1. 数组的交集，并集，差集。

两数组a = [1, 2, 3]，b = [2, 4, 5]，求a，b数组的并集，交集和差集。

1. ES7

//并集

Let union=a.cancat(b.filter(v=> !a.includes(v)))

含义： B过滤含有A数组元素的元素。再与A合并。

Filter用法： Array.filter(function(currentValue,index,arr),thisvalue)

//交集

Let intersection =b.filter(v=>a.inclues(v))

//差集

Let difference =a.concat(b).filter(v=> a.includes(v)&&!b.includes(v))

?????? 不太懂这个函数

1. ES6

Array.from 方法 将类数组对象和可遍历对象转换为数组。

Set方法返回的是集合，不是数组，所以需要Array.from

Let aSet=new Set(a) let bSet=new Set(b)

//并集

Let union=Array.from(new Set(a.concat(b)))

//交集

Let intersection=Array.from(new Set(a.filter(v=>bSet.has(v)))

因为ES6没有includes,所以只能用set.has.

//差集

let differenceNew = Array.from(new Set(a.concat(b).filter(v => aSet.has(v) && !bSet.has(v)))

???????? 不理解

1. 数组去重

6种方法

1. 双层循环，外层循环元素，内层比较值

层循环，外层Array.prototype.distinct=function(){

Var arr=this,

Result=[],

I,j;

1. 双层循环，外层循环元素，内层比较值

利用splice直接在原数组进行操作。

Array.prototype.distinct=function(){  
var arr=this;

I,j;

Len=arr.length;

For(i=0;i<len;i++){

For(j=i+1;j<len;j++){

If(arr[i]=arr[j]){  
arr.splice(j,1);

j--;

}}

Return arr;

};

var a = [1,2,3,4,5,6,5,3,2,4,56,4,1,2,1,1,1,1,1,1,];

var b = a.distinct();

1. 数组递归去重

运用递归方法，先排序，然后开始比较，遇到相同，则删除。

什么是递归？函数内调用自身。

Array.prototype.distinct=function(){

Var arr=this;

Len=arr.lenth;

Arr.sort(function(a,b){return a-b)})

Function loop(index){

If(index>=1){{

If(arr[index]===arr[index-1]){  
arr.splice(index,1)}

Loop(index-1); // 递归loop函数进行去重

}}

Loop(len-1)

Return arr;

var a = [1,2,3,4,5,6,5,3,2,4,56,4,1,2,1,1,1,1,1,1,56,45,56];

var b = a.distinct();

1. 利用indexof去重

Function newArr(array){

Var arrs=[];

For(var i=0;i<arr.lenth;i++){

If(arrs.indexof(i)==-1){  
arrs.push(array[i]);

}}}

Return arrs;

ar arr = [1,1,2,5,5,6,8,9,8];

console.log(newArr(arr)

1. ES6中利用Set去重

Function newArr(arr){

Return Array.from(new Set(arr))

}

var arr = [1,1,2,9,6,9,6,3,1,4,5];

console.log(newArr(arr))

三.JS Map方法、Foreach

(1)现在用map（）函数实现遍历并重组，

Var memo=[{type:’text’,content:’axd’},{type:’number’,content:’123’}]

Var text=[]; number=[];

Memo.map(function(v,k)){

If(v.type==’text’){

Text.push(v.content)

}

If(v.type==’number’){

Number.push(v.content)

}

}

(2)forEach适合于你并不打算改变数据的时候，而只是想用数据做一些事情 – 比如存入数据库或则打印出来

Arr.forEach(function(当前元素（必）)，当前元素的索引值，当前元素所属的数组对象，thisValue(如果这个参数为空，“undefined ”会传递给“this”))

例子： var arr=[1,2,3]

Arr.forEach(function(value,index,arr1)){

Console.log(value,index,arr1)

})

(3) for ..in for..of

四 继承