场景1：  
如果你的技术团队技术选型比较保守，没有新技术的使用场景，比如让你去维护一个管理后台。管理后台，日常就是提交各种表单了，这部分现有的方案，比如form表单提交或者jquery收集信息ajax提交。这时候你可以把vue 当成一个js库来使用，就用来收集form表单，和表单验证。

场景2：  
在场景1中，你尝到了甜头。心中暗爽，还可以这么玩嘞。独乐乐不如众乐乐和大家分享之后，团队开始接受使用vue，小规模推广起来。打怪升级该遇小boss啦，领导说，小伙后台做的越来越有效率了，来承担些常规业务开发，来正规军编制和其他小伙伴一起做新闻列表和新闻详情页吧。  
在这个项目中，你跟大胆一点把 整个页面的dom 用Vue 来管理，你发现jquery 没什么用了，列表用v-for来循环，把评论抽成小组件了。 评论交互比较复杂，但是你的关注点把原来jquery dom操作变成了关注数据的变化，用数据驱动DOM的变化

场景3:  
经历了场景1 场景2，越来越受大家信赖，领导又找你了。你看新闻项目你做的不错，移动站也得重构了，你做个移动端m站吧，正好微信和App分享出去用到。这时候，你需要在做移动端webapp了。 于是你由去了解 webpack vue-router，你发现前端 可以控制路由了。webpack，可以用于前端开发的工程构建。

场景4:  
场景3之后你在技术团队大放异彩，公司越来越器重你。年末领导又来找你了，小伙砸，想不想拿年终奖啊，想的话给我做个新闻直播间吧。需求特别简单：就是滚动播报新闻，用户实时参与评论。有了场景3和之前的经验，发现稍微有点力不从心了，你和后端的接口沟通上越来越频繁，新闻直播间需要大量的数据在组件中共享数据，后来你发现了，vuex 处理数据在组件之间的流动得心应手。场景5:  
年终奖，拿到手了，过年回来升职加薪。带了20人的前端团队，你的精力开始在配合公司其他部门做用户数据增长了。发现场景2中你犯了个错误，虽然整个页面用Vue 管理 开发起来很方便，但是页面白屏时间长，而且类似这样的底层页对seo都不好。开始考虑使用 vue2.0的SSR。为了保障团队高质量的输出，你开始研究如何给 vm写单测...

场景1-5 从最初的只因多看你一眼而用了前端js库，一直到最后的大型项目决方案。能否给你说明白vue 是 The ProgressiveJavaScript Framework  
