<<JavaScript编程精解>>

图书基本信息

书名：<<JavaScript编程精解>>

13位ISBN编号：9787111396659

10位ISBN编号：7111396650

出版时间：2012-10-1

出版时间：机械工业出版社华章公司

作者：Marijn Haverbeke

页数：162

译者：徐涛

前言

前言20世纪70年代，业界首次推出个人计算机时，大多数计算机都内置一种简单的编程语言—通常是BASIC的变体，人与计算机之间的互动需要通过这种语言实现。
这意味着，对天生喜欢钻研技术的人来说，从单纯使用计算机到编程的过渡非常容易。
现在的计算机相比20世纪70年代的功能更加强大，价格也更加便宜，软件接口呈现的是使用鼠标操作的灵活图形界面，而不是语言界面。
这使计算机更容易使用，总的来说，这是一个巨大的进步。
然而，这也在计算机用户与编程世界之间制造了一个障碍—业余爱好者必须积极寻找自己的编程环境，而不是一打开电脑就呈现的环境。
实质上，计算机系统仍然被各种编程语言控制。
大多数的编程语言都比早期个人计算机中的BASIC语言更加先进。
例如，本书的主题—JavaScript语言，就存在于每一款主流Web浏览器中。
关于编程不愤不启，不悱不发。
举一隅不以三隅反，则不复也。
——孔子本书除了介绍JavaScript外，也致力于介绍编程的基本原理。
事实上，这种编程还是比较难的。
编程的基本规则通常都简单明了，但计算机程序构建在这些基本规则之上后，会变得很复杂，产生了其自身的规则和复杂性。
正因为如此，编程并不是那么简单或可预测的。
正如计算机科学的鼻祖高德纳（Donald Knuth）所说，编程是一门艺术，而不是一门科学。
要想从本书里获取最大收获，不能仅仅依靠被动阅读。
一定要集中注意力去理解示例代码，只有确定自己真正理解了前面的内容，才能继续往下阅读。
程序员对其创造的宇宙负全部责任，因为他们是创造者。
以计算机程序的形式，可创造出无限复杂的宇宙。
——Joseph Weizenbaum，《Computer Power and Human Reason》一个程序包含很多含义。
它是程序员敲出的一串字符，是计算机运行的指向力，是计算机内存中的数据，还控制同一个内存上的执行动作。
仅使用熟悉的类推法比较程序与对象往往还不够，因为从表面上看适合该操作的是机器。
机械表的齿轮巧妙地啮合在一起，如果表的制造者技术很棒，它就能够连续多年准确地显示时间。
计算机程序的元素也以类似的方式组合在一起，如果程序员知道自己在做什么，那么这个程序就能够正常运行而不会崩溃。
计算机作为这些无形机器的载体而存在。
计算机本身只会做简单直接的工作。
它们之所以如此有用，是因为它们能够以惊人的速度完成这些工作。
程序可以巧妙地把许多简单动作结合起来，去完成非常复杂的工作。
对有些人来说，编写计算机程序是一种很有趣的游戏。
程序是思想的构筑，它零成本、零重量，在我们的敲打中不断发展。
如果我们不细心，它的规模和复杂性将失去控制，甚至创造者也会感到混乱。
这就是编程的主要问题：控制好程序。
程序在工作时是很不可思议的，编程的艺术就是控制复杂性的技巧，好的程序其复杂性也会降低。
如今，很多程序员认为只要在程序中使用少量易于理解的技术，就可以最有效地降低复杂性。
他们制定了严格的编程规则（最佳实践）及书写格式，那些破坏规则的人被称为“差劲”的程序员。
丰富多彩的编程世界里包含了太多的复杂性！
让我们努力将程序变得简单和可预测，并为所有奇妙和优美的程序制定禁忌规则。
编程技术的前景是广阔的，其多样性使人着迷，它的世界仍有很多未被探索的部分。
编程过程中有很多陷阱和圈套，缺乏经验的程序员会犯各类糟糕的错误，告诫我们需要谨慎，并保持

