回答

1. 对下面数组按年龄和姓名排序，要求不使用 Array sort 方法

const data = [{age:2,name:'mapple'},{age:7,name:'qapple'},{age:3,name:'fapple'},{age:1,name:'tapple1'},{age:7,name:'aapple1'},];

const ageCompare = (a, b) => {

return a.age > b.age;

}

const nameCompare = (a, b) => {

return a.name > b.name;

}

function sortArray (arr, func) {

if (arr.length == 0) return arr;

const mid = arr[0];

let left = [], right = [];

for (let i = 1; i < arr.length; i++) {

if (func(mid, arr[i])) {

left.push(arr[i])

} else {

right.push(arr[i]);

}

}

return sortArray(left, func).concat([mid]).concat(sortArray(right, func));

}

console.log(sortArray(data, ageCompare));

console.log(sortArray(data, nameCompare));

1. 将扁平数组转换为树状结构

const arr = [

{id: 1, name: '部门1', pid: 0}, // 根部门

{id: 2, name: '部门2', pid: 1}, // 根部门 的子部门

{id: 3, name: '部门3', pid: 1}, // 根部门 的子部门

{id: 4, name: '部门4', pid: 3}, // 部门3 的子部门

{id: 5, name: '部门5', pid: 4}, // 部门4 的子部门

]

function transform(arr) {

let tempArr = arr.sort((a, b) => {

if (a.pid > b.pid) {

return 1;

} else if (a.pid < b.pid) {

return -1

} else {

return 0;

}

});

const indexMap = new Map();

let result = [];

tempArr.forEach(element => {

element.children = [];

indexMap.set(element.id, element);

const pid = element.pid;

if (indexMap.has(pid)) {

indexMap.get(pid).children.push(element);

} else {

result.push(element);

}

});

return result;

}

console.log(transform(arr));

1. 按顺序写出输出结果，并说明原理

输出结果如下：

1

2

4

timeStart

timeEnd

success

原理为：先执行 Promise 构造函数内的代码，先输出1，然后忽略定时操作，执行下面的代码，输出2。然后继续执行代码，输出 4。然后执行定时内的代码，输出timeStart 和 timeEnd，Promise 通过 resolve 返回数据。外面通过 then 获取到数据，并输出 success。

1. 使用 React 高阶组件实现条件渲染

function loadWithCondition(Component) {

return function EnhancedComponent({ isLoading, ...props }) {

if (!isLoading) {

return <Component {...props} />;

}

return (

<div>

<p>Loading</p>

</div>

);

};

}

const ListLoadWithCondition = loadWithCondition(List);

function App({ listData, isLoading }) {

return (

<ListLoadWithCondition isLoading={isLoading} listData={listData} />

);

}