第七章 学习第二语言：JavaScript

Block Mode(区块的图形化编程)→CoffeeScript→JavaScript

7.0.1

基本编程概念都是相同的，它适用于不同的编程语言。

接触第二语言能够让学生了解他们作为初学者所学的编程理念与专家们所用的是完全相同的。这一单元介绍了JavaScript，它与CoffeeScript非常接近，并且是专业开发人员最广泛使用的语言之一。在这一单元中，学生们将使用区块来学习JavaScript的语法，并且观察JavaScript和CoffeeScript的许多相似之处。他们会从区块的图形化编程方式直接转变到JavaScript 的文本代码编程中。

7.0.2 主题及要点

7.0 章节介绍

7.0.1 对象

7.0.2 主题及要点

7.0.3 关键术语

7.0.4 关键内容

7.1 学习计划

7.1.1 推荐时间规划

7.1.2 CSTA 标准

7.1.3 教学笔记

7.1.4 学习计划I 用区块的方式编写JavaScript

7.1.5 学习计划II 重建破碎的场景程序演示迭代开发模型

7.0.3 关键术语:

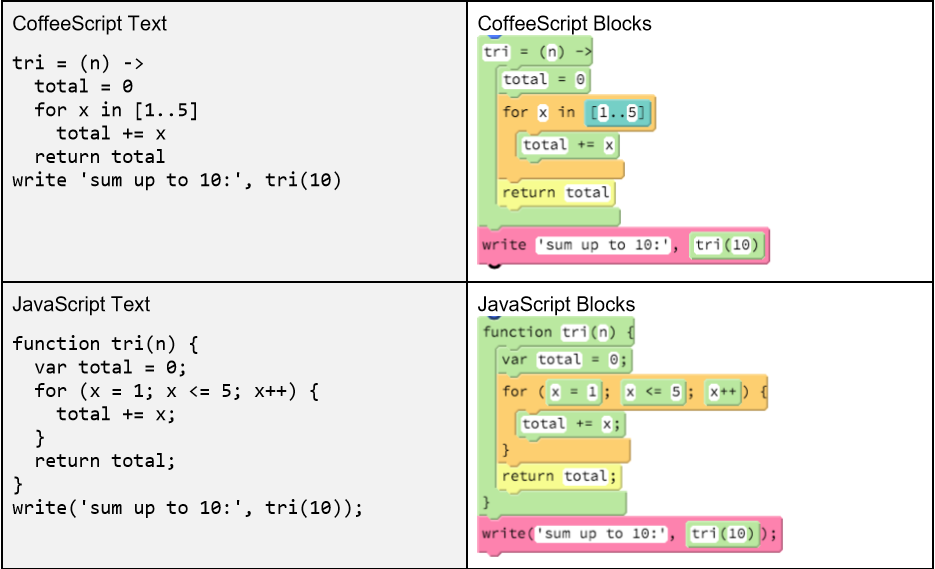
|  |  |
| --- | --- |
| Websites | Block-mode |
| JavaScript | Text-mode |
| Scripting Language | Settings box in Pencil Code |
| Programming Language |  |

7.0.4 关键内容

Pencil Code中的语言

每一种编程语言都有它自己的语法，也就是说，在语言中，有一组特定的单词和标点的来标记或指代功能。如前几章所示，使用一个块编辑器视图能够为学生展现有一种语言的语法规则。

一个感觉如何不同于CoffeeScript的JavaScript，下面会看到两函数定义，其中一个用是CoffeeScript写的，另一个用的是JavaScript。文本方式和区块方式都会进行展示。



虽然看着文本使语言显得非常不同，但是区块写法显示出JavaScript和CoffeeScript是密切相关的语言。一般来说，JavaScript需要更多的标点符号，但大多数代码的结构在两种语言之间基本上是相同的。

JavaScript和CoffeeScript之间选择

在Pencil Code的代码中，一个项目可以切换JavaScript或者CoffeeScript的编写方法通过点击蓝色栏中的“gear”按钮。当你选择JavaScript，它直接运行在你的浏览器。当你选择CoffeeScript，CoffeeScript编译器会在运行之前编译成JavaScript的程序。

CoffeeScript和JavaScript是密切相关的。它们在同一个对象上运行，并且在运行时具有相同的速度和消耗。CoffeeScript是在JavaScript之后被设计出来的，所以CoffeeScript语法有几个优点：

