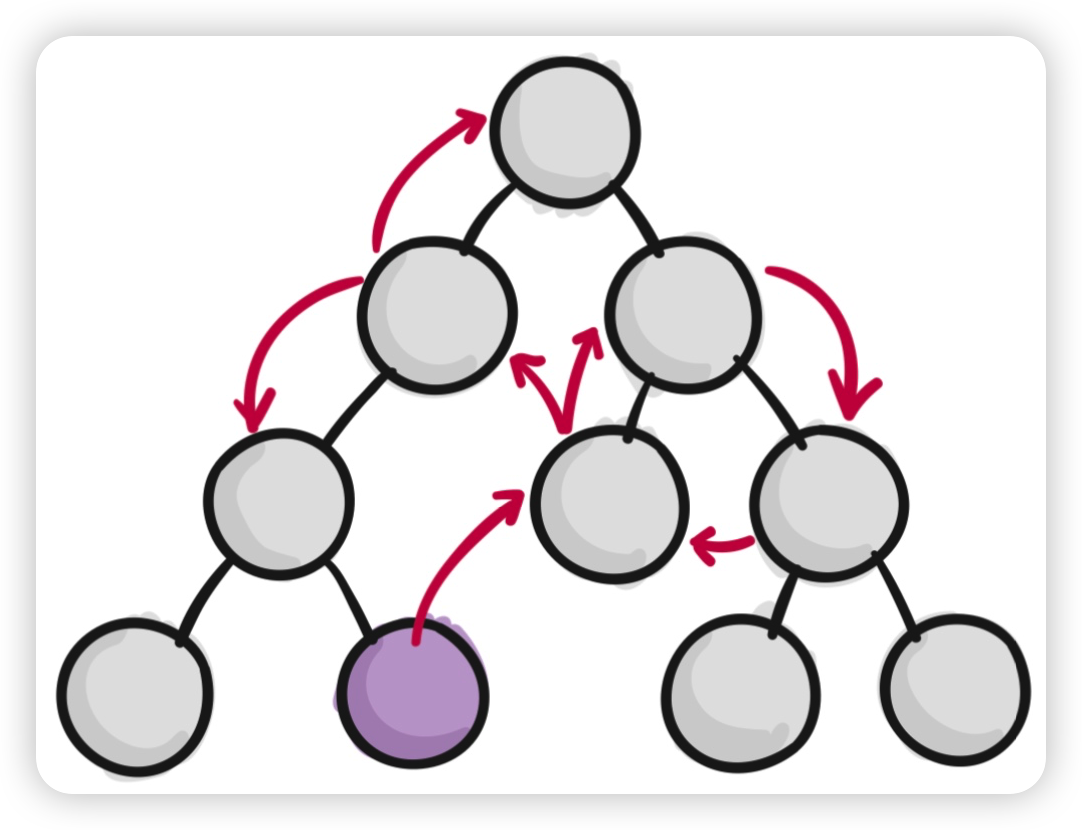
面试题:

