第一图书网, tushu007.com

<<JavaScript王者归来>>

图书基本信息

书名：<<JavaScript王者归来>>

13位ISBN编号：9787302173083

10位ISBN编号：7302173087

出版时间：2008

出版时间：清华大学出版社

作者：月影

页数：712

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

第一图书网, tushu007.com

前言

你好，很高兴你能翻开这本书——《JavaScript王者归来》。
书名中的“王者”，并不是指我这个作者，而是指你们——偶然地停下来，饶有兴趣地翻阅这本书的每一位读者。
正是你们对JavaScript和Web开发的浓厚兴趣和热情，使得Web领域发生了和正在发生着翻天覆地的变化，也使得我有信心和毅力去完成这样一本厚厚的书。
所以，这本书，是献给你们的，现在的和将来的JavaScript王者们！
这是一本什么样的书呢？
作为作者，我很希望它能成为一本带着神奇力量的魔法书，能够将你对技术的兴趣，转化为神奇的魔力，帮助你在Web应用中写出不可思议的奇迹般的漂亮代码。
当然，也许这只是我的一个美好的愿望，但是如果你是带着兴趣和热情来读这本书，并且愿意和我一起探寻JavaScript王国的奥秘，那么，这本书中一定有着能让你着迷的东西。
JavaScript是一位美丽而又难以捉摸的丛林女神，她干净利落而又变化多端，她的美丽能够带给你喜悦和幸福感，她的善变，却又让你深深地陷入困惑之中。
有时候，你本能地想逃避她，内心里又离不开她。
你觉得她单纯，可是你看不透她，你觉得她难以捉摸，可是她却偏偏又有着简单乖巧的一面。
这样的一种语言，是充满魅力的魔法精灵，难道你就不想看清她的真面目，读懂她，让她帮助你创造出激动人心的神迹吗？
来吧，翻开这本书，你的愿望能够实现。
如果你是一位刚刚接触到丰富多彩的Web世界，希望自己也能够制作出那些美丽多彩的页面的人，那么请相信我，这本书是为你准备的，通过阅读它，能够使你具备创建和驾驭属于你自己的Web国度的能力。
如果你是一位偶然地在浏览器页面代码中发现一小段包含在标记之间的小脚本，并且希望弄明白它的人，那么请相信我，这本书是为你准备的，如果你有耐心读完它，你将能够自由地让你的意志在标记之间飞扬。
如果你是一位迷失在无数脚本和特效之间的旅行者，那么请相信我，这本书是为你准备的。
来和我一起在Web的湖泊中畅游吧，我必将带你离开迷宫和陷阱，让你看到脚下的这个王国是多么的美好，你将成为一位英明的王，而不再是无助的迷路人。
如果你是一位饱受脚本困扰的程序员，那么请相信我，这本书是为你准备的，请和我一起打开缠绕在你心头的枷锁，你会发现你手中握着的是一把神奇的利刃，这把神兵将不再割伤你自己，而是被你支配，成为在Web世界里助你开拓疆土的神器。
如果你是一位在脚本泥潭中挣扎的项目经理，那么请相信我，这本书也是为你准备的，我发誓我有能力带你离开我自己曾经挣扎过的地方，当你离开了陷阱，你将发现以前你无暇顾及的世界，原来是那么的美好。
所有翻开了这本书并喜爱着JavaScript的读者们，愿脚本神力与你们同在！
关于版权　　本书中的部分内容来自于网络上公开的文章，所有有出处的文字都尽量标注出原始的出处，包括原文作者、首发网址和译者。
引用为例子的代码在尽可能的情况下得到作者本人的同意。
如果对这部分内容有任何疑义，请及时与作者联系。
本书中的部分文字参考或者直接引用了《JavaScript权威指南》第四版，作者David Flanagan，译者张铭泽等，O’REILLY授权机械工业出版社2006年9月出版。
凡是明确引用此书内容和参考此书内容的部分，笔者也在书中尽量注明（正文中统一用【1】来表示），如果因为引用原文内容而产生的任何问题，请及时与作者联系。
书中相关章节的表格（表XX.AA）大多数来源于《JavaScript权威指南》和互联网，在正文里不再做一一说明，如果对此有任何疑义，请及时与作者联系。
致谢　　虽然这几年来我一直致力于推广JavaScript技术和Web标准，但是如果离开了时时刻刻支

持和鼓励着我的朋友们，这样一本涉及到JavaScript方方面面的书，以我个人微不足道的力量是不可能完成的。
这本书的面市凝聚着无数关心我的朋友们的心血，他们中的一些人是我的同事，另一些人是我在无忧脚本（www.51js.com）结识的伙伴，还有许许多多通过网络联系的未曾谋面的朋友。
他们的每一分鼓励、每一个思想、和每一点意见都是我创作的灵感和力量的源泉，没有他们，就永远也不会有这样一本出自我手的书，在此我要向他们表示由衷的感谢。
　　我在金蝶工作的同事在我写这本书的时候，不但为我提出了各种建议，而且毫无怨言地分担了我的工作，体贴地为我留出足够的创作时间。
在这里我要感谢jimi、张锦、小陆、阿日、建新、大琴、老丁、谢汀以及其他的和我共同奋斗的金蝶MOP团队的兄弟姐妹，谢谢你们的努力工作，谢谢你们对我的默默支持。
　　我创作这本书的大量灵感来源于我和无忧脚本以及CSDN论坛上的朋友们的交流。
我们总是一起探讨关于脚本的深入话题，有时候达成一致意见，也有时候产生分歧甚至激烈的争执。灵感的火花总是在思维碰撞中产生的，和你们的讨论让我学到了很多东西，也直接地决定了这本书的内容组织和观点形成。
在此，我要感谢所有在无忧脚本和CSDN结识的JS高手们，感谢幻宇、宝玉、梅花雪、周爱民、万常华、海浪、梅雪香、dron、stone、刘杰、biyuan、泣红亭、winter和asfman以及其他所有和我一起讨论共同进步的JSers们，谢谢你们。
　　我在04年从学校里毕业的时候，还对JavaScript一无所知，我自学JavaScript使用的第一本教材就是David Flanagan的《JavaScript权威指南》，如果没有这本出色的教材，我可能永远也不会对JavaScript有今天这种程度的理解。
本书的基础部分的许多基本概念、观点和结论，也和《JavaScript权威指南》保持一致。
在此感谢David Flanagan、Brendan Eich和《JavaScript权威指南》，没有你们的帮助，也许我现在还是一位JavaScript的门外汉。
　　这是我写的第一本技术类的书，在此之前，我完全没有创作教程的经验，是我的编辑陈冰鼓励我创作这本书。
而且在我撰写这本书的一年里，他耐心细致地阅读了我每一章节的稿件，提出了许多意见和建议，并一次又一次地督促我改进稿件，直至最终完成。
如果没有陈冰的耐心指导和帮助，我永远也写不出这样一本书。
　　Web技术的飞速发展使得JavaScript这门脚本语言日渐被人们所重视，Web标准化工作的推进，也使得JavaScript变得越来越完善和优美，也变得越来越神奇。
在这里我要感谢所有为Web技术发展做出贡献的技术人员，没有你们的努力，就没有JavaScript生存和发展的土壤。
感谢W3C和ECMA组织孜孜不倦地进行的标准化工作，如果没有这些努力，JavaScript也许到今天仍然只是Web舞台一个不起眼的角落里的小配角。
　　最后，我要感谢我亲爱的父母，没有你们对我的爱，今天的我就不可能拥有我喜欢的事业，写出我喜欢的书，我也要将我的这本书献给你们。
我永远爱着你们。
　　吴亮　　2008年1月10日

