



什么是作用域

什么是作用域？

几乎所有编程语言就是在变量中存储值，并且能读取和修改此值。事实上，在变量中存储值和取出值的能力，给程序赋予了状态。如果没有这样的概念，一个程序虽然可以执行一些任务，但是它们将会受到极大的限制而且不会非常有趣。但是这些变量该存储在哪，又给如何读取？为了完成这个目标，需要制定一些规则，这个规则就是：作用域。

常见的作用域主要分为几个类型：全局作用域、函数作用域、块状作用域、动态作用域。

对象	类型
global/window	全局作用域
function	函数作用域（局部作用域）
{ }	块状作用域
this	动态作用域

[!TIP]
如果一个 变量 或者其他表达式不在“当前的作用域”，那么JavaScript机制会继续沿着作用域链上查找直到全局作用域（global或浏览器中的window）如果找不到将不可被使用。作用域也可以根据代码层次分层，以便子作用域可以访问父作用域，通常是指沿着链式的作用域链查找而不能从父作用域引用子作用域中的变量和引用

全局作用域

变量在函数或者代码块 { } 外定义，即为全局作用域。不过，在函数或者代码块 { } 内未定义的变量也是拥有全局作用域的（不推荐）。

```
var carName = " Volvo";

// 此处可调用 carName 变量
function myFunction() {
    // 函数内可调用 carName 变量
}
```

上述代码中变量 carName 就是在函数外定义的，它是拥有全局作用域的。这个变量可以在任意地方被读取或者修改，当然如果变量在函数内没有明（没有使用 var 关键字），该变量依然为全局变量。

```
// 此处可调用 carName 变量

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

以上实例中 carName 在函数内，但是拥有全局作用域，它将作为 global 或者 window 的属性存在。

[!Warning]
在函数内部或代码块中没有定义的变量实际上是作为 window/global 的属性存在，而不是全局变量。换句话说没有使用 var 定义的变量虽然拥有全局作用域，但是它是可以被 delete 的，而全局变量不可以。

参考：<https://www.runoob.com/js/js-scope.html>

函数作用域

在函数内部定义的变量，就是局部作用域。函数作用域内，对外是封闭的，从外层的作用域无法直接访问函数内部的作用域！