

react hooks 使用

原创

吃瓜群众欢乐多



于 2020-05-07 14:16:43 发布



1990



收藏 6

版权

分类专栏: [React](#)



React

专栏收录该内容

3 订阅

40 篇文章

订阅专

栏

react hooks 使用-官网链接

主要解决，更新state时减少代码

Hook 将组件中相互关联的部分拆分成更小的函数（比如设置订阅或请求数据）

React.FC里可以使用

一、state hook

1.前提

React函数组件-也成为无状态组件，他有明显的缺点，只能展示，不能改变state的值，没有this，不能读取this.state,没有生命周期

- **state hook**那么如何解决函数组件里的更新呢？可以通过引入Hook来解决

2.例子

```
1 import React, { useState } from 'react';
2
3 function Example() {
4   // 声明一个新的叫做 "count" 的 state 变量
5   const [count, setCount] = useState(0);
6   return (
7     <div>
8       <p>You clicked {count} times</p>
9       <button onClick={() => setCount(count + 1)}>
10         Click me
11       </button>
12     </div>
13   );
14 }
```

```
1 const [count, setCount] = useState(0);
2 // count: 当前的state
3 // setCount: 更新 state 的函数
```

3.对比

读取前后

以前: this.state.count

hook: count

更新

旧: this.setState({count:count+1})

新: setCount(count+1)

二、Effect Hook

官网讲解

- **useEffect** 时，就是在告诉 React 在完成对 DOM 的更改后运行你的“副作用”函数
useEffect = componentDidMount、componentDidUpdate 和 componentWillUnmount
useEffect这个函数状态会存在这三个状态中，渲染后，每次更新
常见优势：用于清除某些不需要的副作用，例如订阅外部数据源。这种情况下，清除工作是非常重要的，可以防止引起内存泄露

```
1 import React, { useState, useEffect } from 'react';
2
3 function Example() {
4   const [count, setCount] = useState(0);
5
6   // Similar to componentDidMount and componentDidUpdate:
7   useEffect(() => {
8     // Update the document title using the browser API
9     document.title = `You clicked ${count} times`;
```

```
10   });
11
12   return (
13     <div>
14       <p>You clicked {count} times</p>
15       <button onClick={() => setCount(count + 1)}>
16         Click me
17       </button>
18     </div>
19   );
20 }
```

三、其他

- `useContext`
接收一个 context 对象（`React.createContext` 的返回值）并返回该 context 的当前值。当前的 context 值由上层组件中距离当前组件最近的 `<MyContext.Provider>` 的 `value` prop 决定
- `useReducer`
通过 reducer 来管理组件本地的复杂 state

四：hook使用规则

- 只能在函数最外层调用 Hook。不要在循环、条件判断或者子函数中调用
- 只能在 React 的函数组件中调用 Hook。不要在其他 JavaScript 函数中调用。（还有一个地方可以调用 Hook —— 就是自定义的 Hook 中，我们稍后会学习到