编程题

1 写一个通用的事件侦听器函数

```
 // event(事件)工具集，来源：github.com/markyun
    markyun.Event = {

        // 视能力分别使用dom0||dom2||IE方式 来绑定事件
        // 参数：  操作的元素,事件名称 ,事件处理程序
        addEvent : function(element, type, handler) {
            if (element.addEventListener) {
                //事件类型、需要执行的函数、是否捕捉
                element.addEventListener(type, handler, false);
            } else if (element.attachEvent) {
                element.attachEvent('on' + type, function() {
                    handler.call(element);
                });
            } else {
                element['on' + type] = handler;
            }
        },
        // 移除事件
        removeEvent : function(element, type, handler) {
            if (element.removeEventListener) {
                element.removeEventListener(type, handler, false);
            } else if (element.datachEvent) {
                element.detachEvent('on' + type, handler);
            } else {
                element['on' + type] = null;
            }
        },
        // 阻止事件 (主要是事件冒泡，因为IE不支持事件捕获)
        stopPropagation : function(ev) {
            if (ev.stopPropagation) {
                ev.stopPropagation();
            } else {
                ev.cancelBubble = true;
            }
        },
```

```
        // 取消事件的默认行为
        preventDefault : function(event) {
            if (event.preventDefault) {
                event.preventDefault();
            } else {
                event.returnValue = false;
            }
        },
        // 获取事件目标
        getTarget : function(event) {
            return event.target || event.srcElement;
        }
```

2 如何判断一个对象是否为数组

```
function isArray(arg) {
    if (typeof arg === 'object') {
        return Object.prototype.toString.call(arg) === '[object Array]';
    }
    return false;
}
```

3 冒泡排序

- 每次比较相邻的两个数，如果后一个比前一个小，换位置

```
var arr = [3, 1, 4, 6, 5, 7, 2];

function bubbleSort(arr) {
for (var i = 0; i < arr.length - 1; i++) {
    for(var j = 0; j < arr.length - i - 1; j++) {
        if(arr[j + 1] < arr[j]) {
            var temp;
            temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
        }
    }
}
return arr;
}

console.log(bubbleSort(arr));
```

4 快速排序

> 采用二分法，取出中间数，数组每次和中间数比较，小的放到左边，大的放到右边

快速排序的思想很简单，整个排序过程只需要三步：

- （1）在数据集之中，找一个基准点
- （2）建立两个数组，分别存储左边和右边的数组
- （3）利用递归进行下次比较

```
var arr = [3, 1, 4, 6, 5, 7, 2];

function quickSort(arr) {
    if(arr.length == 0) {
        return [];    // 返回空数组
    }

    var cIndex = Math.floor(arr.length / 2);
    var c = arr.splice(cIndex, 1);
    var l = [];
    var r = [];

    for (var i = 0; i < arr.length; i++) {
        if(arr[i] < c) {
            l.push(arr[i]);
        } else {
            r.push(arr[i]);
        }
    }

    return quickSort(l).concat(c, quickSort(r));
}

console.log(quickSort(arr));
```

5 编写一个方法 求一个字符串的字节长度

- 假设：一个英文字符占用一个字节，一个中文字符占用两个字节

```
function GetBytes(str){

        var len = str.length;

        var bytes = len;
```

```
        for(var i=0; i<len; i++){

            if (str.charCodeAt(i) > 255) bytes++;

        }

        return bytes;

    }

alert(GetBytes("你好,as"));
```

6 bind的用法，以及如何实现bind的函数和需要注意的点

- `bind`的作用与`call`和`apply`相同，区别是`call`和`apply`是立即调用函数，而`bind`是返回了一个函数，需要调用的时候再执行。 一个简单的`bind`函数实现如下

```
Function.prototype.bind = function(ctx) {
    var fn = this;
    return function() {
        fn.apply(ctx, arguments);
    };
};
```

7 实现一个函数clone

> 可以对`JavaScript`中的5种主要的数据类型,包括`Number`、`String`、`Object`、`Array`、`Boolean`）进行值复

- 考察点1：对于基本数据类型和引用数据类型在内存中存放的是值还是指针这一区别是否清楚
- 考察点2：是否知道如何判断一个变量是什么类型的
- 考察点3：递归算法的设计

```
// 方法一：
  Object.prototype.clone = function(){
          var o = this.constructor === Array ? [] : {};
          for(var e in this){
                  o[e] = typeof this[e] === "object" ?
this[e].clone() : this[e];
          }
```

```
          return o;
  }

 //方法二:
   /**
      * 克隆一个对象
      * @param Obj
      * @returns
      */
     function clone(Obj) {
         var buf;
         if (Obj instanceof Array) {
             buf = [];                        //创建一个空的数组
             var i = Obj.length;
             while (i--) {
                 buf[i] = clone(Obj[i]);
             }
             return buf;
         }else if (Obj instanceof Object){
             buf = {};                       //创建一个空对象
             for (var k in Obj) {            //为这个对象添加新的属性
                 buf[k] = clone(Obj[k]);
             }
             return buf;
         }else{                              //普通变量直接赋值
             return Obj;
         }
     }
```