第一图书网, tushu007.com

<<JavaScript王者归来>>

内容概要

《JavaScript王者归来》不仅是一本传播知识的书，更是一本求道的书。
《JavaScript王者归来》分为五个部分循序渐进地与读者讨论了JavaScript的方方面面，从简单的语言基础到丰富的实际应用再到深入剖析语言本质的高级话题，字里行间包含着作者多年工作中对JavaScript实践乃至程序设计思想的深入思考和总结。
《JavaScript王者归来》揭开了JavaScript的面纱，绕过误解和虚幻的表象，引领你探索程序王国的奥妙。
它既是一本为初学者准备的入门级教程，又是一本探寻程序设计思想本源的“魔法典籍”，也是一本Web开发工程师们需要的案头参考书。
《JavaScript王者归来》是你进入脚本王国的一把钥匙，引导你领略脚本魔法的神奇魅力。
它还是一本着眼于未来改变互联网的启蒙读物，在它的引领下，你将在互联网的世界里获得你所希望得到的知识、智慧、成就和快乐。

第一图书网, tushu007.com

作者简介

吴亮，（网名月影），早年曾在微软亚洲研究院做过访问学生、在金蝶软件有限公司先后担任过核心开发工程师、设计师和项目经理，现任百度电子商务事业部Web开发项目经理。
多年来致力于JavaScript技术和Web标准的推广，活跃于国内极有影响力的JavaScript专业网站——无忧脚本（www.51js.com），并任JavaScript版的版主。
平时热爱文学、写作和围棋。

第一图书网, tushu007.com

<<JavaScript王者归来>>

书籍目录

第一部分　概论第1章　从零开始1.1　为什么选择JavaScript?1.1.1　用户的偏好--B/S模式1.1.2　在什么情况下用JavaScript1.1.3　对JavaScript的一些误解1.1.3.1　JavaScript和Java的关系1.1.3.2　披着C外衣的Lisp1.1.3.3　关于JavaScript的思维定势1.1.3.4　JavaScript是为业余爱好者设计的？
1.1.3.5　JavaScript是面向对象的吗1.1.3.6　其他误解1.1.4　警惕！
脚本诱惑1.1.5　隐藏在简单表象下的复杂度1.1.6　令人迷惑的选择--锦上添花还是雪中送炭1.1.7　回到问题上来1.2　JavaScript的应用范围1.2.1　客户端的JavaScript1.2.2　服务器端的JavaScript1.2.3　其他环境中的JavaScript1.3　JavaScript的版本1.3.1　浏览器中的JavaScript版本1.3.2　其他版本1.4　一些值得留意的特性1.4.1　小把戏--神奇的魔法代码1.4.2　为客户端服务--经典Hello World！
的另一种JavaScript实现1.4.3　数据交互--JavaScript的一项强大功能1.4.4　JavaScript表面上的禁忌及如何突破这些禁忌1.5　安全性和执行效率1.5.1　数据安全--永远的敏感话题1.5.2　实战！
攻击与防范1.5.3　不容马虎--时刻关注性能1.6　一个例子--JavaScript编写的计算器1.6.1　从需求分析开始--什么是计算器？
1.6.2　系统设计--如何实现计算器？
1.6.3　系统实现--计算器的最终实现1.6.4　持续改进--迭代的软件开发过程1.7　学习和使用JavaScript的几点建议1.7.1　像程序员一样地思考--程序员的四个境界1.7.2　吝惜你的代码1.7.3　学会在环境中调试1.7.4　警惕那些小缺陷1.7.5　思考先于实践--不要轻易动手写代码1.7.6　时刻回头--圣贤也无法完全预知未来1.8　关于本书的其余部分第2章　浏览器中的JavaScript2.1　嵌入网页的可执行内容2.1.1　在什么地方装载JavaScript代码2.1.2　关于代码的Script标签2.1.3　我的代码什么时候被执行--不同执行期的JavaScript代码2.1.4　拿来主义--引入外部的JavaScript文件2.2　赏心悦目的特效2.2.1　生命在于运动--DHTML的效果2.2.2　换一种风格--CSS的力量2.2.3　用JavaScript操作DOM--一个可拖动窗口的例子2.3　使用JavaScript来与用户交互2.3.1　创建一个弹出式帮助和进度条2.3.2　填错了哦2.4　绕开脚本陷阱2.4.1　现实并不总是完美的2.4.2　不能完全相信你所见到的2.5　总结第3章　开发环境和调试方法3.1　我能用什么来编写脚本--适合编写JavaScript的文本编辑器3.2　来自浏览器的支持3.2.1　主流浏览器3.2.2　非主流浏览器3.3　集成开发环境3.3.1　什么是集成开发环境3.3.2　我需要集成开发环境吗3.3.3　适合JavaScript的集成开发环境3.4　调试工具--提升开发效率的利器3.4.1　什么是调试3.4.2　原始的调试方法--利用输出语句、“反射”机制和调试对象来进行调试3.4.3　适合JavaScript的调试工具3.5　定位代码和调用堆栈3.5.1　Step by Step--单步和断点3.5.2　监视内存3.5.3　追踪问题的源头--查看调用堆栈3.5.4　遇到麻烦了--为什么我跟踪不到代码3.6　浏览器捕获异常3.6.1　异常处理机制--一个try/catch/finally模式的例子3.6.2　异常的种类3.6.3　应该在什么时候“吃掉”异常3.7　总结第二部分　JavaScript核心第4章　语言结构4.1　JavaScript的基本文法4.1.1　字符集4.1.2　大小写敏感4.1.3　分隔符4.1.4　词、句子和段落4.1.5　分号4.1.6　标记4.1.7　注释4.1.8　保留字4.2　常量和变量4.2.1　常量和变量4.2.2　变量的标识符4.2.3　变量的类型4.2.4　变量的声明4.2.5　变量的作用域4.3　表达式和运算符4.3.1　表达式4.3.2　运算符概述4.3.3　算术运算符4.3.4　关系运算符4.3.5　逻辑运算符4.3.6　位运算符4.3.7　赋值运算符4.3.8　其他运算符4.3.8.1　条件运算符4.3.8.2　逗号运算符4.3.8.3　对象运算符4.3.8.4　类型运算符4.3.8.5　void运算符4.3.8.6　函数调用运算符4.4　控制语句4.4.1　表达式语句4.4.2　语句块4.4.3　条件语句4.4.4　循环语句4.4.5　跳转语句4.4.6　异常处理语句4.4.7　其他语句4.4.7.1　var语句4.4.7.2　function语句4.4.7.3　with语句4.4.7.4　空语句4.5　总结第5章　数据类型5.1　基本数据类型5.1.1　数值5.1.2　字符串--一个字符串相关操作的例子5.1.3　布尔型5.2　数组和对象5.2.1　数组5.2.2　对象--一个构造函数的例子5.3　函数类型--一个函数和闭包的例子5.4　神奇的null和undefined5.4.1　null5.4.2　undefined--独一无二的类型5.5　正则表达式5.5.1　正则表达式常量5.5.2　正则表达式对象5.6　值类型和引用类型5.6.1　什么是值和值的引用5.6.2　使用值和使用引用5.6.3　值与引用的相互转换：装箱和拆箱5.7　类型识别与类型转换5.7.1　运行时类型识别--两个运行的类型识别的例子5.7.2　类型的自动转换及其例子5.7.3　强制类型转换及其例子5.7.4　高级用法--一个自定义类型转换的例子5.8　警惕数值陷阱5.8.1　困惑--浮点数的精度问题5.8.2　误差的修正及其例子5.9　总结第6章　函数6.1　函数定义和函数调用6.1.1　函数的定义6.1.1.1　声明式函数定义与函数表达式及其例

