

题目

你知道setState是同步还是异步吗？

回答

- 思路

- setState既可以是异步也可以是同步的，在不同的场景下表现不同

- 异步

- 合成事件（onClick、onChange）

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

class Test extends Component {
  state = { nd: 0 };

  handleClick = () => {
    this.setState({ nd: this.state.nd+ 1 });
    console.log(this.state.nd);
  };
  render() {
    return <div onClick={this.handleClick}>`点击次数:
    ${this.state.nd}`</div>;
  }
}

export default Test;
```

- 生命周期函数

```
componentDidMount() {
  this.setState({ nd: this.state.nd+ 1 });
  console.log(this.state.nd);
}
```

- 同步

- 原生事件

```

handleClick = () => {
  this.setState({ nd: this.state.nd+ 1 });
  console.log(this.state.nd);
};
componentDidMount() {
  document.body.addEventListener('click', this.handleClick);
}

```

■ setTimeout函数

```

componentDidMount() {
  setTimeout(() => {
    this.setState({ nd: this.state.nd+ 1 });
    console.log(this.state.nd);
  }, 1000);
}

```

○ setState的批量更新（异步）

```

class Test extends Component {
  state = { nd: 0 };

  componentDidMount() {
    this.setState({ nd: this.state.nd+ 1 });
    console.log(this.state.nd);

    this.setState({ nd: this.state.nd+ 1 });
    console.log(this.state.nd);

    setTimeout(() => {
      this.setState({ nd: this.state.nd+ 1 });
      console.log(this.state.nd);

      this.setState({ nd: this.state.nd+ 1 });
      console.log(this.state.nd);
    }, 1000);
  }

  render() {
    return <div>{this.state.nd}</div>;
  }
}

```

