1:实现一个功能字符串倒序?

```
chenge hen shuai

//答案

第一种方法
================
"chenge hen shuai".split("").reverse().join("");


第二种方法
================
var a = "chenge hen shuai";
var b = '';
for(var i=a.length-1;i >= 0;i--){
    b += a.charAt(i);
}
console.log(b);
```

2:判断一个字符串中出现次数最多的字符，统计这个次数?

```
//第一种方法
function methodFirst() {
    var str = "晨哥帅不帅";
    var obj = {}; //把每个字符单个存在对象中
    var maxChars = ""; //存储出现次数最多的字符
    for (var i = 0; i < str.length; i++) {
        var chars = str[i];
        //判断字符串如果不存在就以对象形势存起来  1代表第一次
        if (!obj[chars]) {
            obj[chars] = 1;
        } else {
            //如果存在过的话   1++ 累计次数
            obj[chars]++;
        }
        //简写
        // !obj[chars] ? obj[chars] = 1 : obj[chars]++;
```

```
    }


    //循环保存的字符串对象  对比出里面哪个字符最大
    for (var item in obj) {
        if (maxChars == "" || obj[item] > obj[maxChars]) {
            //每次保存次数最大字符
            maxChars = item;
        }
    }
    //最大次数
    console.log(obj[maxChars]);
    //出现最多字符
    console.log(maxChars);
}
//第二种方法
function methodSecond() {
    var str = "晨哥帅不帅";
    var obj = {}; //把每个字符单个存在对象中
    var len = 0; //字符长度
    var maxChars = ""; //存储出现次数最多的字符

    //  /./g 匹配除了换行符 \n 之外的任何单字符
    str.replace(/./g, function(a) {
        obj[a] = (obj[a] || 0) + 1;
        if (obj[a] > len) {
            //最长的长度赋值给len
            len = obj[a];
            //保存最多的字符
            maxChars = a;
        }
    })
    //输出出现次数最多字符
    console.log(maxChars);
}
//第三种方法
function methodThird() {
    var str = "晨哥帅不帅";
    var obj = {}; //把每个字符单个存在对象中
    var len = 0; //字符长度
    var maxChars = ""; //存储出现次数最多的字符

    for (var i = 0; i < str.length; i++) {
        var chars = str[i];
        obj[chars] = (obj[chars] || 0) + 1;
        if (obj[chars] > len) {
            //最长的长度赋值给len
```

```
            len = obj[chars];
            //保存最多的字符
            maxChars = chars;
        }
    }
    //输出出现次数最多字符
    console.log(maxChars);
}
```

3:编程实现javascript在String中写一个trim,要求能够去除一个字符串开始和结尾的空格?

```
//第一种方法
function methodFirst() {
    var str = "   晨哥帅不帅  ";
    function trim(str) {
        //匹配开始位置是否有空格 如果有 则删除 然后递归重新调用trim方法检测
        if (str.charAt(0) == " ") {
            str = str.substr(1, str.length - 1);
            trim();
        }
        //匹配结束位置是否有空格 如果有 则删除 然后递归重新调用trim方法检测
        if(str.lastIndexOf(" ") > 0){
             str = str.substr(0, str.length - 1);
             trim();
        }
    }
    trim(str);
}

//第二种方法
function methodSecond() {
    var str = "   晨哥帅不帅  ";
    // ^是开始位置 /s代表匹配空格 *代表从索引0开始匹配多次   |代表或 $代表结束位置。意思就是从这个字符串开始位置匹配所有的空格或者结束位置匹配所有的空格 替换为空
    return str.replace(/(^\s*)|(\s*$)/g, "");
}
```

4:实现一个数组去重功能?

```
//方法一
function unique(array){
var n = [];
for(var i = 0; i < array.length; i++){
    if (n.indexOf(array[i]) == -1)        n.push(array[i]);
    }
    return n;
}
unique([1,2,3,4,5,2,3,4])



//方法二
var arr = [1,2,3,4,5,2,3];
var obj = {};
for(var i=0;i< arr.length;i++){
    if(obj[arr[i]] == arr[i]){
        arr.splice(i,i+1);
    }else{
        obj[arr[i]] = arr[i];
    }
}
```

5:用JS手写快速排序?

```
//冒泡排序
function sort(arr) {
    var len = arr.length;
    for (var i = 0; i < len; i++) {
        for (var j = 0; j < len - 1 - i; j++) {
            if (arr[j] > arr[j+1]) {        //相邻元素两两对比
                var temp = arr[j+1];        //元素交换
                arr[j+1] = arr[j];
                arr[j] = temp;
            }
        }
    }
    return arr;
}
//快速排序
```

```
  function quickSort(arr){
    //如果数组<=1,则直接返回
    if(arr.length<=1){return arr;}
    var pivotIndex=Math.floor(arr.length/2);
    //找基准，并把基准从原数组删除
    var pivot=arr.splice(pivotIndex,1)[0];
    //定义左右数组
    var left=[];
    var right=[];

    //比基准小的放在left，比基准大的放在right
    for(var i=0;i<arr.length;i++){
        if(arr[i]<=pivot){
            left.push(arr[i]);
        }
        else{
            right.push(arr[i]);
        }
    }
    //递归
    return quickSort(left).concat([pivot],quickSort(right));
}
```

6：如何实现以下函数?

```
console.log(add(2)(5));
console.log(add(2,5));

//答案
var add = function() {
    if(arguments.length ==1){
        return  function(y) {
            return x + y;
        };
     }else{
        return  x+r;
    }
};
```