第一图书网, tushu007.com

子6.1.1.2 JavaScript函数的奥妙——魔法代码6.1.2 函数的调用6.2 函数的参数6.2.1 形参与实参6.2.2 Arguments对象6.2.2.1 一个使用Arguments对象检测形参的例子6.2.2.2 一个使用Arguments对象接收任意个数参数的例子6.2.2.3 一个使用Arguments对象模拟函数重载的例子6.2.3 参数类型匹配--一个利用arguments实现函数重载机制的例子6.3 函数的调用者和所有者6.3.1 函数的调用者6.3.2 函数的所有者--一个为函数指定所有者的例子6.3.3 动态调用--外来的所有者6.4 函数常量和闭包6.4.1 匿名的函数6.4.2 函数引用6.4.3 函数参数和函数返回值及其例子6.4.4 高级用法--闭包作为局部域与延迟求值6.5 高级抽象--Function类型和函数模版6.5.1 动态创建函数--一个利用Function实现Lambda算子的例子6.5.2 模式--函数工厂及其实例6.6 总结第7章 对象7.1 什么是对象7.2 对象的属性和方法7.2.1 对象的内置属性7.2.2 为对象添加和删除属性7.2.3 反射机制--枚举对象属性7.3 对象的构造7.3.1 构造函数--一个双精度浮点数封装类的例子7.3.2 缺省构造和拷贝构造7.3.3 对象常量7.4 对象的销毁和存储单元的回收7.5 JavaScript的内置对象7.5.1 Math对象7.5.2 Date对象--创建一个简单的日历7.5.3 Error对象7.5.4 其他内置对象7.5.5 特殊的对象--全局对象与调用对象7.6 总结第8章 集合8.1 数组和数组元素8.1.1 数组的构造8.1.2 数组常量8.1.3 数组元素8.2 数组对象和方法8.2.1 查找元素8.2.2 添加和删除元素8.2.3 集合操作及其范例8.2.3.1 join()方法8.2.3.2 reverse()方法8.2.3.3 sort()方法8.2.3.4 concat()方法8.2.3.5 slice()方法8.2.3.6 splice()方法8.2.3.7 toSgring()方法和toLocaleString()方法8.3 哈希表8.3.1 什么是哈希表8.3.2 哈希表的构造8.3.3 实现一个简单的HashTable类型8.4 高级用法--集合操作和闭包8.5 总结第9章 字符串9.1 字符串的构造9.1.1 字符串常量9.1.2 转义序列9.1.3 字符串构造函数9.2 字符串的使用9.2.1 比较字符串9.2.2 抽取和检索子串9.2.3 连接拆分字符串9.2.4 字符串的模式匹配--一个字符串格式校验的例子9.2.5 其他方法9.3 字符串与字符数组9.4 字符串与文本处理--JavaScript棋谱阅读器（一）9.4.1 需求分析--什么是棋谱和棋谱阅读器9.4.2 系统设计--棋谱和棋盘数据的字符串描述9.4.3 系统实现--解析和处理棋谱9.4.4 完整的棋谱阅读器9.5 总结第10章 正则表达式10.1 什么是正则表达式10.1.1 正则表达式的概念10.1.2 JavaScript中的正则表达式10.2 正则表达式的规则10.2.1 直接量字符10.2.2 字符类和布尔操作10.2.3 重复10.2.4 选择、分组和引用10.2.5 指定匹配的位置10.2.6 标志——高级模式匹配的规则10.3 模式匹配10.3.1 用于模式匹配的String方法及其例子10.3.2 用于模式匹配的RegExp方法10.3.2.1 一个使用exec()方法从身份证号码获取生日的例子10.3.2.2 一个使用test()方法遍历字符串的例子10.4 关于正则表达式包装对象10.4.1 RegExp对象——利用正则表达式实现全文检索10.4.2 RegExp的实例属性10.5 强大的正则表达式10.5.1 分析正则表达式的局部10.5.2 一个例子--强大的在线编辑器10.5.3 构造新的文法--一个在JSVM中实现JSVM2解析器的例子10.6 高级用法10.7 用正则表达式处理文本10.7.1 创建一个计价公式编辑器10.7.1.1 需求分析--什么是计价公式编辑器10.7.1.2 系统实现--计价公式编辑器的实现10.7.2 创建一个同步滚动歌词播放器10.7.2.1 需求分析--什么是同步滚动歌词播放器10.7.2.2 系统设计与实现--处理LRC歌词10.8 总结第三部分 浏览器与DOM第11章 浏览器对象11.1 Window对象--最基本的浏览器对象11.1.1 Window对象概览11.1.2 Window对象的生命周期11.1.3 Window对象的属性和方法11.1.4 一个多窗口应用的例子11.2 Document对象--浏览器窗口文档内容的代表11.2.1 Document对象概览11.2.2 动态生成的文档11.2.3 Document对象的基本信息11.2.4 Document对象的外观属性11.2.5 Document子对象接口11.2.5.1 一个遍历Anchors对象的例子11.2.5.2 一个颠倒图片的例子11.3 对话框和状态栏11.3.1 创建一个简单对话框11.3.2 其他类型的对话框11.3.2.1 模拟对话框--创建一个窗口对话框及一个对话框阻塞进行的例子11.3.2.2 showModalDialog和showModelessDialog--非W3C或ECMAScrip标准11.3.3 状态栏11.4 框架--上层的Window对象11.4.1 多框架应用11.4.2 框架之间的关系11.4.3 框架的命名11.4.4 子框架中的JavaScript11.4.5 框架的应用--多页签显示11.4.5.1 什么是页签11.4.5.2 页签的实现--创建一个包含页签的页面11.5 表单和表单对象11.5.1 Form对象及其范例11.5.2 定义表单元素11.5.3 客户端表单校验及其例子11.5.4 创建一款通用的客户端表单校验组件11.6 其他内置对象11.6.1 Navigator对象--浏览器总体信息的代表11.6.2 Screen对象--提供显示器分辨率和可用颜色数量信息11.6.3 Location对象--当前窗口中显示文档的URL的代表11.6.4 History对象--一个有趣的对象11.7 总结第12章 文档对象模型12.1 什么是DOM12.1.1 把文档表示为树12.1.2 树的节点12.1.3 DOM对象的通用属性和方法12.1.4 HTML结构和DOM对象的关系--用JavaScript通过DOM来操作HTML文档12.2 DOM与浏览器实

