# TodoList练习

## 我们还是使用reacdemo3项目，新建一个文件夹exercise-todolist，然后在这个文件夹下面新建一个TodoList.jsx组件文件，用快捷键rsc（vscode的是：rafce）创建骨架

|  |
| --- |
| **import *React* from 'react'**;  **const** *TodoList* = () => {  **return** (  <**div**>    </**div**>  ); };  **export default** *TodoList*; |

## 然后在App.jsx中添加对这个组件的引用

|  |
| --- |
| **import** { *useState* } **from 'react'** *// import reactLogo from './assets/react.svg' // import viteLogo from '/vite.svg' // import './App.css'* **import** *FuncComponent* **from "./01-compoment-basic/FuncComponent"**; **import** ClassComponent **from "./01-compoment-basic/ClassComponent"**; **import** ClassEvent **from "./02-class-component-events/ClassEvent"**; **import** ClassState **from "./03class-component-state/ClassState"**; **import** ClassComputed **from "./04class-computed-property/ClassComputed" import** ClassLifeCycle **from "./05class-comp-life-cycle/ClassLifeCycle"**; *// import SideMenu from "./side-menu/SideMenu";* **import** SideMenu2 **from "./side-menu-v2/SideMenu2"**; **import** ComponentStyle **from "./06component-style/ComponentStyle"**; **import** *FunctionState* **from "./07readct-hook-useState/FunctionState"**; **import** *FunctionLifeCycle* **from "./08react-Hook-useEffect/FunctionLifeCycle"**; **import** *FunctionComputed* **from "./09react-Hook-useMemo/FunctionComputed"**; **import** *FunctionComputed2* **from "./09react-Hook-useMemo/FunctionComputed2"**; **import** *FunctionRef* **from "./10react-Hook-useRef/FunctionRef"**; **import** *TodoList* **from "./exercise-todoList/TodoList"**;  **function** *App*() {   **return** (  <>  {*/\*<h1>React demo3</h1>\*/*}  {*/\*<FuncComponent/>\*/*}  {*/\*<ClassComponent/>\*/*}  {*/\*<ClassEvent/>\*/*}  {*/\* <ClassState></ClassState>\*/*}  {*/\* <ClassComputed></ClassComputed>\*/*}  {*/\* <ClassLifeCycle></ClassLifeCycle>\*/*}  {*/\* <SideMenu></SideMenu>\*/*}  {*/\* <SideMenu2/>\*/*}  {*/\* <ComponentStyle></ComponentStyle>\*/*}  {*/\* <FunctionState></FunctionState>\*/*}  {*/\* <FunctionLifeCycle></FunctionLifeCycle>\*/*}  {*/\* <FunctionComputed></FunctionComputed>\*/*}  {*/\* <FunctionComputed2></FunctionComputed2>\*/*}  {*/\* <FunctionRef/>\*/*}  <**TodoList**/>  </>  ) }  **export default** *App* |

## 给这个组件添加一个局部也是文件：todo.module.css,内容如下

|  |
| --- |
| .**done**{  **color**:**deeppink**;  **text-decoration**: **line-through**; } |

## 然后在TodoList.jsx组件文件里面引入局部样式，并且根据item的done属性来给项动态添加样式

|  |
| --- |
| **import *React***, {*useState*} **from 'react'**; **import** styles **from './todo.module.css' let** initTodos = [  {**id**: 1, **value**: **'学习html'**, **done**: **false**},  {**id**: 2, **value**: **'学习css'**, **done**: **true**},  {**id**: 3, **value**: **'学习js'**, **done**: **false**}, ]  **const** *TodoList* = () => {  **let** [todos,setTodos] = *useState*(initTodos)  **return** (  <**div**>  <**h3**>TodoList</**h3**>  <**div**>  <**input type="text"**/>  <**button**>新增</**button**>  </**div**>  <**ul**>  {  todos.map(item=>(<**li key=**{item.**id**} **className=**{item.**done**?styles.done:**null**}>{item.**value**}</**li**>))  }  </**ul**>  <**div**>  <**span**>全部</**span**>  </**div**>  </**div**>  ); };  **export default** *TodoList*; |