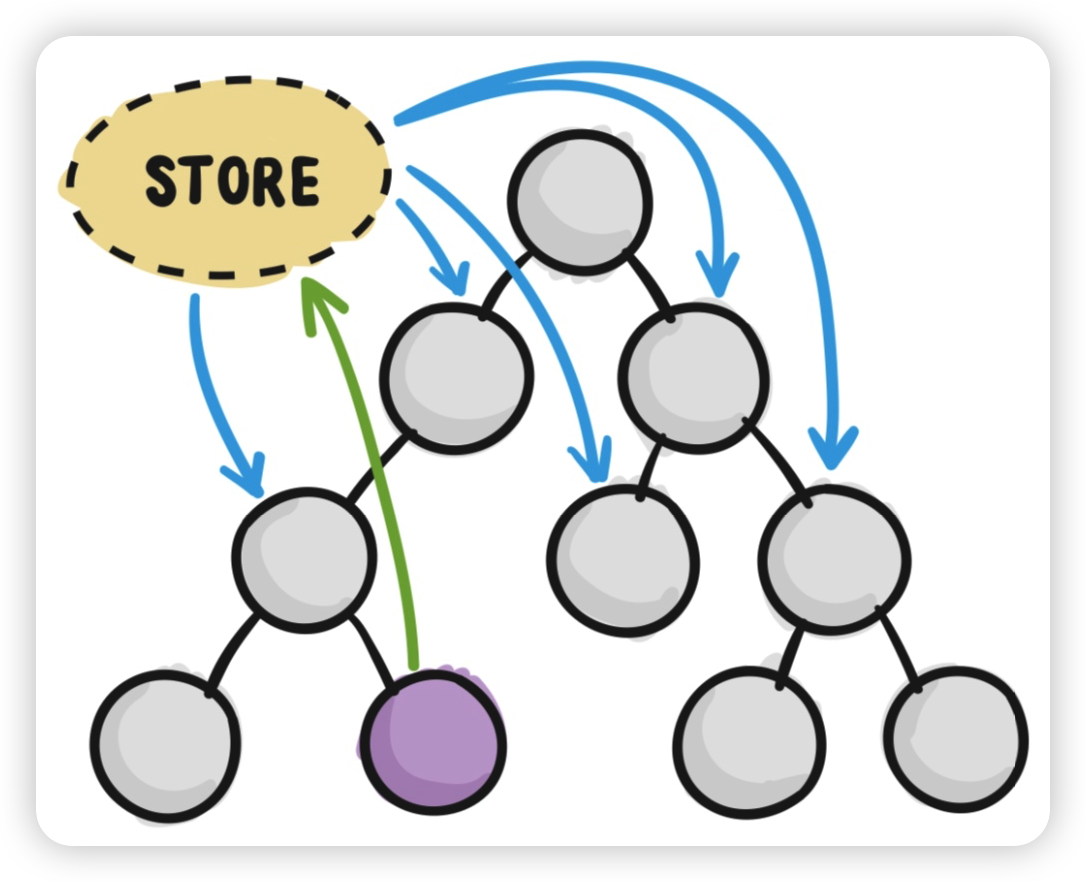
在目前的vue应用中，使用状态管理库进行状态管理时有哪些最佳实践，请列举一至两条。

最佳实践：

1. 分离状态逻辑和业务逻辑（为什么使用状态管理库）
   1. 实际上这个就是我们使用状态管理库的目的，我们使用状态管理库就是为了将组件的状态分离出来，这样可以方便我们维护，也方便组件之间进行状态的共享。
   2. 没有使用状态管理库：



* 1. 使用状态管理库之后:



⭐需要注意：并非所有的vue应用都需要使用状态管理库，这个取决于我们所开发的应用的规模大小，如果只是小规模的vue应用，使用状态管理库反而显得更麻烦。

1. 选择Pinia来进行状态管理

目前vue官方已经推荐开发者使用Pinia来替代Vuex作为状态管理库，可以将Pinia看作是VueX5.x.