现12.2.1　关于DOM HTML API12.2.2　DOM的级别和特性12.2.3　DOM的一致性12.2.4　差异性--浏览器的DOM方言12.3 一组“盒子”--DOM元素12.3.1　嵌套的“盒子”12.3.2“盒子”和“盒子”内容的分类12.4 创建和删除节点12.4.1　构造全新的节点12.4.2　平面展开--通过文档元素直接创建12.4.3　回收空间--删除不用的节点12.5 访问和操纵DOM节点12.5.1　打开每一个盒子--遍历节点12.5.2　弄清层级关系--父子与兄弟12.5.3　如何搜索特定节点12.5.4　克隆节点--一个使用cloneNode()复制表格的例子12.5.5　移动节点及其范例12.5.6　关于添加新行和排序的小技巧12.6 读写数据--添加、修改和删除属性12.7 外观与行为12.7.1　DOM样式属性12.7.2　控制DOM元素的显示与隐藏12.7.3　改变颜色和大小--一个简单有趣的例子12.7.4　改变位置--创建一个绕圆圈旋转的文字12.7.5　编辑控制及其范例12.7.6　改变样式及其范例12.7.7　改变行为12.8 XML DOM12.8.1　什么是XML DOM12.8.2　如何使用XML DOM--一个利用XML实现多级关联下拉选择框的例子12.9 总结第13章　事件处理13.1 什么是事件13.1.1　消息与事件响应13.1.2　浏览器的事件驱动机制13.2 基本事件处理13.2.1　事件和事件类型13.2.2　事件的绑定13.2.3　直接调用事件处理函数13.2.4　事件处理函数的返回值13.2.5　带参数的事件响应及其例子13.2.6“this”关键字13.3 标准事件模型13.3.1　起泡和捕捉--浏览器的事件传播13.3.2　事件处理函数的注册13.3.3　把对象注册为事件处理程序13.3.4　事件模块和事件类型13.3.5　关于Event接口13.3.5.1　Event接口的属性和方法13.3.5.2　UIEvent接口的属性13.3.5.3　MouseEvent接口的属性13.3.5.4　MutationEvent接口13.3.6　混合事件模型13.3.7　合成事件13.4 浏览器的事件处理模型实现13.4.1　Internet Explorer事件模型13.4.1.1　关于IE事件注册13.4.1.2　IE Event对象的属性13.4.1.3　IE中的事件起泡13.4.2　Netscape 4事件模型13.4.2.1　Netscape 4中的事件捕捉及其范例13.4.2.2　Netscape 4 Event对象的属性13.5 回调与用户自定义事件13.5.1　事件处理模式--一个实现简单事件处理模式的例子13.5.2　用户事件接口的定义13.5.3　事件代理和事件注册--一个实现标准事件接口的例子13.5.4　标准模式--事件分派和接收13.6 一个例子--增强数据表格13.6.1　什么是增强数据表格13.6.2　一个采用两重table嵌套方式固定表头的例子13.6.3　可变列宽的实现13.6.4　标记行--呈现有别于其他行的背景色13.6.5　小技巧--将代码添加到样式表13.7 总结第14章　级联样式表14.1 什么是级联样式表14.1.1　CSS样式和样式表14.1.2　CSS的标准化14.1.3　浏览器支持的CSS14.2 JavaScript与CSS14.2.1　CSS和DOM的关系14.2.2　CSS和IE的关系14.2.3　浏览器的CSS兼容性14.3 控制CSS改变页面风格14.3.1　实现结构与表现的分离及其范例14.3.2　使用JwaScript和CSS实现页面多种风格的实时替换14.4 总结第15章　数据存储的脚本化15.1 什么是cookie15.1.1　浏览器和客户端cookie15.1.2　cookie的属性15.2 cookie的客户端存取15.2.1　cookie的存储15.2.2　cookie的读取15.3 cookie的限制15.4 cookie示例--让问候更加温暖15.5 cookie对象的封装15.6 什么是userData15.6.1　浏览器和客户端userData15.6.2　userData的声明15.6.3　userData的属性和方法15.7 userData的客户端存取15.7.1　userData的存储和读取15.7.2　userData的安全性15.8 userData的限制15.9 userData与cookie的对比15.10 userData示例--一个利用userData实现客户端保存表单数据的例子15.11 总结第四部分　数据交互第16章　同步和异步16.1 什么是同步和异步16.2 超时设定和时间间隔16.3 定时器使用--侦听与拦截16.3.1　标准模式--监视器及其范例16.3.2　使用定时器时应当注意的问题16.4 一个例子--漂亮的Web时钟16.4.1　什么是Web时钟？
16.4.2　最简单的Web时钟16.4.3　Web时钟的设计16.4.4　完整的Web时钟源代码16.5 总结第17章　XML DOM和XML HTTP17.1 什么是XML DOM对象17.1.1　XML DOM简介17.1.2　浏览器支持的XML DOM接口17.1.2.1　XML DOM标准接口17.1.2.2　IE的XML DOM组件17.1.2.3　操作XML文档--一个使用MSXML操作XML文档的例子17.2 XML DOM的版本兼容性--XML DOM的跨浏览器应用17.3 XML DOM的错误处理17.3.1　处理错误信息的ParseError对象17.3.2　包含错误信息的文档17.4 XML DOM操作XML文档17.4.1　访问节点17.4.2　创建新节点17.4.3　移动和修改节点及其范例17.4.4　读写节点属性和读写数据17.4.5　保存文档17.5 一个例子--JavaScript棋谱阅读器（二）17.5.1　用XML描述棋谱17.5.2　将XML棋谱转换为SGF棋谱17.6 什么是XML HTTP对象17.6.1　XML HTTP对象简介17.6.2　浏览器支持的XML HTTP对象17.7 通过XML HTTP发送请求17.7.1　建立连接17.7.2　发送请求17.8 读取和设定HTTP头17.8.1　什么是HTTP头17.8.2　读取和设定HTTP头17.9 服务器应答17.9.1　同步和异步应答及其例子17.9.2　包含应答文本内容的ResponseText和ResponseXML17.10 总结第18章　Ajax简介18.1 什么是Ajax18.1.1　Ajax并不神秘18.1.2　Ajax的应用场景18.1.3　Ajax的竞争对手--其他替代技术18.2 Ajax初探--我的第一个Ajax程序18.2.1　从常规应用开始--一个简单的实时聊天室18.2.2　讨厌的页面刷

