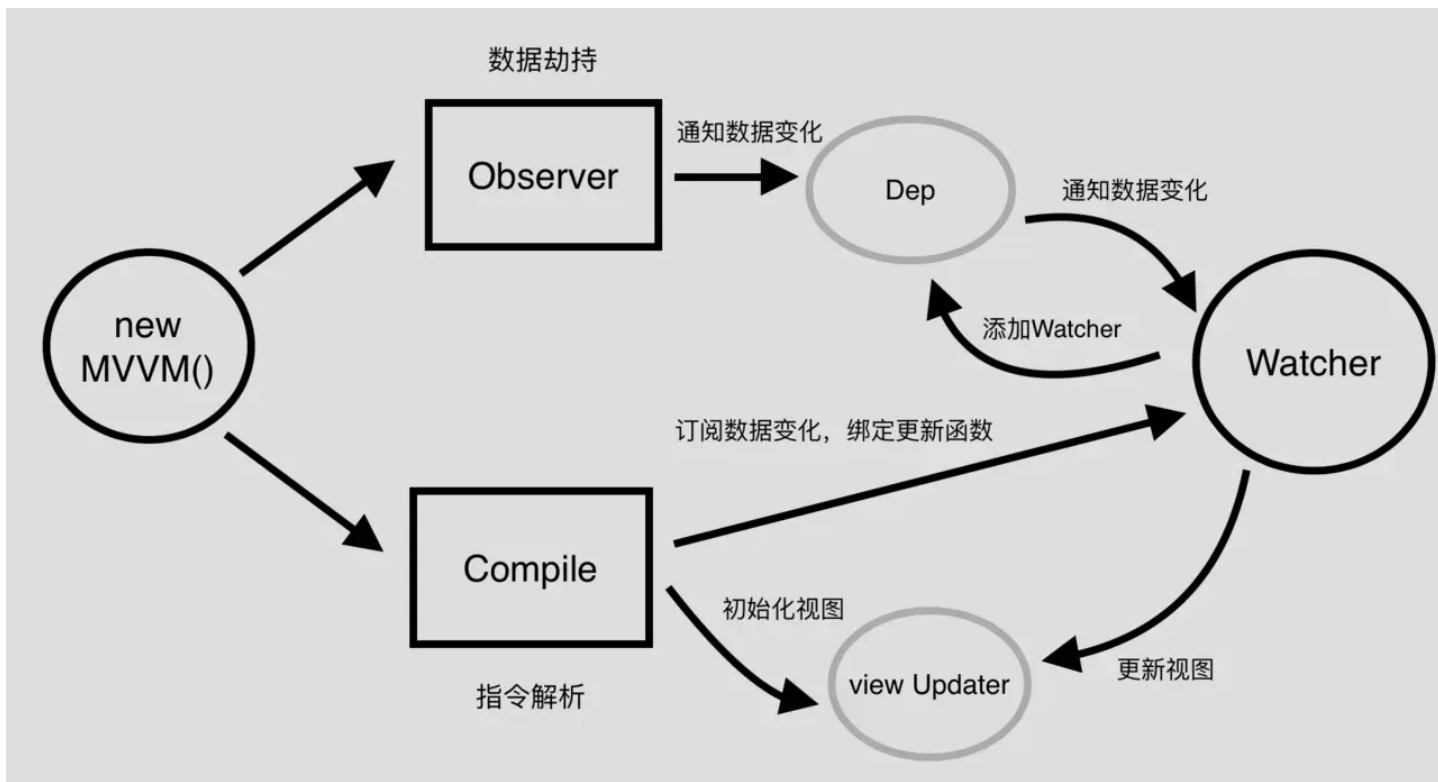


海康威视前端面试题

防抖节流

防抖节流



v-model怎么实现双向绑定

双向绑定是常考题，下面这张图片是官方有过的，每次问到这道题就应该想起这张图片。

- **是什么**
- 模板层的数据变更会导致视图层的更新。视图层的数据更新也会导致模板层的数据改变
- **原理（ViewModel 的原理）**
 - a. 监听器：对所有的数据进行监听
 - b. 解析器：对每个元素节点的指令进行解析
- **双向绑定的原理**
 1. vue的数据源会被劫持，在劫持的过程中为属性做依赖收集，vue中的观察者Watcher负责更新视图，依赖收集到的是观察者Watcher的实例对象。当属性值发生变更时会触发依赖，进而触发视图更新函数。

- 在数据劫持的同时，vue会编译模板，解析指令，当视图层的数据发生变更时，编译器中绑定的函数会被触发，进而获取到最新的数据值，再次通知Watcher去触发依赖。

