高阶组件是参数为组件，返回值为新组件的函数

**高级组件示例**

方式1：组合

// 此函数接收一个组件...

function withSubscription(WrappedComponent, selectData) {

    // ...并返回另一个组件...

    return class extends React.Component {

        constructor(props) {

            super(props);

            this.state = {

                data: selectData(DataSource, props)

            };

        }

        render() {

            // ... 并使用新数据渲染被包装的组件!

            // 请注意，我们可能还会传递其他属性

            return <WrappedComponent data={this.state.data} {...this.props} />;

        }

    };

}

方式2：继承

let wrap = WrapComponet => class extends WrapComponet {

    constructor(props) {

        super(props)

    }

    click() {

        // ...

}

    render() {

        return (

            <div>

                <div onClick={this.click}>点击</div>

                <div>{super.render()}</div>

            </div>

        )

    }

}

let NewComponent = wrap(OldComponent);

**高阶组件解决什么问题**

有的时候，2个组件之间的处理逻辑大部分都是相同的，只有小部分逻辑或渲染UI不相同，这时候想要复用组件，可以使用高级组件

为什么不使用继承？继承有很多不便之处，没想明白

**使用原则**

高级组件的目的是扩展而不是继承，即传入的WrapComponet组件是一个已经封装好的组件，他并不关心自己是否会被传入高级组件中生成一个新的组件。