新18.2.3　无刷新解决方案--改进的聊天室18.2.4　异步工作--迟滞感的解决方案18.3 Ajax原理剖析18.3.1　XML HTTP实时通信及一个简单封装了AjaxProxy对象的例子18.3.2　数据动态显示--Ajax改善交互体验的重要特点18.3.3　发挥XML的能力18.3.4　用JavaScript绑定一切18.3.5　应用背后的标准18.4 Ajax范例--实时聊天工具18.4.1　什么是实时聊天工具18.4.2　需求分析--实时聊天功能的实现要点18.4.3　系统实现--实时聊天功能的实现18.4.4　小结18.5 总结第19章　标准和兼容性19.1 标准化组织19.1.1　W3C和DOM标准19.1.2　ECMA和JavaScript标准19.1.3　互联网标准19.2 平台和浏览器的兼容性19.2.1　最小公分母法19.2.2　防御性编码19.2.3　客户端探测器19.2.4　特性检测19.2.5　实现标准19.2.6　适度停止运行19.3 语言版本的兼容性19.3.1　language属性19.3.2　版本测试19.4 如何实现跨浏览器应用19.4.1　取舍--划定支持范围19.4.2　基础模块设计--独立兼容性检测19.4.3　划分运行级别19.4.4　给出正确的信息--不要让你的代码保持沉默19.4.5　充分的应用测试--“兼容性魔鬼”总会趁你不注意时“踢你的狗”19.4.6　靠近标准和就近原则　19.5 展望未来19.6 总结第20章　信息安全20.1 用户的隐私信息20.2 禁止和受限制的操作20.2.1　受限制的属性20.2.2　受限制的操作20.2.3　脚本安全级别20.2.4　脚本调试20.3 警惕幕后的攻击者20.3.1　攻击的手段20.3.2　隐匿的数据流20.3.3　页面伪装20.3.4　发现蛛丝马迹20.3.5　防范的手段20.3.5.1　传输数据的加密20.3.5.2　对用户隐藏源代码20.4 同源策略20.4.1　什么是同源策略20.4.2　同源策略的利弊20.4.3　突破同源策略20.5 安全区和签名脚本20.5.1　可配置的安全策略方案20.5.2　Internet Explorer的安全区20.5.3　Netscape的签名脚本20.6 代码本身的安全--加密和混淆20.6.1　为什么要加密和混淆20.6.2　客户端的加密技术及其例子20.6.3　代码混淆原理20.6.4　JavaScript代码混淆工具--一个代码混淆算法的例子20.6.5　加密和混淆的结合使用20.7 总结第五部分　超越JavaScript第21章　面向对象21.1 什么面向对象21.1.1　类和对象21.1.2　公有和私有--属性的封装21.1.3　属性和方法的类型21.2 神奇的prototype21.2.1　什么是prototype21.2.2　prototype的使用技巧21.2.2.1　给原型对象添加属性21.2.2.2　带默认值的Point对象21.2.2.3　delete操作将对象属性恢复为默认值21.2.2.4　使用prototype巧设getter21.2.2.5　delete操作恢复原型属性的可见性21.2.2.6　使用prototype创建大量副本21.2.2.7　使用prototype定义静态方法21.2.3　prototype的实质及其范例21.2.4　prototype的价值与局限性21.3 继承与多态21.3.1　什么是继承21.3.2　实现继承的方法21.3.2.1　构造继承法及其例子21.3.2.2　原型继承法及其例子21.3.2.3　实例继承法及其例子21.3.2.4　拷贝继承法及其例子21.3.2.5　几种继承法的比较21.3.2.6　混合继承法及其例子21.3.3　单继承与多重继承21.3.4　接口及其实现21.3.5　多态及其实现21.4 构造与析构21.4.1　构造函数21.4.2　多重构造21.4.3　析构21.5 疑团！“this”迷宫21.5.1　无数个陷阱--令人困扰的“this”谜团21.5.1.1　this代词的运用21.5.1.2　this“陷阱”21.5.1.3　this代词的异步问题21.5.2　偷梁换柱--不好的使用习惯21.5.3　异步调用--谁动了我的“this”21.5.4　揭开真相--JavaScript的“this”本质21.5.5　困难不再--利用闭包修正“this”引用21.6 包装对象21.6.1　区分值和引用21.6.2　装箱与拆箱21.7 元类，类的模板21.7.1　什么是元类21.7.2　元类--构造类的类21.7.3　为什么要用元类21.7.4　类工厂21.7.4.1　什么是类工厂21.7.4.2　为什么要建立类工厂21.8 谁才是造物主21.8.1　万物适用的准则21.8.2　抽象的极致--一个抽象模式的例子21.8.3　返璞归真，同源架构21.9 总结第22章　闭包与函数式编程22.1 动态语言与闭包22.1.1　动态语言22.1.2　语法域和执行域22.1.3　JavaScript的闭包--一个体现闭包本质的例子22.2 闭包的特点与形式22.2.1　闭包的内在--自治的领域22.2.2　访问外部环境--一个用闭包改变外部环境的例子22.2.3　闭包和面向对象22.2.4　其他形式的闭包22.3 不适合使用闭包的场合22.4 函数式编程22.4.1　什么是函数式编程22.4.1.1　函数是第一型22.4.1.2　闭包与函数式编程22.4.1.3　科里化（Currying）--一个有趣的概念22.4.1.4　延迟求值和延续--一个Fibonacci无穷数列的例子22.4.2　函数式编程、公式化与数学模型--一个抛物线方程的例子22.4.3　函数式编程的优点22.4.3.1　单元测试方面的优点22.4.3.2　调试方面的优点22.4.3.3　并行方面的优点22.4.3.4　代码热部署方面的优点22.4.3.5　机器辅助的推理和优化22.4.4　函数式编程的缺点22.4.4.1　闭包的副作用22.4.4.2　递归的形式22.4.4.3　延迟求值的副作用22.5 闭包与面向对象22.5.1　私有域22.5.2　名字空间管理22.5.3　友元--一个非常有趣的概念22.6 Python风格的JavaScript代码22.6.1　最简约代码22.6.2　轻量级重用22.6.2.1　JSON22.6.2.2　Functional22.6.2.3　迭代函数--一个Array迭代函数的例子22.6.3　模块管理及其范例22.7 总结第23章　模块级管理23.1 模块化管理23.1.1　模块化--代码的重用23.1.2　JavaScript的模块管理23.2 开放封闭原则和面向接口23.2.1　开放封闭原则23.2.2　面向接口23.3 名字空间管理23.3.1　什么是名字空间23.3.2　为什么要用名字空间23.3.3　JavaScript的名字空间