### 效果

|  |
| --- |
|  |

## 现在我们需要一个功能，就在在文本框里面输入数据，点击按钮会把数据添加到下面的列表，代码如下TodoList.jsx

|  |
| --- |
| **import *React***, {*useRef*, *useState*} **from 'react'**; **import** styles **from './todo.module.css' let** initTodos = [  {**id**: 1, **value**: **'学习html'**, **done**: **false**},  {**id**: 2, **value**: **'学习css'**, **done**: **true**},  {**id**: 3, **value**: **'学习js'**, **done**: **false**}, ]  **const** *TodoList* = () => {  **let** [todos,setTodos] = *useState*(initTodos)  **let** inputRef = *useRef*(**null**)  **let** addTodo = ()=>{  **let** addValue = inputRef.**current**.**value**;  setTodos([  ...todos,  {  **id**:todos.**length**+1,  **value**:addValue,  **done**:**false**}  ]);  inputRef.**current**.**value**=**''**  }  **return** (  <**div**>  <**h3**>TodoList</**h3**>  <**div**>  <**input type="text" ref=**{inputRef}/>  <**button onClick=**{addTodo}>新增</**button**>  </**div**>  <**ul**>  {  todos.map(item=>(<**li key=**{item.**id**} **className=**{item.**done**?styles.done:**''**}>{item.**value**}</**li**>))  }  </**ul**>  <**div**>  <**span**>全部</**span**>  </**div**>  </**div**>  ); };  **export default** *TodoList*; |

## 添加功能:给每一个li添加点击事件，点击li会改变它的样式

|  |
| --- |
| **import *React***, {*useRef*, *useState*} **from 'react'**; **import** styles **from './todo.module.css' let** initTodos = [  {**id**: 1, **value**: **'学习html'**, **done**: **false**},  {**id**: 2, **value**: **'学习css'**, **done**: **true**},  {**id**: 3, **value**: **'学习js'**, **done**: **false**}, ]  **const** *TodoList* = () => {  **let** [todos,setTodos] = *useState*(initTodos)  **let** inputRef = *useRef*(**null**)  **let** addTodo = ()=>{  **let** addValue = inputRef.**current**.**value**;  *// todos.push({id:todos.length+1,value:addValue,done:false})  // console.log(todos)* setTodos([  ...todos,  {  **id**:todos.**length**+1,  **value**:addValue,  **done**:**false**}  ]);  inputRef.**current**.**value**=**''** }  *//点击事件处理函数* **const** doneChange = (id)=>{  setTodos(todos.map(item=>{  **if**(item.**id** === id){  **return** {  ...item,  **done**:!item.**done** }  }  **return** item  }))  *// console.log(newTodos)  // setTodos(newTodos)* }  **return** (  <**div**>  <**h3**>TodoList</**h3**>  <**div**>  <**input type="text" ref=**{inputRef}/>  <**button onClick=**{addTodo}>新增</**button**>  </**div**>  <**ul**>  {  todos.map(item=>  <**li onClick=**{()=>doneChange(item.**id**)}  **className=**{item.**done**?styles.done:**''**} **key=**{item.**id**} >  {item.**value**}  </**li**>)  }  </**ul**>  <**div**>  <**span**>全部</**span**>  </**div**>  </**div**>  ); };  **export default** *TodoList*; |

## 效果：

|  |
| --- |
|  |

### 需要注意：

### 1.如果处理方法需要传参，一定要把它放在箭头函数里面调用，否则会报错

### 2.react的原则是不要去该原数组或者数据，需要返回新数组或者数据，要用setXXX来设置

## 我们继续添加功能，用来过滤完成事项的情况，分为：未完成，已完成和全部，点击超链接来事项过滤，此时需要使用就是属性来实现

|  |
| --- |
| **import *React***, {*useMemo*, *useRef*, *useState*} **from 'react'**; **import** styles **from './todo.module.css' let** initTodos = [  {**id**: 1, **value**: **'学习html'**, **done**: **false**},  {**id**: 2, **value**: **'学习css'**, **done**: **true**},  {**id**: 3, **value**: **'学习js'**, **done**: **false**}, ] **let** initResult = [  {**id**: 1, **value**: **'全部'**},  {**id**: 2, **value**: **'已完成'**},  {**id**: 3, **value**: **'未完成'**}, ]  **const** *TodoList* = () => {  **let** [todos,setTodos] = *useState*(initTodos)  **let** [resultList,setResultList] = *useState*(initResult)  **let** [selectedId,setSelectedId] = *useState*(1)  **let** inputRef = *useRef*(**null**)  *//计算属性* **var** filteredTodos = *useMemo*(()=>{  **switch** (selectedId) {  **case** 2:  **return** todos.filter(item=> item.**done**)  **break  case** 3:  **return** todos.filter(item=> !item.**done**)  **break  default**:  **return** todos  **break** }  },[selectedId,todos]);   **let** addTodo = ()=>{  **let** addValue = inputRef.**current**.**value**;  *// todos.push({id:todos.length+1,value:addValue,done:false})  // console.log(todos)* setTodos([  ...todos,  {  **id**:todos.**length**+1,  **value**:addValue,  **done**:**false**}  ]);  inputRef.**current**.**value**=**''** }  *//点击事件处理函数* **const** doneChange = (id)=>{  setTodos(todos.map(item=>{  **if**(item.**id** === id){  **return** {  ...item,  **done**:!item.**done** }  }  **return** item  }))  }   **return** (  <**div**>  <**h3**>TodoList</**h3**>  <**div**>  <**input type="text" ref=**{inputRef}/>  <**button onClick=**{addTodo}>新增</**button**>  </**div**>  <**ul**>  {  filteredTodos.map(item=>  <**li onClick=**{()=>doneChange(item.**id**)}  **className=**{item.**done**?styles.done:**''**} **key=**{item.**id**} >  {item.**value**}  </**li**>)  }  </**ul**>  <**div**>  {resultList.map(item=>{  **if**(item.**id** === selectedId){  **return** <**span key=**{item.**id**}>{item.**value**}</**span**>  }  **return** <**a href="#" key=**{item.**id**} **onClick=**{(e)=>{  e.preventDefault()  setSelectedId(item.**id**)}  }>{item.**value**}</**a**>  })}   </**div**>  </**div**>  ); };  **export default** *TodoList*; |

