1. 列表渲染中的替换数组

example1.items = example1.items.filter(function (item) {

return item.message.match(/Foo/)

})

和对象变更检查注意事项中的Vue.set( )；

1. 事件处理中的事件修饰符 .passive（该修饰符表示就是设置{passive:true}，表示处理事件函数中不会调用preventDefault函数，就会减少了额外的监听，从而提高了性能）；
2. Prop中传一个对象的所有属性，prop验证类型检测中自定义构造函数；
3. 非prop的attribute的理解（增加到组件的根元素），禁用attribute中的inheritAttrs: false 结合 $attrs 的使用（结合面试题2中来理解）；
4. 自定义组件的v-model的 model选项，原生事件绑定 .native修饰符和$listeners属性，.sync修饰符（类似于 ’update: myPropName’ 模式触发事件）；
5. 插槽的使用（内容分发的出口，默认插槽、具名插槽…）；
6. 特殊属性is 和内置组件 component，渲染一个“元组件”为动态组件，依 is 的值，来决定哪个组件被渲染。在动态组件上使用keep-alive，及keep-alive的使用（看api）；
7. 程序化事件监听器中的

this.$once('hook:beforeDestroy', function () {

picker.destroy()

})

1. 混入对象的选项合并；
2. 自定义指令基础使用；
3. 渲染函数的基本使用；
4. 导航守卫和数据获取（比如：动态路由匹配中当路由路径未改变，路由的参数变化监听beforeRouteUpdate）；