

# Daily\_Learning(2019-11-27)

---

```
var arr = [ [1, 2, 2], [3, 4, 5, 5], [6, 7, 8, 9, [11, 12, [12, 13, [14] ] ] ], 10];
```

1.编写一个程序将数组扁平化去并除其中重复部分数据，最终得到一个升序且不重复的数组(携程)

解法一：

```
1 Array.from(new Set(arr.flat(Infinity))).sort((a,b)=>{ return a-b})
```

解法二（推荐）：

```
1 arr.toString().split(",").sort((a,b)=>{ return a-b})
```

2.实现 (5).add(3).minus(2) 功能（百度）。

例：5 + 3 - 2，结果为 6

```
1 Number.prototype.add = function(n) {  
2   return this.valueOf() + n;  
3 };  
4 Number.prototype.minus = function(n) {  
5   return this.valueOf() - n;  
6 };
```

第二题拓展（可不看）：

```
1 Number.MAX_SAFE_DIGITS = Number.MAX_SAFE_INTEGER.toString().length-2  
2 Number.prototype.digits = function(){  
3   let result = (this.valueOf().toString().split('.')[1] || '').length  
4   return result > Number.MAX_SAFE_DIGITS ? Number.MAX_SAFE_DIGITS :  
   result  
5 }  
6 Number.prototype.add = function(i=0){  
7   if (typeof i !== 'number') {  
8     throw new Error('请输入正确的数字');  
9   }  
10   const v = this.valueOf();  
11   const thisDigits = this.digits();
```

```

12     const iDigits = i.digits();
13     const baseNum = Math.pow(10, Math.max(thisDigits, iDigits));
14     const result = (v * baseNum + i * baseNum) / baseNum;
15     if(result>0){ return result > Number.MAX_SAFE_INTEGER ?
Number.MAX_SAFE_INTEGER : result }
16     else{ return result < Number.MIN_SAFE_INTEGER ?
Number.MIN_SAFE_INTEGER : result }
17 }
18 Number.prototype.minus = function(i=0){
19     if (typeof i !== 'number') {
20         throw new Error('请输入正确的数字');
21     }
22     const v = this.valueOf();
23     const thisDigits = this.digits();
24     const iDigits = i.digits();
25     const baseNum = Math.pow(10, Math.max(thisDigits, iDigits));
26     const result = (v * baseNum - i * baseNum) / baseNum;
27     if(result>0){ return result > Number.MAX_SAFE_INTEGER ?
Number.MAX_SAFE_INTEGER : result }
28     else{ return result < Number.MIN_SAFE_INTEGER ?
Number.MIN_SAFE_INTEGER : result }
29 }
30 JS的经典的浮点数陷阱，为了避免出现小数不准确的情况。做了拓展。一个阿里大佬的思路。

```

```

> 0.1 + 0.2
< 0.30000000000000004

```