管理23.4 调用依赖23.4.1　模块的依赖性23.4.2　模块依赖的管理23.5 用代码管理代码23.5.1　运行时环境的管理23.5.2　托管代码--一个简单的托管代码“容器”23.5.3　一个完整的代码管理容器23.6 总结第24章　动态构建24.1 让代码去写代码24.1.1　脚本的动态解析24.1.2　语法扩展--创造属于自己的语言24.2“发明”语法24.2.1　正则表达式和语法解析及例子24.2.2　一个简单的语法解析器实现24.2.2.1　什么是JavaScript 2.0语法24.2.2.2　JavaScript 2.0语法的部分关键特性实现24.3 实现自己的方言--LispScript24.3.1　从JavaScript到Lisp24.3.2　最初的工作--一般JavaScript代码24.3.3　公理，表达式24.3.4　函数式编程的七条基本公设24.3.4.1“引用”公设24.3.4.2“原子”公设24.3.4.3“等值”公设24.3.4.4“表头”公设24.3.4.5“余表”公设24.3.4.6“和表”公设24.3.4.7“条件”公设24.3.5　函数文法24.3.6　使用LispScript定义新函数24.3.7　一个惊喜--_eval24.3.8　其他的扩展24.3.9　小结24.3.10　运行环境和代码容器——看看“新发明”的LispScript的实际表现24.4 总结第25章　执行效率25.1 为什么要讨论执行效率25.1.1　来自客户的抱怨--JavaScript能有多慢25.1.2　代码慢下来是谁的错25.2 封闭的代价25.2.1　过度封装的性能问题25.2.2　信息隐藏的利弊25.2.3　构造对象的开销25.3 盒子里的流火25.3.1　DOM的内存开销25.3.2　浏览器的内存管理25.3.3　看清一个事实--内存泄漏的存在25.3.4　注意--及时关闭你的“盒子”25.3.5　一些误会的澄清25.3.5.1　脚本动态创建DOM导致内存泄漏25.3.5.2　闭包导致内存泄漏25.4 动态--魔鬼与天使25.4.1　动态解析的性能分析--一个动态特性的效率测试25.4.2　开发效率与执行效率--永远的困难选择25.4.3　优美与适用--学会经受魔鬼的诱惑25.4.4　扮演客户眼中的天使25.5 让代码轻舞飞扬25.5.1　简单就是美--为代码瘦身25.5.2　最完美的运用是不用25.5.3　高度抽象是为了简化问题25.5.4　逻辑和表达同样重要25.5.5　保持代码的严谨25.5.6　漂亮的书写风格--让阅读者心情愉快25.6 总结第26章　应用框架26.1 应用框架概览26.1.1　什么是应用框架26.1.2　应用框架的组成部分26.1.2.1　类库26.1.2.2　核心模块26.1.2.3　环境配置26.1.2.4　使用手册26.2 为什么要设计应用框架26.2.1　应用框架的适用范围26.2.2　应用框架的利弊26.3 如何设计应用框架26.3.1　把握设计的目标26.3.2　应用框架的设计准则26.3.3　什么样的应用框架才是成熟的应用框架26.3.4　应用框架的设计方法26.3.5　实战！
设计一个简单的应用框架26.3.5.1　自描述26.3.5.2　基础接口和语义型代码26.3.5.3　核心对象的原型扩展26.3.5.4　简单方法26.3.5.5　名字空间26.3.5.6　支持标准和跨浏览器26.3.5.7　事件模型--Silverna的事件模型26.3.5.8　应用模式26.3.5.9　提供Ajax组件26.3.5.10　内存管理和其他26.4 框架的实际应用--在Silverna 2.0框架上开发的Widgets26.5 已存在的应用框架26.5.1　Prototype26.5.2　JQuery26.5.3　Dojo26.5.4　JSVM26.5.5　其他框架26.5.5.1　Bindows（成立于2003年）26.5.5.2　BackBase（成立于2003年）26.5.5.3　DOJO（开发中，成立于2004年9月）26.5.5.4　Open Rico（开发中，成立于2005年5月，基于早期的一个proprietary框架）26.5.5.5　qooxdoo（开发中，成立于2005年5月）26.5.5.6　Tibet（开发中，创建于2005年6月）26.5.5.7　AJFORM（创建于2005年6月）26.6 总结

第一图书网, tushu007.com

章节摘录

第一部分 概论　第1章　从零开始　程序设计之道无远弗届，御晨风而返。
——杰弗瑞·詹姆士　在人类漫漫的历史长河里，很难找到第二个由简单逻辑和抽象符号组合而成的，具有如此宏大信息量和丰富多彩内涵的领域。
从某种意义上说，当你翻开这本书的时候，你已经踏入了一个任由你制定规则的未知世界。
尽管你面对的仅仅是程序设计领域的冰山一角，但你将透过它，去领悟“道”的奥秘。
在接下来的一段时间内，你会同我一起，掌握一种简单而优雅的神秘语言，学会如何将你的意志作用于它。
这种语言中所蕴涵着的亘古之力，将为你开启通往神秘世界的大门……　1.1　为什么选择JavaScript？