* CoffeeScript语法比JavaScript要求更少的标点符号，所以更容易进行没有语法错误的编写。
* CoffeeScript的直接支持更多的编程概念；例如，它有类和等待的语法
* CoffeeScript的使用有意义的缩进，这意味着它是不可能隐藏令人迷惑的嵌套错误。
* CoffeeScript避免了像JavaScript中近似零的值相等和偶然的全局变量这类常见错误

在Pencil Code中的的默认语言是CoffeeScript的，因为很容易的学会读和写CoffeeScript文本编码。为什么程序员会选择程序在JavaScript呢？因为JavaScript具有三个显著优势：

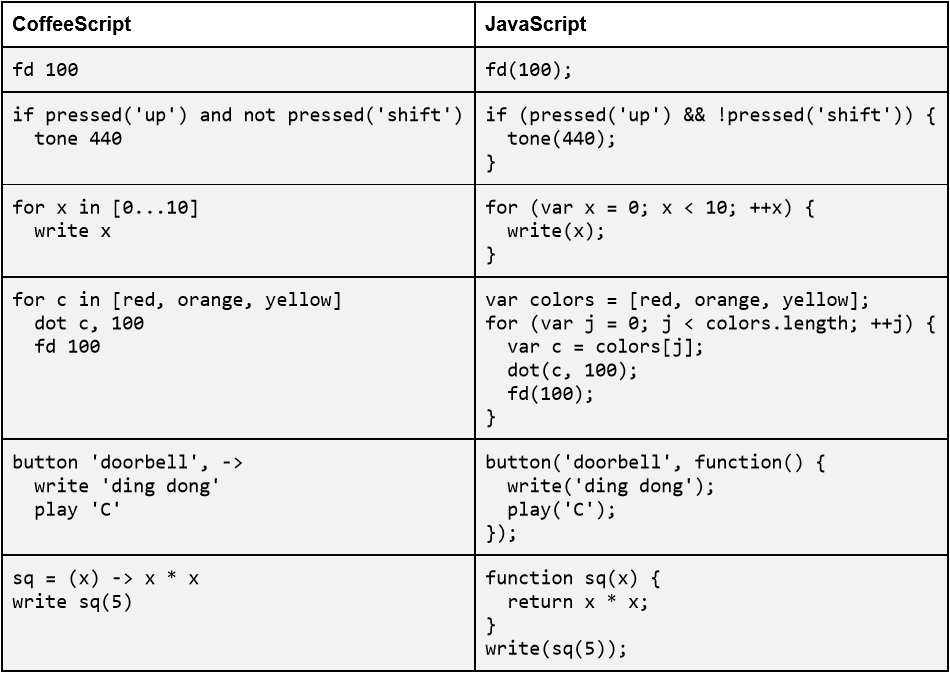
* JavaScript是一种由国际委员会设计的官方标准，它可以不用转换直接运行在Web浏览器中。
* JavaScript的程序员社区比CoffeeScript的社区大。
* 许多人都致力于提高JavaScript的未来版本，他们意识到了CoffeeScript中的改进。

JavaScript的不断发展，和JavaScript的未来版本将包括一些在CoffeeScript的创新。所以即使JavaScript狂爱人员，也值得了解CoffeeScript，因为它的许多思想代表JavaScript的未来。

然而对于CoffeeScript的爱好者，也值得知道JavaScript，因为其有更大的开发者社区。

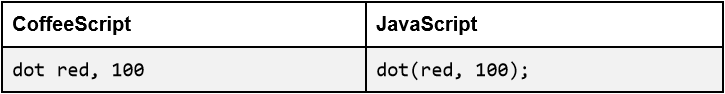
CoffeeScript和JavaScript之间的差异

JavaScript和CoffeeScript是密切相关的语言，并且了解一些额外的标记符号是必需的，有助于你直接从一个到另一个的理解。这里是一些CoffeeScript和JavaScript之间差异的一个总结，每一行代码都是等效的，我们将在下面进行详细讨论。



小括号和分号

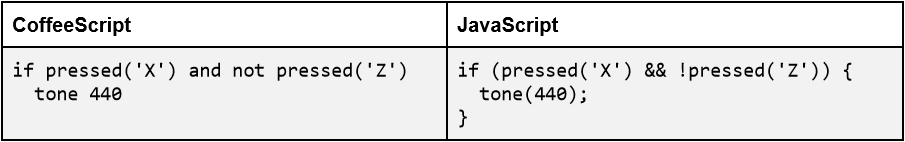
JavaScript需要在函数名之后加上括号来运行一个函数调用。这些括号在CoffeeScript中是可选的，但是在JavaScript中是必须要写的。



