

正确使用setState

正确使用setState

课堂目标

资源

起步

正确使用setState

关于 `setState()` 你应该了解三件事：

不要直接修改 State

State 的更新可能是异步的

State 的更新会被合并

课堂目标

1. 掌握setState用法

资源

1. [State & 生命周期](#)

起步

正确使用setState

`setState(partialState, callback)`

1. `partialState`: `object` | `function` function情况下需要return一个操作后的自己
用于产生与当前state合并的子集。
2. `callback`: `function`
state更新之后被调用。

关于 `setState()` 你应该了解三件事：

不要直接修改 State

例如，此代码不会重新渲染组件：

```
// 错误示范
this.state.comment = 'Hello';
```

而是应该使用 `setState()`:

```
// 正确使用
this.setState({comment: 'Hello'});
```

State 的更新可能是异步的

出于性能考虑, React 可能会把多个 `setState()` 调用合并成一个调用。

观察以下例子中log的值和button显示的counter。

```
import React, { Component } from "react";

export default class SetStatePage extends Component {
  constructor(props) {
    super(props);
    this.state = {
      counter: 0
    };
  }

  changeValue = v => {
    this.setState({
      counter: this.state.counter + v
    });
    console.log("counter", this.state.counter);
  };

  setCounter = () => {
    this.changeValue(1);
    //this.changeValue(2);
    //console.log("counter", this.state.counter);
  };

  render() {
    const { counter } = this.state;
    return (
      <div>
        <h3>SetStatePage</h3>

        <button onClick={this.setCounter}>{counter}</button>
      </div>
    );
  }
}
```

```
}
```

如果要获取到最新状态值有以下方式：（同步）

1. 在回调中获取状态值 即在state更新之后再去执行回调

```
changeValue = v => {  
  this.setState(  
    {  
      counter: this.state.counter + v  
    },  
    () => {  
      console.log("counter", this.state.counter);  
    }  
  );  
};
```

2. 使用定时器:

```
setTimeout(() => {  
  this.setCounter();  
}, 0);
```

3. 原生事件中修改状态

```
componentDidMount(){  
  document.body.addEventListener('click', this.changeValue, false)  
}
```

总结：setstate只有在合成事件和生命周期函数中是异步的，在原生事件和setTimeout中都是同步的，这里的异步其实是批量更新。

State 的更新会被合并

```
changeValue = v => {  
  this.setState({  
    counter: this.state.counter + v  
  });  
};  
setCounter = () => {  
  this.changeValue(1);  
  this.changeValue(2);  
};
```

如果想要链式更新state:

```
changeValue = v => {  
  this.setState(state => ({ counter: state.counter + v }));  
};  
setCounter = () => {  
  this.changeValue(1);  
  this.changeValue(2);  
};
```