## 另外我们还需要一个统计已完成的事项的比例功能，还需要一个计算属性

|  |
| --- |
| **import *React***, {*useMemo*, *useRef*, *useState*} **from 'react'**; **import** styles **from './todo.module.css' let** initTodos = [  {**id**: 1, **value**: **'学习html'**, **done**: **false**},  {**id**: 2, **value**: **'学习css'**, **done**: **true**},  {**id**: 3, **value**: **'学习js'**, **done**: **false**}, ] **let** initResult = [  {**id**: 1, **value**: **'全部'**},  {**id**: 2, **value**: **'已完成'**},  {**id**: 3, **value**: **'未完成'**}, ]  **const** *TodoList* = () => {  **let** [todos,setTodos] = *useState*(initTodos)  **let** [resultList,setResultList] = *useState*(initResult)  **let** [selectedId,setSelectedId] = *useState*(1)  **let** inputRef = *useRef*(**null**)  *//计算属性* **var** filteredTodos = *useMemo*(()=>{  **switch** (selectedId) {  **case** 2:  **return** todos.filter(item=> item.**done**)  **break  case** 3:  **return** todos.filter(item=> !item.**done**)  **break  default**:  **return** todos  **break** }  },[selectedId,todos]);   **let** finshedTotal = *useMemo*(()=>{  **return** todos.filter(item=>item.**done**).**length** },[todos])    **let** addTodo = ()=>{  **let** addValue = inputRef.**current**.**value**;  *// todos.push({id:todos.length+1,value:addValue,done:false})  // console.log(todos)* setTodos([  ...todos,  {  **id**:todos.**length**+1,  **value**:addValue,  **done**:**false**}  ]);  inputRef.**current**.**value**=**''** }  *//点击事件处理函数* **const** doneChange = (id)=>{  setTodos(todos.map(item=>{  **if**(item.**id** === id){  **return** {  ...item,  **done**:!item.**done** }  }  **return** item  }))  *// console.log(newTodos)  // setTodos(newTodos)* }   **return** (  <**div**>  <**h3**>TodoList</**h3**>  <**div**>  <**input type="text" ref=**{inputRef}/>  <**button onClick=**{addTodo}>新增</**button**>  </**div**>  <**ul**>  {  filteredTodos.map(item=>  <**li onClick=**{()=>doneChange(item.**id**)}  **className=**{item.**done**?styles.done:**''**} **key=**{item.**id**} >  {item.**value**}  </**li**>)  }  </**ul**>  <**div**>  {resultList.map(item=>{  **if**(item.**id** === selectedId){  **return** <**span key=**{item.**id**}>{item.**value**}</**span**>  }  **return** <**a href="#" key=**{item.**id**} **onClick=**{(e)=>{  e.preventDefault()  setSelectedId(item.**id**)}  }>{item.**value**}</**a**>  })}  <**span**> 进度：{finshedTotal}/{todos.**length**}</**span**>  </**div**>  </**div**>  ); };  **export default** *TodoList*; |

### 效果

|  |
| --- |
|  |

## 总结：这个练习用到了数组操作中的map方法和filter方法，

## 计算属性的使用，获取节点的值，局部样式的使用，事件传参等等

## 注意：react建议不要直接修改数据，需要使用Hook中useState函数返回的setXXX方法来传递新数据实现数据修改