<https://blog.csdn.net/qq_42946917/article/details/89024468>

**1.判断一段字符串是否为回文字符串**？

**回文字符串：即字符串从前往后读和从后往前读字符顺序是一致的。**

如：字符串abccba，从前往后读是a-b-c-c-b-a；从后往前读也是a-b-c-c-b-a

方法一: 1.字符串转数组;2.用reverse()函数颠倒;3.使用join(‘’)拼接成字符串

function checkPalindrom(str) {

return str == str.split('').reverse().join('');

}  
方法二: 从后面往前面遍历字符串，组装成新字符串与原字符串进行比较

function checkPalindrom (str){

var len = str.length; var str1 = "";

for(var i=len-1; i>=0;i--){

str1+=str[i];

}

return str1 == str

}

**2.数组去重**

方法一: 直接双重遍历：一层遍历至倒数第二项，二层遍历从一层的下一项起，移除后--

var arr = [1,2,3,3,4,3,5,3,7,5];

function unique(arr) {

for(var i = 0; i < arr.length-1; i++){//一层遍历至倒数第二项

for(var j = i+1; j < arr.length; j++){//二次从一层的下一项开始

if(arr[i]==arr[j]){

arr.splice(j,1);//console.log(arr[j]);

j--; //减去移除项的影响

}

}

}

return arr;

}  
方法二: 用indexOf()方法：新建一个数组，遍历原数组，不存在于新数组中就push

function unique(arr) {

if (!Array.isArray(arr)) {

console.log('type error!')

return

}

var array = [];

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

if (array .indexOf(arr[i]) === -1) {

array .push(arr[i])

}

}

return array;

}方法三: 用fifter方法

function unique(arr) {

return arr.filter(function(item, index, self) {

//当前元素，在原始数组中的第一个索引==当前索引值，否则返回当前元素

return self.indexOf(item, 0) === index;

});

}方法四:使用ES6的Set