

数组

切分数组

每3个分成一个数组，放入一个数组

```
var data = ['法国','澳大利亚','智利','新西兰','西班牙','加拿大','阿根廷','美国','0','国产','波多黎各','英国','比利时','德国','意大利','意大利',];
var result = [];
for(var i=0, len=data.length; i<len; i+=3) {
    result.push(data.slice(i, i+3));
}

//[['法国','澳大利亚','智利'], ['新西兰','西班牙','加拿大'], ['阿根廷','美国','0'], ['国产','波多黎各','英国'], ['比利时','德国','意大利'], ['意大利'], ]
```

删除指定元素

```
var arr=[1,2,3,4,5];
arr.splice(2,1);
//[1,2,4,5]
```

去重

1. 遍历数组法

```
function unique1(array) {
var n = []; //一个新的临时数组
for(var i = 0; i < array.length; i++) {
//如果当前数组的第i已经保存进了临时数组，那么跳过，否则把当前项push到临时数组里面
if (n.indexOf(array[i]) == -1) n.push(array[i]);
}
return n;
}
```

2. 数组下标判断法

```
function unique2(array) {
var n = [array[0]]; //保存进第一项
//从第二项开始遍历
for(var i = 1; i < array.length; i++) {
//如果当前数组的第i项在当前数组中第一次出现的位置不是i，那么忽略掉。否则存入结果数组
if (array.indexOf(array[i]) == i) n.push(array[i]);
}
return n;
}
```

3. 排序后相邻去除法

```
function unique3(array) {
    array.sort();
    var re=[array[0]]; //保存进第一项
    for(var i = 1; i < array.length; i++){
        if( array[i] !== re[re.length-1]) re.push(array[i]);
    }
}
```

```
    return re;
}
```

4. 对象键值对法（速度最快，占用的内存大）

新建一js对象以及新数组，遍历传入数组时，判断值是否为js对象的键，不是的话给对象新增该键并赋值为1。

```
function unique4(ar) {
    var tmp = {},
        ret = [];

    for (var i = 0, j = ar.length; i < j; i++) {
        if (!tmp[ar[i]]) {
            tmp[ar[i]] = 1;
            ret.push(ar[i]);
        }
    }

    return ret;
}
```

5. 优化遍历数组法（推荐）

获取没重复的最右一值放入新数组，检测到有重复值时终止当前循环同时进入顶层循环的下一轮判断。

```
function unique5(array) {
    var r = [];
    for(var i = 0, l = array.length; i < l; i++) {
        for(var j = i + 1; j < l; j++) {
            if (array[i] === array[j]) j = ++i;
        }
        r.push(array[i]);
    }
    return r;
}
```