# 数组操作必要性

## 在react中经常会涉及到数组的新增，修改和删除操作。

### 我们通过一个实例来演示一下数组的操作技巧

### 还是reactdemo3项目，在src新建一个文件夹17react-array-op，然后在这个文件夹下面新建一个ArrayOpSkills.jsx组件文件,用rsc创建骨架或者使用webstorm设置模板文件，然后在App.jsx文件里面添加对组件的引用 .

### 1)数组的新增，添加如下代码

|  |
| --- |
| **import *React***, {*useState*} **from 'react'**;  **const** *ArrayOpSkills* = () => {  **let** [students,setStudents] = *useState*([  {  **id**:1,  **name**:**'李连杰'** },  {  **id**:2,  **name**:**'赵丽颖'** },  {  **id**:3,  **name**:**'周迅'** }  ])   **let** add = ()=>{  setStudents([  ...students,  {  **id**:students.**length**+1,  **name**:**'赵晓虎'** }  ])  }  **let** edit = ()=>{   }  **let** del = ()=>{   }  **return** (  <**div**>  <**h3**>学生列表</**h3**>  {students.map(item=>{  **return** (<**li key=**{item.**id**}>{item.**name**}</**li**>)  })}  <**button onClick=**{add}>新增</**button**>  <**button onClick=**{edit}>修改</**button**>  <**button onClick=**{del}>删除</**button**>  </**div**>  ); };  **export default** *ArrayOpSkills*; |

#### 核心技巧：调用setStudents，在里面传递一个新数组，在数组里面保留旧数组的数据使用...操作符，然后在下面添加新数据

### 2）数组的修改，添加代码如下，使用map方法

|  |
| --- |
| **import *React***, {*useState*} **from 'react'**;  **const** *ArrayOpSkills* = () => {  **let** [students,setStudents] = *useState*([  {  **id**:1,  **name**:**'李连杰'** },  {  **id**:2,  **name**:**'赵丽颖'** },  {  **id**:3,  **name**:**'周迅'** }  ])   **let** add = ()=>{  setStudents([  ...students,  {  **id**:students.**length**+1,  **name**:**'赵晓虎'** }  ])  }  **let** update = (id)=>{  setStudents(students.map(item=>{  **if**(item.**id**==id){  **return** {  ...item,  **name**:**'xiaoming'** }  }  **return** item  }))  }  **let** del = (id)=>{  setStudents(students.filter(item=>item.**id**!=id))  }  **return** (  <**div**>  <**h3**>学生列表</**h3**>  {students.map(item=>{  **return** (<**li key=**{item.**id**}>  {item.**name**}  <**button onClick=**{()=>update(item.**id**)}>修改</**button**>  {*/\*<button onClick={()=>del(item.id)}>删除</button>\*/*}  </**li**>)  })}  <**button onClick=**{add}>新增</**button**>  </**div**>  ); };  **export default** *ArrayOpSkills*; |

### 效果：

#### 修改前

|  |
| --- |
|  |

#### 修改后

|  |
| --- |
|  |

### 3）数组的删除。这里使用setStudents传递一个调用students.filter方法返回的新数组，在这个方法里面过滤出id和当前id不相等的元素

|  |
| --- |
| **import *React***, {*useState*} **from 'react'**;  **const** *ArrayOpSkills* = () => {  **let** [students,setStudents] = *useState*([  {  **id**:1,  **name**:**'李连杰'** },  {  **id**:2,  **name**:**'赵丽颖'** },  {  **id**:3,  **name**:**'周迅'** }  ])   **let** add = ()=>{  setStudents([  ...students,  {  **id**:students.**length**+1,  **name**:**'赵晓虎'** }  ])  }  **let** update = (id)=>{  setStudents(students.map(item=>{  **if**(item.**id**==id){  **return** {  ...item,  **name**:**'xiaoming'** }  }  **return** item  }))  }  **let** del = (id)=>{  setStudents(students.filter(item=>item.**id**!=id))  }  **return** (  <**div**>  <**h3**>学生列表</**h3**>  {students.map(item=>{  **return** (<**li key=**{item.**id**}>  {item.**name**}  <**button onClick=**{()=>update(item.**id**)}>修改</**button**>  <**button onClick=**{()=>del(item.**id**)}>删除</**button**>  </**li**>)  })}  <**button onClick=**{add}>新增</**button**>  </**div**>  ); };  **export default** *ArrayOpSkills*; |

### 效果：

#### 删除前

|  |
| --- |
|  |

#### 删除后

|  |
| --- |
|  |