8 下面这个ul，如何点击每一列的时候alert其index

> □考察闭包

```
 <ul id="test">
     <li>这是第一条</li>
     <li>这是第二条</li>
     <li>这是第三条</li>
 </ul>
  // 方法一:
  var lis=document.getElementById('2223').getElementsByTagName('li');
  for(var i=0;i<3;i++)
  {
```

```
      lis[i].index=i;
      lis[i].onclick=function(){
          alert(this.index);
   }

 //方法二：
 var
lis=document.getElementById('2223').getElementsByTagName('li')
;
 for(var i=0;i<3;i++)
 {
     lis[i].index=i;
     lis[i].onclick=(function(a){
         return function() {
             alert(a);
         }
     })(i);
 }
```

9 定义一个log方法，让它可以代理console.log的方法

```
可行的方法一：

function log(msg)  {
    console.log(msg);
}

log("hello world!") // hello world!
```

> 如果要传入多个参数呢？显然上面的方法不能满足要求，所以更好的方法是：

```
function log(){
    console.log.apply(console, arguments);
};
```

10 输出今天的日期

> 以`YYYY-MM-DD`的方式，比如今天是2014年9月26日，则输出2014-09-26

```
var d = new Date();
  // 获取年，getFullYear()返回4位的数字
  var year = d.getFullYear();
  // 获取月，月份比较特殊，0是1月，11是12月
  var month = d.getMonth() + 1;
  // 变成两位
  month = month < 10 ? '0' + month : month;
  // 获取日
  var day = d.getDate();
 day = day < 10 ? '0' + day : day;
 alert(year + '-' + month + '-' + day);
```

11 用js实现随机选取10–100之间的10个数字，存入一个数组，并排序

```
var iArray = [];
 funtion getRandom(istart, iend){
         var iChoice = istart - iend +1;
         return Math.floor(Math.random() * iChoice + istart;
 }
 for(var i=0; i<10; i++){
         iArray.push(getRandom(10,100));
 }
 iArray.sort();
```

12 写一段JS程序提取URL中的各个GET参数

> 有这样一个`URL`：`http://item.taobao.com/item.htm?a=1&b=2&c=&d=xxx&e`，请写一段JS程序提取URL中的各个GET参数(参数名和参数个数不确定)，将其按`key-value`形式返回到一个`json`结构中，如`{a:'1', b:'2', c:'', d:'xxx', e:undefined}`

```
function serilizeUrl(url) {
     var result = {};
     url = url.split("?")[1];
     var map = url.split("&");
     for(var i = 0, len = map.length; i < len; i++) {
         result[map[i].split("=")[0]] = map[i].split("=")[1];
     }
     return result;
 }
```

13 写一个function，清除字符串前后的空格

> 使用自带接口trim()，考虑兼容性：

```
if (!String.prototype.trim) {
    String.prototype.trim = function() {
        return this.replace(/^\s+/, "").replace(/\s+$/,"");
    }
}

 // test the function
 var str = " \t\n test string ".trim();
 alert(str == "test string"); // alerts "true"
```

14 实现每隔一秒钟输出1,2,3...数字

```
for(var i=0;i<10;i++){
  (function(j){
     setTimeout(function(){
       console.log(j+1)
     },j*1000)
   })(i)
}
```

15 实现一个函数，判断输入是不是回文字符串

```
function run(input) {
  if (typeof input !== 'string') return false;
  return input.split('').reverse().join('') === input;
}
```

16、数组扁平化处理

> 实现一个flatten方法，使得输入一个数组，该数组里面的元素也可以是数组，该方法会输出一个扁平化的数组

```
function flatten(arr){
    return arr.reduce(function(prev,item){
        return prev.concat(Array.isArray(item)?
flatten(item):item);
    },[]);
}
```

17、实现一个函数clone，可以对JavaScript中的5种主要的数据类型（包括Number、String、Object、Array、Boolean）进行值复制

```
Object.prototype.clone = function(){
    var o = this.constructor === Array ? [] : {};
    for(var e in this){
      o[e] = typeof this[e] === "object" ? this[e].clone() : this[e];
    }
    return o;
  }
```