## 数组去重:第二种最快

**1、**

Array.prototype.unique1 = function()

{

var n = []; //一个新的临时数组

for(var i = 0; i < this.length; i++) //遍历当前数组

{

//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(this[i]) == -1) n.push(this[i]);

}

return n;

}

Var arr=[1,2,3,3,3,4,2,1,5,6];

Alert(arr.unique1());

**2、**

Array.prototype.unique2 = function()

{

var n = {},r=[]; //n为hash表，r为临时数组

for(var i = 0; i < this.length; i++) //遍历当前数组

{

if (!n[this[i]]) //如果hash表中没有当前项

{

n[this[i]] = true; //存入hash表

r.push(this[i]); //把当前数组的当前项push到临时数组里面

}

}

return r;

}

**3、**

Array.prototype.unique3 = function()

{

var n = [this[0]]; //结果数组

for(var i = 1; i < this.length; i++) //从第二项开始遍历

{

//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，

//那么表示第i项是重复的，忽略掉。否则存入结果数组

if (this.indexOf(this[i]) == i) n.push(this[i]);

}

return n;

}

## 排序算法

**1、冒泡算法：**

var arr=[10,2,5,1,7,6,3];

function sort(array){

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

for(var j=i;j<array.length;j++){

if(array[i]>array[j]){

var temp=array[i];

array[i]=array[j];

array[j]=temp;

}

}

}

return array;

}

alert(sort(arr));