## 基本概念

JavaScript ( JS ) 是一种具有函数优先的轻量级解释型或即时编译型的编程语言。

虽然它是作为开发Web 页面的脚本语言而出名的，但是它也被用到了很多非浏览器环境中，例如 Node.js、 Apache CouchDB 和 Adobe Acrobat。

JavaScript 是一种基于原型编程、多范式的动态脚本语言，并且支持面向对象、命令式和声明式（如函数式编程）风格。

## 变量（Variable）

[变量](https://developer.mozilla.org/en-US/docs/Glossary/Variable" \o "变量: A variable is a named location for storing a value. That way an unpredictable value can be accessed through a predetermined name.) 是存储值的容器。要声明一个变量，先输入关键字 let ，然后输入合适的名称：



行末的分号表示当前语句结束，不过只有在单行内需要分割多条语句时，这个分号才是必须的。（有些人认为每条语句末尾加分号是一种好的风格。

JavaScript 对大小写敏感，myVariable 和 myvariable 是不同的。

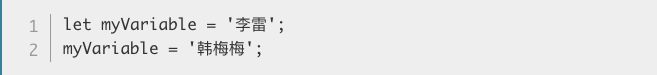
变量定义后可以进行赋值：



可以直接通过变量名取得变量的值：



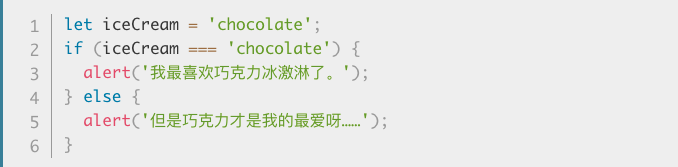
变量在赋值后是可以更改的：



变量可以有不同的数据类型：



## 条件语句



if ( ... ) 中的表达式进行测试，用（上文所提到的）等于运算符来比较变量 iceCream 与字符串 'chocolate' 是否相等。 如果返回 true，则运行第一个代码块；如果返回 false，则跳过第一块直接运行 else 之后的第二个代码块。

函数

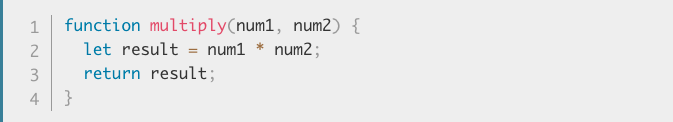
1. 
2. 

document.querySelector 和 alert 是浏览器内置的函数，随时可用。

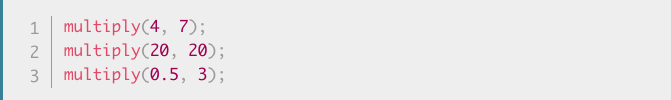
如果代码中有一个类似变量名后加小括号 () 的东西，很可能就是一个函数。函数通常包括[参数](https://developer.mozilla.org/en-US/docs/Glossary/Argument" \o "参数: An argument is a value (primitive or object) passed as input to a function.)，参数中保存着一些必要的数据。它们位于括号内部，多个参数之间用逗号分开。

alert() 函数在浏览器窗口内弹出一个警告框，还应为其提供一个字符串参数，以告诉它警告框里要显示的内容。

任何人都能定义自己的函数。下面的示例是为两个参数进行乘法运算的函数：



尝试在控制台运行这个函数，不妨多试几组参数，比如：



注：[return](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Statements/return) 语句告诉浏览器当前函数返回 result 变量。这是一点很有必要，因为函数内定义的变量只能在函数内使用。这叫做变量的 [作用域](https://developer.mozilla.org/en-US/docs/Glossary/Scope" \o "作用域: The current context of execution. The context in which values and expressions are \"visible,\" or can be referenced. If a variable or other expression is not \"in the current scope,\" then it is unavailable for use. Scopes can also be layered in a hierarchy, so that child scopes have access to parent scopes, but not vice versa.)。

## 事件

事件能为网页添加真实的交互能力。它可以捕捉浏览器操作并运行一些代码做为响应。最简单的事件是 [点击事件](https://developer.mozilla.org/zh-CN/docs/Web/Events/click)，鼠标的点击操作会触发该事件。 可尝试将下面的代码输入到控制台，然后点击页面的任意位置：



## JavaScript对象

在 JavaScript 中，大多数事物都是对象, 从作为核心功能的字符串和数组，到建立在 JavaScript 之上的浏览器 [API](https://developer.mozilla.org/en-US/docs/Glossary/API" \o "API: An API (Application Programming Interface) is a set of features and rules that exist inside a software program (the application) enabling interaction with it through software - as opposed to a human user interface. The API can be seen as a simple contract (the interface) between the application offering it and other items, such as third party software or hardware.) 。还可以自己创建对象，将相关的函数和变量高效地封装打包成便捷的数据容器。

**对象基础**

对象是一个包含相关数据和方法的集合（通常由一些变量和函数组成，我们称之为对象里面的属性和方法），让我们通过一个例子来了解它们。

首先, 将 [oojs.html](https://github.com/mdn/learning-area/blob/master/javascript/oojs/introduction/oojs.html) 文件复制到本地. 此文件包含非常少 — 一个供我们写源代码的 [<script>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/script" \o "HTML <script> 元素用于嵌入或引用可执行脚本。)标签, 一个供我们输入示例指令的 [<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input" \o "HTML <input> 元素用于为基于Web的表单创建交互式控件，以便接受来自用户的数据。) 标签，当页面被渲染时, 一些变量定义, 一个输出任何输入到[<input>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/input" \o "HTML <input> 元素用于为基于Web的表单创建交互式控件，以便接受来自用户的数据。)的内容输出到[<p>](https://developer.mozilla.org/zh-CN/docs/Web/HTML/Element/p" \o "HTML <p>元素（或者说 HTML 段落元素）表示文本的一个段落。该元素通常表现为一整块与相邻文本分离的文本，或以垂直的空白隔离或以首行缩进。另外，<p> 是块级元素。)标签的函数。我们用这个文件做为基础探索对象的基础语法.

