1. Data=[{children:[{id:1}]},{children:[{id:5}]},{children:[{id:4}]},{children:[{id:11}]},{children:[{id:7}]},{children:[]},]根据id手动排序(顺序)以children的第一个元素的id顺序排,如果children没有元素则放在第一个(冒泡排序)

// 编写方法，实现冒泡

var arr = [21,45, 68, 72 , 51 , 97];

//外层循环，控制趟数，每一次找到一个最大值

for (var i = 0; i < arr.length - 1; i++) {

// 内层循环,控制比较的次数，并且判断两个数的大小

for (var j = 0; j < arr.length - 1 - i; j++) {

// 白话解释：如果前面的数大，放到后面(当然是从小到大的冒泡排序)

if (arr[j] > arr[j + 1]) {

var temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

console.log(arr);

1. 全部用原生写,不能用js封装的函数

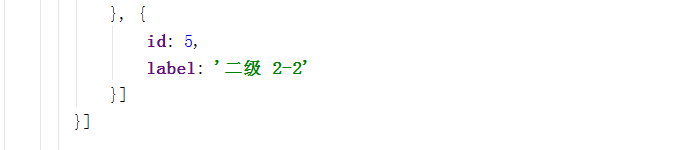
(1)手动对一个数组进行去重

(2)对两个数组进行取交集

(3)对两个数组进行取差集





  
,parent\_id的值为父级的id的值,parent\_id为0时为一级,将上面的数据根据parent\_id分层

1. 通过js实现树形控件的展开收起,追加,通过控制数组的值,来控制显示和影藏(需要演示,$set)

Old=[1,2,3,1-1,1-2,2-1,2-2,3-1,3-2]

Show=[1,2,3]

展开2的二级

Show=[1,2,3,2-1,2-2]

展开3的二级

Show=[1,2,3,2-1,2-2,3-1,3-2]

收起3的二级

Show=[1,2,3,2-1,2-2]

收起2的二级

Show=[1,2,3]