**数据结构Set**

集合的基本概念：集合是由一组无序且唯一（即不能重复）的项组成的。

Set的特点：key和value相同，没有重复的项

{[1, 1], [2, 2], ['a', 'a'], ['b', 'b']}

ES6提供了数据结构set，它类似于数组，但是成员的值都是唯一的，没有重复的值。

set创建：

不传参：

const s = new Set();

console.log(s); // Set []

传参：传进一个数组或类数组

const s = new Set([1, 2, 3]);

console.log(s); // Set [ 1, 2, 3 ]

set属性：

size: 即有几项

const s = new Set([1, 2, 3]);

console.log(s.size); // 3

set类的方法

1、set.add(value); 添加一个数据(可链式添加)，并返回set结构本身

const s = new Set([1, 2, 3]);

console.log(s.add('a').add('b')); // Set [ 1, 2, 3, "a", "b" ]

2、set.delete(value); 删除指定数据，返回一个布尔值，表示删除是否成功

const s = new Set([1, 2, 3]);

console.log(s.delete(3)); // true

console.log(s); // Set [ 1, 2 ]

console.log(s.delete(3)); // false

3、set.has(value); 判断该值是否为set的成员，返回一个布尔值

const s = new Set([1, 2, 3]);

console.log(s.has(3)); // true

console.log(s.has(4)); // false

4、set.clear(); 清除所有数据，没有返回值

const s = new Set([1, 2, 3]);

console.log(s.clear()); // undefined

console.log(s);

5、keys() 返回键名的遍历器

const s = new Set(['a', 'b', 'c']);

console.log(s.keys()); //  {"a", "b", "c"}

const s = new Set(['1', '2', '3']);

console.log(s.keys()); //  {"1", "2", "3"}

6、values() 返回键值的遍历器(在火狐下返回不出来)

const s = new Set(['1', '2', '3']);

console.log(s.values()); //  {"1", "2", "3"}

7、entries() 返回键值对的遍历器

const s = new Set(['1', '2', '3']);

console.log(s.entries()); // {"1", "2", "3"}

8、forEach(); 使用回调函数遍历每个成员，不会修改set数据本身

回调函数接收三个参数：键名，键值，set本身

const s = new Set(['1', '2', '3']);

s.forEach(function(value, key, set) {

console.log(key);

console.log(value);

console.log(set);

});

案例：数组去重，先放入set中，再把它转成数组

let arr = [1, 2, 3, 4, 2, 3];

let s = new Set(arr);

console.log(s); // Set [ 1, 2, 3, 4 ]

console.log(Array.from(s)); // Array [ 1, 2, 3, 4 ]

function arrayRepetition(array) {

return Array.from(new Set(array));

}

let arr = [1, 2, 3, 1, 2, 3];

console.log(arrayRepetition(arr)); // Array [ 1, 2, 3 ]

let arr = [{}, 1, 'a', 1, 'a', 'b', []];

console.log([...new Set(arr)]); // Array [ {}, 1, "a", "b", [] ]

思考？下面这个去重以后会是什么样？

let arr = [1, 2, 3, 4, 2, 3, NaN, NaN, {}, {}, [], []];