

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>

</body>
</html>
<script>
```

```
// js对象属性 通过点(.) 和 方括号([]) 的不同之处
// 1、点操作符：静态的。右侧必须是一个以属性名称命名的简单标识符。属性名用一个标识符来表示。标识符必须直接出现再js程序中，
//它们不是数据类型，因此程序无法修改它们。
// 2、中括号操作符：动态的。方括号里必须是一个计算结果为字符串的表达式，属性名通过字符串表示。字符串是js的数据类型，
//再程序运行时可以修改和创建它们。
// 主要有以下区别：
```

```
//1、[]--可以用变量作为属性名或访问,而点方法不可以；
```

```
var obj = {};
obj.name = '张三';
var myName = 'name';
console.log(obj.myName);//undefined,访问不到对应的属性
console.log(obj[myName]);//张三
```

```
var person = {
  name: 'gogo'
};
console.log(person["name"]);//gogo
console.log(person.name); //gogo
// 优点：1,可以通过变量来访问属性
var propertyName = 'name';
console.log(person[propertyName]); //gogo
var propertyName2 = 'name2';
console.log(person[propertyName2]); //undefined
```

```
//2、[]中括号法--可以用数字作为属性名,而点语法不可以；
```

```
var obj1={};
// obj1.1=1;//Unexpected number
obj1[2]=2;
// console.log(obj1.1)
console.log(obj1[2]);//2
// console.log(obj1.2)
console.log(obj1){2: 2}
```

```
// 3、[] 可以动态访问的属性名，可以在程序运行时创建和修改属性，点操作符就不行！
```

```
// （即 []--可以动态设置和获取）
```

```
var customer={};
var addr=['北京','上海','广州','深圳'];
```

```

for(i=0;i<4;i++){
    customer["address"+i]=addr[i];
}
console.log(addr);
console.log(customer);
var str = "";
for(i=0;i<4;i++){
    str += customer["address" + i] + "\t";
}
console.log(str);

```

//4,如果属性名中包含会导致语法错误的字符,或者属性名是关键字或者保留字,也可以使用方括号表示法。

//如: (属性名是关键字或者保留字--都可以点语法不严密,不报错,写法提示有错)

```

person['first name'] = 'gogo2'; //first name包含一个空格
console.log(person['first name']);
// console.log(person.first name)//书写都通不过
person['for'] = 'gogo_for'; //first name包含一个空格
person.if = 'gogo_if'; //first name包含一个空格
console.log(person['for']);//gogo_for
console.log(person.for);//gogo_for
console.log(person['if']);//gogo_if
console.log(person.if);//gogo_if

```

//简单利用: 在数组原型链上增加一个去重得的方法,并能实现链式写法。

```

Array.prototype.myDistinct=function () {
    var obj={};
    for(var i=0;i<this.length;i++){
        var cur=this[i];
        if(obj[cur]==cur){//对象的属性名不能重复,重复就是修改;让对象的属性名和属性值相同,借以保存不重复的数组元素
            this[i]=this[this.length-1];
            this.length--;
            i--;
            continue;
        }
        obj[cur]=cur;
    }
}

// console.log(obj); //{2: 2, 3: 3, 4: 4, 5: 5}
obj=null;
return this;

};
var arr=[5,3,3,4,5,4,2];
arr.myDistinct().sort().pop();
console.log(arr); //[2, 3, 4]
var arr1=[3,'a',4,5,4,'b','a'];
console.log(arr1.myDistinct()); //[3, "a", 4, 5, "b"]

```

</script>