如同Javascript中的很多东西一样，创建一个对象通常先定义初始化变量。 尝试在您已有的文件中JavaScript代码下面输入以下内容, 保存刷新页面:

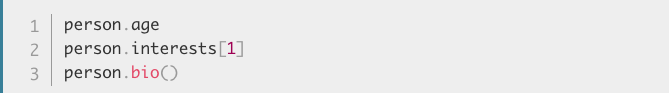


如果在浏览器控制台输入person，然后按下Enter(确认)键，会得到如下结果：



**点表示法**

在上面的例子中，使用了点表示法(dot notation)来访问对象的属性和方法。对象的名字表现为一个命名空间(namespace)，它必须写在第一位——当想要访问对象内部的属性或方法时，然后是一个点(.)，紧接着是想要访问的项目，标识可以是简单属性的名字(name)，或者是数组属性的一个子元素，又或者是对象的方法调用。如下所示：

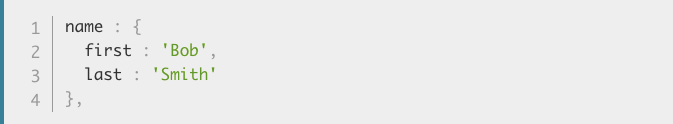


**子命名空间**

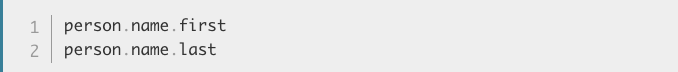
可以用一个对象来做另一个对象成员的值。例如将name成员



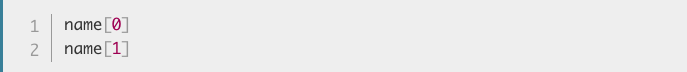
改成



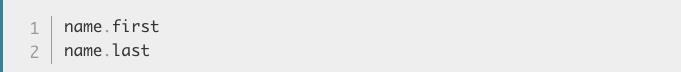
这样，就创建了一个子命名空间。看上去有点复杂，但用起来很简单，只需要链式的再使用一次点表示法，像这样：



注意：需要改变你之前的代码，从



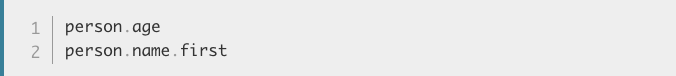
改成



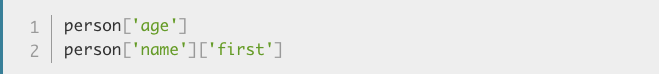
否则，方法将不再有效。

**括号表示法**

另外一种访问属性的方式是使用括号表示法(bracket notation)，替代这样的代码



使用如下所示的代码：



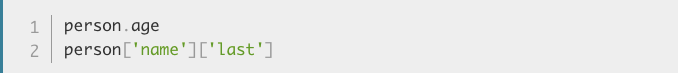
因为看起来很像访问一个数组的元素，使用了关联了值的名字，而不是索引去选择元素，所以对象有时被称之为关联数组(associative array)——对象做了字符串到值的映射，而数组做的是数字到值的映射。

**设置对象成员**

目前仅仅看到了如何访问对象的成员，而其实也可以设置对象成员的值，通过声明想要设置的成员，像这样：



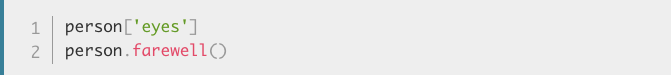
尝试这些代码，然后再查看这些成员是否已经被改变了



设置成员并不意味着只能更新已经存在的属性的值，完全可以创建新的成员：

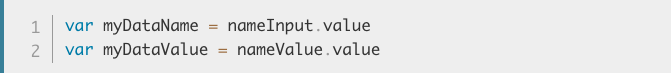


现在可以测试新创建的成员



括号表示法一个有用的地方是 它不仅可以动态的去设置对象成员的值，还可以动态的去设置成员的名字。

比如说，我们想让用户能够在他们的数据里存储自己定义的值类型，通过两个input框来输入成员的名字和值，通过以下代码获取用户输入的值：



可以这样把这个新的成员的名字和值加到person对象里：



**this的含义**



关键字"this"指向了当前代码运行时的对象( 原文：the current object the code is being written inside )——这里即指person对象。

"this"非常有用：它保证了当代码的上下文(context)改变时变量的值的正确性（比如：不同的person对象拥有不同的name这个属性，很明显greeting这个方法需要使用的是它们自己的name）。

用两个简单的person对象来说明：



在这里，person1.greeting()会输出："Hi! I'm Chris."；person2.greeting()会输出："Hi! I'm Brain."，在字面量的对象里this看起来不是很有用，但是当动态创建一个对象（例如使用构造器）时它非常有用。