在一些人眼里，程序设计是一件神秘而浪漫的艺术工作，对他们来说，一旦选定某种编程语言，就会像一个忠贞的信徒一样坚持用它来完成任何事情，然而我不是浪漫的艺匠，大多数人也都不是，很多时候我们学习一种新技术的唯一目的，只是为了把手中的事情做得更好。
所以，。
当你面对一项陌生的技术时，需要问的第一个问题往往是，我为什么选择它，它对我来说，真的如我所想的那么重要吗？

好，让我们带着问题开始。

1.1.1 用户的偏好——B/S模式　如果你坚持站在专业人员的角度，你就很难理解为什么B/S模式会那么受欢迎。
如果你是一个资深的程序员，有时候你甚至会对那些B/S模式的东西有一点点反感。
因为在你看来，浏览器、表单、DOM和其他一切与B/S沾边的东西，大多是行为古怪而难以驾驭的。
以你的经验，你会发现实现同样的交互，用B/S来做通常会比用任何一种客户端程序来做要困难得多。

如果你尝试站在用户的角度，你会发现为什么大多数最终用户对B/S模式却是如此的青睐。
至少你不必去下载和安装一个额外的程序到你的电脑上，不必为反复执行安装程序而困扰，不必整天被新的升级补丁打断工作，不必理会注册表、磁盘空间和一切对普通用户来说有点头疼的概念。
如果你的工作地点不是固定的办公室，你日常工作的PC也不是固定的一台或者两台，那么，B/S的意义对你而言或许比想象的还要大。

媒体关注与评论

网友评论　　emu：这是我多年来最期盼的一本JS的书。
月影式的思考问题的方式和角度，总是让人耳目一新。
我首先想说的是，作者不是个正常人来着。
如果说像我这样的正常人看一个问题，可以构造出来一个数学模型，用一个函数来描述它fn(a，b，c，d...)={do sth.}，在确定了n个参数后，问题就可以解决；那么到了月影手里，这个问题就变成了n维的问题空间，在只确定了部分条件的时候，就可以把这个n维空间在一个n-m维空间上做一个投影，转化为另一个问题来看待（22.4节关于Currying的内容，是我看过的最精彩的关于函数式编程的讲解之一）。
一个脚本开发的问题，到了月影手里，也许实质上是个滤波器的问题（见本书第一章），也许和波粒二像性有关（看月影以前的博文），也许实质上是受到测不准原理的限制，也许和多维时空有密不可分的关系（都是某次和月影聊天时的话题），也许背后更有不可捉摸的“道”。
月影看待问题的目光，总是这样穿透表象，深入实质，并升华了问题本身，最后不但可以解决这一个问题，还可以解决相关的一系列问题，而且在解决问题的过程中，我们可能已经解决了一系列的问题。
这种月影式的风格，让我们每每击案叫绝。
这些年来，在JavaScript和DOM上面基于对象和函数式编程，网友们有过无数的尝试和探索。但是这样的探索之路是如此艰难，限于精力和能力，我们往往只能在得到零星小小的突破后便折返。我们也一直期待有达人可以穿透那些荆棘后把沿途的这些风景整理出来给后人分享，但是我们都知道这些风景只要看一眼已经是如此之难，要什么样的人，才能攀下这么多的高峰，并把它们一一系统的整理出来，呈现在大家面前？
aoao：是不是看到书的文字过于华丽，是不是认为这不是技术人员写的书？
别急~华丽的语言并不与技术冲突，虽然很难想像得到有一本技术类的书读起来可以用舒服来形容，很不巧，刚好这本书就是。
是不是找不到有专门介绍IE或Firefox兼容的一章，是不是正想放弃这本书？
别急~这不是一本只专注解决特定问题的书，而是一本让您看到JavaScript真相的书，当你了解了事实的真相时，问题自然迎刃而解。
dh20156：许久不曾买书，嗷嗷和月影的书是我至今为止最为期待的两本！
月影的这本书从开篇便用经验及趣味的脚本来吸引读者，加上引人入胜的描述方式，让人心痒不止，不睹不快！
再谈谈本书的设计，理论知识由浅到深的推进，实战则从最简单的应用直到当前最流行应用的深入分析，更值得一提的是，作者无时无刻不在引导着读者朝着一个真正的程序员的方向去看待问题、思考问题、解决问题！
可以说，本书是一本“老少皆宜”的JavaScript参考手册！
无论是新手还是已从事JavaScript工作多年的人，都可以从中获得巨大的帮助！
希望本书能够帮到更多的人，为Web开发带来更多新突破！
winter：这是一本求“道”的书。
市面上关于JavaScript的书极多，且不乏精品，然而多执著于“术”。
若讲JavaScript的语法，ECMA262标准的文档已经足够，若讲JavaScript与DOM应用，MSDN和W3C上的DOM参考也足够。
JavaScript是一门很有趣的语言，每取得一点更深入的理解，我便以为自己已经彻悟。
从最初的函数使用、面向对象、原型继承、函数对象到后来作用域链、functional、预解析、表达式解析机制、动态语言特性，每在此道路上前进一步，我都曾以为自己到达了顶峰。
然而，事实上是，简单的语法也能承载深刻的思想。
巧妙的设计让JavaScript成为一门兼有元类（meta-class）、类（class-based）、原型（prototype-based）三种面向对象特性和functional特性的动态语言。

