1、React 中 keys 的作用是什么？

答：key是React中用于追踪哪些列表中元素被修改、删除或者被添加的辅助标识。在diff算法中，key用来判断该元素节点是被移动过来的还是新创建的元素，减少不必要的元素重复渲染。

2、React 的工作原理？

答：React 会创建一个虚拟 DOM(virtual DOM)。当一个组件中的状态改变时，React 首先会通过 "diffing" 算法来标记虚拟 DOM 中的改变，第二步是调节(reconciliation)，会用 diff 的结果来更新 DOM。

3、类组件(Class component)和函数式组件(Functional component)之间有何不同？

答：类组件不仅允许你使用更多额外的功能，如组件自身的状态和生命周期钩子，也能使组件直接访问 store 并维持状态

当组件仅是接收 props，并将组件自身渲染到页面时，该组件就是一个 '无状态组件(stateless component)'，可以使用一个纯函数来创建这样的组件。这种组件也被称为哑组件(dumb components)或展示组件。

4、何为高阶组件？

答：高阶组件就是一个以组件作为参数，并返回一个组件的函数。高阶组件的作用就为了提高组件之间的代码复用。如果组件有某些相同的逻辑，那我们可以将这些逻辑抽离出来，放到高阶组件中进行复用，高阶组件和参数组件使用props传递数据。

5、使用箭头函数(arrow functions)的优点是什么？

答：1、作用域安全：在箭头函数之前，每一个新创建的函数都有定义自身的 this 值(在构造函数中是新对象；在严格模式下，函数调用中的 this 是未定义的；如果函数被称为“对象方法”，则为基础对象等)，但箭头函数不会，它会使用封闭执行上下文的 this 值。

2、简单：箭头函数易于阅读和书写

3、清晰：当一切都是一个箭头函数，任何常规函数都可以立即用于定义作用域。开发者总是可以查找 next-higher 函数语句，以查看 this 的值。

6、为什么建议传递给 setState 的参数是一个 callback 而不是一个对象？

答：因为 this.props 和 this.state 的更新可能是异步的，不能依赖它们的值去计算下一个 state。

7、何为 JSX？

答：JSX 是 JavaScript 语法的一种语法扩展，并拥有 JavaScript 的全部功能。JSX 生产 React "元素"，你可以将任何的 JavaScript 表达式封装在花括号里，然后将其嵌入到 JSX 中。在编译完成之后，JSX 表达式就变成了常规的 JavaScript 对象，这意味着你可以在 if 语句和 for 循环内部使用 JSX，将它赋值给变量，接受它作为参数，并从函数中返回它。

8、何为 redux？

答：Redux 的基本思想是整个应用的 state 保持在一个单一的 store 中。store 就是一个简单的 javascript 对象，而改变应用 state 的唯一方式是在应用中触发 actions，然后为这些 actions 编写 reducers 来修改 state。整个 state 转化是在 reducers 中完成，并且不应该有任何副作用。

9、何为纯函数？

答：一个纯函数是一个不依赖于且不改变其作用域之外的变量状态的函数，这也意味着一个纯函数对于同样的参数总是返回同样的结果。

10、react生命周期的三个阶段？

答：初始化阶段；更新阶段；销毁阶段。

11、call、apply、bind 的用法以及区别？

答：相同之处：改变函数体内 this 的指向。

不同之处： call、apply的区别：接受参数的方式不一样。

bind：不立即执行。而apply、call 立即执行。

12、将以下函数改为箭头函数。

const fn = function(a){

console.log(a);

};

fn(2);

答：const fn = (a) => {

console.log(a);

};

fn(2);

13、var、const、let的区别？

答：1.var定义变量，没有块的概念，可以跨块访问，不能跨函数访问，不初始化出现undefined，不会报错。

2.let定义变量，只能在块作用域里访问，也不能跨函数访问，对函数外部无影响。

3.const定义常量，只能在块作用域里访问，也不能跨函数访问，使用时必须初始化(即必须赋值)，而且不能修改。