1. 谈一谈你对MVVM的理解。

MVVM就是 model view viewModel 的缩写。Model代表了数据层，只关心数据。view代表了视图层，只关心视图。而viewModel代表的就是view 和 model的桥梁，是业务逻辑。viewModel提供view所需要的数据，响应view中的用户操作带来的变化。利用双向绑定，似的model数据变化时，viewModel自动更新，，而viewModel更新后，view也会自动变化。总结一句话，MVVM简化了界面与业务的依赖，解决了数据频繁更新。

1. V-if和v-show的区别。
2. dom层：v-if操作dom添加删除，为false时，不在html中生成dom。v-show只会控制css display，始终存在于html中 。
3. 原理层：v-if 逻辑生成的render函数时 (show)?\_c(xxx):\_e();v-show只会处理style
4. 双向绑定的原理

Vue.js是数据劫持+发布订阅者模式实现双向绑定。通过Object.defineProperty()来劫持各个属性的getter和setter，在数据发生变化时，发布消息给订阅者，触发响应的监听回调。

具体实现分为Object.defineProperty，complie，watcher

1. vue遍历data属性，给属性加上getter，setter，这样在改变属性时就可以触发setter，实现监听数据变化。
2. Complie解析模板指令，将模板中的变量替换为数据，然后初始化渲染页面视图，并为每个对应的节点绑定对应的更新函数，添加监听数据的订阅者，一旦setter触发，则调用对应的更新函数，更新视图。
3. Watcher是Observer和complie的桥梁，主要功能：1）把自己添加到属性订阅器（dep）中添加自己。2）包含一个update方法。3）当属性发生变化时dep.notice()通知时，调用自生的update方法，触发complie绑定的回调，实现视图更新。

总结一句话：通过Observer来监听自己的model数据变化，通过Compile来解析编译模板指令，最终利用Watcher搭起Observer和Compile之间的通信桥梁，达到数据变化 -> 视图更新；视图交互变化(input) -> 数据model变更的双向绑定效果

1. vuex原理
2. Event Loop
3. 宏事件 微事件