第一图书网, tushu007.com

回忆起来，对这样的语言，说我已到达顶峰，是十分可笑的。
　　这本书虽然已经详细到足可以代替参考手册来查阅，但是，它是一门讲思想的书。
她将以优美的文字和同样优美的代码向你展示JavaScript的独特魅力。
读过这本书，你不但能了解JavaScript，还会对整个编程的思想有一个新的认识，对于面向对象语言、函数式语言、动态语言有全新的理解。
　　从另一个角度说，思想也并非作者所追求的“道”的终极。
透漏一下，除了项目经理/资深程序员外，作者还有很多身份：论坛版主、文学爱好者（写的小说挺不错、就是更新慢了点）、理论物理爱好者（是的，你将会看到他经常把程序跟理论物理联系起来）。
读过这本书，你也许能理解作者所追求的“道”。
　　多说无益，赶紧弄一本回家看吧。
　　leisang：正在看样章，非常不错。
前面的建议确实很有意义，一个JS初学者首先要学的不是如何炫耀各种网页特技，而是限制自己的欲望，保证只在必要的地方使用。
　　月影：那你觉得我写这本书是为什么？
为名？
为利？
还是什么？
　　这本书的市场定位和读者群又是什么？
　　我和出版社的关系是什么？
出版社为什么让我写这本书？
他们对市场有过怎样的研究？
我作为签约作者，又如何去对待这本书，如何在这之间取舍？
如何同时对出版社和读者负责？
　　很多事情不是由我能够决定的，而你呢，要理解他人的付出和他人的难处……　　你知道一本600页的书一年时间完成需要多少付出吗？
　　你知道被人期待而恨自己能力不能再高一点的那种感觉吗？
　　你知道自己累死累活写出来的书，本以为能够虚心接受心平气和的指点和批评的时候突然受到劈头盖脸的一通乱棒时的那种感觉吗？
　　不是我接受不了尖锐的批评，而是我希望评论时能够稍微体谅一下我这个作者，平和地告诉我哪里不好，哪里要修改……而不是一上来就说我这本书是没有价值的……　　你的技术和知识面我还是比较认可的，你的批评我也虚心接受，但你前面的一些过于尖锐的言辞我接受不了，抱歉了~　　（策划编辑点评：上面这段话是作者对一个在论坛上以极不友好的言词和视角攻击他的书的网友的回复，之所以把这段话放在这里，是因为从这段话中可以感受到作者对自己作品倾注的感情。
令我感动的是这句话——“你知道被人期待而恨自己能力不能再高一点的那种感觉吗？”）　　redcarpet：作者乃我的围棋师傅，我相信他的文章就如他教的围棋一样，让JS的学习者能进入神奇的JS之门。
　　希望JS迷们，能接近那传说中的“神之一技”的境界。
　　个人觉得最有看点的部分：第五部分 第二十二章 看点：闭包与函数式编程。
　　还有一点小小建议：　　新学JS的朋友，可以试着按如下顺序看：第一部分 第三章 开发环境和调试方法，第二部分 全部，第三、四部分，第五部分 最后看。
　　刚接触任何一门语言，碰到的问题就是编辑环境了，好多初学者自己摸索的话，会走不少的弯路。
JS也不例外，先了解开发环境和调试方法，磨刀不误砍柴功。
　　刚学JS，不要立刻陷入具体使用环境中的问题，先把语言的核心、思想学好了，再看客户端、服务端上的JS。
第二部分学透了，就会为后续的章节打下了比较好的基础。
　　voocoo：强烈支持一下，久闻月影大侠的名字，出书了我一定会买一本，光看目录就够吸引

我了！

greatghoul：这位兄台可就不对了，入门书是很多，但真正优秀的不是很多，现在比的就是谁的书能让读者更容易的入门，讲得那么专业，小孩子吓都吓死了，还入门个pp呀。

月版在论坛里名声也算不小，解决的问题也算不少，而且还有大公无私的精神（上面有所提及），大家都喜欢他（她）的生动的语言，读他（她）的书就好像在论坛里面和他（她）交流一样，在这样具有亲和力的环境中入门，是非常享受的。

这位兄台或许书读的很多，但对入门书籍方面只是想当然，不是书写得专业就行，像《Thinking in Java》就写得很有亲和力，但依然受很多所谓专业人士的抨击，然而读者喜欢。

月版要致力于写类似于Thinking in JavaScript的东西呀。

秦皇也爱JS：说实话，这本书非常好看，月影的写作水平和文字功底是非常之高啊。

像《JavaScript权威指南》，有时候看着看着就看不下去了，但这本书的样章却让我看得兴致盎然，不忍释卷。

里面竟然还穿插了我喜欢的《明朝那些事儿》的句子，呵呵，真是奇峰迭起。

比起呆板严肃的译版，我还是更喜欢这本。

顶！

编程浪子：一本有思想的书，其价值已超出内容本身，以前那些JS的书看着就乏味，教条式太严重。

还没看到书，不敢把作者评得那么好，不过我相信月影的实力，样章就写得不错，可谓“得道中人”呀。

xpnew：月影老大出书了，我们一定要支持！

确实很感动啊。

myhome：不适合你读，但很适合我读！

没人要你读，综合你的一些言论和表现，得出的结论是你没资格读这书，因为你根本就不懂作者写的是什么！

（策划编辑点评：在论坛上以极不友好的言词和视角攻击作者所写的这本书的那个网友的言词，激起了其他很多网友的反感，这是其中一个网友对那位攻击者的回复。）

果蝇：其实我更希望月影把书写成小说或散文的形式，有什么不可以的呢！

我们需要的是生动有趣的语言和奇思妙想，也许不是很技术，但可以给读者以遐想、启发，可以开阔我们的思路，我想如果书能达到这个目的，那就是一本难得的好书了。

我想这也是无忧能如此吸引人的原因。

看过月版的一些帖子，对月版的书有更多的期待。

易中天曾把自己的文章给出版社看，主编退稿说你这个是散文，不是学术著作。

事实证明学术这些东西不只是在高高的象牙塔尖上，也证明一些人脑筋确实死得很，不必太在意。

月影应该感到荣幸有这么多的关注，不管是好是坏，都是支持。

solidluck：你就放肆的写吧，让我们也放肆的看。

真不聪明：个人觉得好书在于对基础讲解的深刻，而不是用最新的技术来引诱你。

xuchaofeng：在哪儿有买的，期待中……

monfs：快出呀，等不急了，这是一个历史时刻！

fuhao19830308：这书要出了可得通知一声，看了样章就有了看下去的欲望了！

LeoZ：我也每天上来看一下是否有出版，可是每次都没消息，我也继续等。

samon127：什么时候出版啊，等了好久了：）

ansonvili：什么时候出啊，等不急了。

freeren：看了样章，很贴切，很易理解，期待书的发行！

ravioli：每天以泪洗面的等啊等。

fchnipj：等的好辛苦，月版主能说说具体的时间吗？

好想看呀。

第一图书网, tushu007.com

（策划编辑点评：本来我希望这本书在2008年的春节后就能出版，而实际上却由于种种原因一直拖到了6月份这本书才面世。
说声抱歉，读者们，让你们久等了。
）

第一图书网, tushu007.com

<<JavaScript王者归来>>

编辑推荐

脚本神力与你同在 无忧版主月影力作。
《JavaScript王者归来》是一本寻求JavaScript解决之道的书。
《JavaScript王者归来》是一本探寻程序设计思想本源的“魔法典籍”。
从简单的语言基础到丰富的实际应用再到语言本质的深入剖析。
字里行间包含作者对JavaScript实践乃至程序设计思想的深入思考和总结　　脚本攻击就在黎明发起。

第一图书网, tushu007.com

<<JavaScript王者归来>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com