

实际工作中对 React 做过哪些优化？

- 使用 CSS 模拟 v-show

```
<MyComponent style={{ display: flag ? 'block' : 'none'}}/>
```

- 循环使用 key

```
const items = list.map(item => <li key={item.id}>{item.text}</li>)
```

- 使用 fragment 减少层级

减少不必要的标签，如果是要组合多个一级标签，可以用空标签包裹<></>。

- JSX 中不要定义函数

可以先定义出来，在 render（class 组件）或返回时（函数组件）中使用。

```
/* Bad */
<button onClick={() => {...}}>点击</button>

// good
class MyComponent extends React.Component {
  clickHandler = () => { /**/ };
  render() {
    return <button onClick="this.clickHandler">点击</button>
  }
}
```

- 要在构造函数中使用 bind this

```
class MyComponent extends React.Component {
  constructor() {
    // 要在构造函数中 bind this，而不是在 JSX 中
    this.clickHandler1 = this.clickHandler1.bind(this)
  }
  clickHandler1() { /* 如果 JSX 中直接调用，则 this 不是当前组件。所以要 bind this */ }
  clickHandler2 = () => { /* 使用箭头函数，不用 bind this */ }
  render() {
    return <>
      <button onClick={this.clickHandler1}>点击</button>
    </>
  }
}
```

- 使用 shouldComponentUpdate 判断组件是否要更新

class 组件使用 shouldComponentUpdate 函数组件使用 React.memo

- hooks 缓存数据和函数

useMemo useCallback

- 路由懒加载 (同 Vue)
- 异步组件 (同 Vue)
- SSR, next.js

连环问：在使用 React 过程中遇到过哪些坑？

- 自定义组件，首字母要大些。
- JS 关键字冲突

```
{/*for 要改成 htmlFor, class 要改成 className */}  
<label htmlFor="input-name" className="xxx">  
  姓名<input id="inputName"/>  
</label>
```

- JSX 的数据类型

```
{/*用引号表示纯字符串，用大括号表示 js 表达式*/}  
<Demo position="1" flag="true" />  
<Demo position={1} flag={true} />
```

- setState 是异步更新的

使用 setState 改变了变量的值后，如果要知道改变后的值，需要在 setState 方法中传递一个函数来获取。直接去打印他的值是不行的。

```
const curNum = this.state.num;  
this.setState({  
  num: curNum + 1  
}, () => {  
  console.log("curNum", this.state.num); {/*正确*/}  
});  
  
console.log("curNum", this.state.num); {/*错误*/}
```