Demo  
替换指定元素  
  
  
js splice使用总结（对js数组的增删改），替换数组的指定位置的数据，以及js数组的位置移动，向上（前）移动，或者向下（后）移动，el-table上移下移  
  
新林。  
  
于 2020-01-01 16:43:03 发布  
  
3725  
收藏 12  
分类专栏： javascript 文章标签： js数组的splice总结 js实现数组的位置移动 js数组向上移，向下移动 el-table数据上移下移  
版权  
  
javascript  
专栏收录该内容  
2 篇文章0 订阅  
订阅专栏  
splice()：插入、删除和替换  
3 个参数：起始位置、要删除的项数和要插入的任意数量的项  
再简单说splice有三个位置的参数，第一个index是数组的位置，第二个是这个位置要删除的项数、，第三个位置以及后面的第四五等位置都表示插入的数组项  
  
看代码理解吧  
  
数组数据：  
  
let userArray=[  
 {"id":1,"name":"小明"},  
 {"id":2,"name":"小红"},  
 {"id":3,"name":"小花"}  
]  
splice插入  
  
//深拷贝上面的对象数组，为了不影响后面的数组  
let array1 = JSON.parse(JSON.stringify(userArray));  
  
//例：splice插入  
let user = {"id":4,"name":"小黑"};  
let user2 = {"id":5,"name":"小华"};  
//表示在2数组的最后的位置，删除0个数据项，然后插入user的新数组项  
array1.splice(3,0,user,user2);  
console.log(array1);//可以看到数组最后面多了一个小黑,小华  
  
8  
  
  
  
splice删除  
//表示在数组的位置2，往后删除一项  
let array2 = JSON.parse(JSON.stringify(userArray));  
array2.splice(2,1);  
console.log(array2);//可以看到数组的小花被删除掉了  
  
  
splice替换  
//例：替换数组中index=1的位置的小红替换为小东  
let array3 = JSON.parse(JSON.stringify(userArray));  
let xiaodong = {"id":6,"name":"小东"};  
//表示在数组index位置1处往后删除一个数据项再新增一个数据项（即先把小红删了然后再新增一个小东）  
array3.splice(1,1,xiaodong);  
console.log(array3);//可以看到数组的小红被替换成小东了  
6  
  
  
数组里面的位置移动  
上移（向前）移动  
//向上移动，把小花往上移动  
let array4= JSON.parse(JSON.stringify(userArray));  
this.up(2,array4);  
console.log(array4);  
  
//向上移动  
function up(index,arr){  
 if(index == 0){//第一个节点不移动  
 }else{  
 let curObj = arr[index];//当前位置对象  
 let preObj = arr[index-1];//上一个位置对象  
 arr.splice(index,1,preObj);//上一个节点移动到当前节点  
 arr.splice(index-1,1,curObj);//当前节点往上移动  
 }  
  
}  
  
  
下移（向后）移动  
//向下移动，把小明往下移动  
let array5= JSON.parse(JSON.stringify(userArray));  
this.down(0,array5);  
console.log(array5);  
  
function down(index,arr){  
 if(index == arr.length-1){//最后节点不移动  
 }else{  
 let curObj = arr[index];//当前位置对象  
 let downObj = arr[index+1];//下一个位置对象  
 arr.splice(index,1,downObj);//下一个节点移动到当前节点  
 arr.splice(index+1,1,curObj);//当前节点往下移动  
 }  
  
}  
15  
  
  
可以自己测试一下，把下面这段代码粘贴到文本框修改后缀为html即可调试  
  
<script>  
  
console.log("splice使用demo");  
let userArray=[  
 {"id":1,"name":"小明"},  
 {"id":2,"name":"小红"},  
 {"id":3,"name":"小花"}  
]  
//深拷贝上面的对象数组，为了不影响下面的数组  
let array1 = JSON.parse(JSON.stringify(userArray));  
  
//例：splice插入，替换数组中index=1的位置的小红为小黑  
let user = {"id":4,"name":"小黑"};  
let user2 = {"id":5,"name":"小华"};  
//表示在数组的最后的位置，删除0个数据项，然后插入user的新数组项  
array1.splice(3,0,user,user2);  
console.log(array1);//可以看到数组最后面多了一个小黑，小华  
  
//splice删除  
let array2 = JSON.parse(JSON.stringify(userArray));  
array2.splice(2,1);  
console.log(array2);//可以看到数组的小花被删除掉了  
  
//splice替换  
let array3 = JSON.parse(JSON.stringify(userArray));  
let xiaodong = {"id":6,"name":"小东"};  
array3.splice(1,1,xiaodong);  
console.log(array3);//可以看到数组的小红被替换成小东了  
  
