

1. 对象

1.1. 对象的概念

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

对象是由属性和方法组成的：

- 属性：事物的特征，在对象中用属性来表示（常用名词）
- 方法：事物的行为，在对象中用方法来表示（常用动词）

1.2. 创建对象的三种方式

JavaScript 中，现阶段我们可以采用三种方式创建对象（`object`）

- 利用字面量创建对象
- 利用 `new Object` 创建对象
- 利用构造函数创建对象

1.2.1. 利用字面量创建对象

1.2.1.0.1. 例子：

```
<script>
  // 1、利用字面量创建对象
  // 创建一个空的对象
  var obj = {};
</script>
```

```
<script>
  var obj = {
    username: '你的名字',
    age : 18,
    sex : '男性',
    sayHi : function () {
      console.log('HiHiHi');
    }
  }
</script>
```

对象字面量：就是花括号 `{ }` 里面包含了表达这个具体事物（对象）的属性和方法

`{ }` 里面采取键值对的形式表示

- 键：相当于属性名
- 值：相当于属性值，可以是任意类型的值（数字类型、字符串类型、布尔类型，函数类型等）

注意：多个属性或者方法中间用逗号隔开，方法冒号后面跟的是一个匿名函数

1.2.2. 调用对象

- 对象里面的属性调用：对象.属性名，这个小点.就理解为“的”
- 对象里面属性的另一种调用方式：对象['属性名']，注意方括号里面的属性必须加引号，我们后面会用
- 对象里面的方法调用：对象.方法名()，注意这个方法名字后面一定加括号

```
<script>
  var obj = {
    username: '你的名字',
    age : 18,
    sex : '男性',
```

```

        sayHi : function () {
            console.log('HiHiHi');
        }
    }
    //对象名.属性名
    console.log(obj.username,obj.age);
    // 对象名['属性名']
    console.log(obj['age']);
    // 对象名.方法名() ,注意, 方法名后面一定要带括号
    obj.sayHi();
</script>

```

1.2.2.0.1. 练习：使用字面量，创建一个关于你的属性 对象。

1.2.3. 利用 **new Object** 创建对象

跟之前的 `new Array()` 原理一致： `var 对象名 = new Object();`

使用的格式： `对象.属性 = 值`

```

<script>
    // 利用 new object 创建对象
    var obj = new Object(); //创建了一个空的对象
    obj.name = '张三丰';
    obj.age = 18;
    obj.sex = '男';
    obj.sayHi = function() {
        console.log('hi~');
    }

    //1.利用等号赋值的方法添加对象
    //2.每个属性和方法之间用分号结束
    console.log(obj.name);
    console.log(obj['sex']);
    obj.sayHi();
</script>

```

1.2.3.0.1. 练习：使用 **new object**，创建一个关于你的属性 对象

1.2.4. 使用构造函数创建对象

构造函数：是一种特殊的函数，主要用来初始化对象，即为对象成员变量赋初始值，它总与 **new** 运算符一起使用。我们可以把对象中一些公共的属性和方法抽取出来，然后封装到这个函数里面。

在 js 中，使用构造函数要时时要注意以下两点：

- 构造函数用于创建某一类对象， **其首字母要大写**
- 构造函数要和 **new** 一起使用才有意义

```
<script>
//构造函数的语法格式
function 构造函数名() {
    this.属性 = 值;
    this.方法 = function() {}
}
new 构造函数名();
</script>
```

1.2.4.0.1. 例子：

```
<script>
function Star(uname,age,sex) {
    this.name = uname;
    this.age = age;
    this.sex = sex;
    this.sing = function(sang){
        console.log(sang);
    }
}
var ldh = new Star('刘德华',18,'男');
console.log(typeof ldh) // object对象, 调用函数返回的是对象
console.log(ldh.name);
```

```
console.log(ldh['sex']);  
ldh.sing('冰雨');  
//把冰雨传给了sang  
  
var zxy = new Star('张学友',19,'男');  
</script>
```

注意：

1. 构造函数名字首字母要 **大写**
2. 构造函数 **不需要return** 就可以返回结果
3. 调用构造函数 **必须使用 new**
4. 我们只要new Star() 调用函数就创建了一个对象
5. 我们的属性和方法前面 **必须加this**

1.3. 遍历对象

使用 **for...in** 语句用于对 **数组** 或者 **对象** 的属性进行循环操作

1.3.1. 语法结构

```
<script>  
  for(变量 in 对象名字){  
    // 在此执行代码  
  }  
</script>
```

1.3.1.0.1. 例子：

```
<script>  
  var obj = {  
    username: '你的名字',  
    age : 18,  
    sex : '男性',  
    sayHi : function () {  
      console.log('HiHiHi');  
    }  
  }  
</script>
```

```
    }  
  }  
  for(var k in obj){  
    console.log(k);      //k变量输出 属性名  
    console.log(obj[k]);  //使用 obj[k]   输出 属性值  
  }  
</script>
```