Vue框架问题（技术支持组提供）答案（技术支持组提供）

1、关于Vue实例的生命周期

实例从  
beforeCreate -> 实例创建之前  
created -> 实例创建完毕  
beforeMount -> 实例挂载之前  
mounted -> 实例挂载完毕  
 beforeUpdate -> 如果有更新，在mount阶段进行，更新之前  
 updated 如果有更新，在mount阶段进行，更新完毕  
beforeDestory -> 实例销毁之前  
destoryed 实例销毁完毕  
  
钩子函数大部分情况下用来处理业务方面相关的逻辑，在Vue中没有控制器一说，通过钩子函数实现这个功能

2、在做文本渲染的时候如何保证将数据原样输出？

数据渲染三种实现方式：  
v-html：可以输出真正的HTML，但考虑到XSS，不建议这么做  
v-text：纯文本输出  
{{}}：与v-text功能相同，写法不同而已

3、Vue中的过滤器是如何来传递参数的？

过滤器定义：在Vue实例中通过filters对象下进行设置指定的过滤器  
过滤器使用：在渲染数据的时候通过 | 管道来进行对过滤器的使用  
过滤器传参：默认数据在使用过滤器的时候，数据源作为过滤器的第一个参数，如果需要传递更多的参数，在过滤器后面添加（参数2，参数3）即可

4、Vue中methods和computed有什么区别？应用场景如何？

首先，methods和computed都可以实现相同的功能，不同的是computed中所计算的依赖值没有发生变化，那么computed将不会执行，只会使用缓存，而methods每次都会执行，两者之间存在一个惰性的问题  
如果我们希望结果值是每次都需要精确算出来的，无论依赖值是否改变，那么我们使用methods，如果我们希望可以用缓存来缓解因为大量计算带来的时间及性能问题，那么使用computed

5、Vue中如何高效复用一个元素，如何解除对元素的复用？

Vue会尽可能高效的渲染元素，通常会复用已有元素而不是从头进行渲染，那么解除这种渲染方式，我们可以在当前元素上面增加key属性，表示他的唯一性，那么Vue的渲染的时候就会重新对元素渲染而不是进行复用了

6、Vue如何处理原生的DOM事件？

Vue在绑定事件的时候通过v-on来进行绑定，通过对处理函数传参的方式来获取原生的事件对象，这个参数是特殊表明的，为$event，通过原生的DOM事件对象来调用preventDerault()和stopPropagation()是很常见的问题

7、除了通过原生的DOM事件对象来解决冒泡等常见问题，Vue自身有什么方式来解决这类需求？

Vue提供了一系列的事件修饰符：  
.top：阻止事件冒泡  
.prevent：阻止默认行为  
.capture：使用事件捕获行为  
.self：表明必须当期那元素本身触发事件监听器  
.once：事件只触发一次

8、Vue为什么要在HTML中来监听事件？

Vue事件处理是绑定在当前视图的ViewModel上的，不会带来维护上的困难，并且我们不需要再ViewModel中操作任何与DOM相关的事情，对DOM完全解耦，在当前的ViewModel被销毁的时候，我们也不需要考虑事件处理器如何清理，因为Vue会自动删除这些处理器

9、Vue中父子组件是如何通信的？

父->子：  
父组件通过绑定属性的方式，先将数据绑定到组件上面  
子组件通过prop属性来接受传递过来的数据  
子->父  
子组件通过v-on注册事件，在事件处理函数中通过$emit来进行发射一个事件以及携带的参数  
父组件在子组件元素上面通过v-on注册子组件发射的事件，通过事件处理函数来接受传递的数据  
<https://cn.vuejs.org/v2/guide/components.html>

10、非父子组件如何进行通信？

如果项目较简单，非父子组件通信不是很复杂，我们可以借助一个新的Vue实例，分别通过$emit和$on来进行发射事件和监听事件来达到数据传递的目的，如何项目比较庞大，数据交互复杂，我们就要考虑使用vuex来实现数据的管理了

11、Vue如何实现目标的过渡？

Vue中在渲染目标元素的时候，先会嗅探目标元素有没有应用了CSS过渡或者动画，如果有应用，在恰当的实际进行添加和删除CSS类名。  
对于要想参与过渡的元素，我们通过transtion元素进行外层包裹即可，如过渡的实现方式要自定义，可以对transtion进行添加name属性，通过enter，enter-active，leave，leave-active四类样式上面实现具体的效果

12、Vue中双向数据绑定是如何实现的？

首先双向数据的绑定需要通过v-model来将数据关联起来，其次在data中一定要定义好通过v-model关联的数据，Vue在实现数据监听只能够监听到data中的数据，如果在data中没有定义改数据，那么将无法实现监听，同时也无法实现表单的双向数据绑定

13、Vue中如何实现路由的分发?

Vue实现路由分发借助官方推荐vue-router，通过配置vue-router的配置文件项，通过实例化一个router对象，关于router配置项，会有如下几个option:  
mode：路由的模式  
base：路由的根路径  
routes：路由详细参数   
 path：路由  
 name：命名路由  
 component：对应路由要加载的组件  
vue-router通过匹配不同的路由来动态加载不同的组件

14、Vue通过vue-router配置了路由规则，但组件未显示，为什么？

在使用vue-router的时候，通过router-link来当做路由跳转的开始，配置文件所加载的内容在匹配到当前路由的时候，加载的component会在router-view中来做显示，如果只有使用router-link而当前组件中没有router-view的话，组件是显示不出来的

15、Vue在通过vue-router进行组件加载的时候，如何传递参数？

在router-link中，通过:to来进行配置路由的走向以及参数等，通产情况下我们通过对象的方式来进行传递参数，传递的下表为params:{属性名:属性值}，在路由所匹配到的component组件中，通过this.$route.params.属性名来进行获取参数的值

16、Vue如何对数据进行监听？

在实例化Vue对象的时候，option选项中有watch可以用来实现对data数据中的数据进行监听，监听的方式如下：  
data数据：function(旧数据，新数据){}  
我们可以在函数中实现对数据的监听以及修改数据等操作

17、Vue中的修饰符有哪些？都有什么作用？

.lazy：默认情况下表单在实现双向数据绑定是监听输入框的input事件，通过lazy可以将input事件转换成change事件来进行监听  
.number：自动将用户数据的值转换为number类型，如果转换结果是NaN的话，那么直接返回原来值  
.trim：将用户输入的数据自动过滤收尾空格

18、如何确保一个回调在DOM更新循环结束后能够准确的调用？

如果我们希望在虚拟DOM更新完毕后进行处理业务逻辑，那么就需要通过Vue实例对象的nextTick()进行处理，nextTick里的处理方法确保DOM已经经过更新

19、在Vue中如何精确获取到想要的DOM元素？在虚拟DOM中提倡这么做么？

在Vue中不提倡直接来操作DOM元素，但如果有这方面的需求，我们可以通过在指定元素上面添加ref属性，类似ID的属性，通过this.$refs.属性值来进行获取指定的DOM元素

20、如何提高Vue在渲染组件或元素的效率？

对于重复性的数据不进行变化的组件，可以通过keep-alive来进行保留组件的状态，以避免重新渲染这个组件，对组件进行复用，提高渲染的效率