加密算法

项目里的加密算法用的是md5。但是为了安全性应该使用其他加密算法。

- md5是一个哈希函数，用于将任意长度的输入转换为固定长度的输出，可用于加密。
 - 缺点：碰撞攻击：用两个不同的输入值经过md5算法得到相同的散列值
 - 预图攻击：通过已知的散列值，找到对应的原始数据。
- AES（对称加密算法）
 - 使用它需要引入CryptoJS库，基础加密使用 `const ciphertext = CryptoJS.AES.encrypt(content, key).toString();` content是需要加密的内容，key为密钥。
- RSA（非对称加密算法）
 - 需要引入jsencrypt库，需要创建jsencrypt实例并且定义公钥和私钥
 - 用于数据加密和数字签名，密钥太简单安全性不足。
- HMAC
 - 需要引入crypto模块，用于验证消息的完整性和真实性。
 - 需要选择合适的哈希函数和密钥，需要填充消息，其实就是确保消息长度达到算法要求。

除了本地存储，还有什么存储方法

本地存储，字节面试里面我总结很清楚了。背就完了。

路由守卫

路由守卫

计算属性有哪些？区别是什么

严格来讲只有computed是计算属性，watch是侦听器，methods是用来处理事件。但是他们都能对数据处理和计算。

- computed
 - 用于创建派生数据，即基于已有的响应式数据进行计算
 - 会对相关依赖进行缓存，只有在相关依赖发生改变时才会重新计算。
- watch:

- 用于监听数据的变化，当监测的数据发生变化时执行指定的回调函数。
- 适合用于需要在数据变化时执行一些异步操作、复杂逻辑处理或者非响应式数据的监控。
- **methods:**
 - 用于定义可以在 Vue 实例中调用的方法，这些方法可以包含任意的 JavaScript 代码。
 - 适合用于处理用户交互或者执行一些逻辑操作，方法中的逻辑不会被缓存，每次调用方法都会重新执行其中的逻辑。

因此，它们的本质区别在于 **computed** 用于派生数据的计算，**watch** 用于监听数据的变化并执行相应操作，**methods** 用于定义可以在 Vue 实例中调用的方法。

路由懒加载，图片懒加载怎么实现？

路由懒加载

父子组件将父组件方法传到子组件？

其实就是组件通信方式

Promise处理异步？.all方法

在处理多个异步操作时可能会存在函数里面调用其他函数的嵌套情况，代码量一多就会出现回调地狱。这样代码维护困难，排查错误困难。

通过 Promise 的 then 方法，你可以按顺序连接多个异步操作，并在每个 then 方法中处理对应操作的结果，而不需要层层嵌套回调函数。

- **all方法**
- 他是一种处理异步的操作，是一种静态方法，接收Promise对象组成的数组作为参数，可以将多个 Promise 包装成一个新的 Promise 对象， Promise.all 可以并行执行多个异步操作，这是其最大的优势。它不会等待每个 Promise 依次完成，而是同时开始所有的异步操作。，都成功会返回一个对象，其中有一个失败整个调用都会失败。

箭头函数的this值指向

this的指向

深浅拷贝

深浅拷贝

axios

这个我们使用的时候只用安装引入就可以了，可以用它对我们的接口请求进行封装，也可以用于对服务器返回的数据异常捕捉等。功能非常强大。

Axios 是一个流行的基于 Promise 的 HTTP 客户端，可以在浏览器和 Node.js 环境中使用。它可以用于进行 HTTP 请求并处理响应数据，提供了许多功能，比如拦截请求和响应、转换请求和响应数据等。

以下是 Axios 的一些主要特点和用法：

1. **Promise API**：Axios 使用 Promise API 处理异步请求，支持 `async/await` 语法，使得处理异步操作更加简洁和易读。
2. **浏览器和 Node.js 都可用**：Axios 可以在浏览器和 Node.js 环境中使用，这使得它成为编写跨平台应用程序的理想选择。
3. **拦截器**：Axios 允许你在请求或响应被 `then` 或 `catch` 处理之前拦截它们。这意味着你可以对它们进行修改、日志记录或错误处理等操作。
4. **请求和响应转换**：Axios 允许你在请求和响应时对数据进行自定义转换，例如将 JavaScript 对象转换为 JSON 数据或将 JSON 数据转换为 JavaScript 对象。
5. **取消请求**：Axios 允许你使用取消令牌来取消请求，这对于需要在用户离开页面时取消请求非常有用。
6. **客户端支持防止 CSRF 攻击**：Axios 可以支持在浏览器中防止跨站请求伪造（CSRF）攻击。