**类组件和函数组件之间的区别是啥？**

函数组件的性能比类组件的性能要高，类组件使用的时候要实例化，而函数组件直接执行函数取返回结果即可。

函数组件没有this 没有生命周期 没有state状态

**state 和 props 区别是啥？**

state 是组件自己管理数据，控制自己的状态，可变；  
props 是外部传入的数据参数，不可变；

**在构造函数调用 super 并将 props 作为参数传入的作用是啥？super(props);**

在调用 super() 方法之前，子类构造函数无法使用this引用，ES6 子类也是如此。将 props 参数传递给 super() 调用的主要原因是在子构造函数中能够通过this.props来获取传入的 props。

**什么是 React Context?**

Context 通过组件树提供了一个传递数据的方法，从而避免了在每一个层级手动的传递 props 属性

export const Context = React.createContext();

export const Provider = Context.Provider;

export const Consumer = Context.Consumer;

**如何避免组件的重新渲染？**

React.memo():这可以防止不必要地重新渲染函数组件

React.memo()可接受2个参数，第一个参数为纯函数的组件，第二个参数用于对比props

function Child({seconds}){

  console.log('I am rendering');

  return (

    <div>I am update every {seconds} seconds</div>

  )

};

function areEqual(prevProps, nextProps) {

  if(prevProps.seconds===nextProps.seconds){

    return true

  }else {

    return false

  }

}

export default React.memo(Child,areEqual)

PureComponent:这可以防止不必要地重新渲染类组件

export default class a extends PureComponent {}

**什么是纯函数？**

纯函数是不依赖并且不会在其作用域之外修改变量状态的函数。本质上，纯函数始终在给定相同参数的情况下返回相同结果。

**当调用setState时，React render 是如何工作的？**

虚拟 DOM 渲染:当render方法被调用时，它返回一个新的组件的虚拟 DOM 结构。当调用setState()时，render会被再次调用，因为默认情况下shouldComponentUpdate总是返回true。  
原生 DOM 渲染:React 只会在虚拟DOM中修改真实DOM节点，而且修改的次数非常少