

高阶函数

1 普通的高阶函数

```
1 import React from 'react'
2
3 export default (wrappeCompoent, name) => {
4   class NewComponent extends React.Component {
5     constructor () {
6       super()
7       this.state = {
8         data: null
9       }
10    }
11
12    componentWillMount () {
13      let data = localStorage.getItem(name)
14      this.setState({data})
15    }
16
17    render () {
18      return <wrappeCompoent data={ this.state.data } />
19    }
20  }
21
22  return NewComponent
23 }
```

2 复杂的高阶组件

```
1 import React from 'react';
2 import PropTypes from 'prop-types'
3
4 // connect 现在是接受一个参数 mapStateToProps 然后返回一个函数，这个函数接受一个参数
  为组件
5 // 这个返回的函数才是高阶组件
6 // eslint-disable-next-line no-undef
7 export const connect = (mapStateToProps, mapDispatchToProps) =>
  (wrappedComponent) => {
8   class Connect extends React.Component {
9     static contextTypes = {
10       store: PropTypes.object
11     }
12
13     constructor () {
14       super()
15       this.state = {
16         allProps: {}
17       }
18     }
19   }
```

```

20   componentWillMount () {
21     const { store } = this.context
22     this._updateProps()
23     store.subscribe( () => this._updateProps() )
24   }
25
26   _updateProps () {
27     const { store } = this.context
28     // 额外传入this.props
29     let stateProps = mapStateToProps
30       ? mapStateToProps(store.getState(), this.props)
31       : {} // 防止 mapStateToProps 没有传入
32     let dispatchProps = mapDispatchToProps
33       ? mapDispatchToProps(store.dispatch, this.props)
34       : {} // 防止 mapDispatchToProps 没有传入
35     this.setState({
36       // 整合普通的props 和从 state 生成的props
37       allProps: {
38         ...stateProps,
39         ...this.props,
40         ...dispatchProps
41       }
42     })
43   }
44
45   // 接收一个组件作为参数 把此组件当成子组件进行渲染
46   render () {
47     // { ...stateProps 意思就是把这个对象里面的属性全部通过props方式传递进去 }
48     return <wrappedComponent {...this.state.allProps} />
49   }
50 }
51
52 // 返回一个组件
53 return Connect
54 }
55

```

2.1 使用

```

1  import React from 'react'
2  import PropTypes from 'prop-types'
3  import ThemeSwitch from './ThemeSwitch'
4  import { connect } from './Connect'
5
6  class Content extends React.Component {
7    static propTypes = {
8      themeColor: PropTypes.string
9    }
10
11    render () {
12      return (
13        <div>
14          <p style={{ color: this.props.themeColor }}>
15            React.js 小书内容
16          </p>
17          <ThemeSwitch />
18        </div>

```

```
19     )
20   }
21 }
22
23 const mapStateToProps = (state) => {
24   return {
25     themeColor: state.themeColor
26   }
27 }
28
29 const mapDispatchToProps = (dispatch) => {
30   return {
31     onSwitchColor: (color) => {
32       dispatch({ type: 'CHANGE_COLOR', themeColor: color })
33     }
34   }
35 }
36
37 export default Content = connect(mapStateToProps, mapDispatchToProps)
38   (Content)
```