

# JS对象的创建和使用

JavaScript 是一种面向对象的编程语言，在 JavaScript 中几乎所有的东西都是对象。因此，要想有效的使用 JavaScript，首先需要了解对象的工作原理以及如何创建并使用对象。

我们可以将对象看作是一个属性的无序集合，每个属性都有一个名称和值（键/值对）。通过《[JS数据类型](#)》一节我们知道，数组是值的集合，每个值都有一个数字索引（从零开始，依次递增）。对象类似与数组，不同的是对象中的索引是自定义的，例如 name（姓名）、age（年龄）、gender（性别）等。

## JS 创建对象

您可以使用花括号 `{}` 来创建对象，`{}` 中用来定义对象中的属性。属性是一个个 `键:值` 对的组合，其中键（属性名称）始终是字符串类型的，而值（属性值）则可以是任意类型，例如字符串、数组、函数或其它对象等。不同的属性之间使用逗号进行分隔。示例代码如下：

```
01.  var person = {
02.      name: "Peter",
03.      age: 28,
04.      gender: "Male",
05.      displayName: function() {
06.          document.write(this.name);
07.      }
08.  };
```

提示：在对象中定义的函数通常被称为方法。

上面示例中创建了一个名为 person 的对象，该对象中包含三个属性 name、age、gender 和一个方法 displayName()。displayName() 方法中的 this.name 表示访问当前对象中的 name 属性，会被 JavaScript 解析为 person.name。

在定义对象时，属性名称虽然是字符串类型，但通常不需要使用引号来定义，但是以下三种情况则需要为属性名添加引号：

- 属性名为 JavaScript 中的保留字；
- 属性名中包含空格或特殊字符（除字母、数字、\_ 和 \$ 以外的任何字符）；
- 属性名以数字开头。

示例代码如下：

```
01.  var person = {
02.      "first name": "Peter",
03.      "current age": 28,
04.      gender: "Male"
```

```
05.  };
```

## 访问对象的属性

要访问或获取属性的值，您可以使用 `对象名.属性名` 或者 `对象名["属性名"]` 的形式，如下例所示：

```
01.  var person = {
02.      name: "Peter",
03.      age: 28,
04.      gender: "Male",
05.      displayName: function() {
06.          document.write(this.name);
07.      }
08.  }
09.
10.  document.write("姓名: " + person.name + "<br>");    // 输出: 姓名: Peter
11.  document.write("年龄: " + person["age"]);           // 输出: 年龄: 28
```

在访问对象属性时，使用 `对象名.属性名` 的形式更易于代码的编写，但并不是所有情况下都可以使用。如果属性名中包含空格或者特殊字符，则不能使用 `对象名.属性名` 的形式来访问对象属性，必须使用 `对象名["属性名"]` 的形式才行，如下例所示：

```
01.  var person = {
02.      "first name": "Peter",
03.      "current age": 28,
04.      gender: "Male"
05.  };
06.
07.  document.write(person["first name"]);    // 输出: Peter
08.  document.write(person.current age);     // 报错: missing ) after argument
      list
```

使用 `对象名["属性名"]` 的形式访问对象属性相对比较灵活，您除了可以直接通过属性名访问属性外，还可以将属性名称赋值给变量，然后再通过这个变量来访问属性的值，如下所示：

```
01.  var person = {
02.      name: "Peter",
03.      age: 28,
04.      gender: "Male"
05.  };
06.
07.  var key = "age";
08.  document.write(person[key]); // 输出: 28
```

## 设置修改对象的属性

使用 `对象名.属性名` 或者 `对象名["属性名"]` 的形式除了可以获取对象的属性值外，也可以用来设置或修改对象的属性值，如下例所示：

```
01.  var person = {  
02.      name: "Peter",  
03.      age: 28,  
04.      gender: "Male"  
05.  };  
06.  
07.  person.phone = "15012345678";  
08.  person.age = 20;  
09.  person["name"] = "Peter Parker";  
10.  
11.  for (var key in person) {  
12.      document.write(key + ": " + person[key] + "<br>")  
13.  }
```

输出结果如下所示：

```
name: Peter Parker  
age: 20  
gender: Male  
phone: 15012345678
```

## JS 删除对象的属性

您可以使用 `delete` 语句来删除对象中的属性，如下例所示：

```
01.  var person = {  
02.      name: "Peter",  
03.      age: 28,  
04.      gender: "Male",  
05.      phone: "15012345678"  
06.  };  
07.  
08.  delete person.gender;  
09.  delete person["phone"];  
10.  
11.  for (var key in person) {  
12.      document.write(key + ": " + person[key] + "<br>")  
13.  }
```

输出结果如下所示：

```
name: Peter  
age: 28
```

提示：delete 语句是从对象中删除指定属性的唯一方式，而将属性值设置为 undefined 或 null 仅会更改属性的值，并不会将其从对象中删除。

## JS 调用对象的方法

您可以像访问对象中属性那样来调用对象中的方法，如下例所示：

```
01.  var person = {  
02.      name: "Peter",  
03.      age: 28,  
04.      gender: "Male",  
05.      displayName: function() {  
06.          document.write(this.name);  
07.      }  
08.  };  
09.  
10.  person.displayName();           // 输出: Peter  
11.  person["displayName"]();       // 输出: Peter
```