


React.Component、React.PureComponent、React.FC的使用与区别

原创 吃瓜群众欢乐多 于 2020-05-07 17:38:57 发布 8114 收藏 13 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](#)

1. React.Component

最常见的，包含最常用的render(),componentDidMount(),shouldComponentUpdate...

- shouldComponentUpdate(nextProps, nextState)
判断 React 组件的输出是否受当前 state 或 props 更改的影响。意思就是只要组件的 props 或者 state 发生了变化，就会重新构建 virtual DOM，然后使用 diff算法进行比较，紧接着根据比较结果决定是否重新渲染整个组件。shouldComponentUpdate函数是重渲染时render()函数调用前被调用的函数，它接受两个参数：nextProps和nextState，分别表示下一个props和下一个state的值。并且，当函数返回false时候，阻止接下来的render()函数的调用，阻止组件重渲染，而返回true时，组件照常重渲染。
- React.Component 并未实现 shouldComponentUpdate(), 如何利用shouldComponentUpdate钩子函数优化react性能
- demo1.tsx

```
1 import React, { PureComponent, Component } from 'react';
2 class CompareComponent extends Component<any, any>{
3   state = {
4     isShow: false,
5     count:1
6   }
7   shouldComponentUpdate(nextProps, nextState){
8     if(nextState.isShow===this.state.isShow){
9       return false // false则不会调用render
10    }
11    return true
12  }
13  changeState = () => {
14    this.setState({
15      isShow: true
16    })
17  };
18
19  handleAdd={()=>{
20    const {count} = this.state
21    this.setState({
22      count:count+1
23    })
24  }
25
26  render() {
27    console.log('render');
28    return (
29      <div>
30        <button onClick={this.changeState}>点击赋值</button>
31        <div>{this.state.isShow.toString()}</div>
32
33        <button onClick={this.handleAdd}>点击{this.state.count}</button>
34      </div>
35    );
36  }
37 }
38 export default CompareComponent
```

[参考链接](#)

2.React.PureComponent

React.PureComponent 与 React.Component 几乎完全相同，也包括render,生命周期等等。但 React.PureComponent 通过props和state的浅对比来实现 shouldComponentUpdate()。如果对象包含复杂的数据结构，它可能会因深层的数据不一致而产生错误的否定判断(表现为对象深层的数据已改变视图却没有更新)

- React.PureComponent在某些条件下，render不用重新渲染

- PureComponent中不能有shouldComponentUpdate
- demo2.tsx

```
1 import React, { PureComponent, Component } from 'react';
2
3 class CompareComponent extends PureComponent {
4   state = {
5     isShow: false,
6     count: 1
7   }
8   changeState = () => {
9     this.setState({
10       isShow: true
11     })
12   };
13   handleAdd = () => {
14     const { count } = this.state
15     this.setState({
16       count: count + 1
17     })
18   }
19   render() {
20     console.log('render');
21     return (
22       <div>
23         <button onClick={this.changeState}>点击赋值</button>
24         <div>{this.state.isShow.toString()}</div>
25
26         <button onClick={this.handleAdd}>点击{this.state.count}</button>
27       </div>
28     );
29   }
30 }
31 export default CompareComponent
```

3. React.Component、React.PureComponent的区别

通过查看render的渲染次数会发现规律

shouldComponentUpdate返回true的话，React.Component无论state发生新的改变，都会重新渲染render。若是返回false，则state在满足条件下返回false的话，不会重新渲染render

- PureComponent = shouldComponentUpdate(false) + React.Component

4. React.FC

React.FC<>的在typescript使用的一个泛型,这个泛型里面可以使用useState
这个里面无生命周期。

通过useState可以实现在函数组件中实现值的更新

useState在前两者中不能使用，只能在函数组件或者泛型里面使用。具体如何使用，可参考我[这篇文章](#)

```
1 import React, { useState } from 'react';
2
3 export interface NoticeProps {
4   name: string;
5   address: string
6 }
7
8 const Notice: React.FC<NoticeProps> = (props) => {
9   const [ name, setName ] = useState('angle');
10   const [ count, setCount ] = useState(1);
11   const addCount = () => {
12     setCount(count + 1)
13   }
14
15   console.log(count)
16   return <div>
17     <p>我是message页面, name是: {name}</p>
18     <p>我是message页面, count是: {count}</p>
19     <button onClick={() => setName('jane')}>点击我改变name</button>
20     <br />
21     <button onClick={addCount}>点击我改变count</button>
22   </div>;
23 }
24 export default Notice;
25
```