

# JavaScript 对象

Tags

[1 对象 Object](#)

[2 创建对象的三种方式](#)

[3 构造函数和对象](#)

[4 new关键字执行过程](#)

[5 遍历对象](#)

## 1 对象 Object

在JS中，对象是一组无序的相关属性和方法的几何，所有的事物都是对象，例如字符串、数组、数值、函数等。

对象由属性和方法构成。

## 2 创建对象的三种方式

- 利用字面量创建对象

对象字面量：即花括号{}里面包含这个具体事物的属性和方法。

注：小括号用于运算，中括号用于数组。

```
//字面量创建对象
var obj = {
    name: 'yyh',
    age : 18,
    lover : 'zjd',//属性
    sayHi: function(){//方法
        console.log('hi!');
    }
}

//使用
console.log(obj.name);
console.log(obj['age']);
obj.sayHi();
```

- 利用 new Object 创建对象

### ▲注意利用等号赋值

```
var obj = new Object(); //创建了一个空对象
obj.name = 'yyh';
obj.age = 18; //用等号
obj.sayHi = function(){
    ...
}
```

- 构造函数

使用构造函数：前面两种创建对象的方式一次只能创建一个对象。构造函数可以用于封装对象。

```
function 构造函数名(){
    this.属性 = 值;
    this.方法 = function(){}
}

new 构造函数名();
```

### ▲

1. 构造函数首字母要大写
2. 要调用构造函数，必须使用new

```
function Star(name, age, sex){
    this.name = name;
    this.age = age;
    this.sex = sex;
}

var ldh = new Star('刘德华', 18, 'male');
```

## 3 构造函数和对象

- 构造函数约等于class。
- 通过new关键字创建对象的过程也称为对象实例化。

## 4 new关键字执行过程

---

1. new 构造函数会在内存中创建了一个空的对象。
2. this会指向刚才创建的空对象。
3. 执行构造函数代码，给空对象添加属性和方法。
4. 返回这个对象。(无须return)

## 5 遍历对象

---

```
for (var key in obj){  
    console.log(key); //输出属性名  
    console.log(obj[key]); //属性值  
}
```