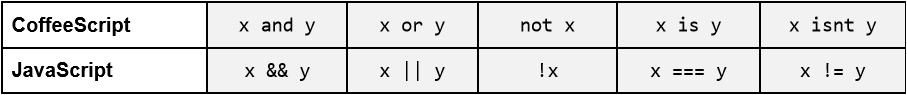
JavaScript也建议在每个完整的语句结束添加分号（包括函数调用，返回语句和break和continue语句）。然而，不是每一行代码都是一个完整的语句。例如，第一行的if是一个不完整的表述，则不应该函数体单独使用分号。

布尔表达式的标点符号

JavaScript还需要在if, while和switch后加小括号来包裹条件表达，在CoffeeScript中，并不需要这些括号。



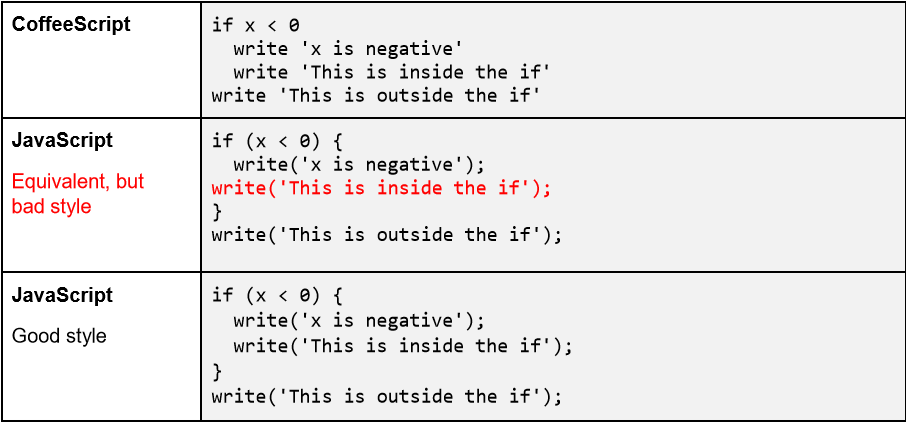
and, or, not , is 和 isnt这些在CoffeeScript中的关键字在JavaScript中是不支持的，取而代之的是JavaScript中使用符号来表示布尔表达式。相应的标点符号的意义和使用单词代替在CoffeeScript中是完全相同的。



CoffeeScript允许所有这些JavaScript的标点符号，但在编写CoffeeScript的时候通常使用的英语单词来提高代码的可读性。

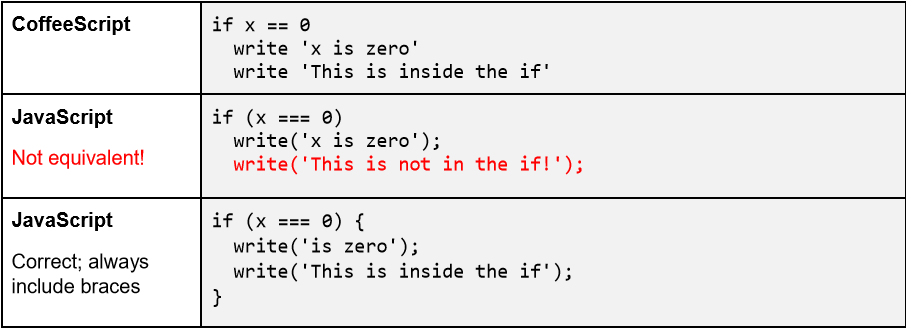
缩进和括号

JavaScript使用大括号表示嵌套。而缩进是允许的，但是缩进对于JavaScript解释器没有任何意义。它主要使用大括号来确保代码符合缩进。



一些初学者，一旦发现JavaScript对缩进不敏感，就会写出完全没有缩进JavaScript代码。这是一个糟糕的想法，它会导致代码混乱，就像上面的例子。好的代码不仅是功能性的，而且是可读的。

如果大括号省略在JavaScript中，控制流语句像if, while, 或者只适用于条件语句后的一行代码的情况。在下面的JavaScript的红线将打印“这不是在if中！”不管x的值为多少。JavaScript中正确的书写等同于的方法是在最后一列写上大括号。



千万不要在JavaScript中省略大括号中！JavaScript解释器不会注意缩进，所以省略大括号是是会发生错误的，如上图。

帮助学生理解JavaScript中的良好编码风格是很重要的，需要考虑使用和大括号一致的缩进。程序应该用大括号后的条件和循环和缩进必须匹配括号使其他程序员可以阅读并理解。缩进的数量应匹配嵌套括号的数量。

理解循环中的“Three-Clause”