头脑清醒。
学习编程时总是需要探索新的挑战、新的领域，拒绝不断探索的程序员必定会停滞不前、忘记编程的快乐、并失去编程的意志（或成为管理人员）。
语言为何很重要在计算机诞生初期并没有编程语言。
程序看起来就像这样：这是一个从1加到10并输出结果（1 + 2 +⋯+ 10 = 55）的程序。
它可以在一个非常简单、理想化的计算机上运行。
为早期的计算机编制程序时，必须在正确的位置设置一排排的开关或者在纸带上打上一系列有规律的孔点，这样才能将程序传递给计算机。
可以想象这个过程有多么繁琐和易出错。
即使编写简单的程序也需要使用很多脑力和规则，编写复杂的程序更是不可想象。
当然，手动输入这些二进制位（即以上这些1和0的统称）的神秘组合，让程序员感觉自己像巫师一样拥有强大的魔力，而且还能够获得工作满足感，因此这点还是很值得的。
程序的每一行都包含一条单独的指令。
可以用语言这样描述：1）将数字0保存在第0个存储单元；2）将数字1保存在第1个存储单元；3）将第1个存储单元的值保存在第2个存储单元；4）将第2个存储单元中的值减去数字11；5）如果第2个存储单元中的值是数字0，则继续执行指令9；6）将第1个存储单元的值添加至第0个存储单元；7）将数字1添加至第1个存储单元；8）继续执行指令3；9）输出第0个存储单元的值。
虽然这比二进制位易读，但仍然令人不快。
用名称代替指令和存储单元的数字或许更有帮助。
在这里不难看出程序是如何运行的。
前两行代码为两个存储单元赋予初始值：total用于创建计算的结果，count记录当前看到的数字。
使用compare的行可能是最令人费解的地方。
该程序的目的是判断count是否等于11，从而确定能否停止运行。
由于该机器相当原始，它只能测试一个数字是否为零，并在此基础上做出判断（跳转），因此它使用标记compare的存储单元来计算count的值，即11，并在该值的基础上做出判断。
后面两行代码将count的值添加到结果total上，当程序判断count不是11时，为count加1。
下面是用JavaScript编写的有同样效果的程序。
这段程序有了更多的改进。
最重要的是，不再需要指定程序来回转换的方式，while这个神奇的单词会帮助解决这个问题。
只要满足给定的条件：count

<<JavaScript编程精解>>

内容概要

如果你只想阅读一本关于JavaScript的图书，那么本书应该是你的首选。
本书由世界级JavaScript程序员撰写，JavaScript之父和多位JavaScript专家鼎力推荐。
本书适合作为系统学习JavaScript的参考书，它在写作思路上几乎与现有的所有同类书都不同，打破常规，将编程原理与运用规则完美地结合在一起，而且将所有知识点与一个又一个经典的编程故事融合在一起，读者可以在轻松的游戏式开发中学会JavaScript程序设计，趣味性十足，可操作性极强。

全书一共12章：第1~3章介绍了JavaScript的基本语法，旨在帮助读者编写出正确的JavaScript程序，包含数字、算术、字符串、变量、程序结构、控制流程、类型、函数、对象和数组等最基础和最核心的内容；第4~7章讲解了JavaScript编程中的高级技术，目的是帮助读者编写更复杂的JavaScript程序，主要涉及错误处理、函数式编程、面向对象编程、模块化等重要内容；第8~12章则将重心转移到JavaScript环境中可用的工具上，分别详细讲解了正则表达式、与Web编程相关的知识、文档对象模型、浏览器事件和HTTP请求等。

<<JavaScript编程精解>>

作者简介

Marijn Haverbeke　世界级JavaScript程序员，通晓多种编程语言，在Web开发方面积累了丰富的经验，在JavaScript领域颇有影响力。
如今，他专注于动态语言下的数据库系统的研究和Web API的设计。
此外，他创建并维护着多个流行的开源项目。