  
//向上移动，把小花往上移动  
let array4= JSON.parse(JSON.stringify(userArray));  
this.up(2,array4);  
console.log(array4);  
  
//向下移动，把小明往下移动  
let array5= JSON.parse(JSON.stringify(userArray));  
this.down(0,array5);  
console.log(array5);  
  
  
function up(index,arr){  
 if(index == 0){//第一个节点不移动  
 }else{  
 let curObj = arr[index];//当前位置对象  
 let preObj = arr[index-1];//上一个位置对象  
 arr.splice(index,1,preObj);//上一个节点移动到当前节点  
 arr.splice(index-1,1,curObj);//当前节点往上移动  
 }  
  
}  
  
function down(index,arr){  
 if(index == arr.length-1){//最后节点不移动  
 }else{  
 let curObj = arr[index];//当前位置对象  
 let downObj = arr[index+1];//下一个位置对象  
 arr.splice(index,1,downObj);//下一个节点移动到当前节点  
 arr.splice(index+1,1,curObj);//当前节点往下移动  
 }  
  
}  
</script>  
  
拓展记录  
例如用了elment-ui的el-table,想要在el-table实现上下移动，首先增加上移、下移按钮  
  
  
<div style="padding-bottom: 10px">  
<el-button class="btn" plain type="primary" size="mini" @click="upColumn">上移</el-button>  
<el-button class="btn" plain type="primary" size="mini" @click="downColumn">下移</el-button>  
</div>  
<el-table :data="columnList" max-height="350" border style="width: 100%" ref="mainTable"  
highlight-current-row  
@row-click="(row, column, event) => rowClick(row, column, event, 'mainTable')"  
>  
<el-table-column type="index" width="50" label="序号" ></el-table-column>  
<el-table-column width="180" prop="dbFieldName" label="字段名称" >  
</el-table-column>  
...  
...  
</el-table>  
实现的方法  
  
//上移  
upColumn(){  
 let selected = this.$refs.mainTable.selection;  
 if (selected.length < 1){  
 this.$message({  
 message: '请选择一条记录',  
 type: 'warning',  
 duration: 1500  
 })  
 return  
 }  
 let arr = this.columnList;  
 let index = arr.indexOf(selected[0]);  
 if (index === 0){ // 第一个节点不移动  
 } else {  
 let curObj = arr[index];// 当前位置对象  
 let preObj = arr[index - 1];// 上一个位置对象  
 arr.splice(index, 1, preObj);// 上一个节点移动到当前节点  
 arr.splice(index - 1, 1, curObj);// 当前节点往上移动  
 }  
},  
//下移  
downColumn(){  
 let selected = this.$refs.mainTable.selection;  
 if (selected.length < 1){  
 this.$message({  
 message: '请选择一条记录',  
 type: 'warning',  
 duration: 1500  
 })  
 return  
 }  
 let arr = this.columnList;  
 let index = arr.indexOf(selected[0]);  
 if (index === arr.length - 1){ // 最后节点不移动  
 } else {  
 let curObj = arr[index];// 当前位置对象  
 let downObj = arr[index + 1];// 下一个位置对象  
 arr.splice(index, 1, downObj);// 下一个节点移动到当前节点  
 arr.splice(index + 1, 1, curObj);// 当前节点往下移动  
 }  
}  
重点逻辑  
  
// table单选函数  
rowClick(row, column, event, tableName) {  
 //清空之前选择的行  
 this.$refs[tableName].clearSelection();  
 //选择当前行（只有用了toggleRowSelection，）  
 this.$refs[tableName].toggleRowSelection(row);  
},  
备注：  
只有用了this.r e f s [ t a b l e N a m e ] . t o g g l e R o w S e l e c t i o n ( r o w ) 选 择 了 当 前 行 ， t h i s . refs[tableName].toggleRowSelection(row)选择了当前行，this.refs[tableName].toggleRowSelection(row)选择了当前行，this.refs.mainTable.selection才会获得当前行的值  
然后用 let index = arr.indexOf(selected[0])就能获取当前行的序号，剩下的上移下移就能复用上面一开始的代码段了  
————————————————  
版权声明：本文为CSDN博主「新林。」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。  
原文链接：https://blog.csdn.net/qq\_21187515/article/details/103792701

Every some

if(this.treeType === 3) { // 项目级别每个都为真，那么目录不用调授权接口查询授权信息  
 this.hasAllPer = **Data.length &&** Data.every(item => {  
 return item.CanEdit && item.CanDownload && item.CanView && item.CanAuth  
 })  
}

for (const key in this.permission) { // res为[]，则permission属性值都为false

this.permission[key] = res.some(item => item[key])

}