

JavaScript 作用域

作用域可访问变量的集合。

JavaScript 作用域

在 JavaScript 中, 对象和函数同样也是变量。

在 JavaScript 中, 作用域为可访问变量, 对象, 函数的集合。

JavaScript 函数作用域: 作用域在函数内修改。

JavaScript 局部作用域

变量在函数内声明, 变量为局部作用域。

局部变量: 只能在函数内部访问。

实例

```
// 此处不能调用 carName 变量
function myFunction() {
  var carName = "Volvo";
  // 函数内可调用 carName 变量
}
```

[尝试一下 »](#)

因为局部变量只作用于函数内, 所以不同的函数可以使用相同名称的变量。

局部变量在函数开始执行时创建, 函数执行完后局部变量会自动销毁。

JavaScript 全局变量

变量在函数外定义, 即为全局变量。

全局变量有 **全局作用域**: 网页中所有脚本和函数均可使用。

实例

```
var carName = "Volvo";
// 此处可调用 carName 变量
function myFunction() {
  // 函数内可调用 carName 变量
}
```

[尝试一下 »](#)

如果变量在函数内没有声明 (没有使用 var 关键字), 该变量为全局变量。

以下实例中 carName 在函数内, 但是为全局变量。

实例

```
// 此处可调用 carName 变量
function myFunction() {
  carName = "Volvo";
  // 此处可调用 carName 变量
}
```

[尝试一下 »](#)

JavaScript 变量生命周期

JavaScript 变量生命周期在它声明时初始化。

局部变量在函数执行完毕后销毁。

全局变量在页面关闭后销毁。

函数参数

函数参数只在函数内起作用，是局部变量。

HTML 中的全局变量

在 HTML 中, 全局变量是 window 对象: 所有数据变量都属于 window 对象。

实例

```
//此处可使用 window.carName
function myFunction() {
  carName = "Volvo";
}
```

[尝试一下 »](#)

你知道吗？



你的全局变量，或者函数，可以覆盖 window 对象的变量或者函数。
局部变量，包括 window 对象可以覆盖全局变量和函数。

[← JavaScript JSON](#)[JavaScript 事件 →](#)**4 篇笔记****写笔记**