相比Vuex，Pinia真的真的真的很轻量，大小只有1kb左右，基本可以忽略

当然相比之前的vuex，还有一些其他优点：

<https://pinia.vuejs.org/zh/introduction.html#comparison-with-vuex>

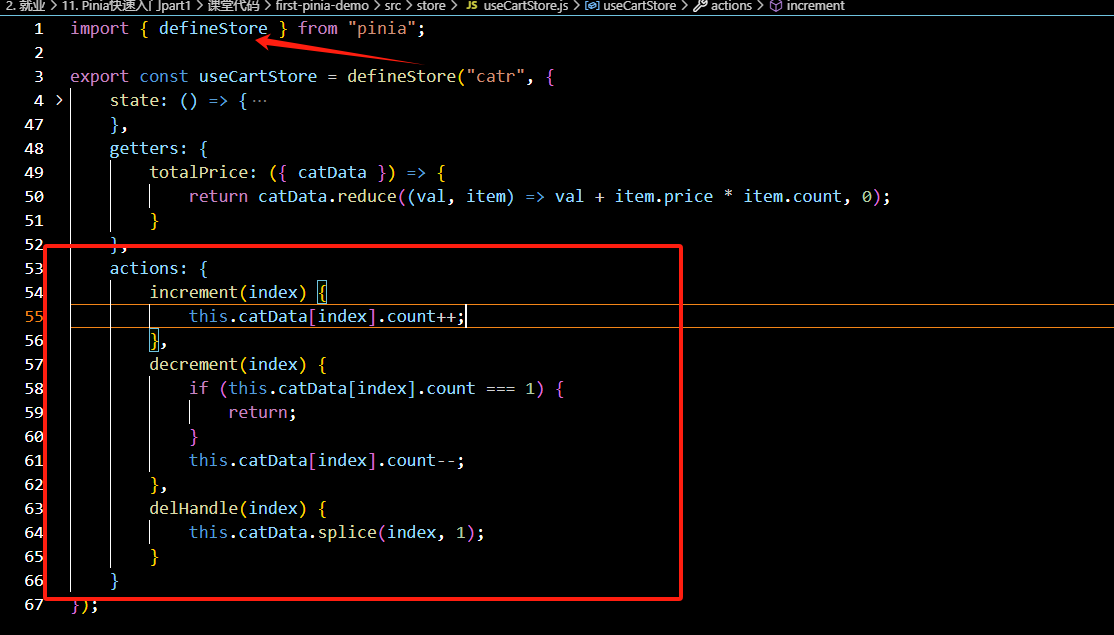
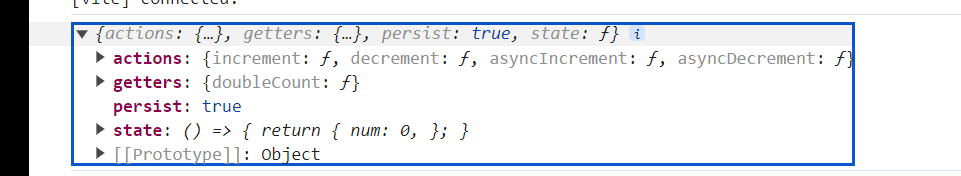
另外如果你之前的项目使用的是vuex可以看一下官方的迁移指南

1. 避免直接操作store的状态
   1. 虽然我们可以直接操作store的状态，但是在Pinia我们最好还是避免直接操作store里面的状态，而是通过对应的getters，来读取，actions来修改。
   2. 好处就是提高了代码的可维护性，因为数据的改变始终来自于actions的方法，而不是分散于组件的各个部分。
2. 使用TypeScript
   1. Pinia本身就是使用typeScript编写的，因此我们在使用pinia的时候，能够非常方便的，非常自然的使用typescript，使用typescript可以更好的提供类型检查和代码提示，让我们的代码更加可靠和易于维护。
   2. 官方文档对应：

https://pinia.vuejs.org/zh/core-concepts/state.html#typescript

1. 将状态划分为多个模块
   1. 在一个大型应用中，如果将所有组件的状态放置在一个状态仓库中，那么会显得该状态仓库非常的臃肿，因此一般在大型的项目中，是一定会将状态仓库进行拆分的。
   2. 在早期的vuex中就已经支持将状态仓库按照不同的功能模块进行拆分，只不过在vuex时期，状态仓库拆分时按照的是嵌套的方式进行代码组织的。
   3. 在Pinia中，组织状态仓库的形式不在采用像Vuex一样的嵌套，而是采用的是扁平化的设计，每一个状态仓库都是独立的，这个也是Pinia名字的来源。

补充内容：

1. 辅助函数
   1. 
   2. 
2. 订阅state以及action
   1. 
3. 插件选项
   1. 
   2. 在插件中如何获取传递给插件的数据