<<JavaScript编程精解>>

书籍目录

章节摘录

版权页： 插图： 应用于模拟环境中时，这种新动物（Clever Lichen Eater）比之前的简单同类动物（Lichen Eater）活得要长久。
如果给定一个足够大的生活空间，部分环境食物充足，部分环境食物不足，而不是每个地方都一样，那么这个生态系统似乎可以保持稳定。
6.4原型继承 整个生态圈应该为我们在实际程序中如何使用对象提供一些启发。
本章剩余部分将针对Java Script里的继承和继承相关的问题做更深入的讲解。
首先，来复习一些理论。
我们经常听到一些学习面向对象编程的学生讨论继承的正确和错误使用，这些谈话通常很冗长也很精细。
重点要记住：继承归根结底只是一种使程序员可以偷懒的技巧（这里使用“偷懒”是最贴切的说法），可以编写更少的代码。
因此，继承的使用是否正确的问题可以归结为代码是否正常工作以及如何避免不必要重复工作的问题。
尽管如此，这些学生采用的原则为我们思考继承相关的内容提供了很好的途径。
继承是创建新类型对象——子类型（subtype），子类型基于父类型（supertype）。
子类型拥有父类型所有的属性和方法（从父类型继承得到），然后修改其中的部分内容或者添加新的内容。
继承最好在子类型模型可以被视为父类型对象的时候使用。
因此，钢琴类型（Piano）可以是乐器类型（Instrument）的子类型，因为钢琴是一种乐器。
钢琴有一组钢琴键，我们可能想让钢琴作为Array的子类型，但钢琴不是数组，像数组一样实现它一定会导致各种各样的愚蠢错误。
例如，钢琴也有踏板，为什么piano[0]给出是第一个键，而不是第一个踏板呢？
实际情况是，钢琴有钢琴键也有踏板，因此最好给piano对象一个属性keys（钢琴键），再给一个属性pedals（踏板），两者都拥有数组。
也可以让一个子类型作为另外一个子类型的父类型。
有些问题最好的解决方法就是构建一个复杂的类型家族树。
必须注意不要有太多继承。
继承有自己的组合代码方式，过度使用会使代码之间的衔接混乱，并且很难修改。
6.4.1类型定义工具 使用new关键字和构造函数的prototype属性都是定义类型的特定方式，这些是我们到目前为止一直使用的。
对于简单的对象（比如生态圈生物），这种方式是非常有效的。
可是，当程序过度使用继承时，这种创建对象的方法很快就显得笨拙了。

媒体关注与评论

编程原理与运用规则的简练、完美融合。
我喜欢游戏式的程序开发教程。
本书再次点燃了我学习编程的热情。
对了，是JavaScript！
——BrendanEich，JavaScript之父因为这本书，我成为了更棒的架构师、作家、咨询师和开发者。
——AngusCroll，Twitter开发者如果你决定只买一本有关JavaScript的书，那么就应是MarijnHaverbeke的这本书。
——JoeydeVilla，GlobalNerdy本书不仅是学习JavaScript最棒的教材之一，也是通过学习JavaScript进而学习现代编程的优秀图书。
当有人问我如何学好JavaScript时，我会推荐这本书。
——ChrisWilliams，美国JSConf组织者我读过的最棒的JavaScript书籍之一。
——ReyBango，微软Client-Web社区项目经理和jQuery团队成员这本书不仅是一本非常不错的JavaScript指导书，而且是一本很棒的编程指导书。
——BenNadel，Epicenter咨询公司首席软件工程师真是本好书。
——DesignShack这本书对编程基本原理的详述以及对栈和环境等概念的解释非常到位。
注重细节使本书从其他的JavaScript书中脱颖而出。
——Designorati学习JavaScript的好书。
——CraigBuckler，OptimalWorksWebDesign公司

编辑推荐

《JavaScript编程精解》编辑推荐：世界级JavaScript程序员力作，JavaScript之父Brendan Eich高度评价并强力推荐。
JavaScript编程原理与运用规则完美融合，读者将在游戏式开发中学会JavaScript程序设计，是系统学习JavaScript程序设计的首选之作。

名人推荐

编程原理与运用规则的简练、完美融合。
我喜欢游戏式的程序开发教程。
本书再次点燃了我学习编程的热情。
对了，是JavaScript！
——Brendan Eich，JavaScript之父因为这本书，我成为了更棒的架构师、作家、咨询师和开发者。
——Angus Croll，Twitter开发者如果你决定只买一本有关JavaScript的书，那么就应是Marijn Haverbeke的这本书。
——Joey deVilla，Global Nerdy本书不仅是学习JavaScript最棒的教材之一，也是通过学习JavaScript进而学习现代编程的优秀图书。
当有人问我如何学好JavaScript时，我会推荐这本书。
——Chris Williams，美国JSConf组织者我读过的最棒的JavaScript书籍之一。
——Rey Bango，微软Client-Web社区项目经理和jQuery团队成员这本书不仅是一本非常不错的JavaScript指导书，而且是一本很棒的编程指导书。
——Ben Nadel，Epicenter咨询公司首席软件工程师真是本好书。
——Design Shack这本书对编程基本原理的详述以及对栈和环境等概念的解释非常到位。
注重细节使本书从其他的JavaScript书中脱颖而出。
——Designorati学习JavaScript的好书。
——Craig Buckler，OptimalWorks Web Design公司

<<JavaScript编程精解>>

版权说明