第一图书网, tushu007.com

<<JavaScript高级程序设计:第2 >

图书基本信息

书名：<<JavaScript高级程序设计:第2版>>

13位ISBN编号：9787115230959

10位ISBN编号：7115230951

出版时间：2010-7

出版时间：人民邮电出版社

作者：Nicholas Zakas

页数：600

译者：李松峰,曹力

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

前言

从诞生至今的大部分时间里，恐惧、咒骂、蔑视和误解一直与JavaScript如影随形。
JavaScript刚刚问世那几年，很多“严肃的程序员”都认为它不够严肃。
相比之下，COM泡沫时期加入Web开发行列的许多文科生，则普遍觉得JavaScript深不可测、晦涩难懂。
就算那些耐力和韧性俱佳者能够把JavaScript琢磨得很透，但仍然摆脱不掉竞争中的各种浏览器提供的不同实现给他们带来的麻烦。
凡此种种，最终导致粗制滥造的脚本越来越多。
另一方面，拜Web前端代码的无比开放性所赐，各种坏习惯不断从一个站点被粘贴进另一个站点的源代码中。
那些实现活该臭名昭著，可是，JavaScript这门语言也因此被严重拖累，背上了不该有的坏名声。
2001年前后（随着Interact Explorer 6的发布），浏览器实现已经大为改进，Web开发实践也开始得到改善，呈现出了二者水乳交融的局面。

内容概要

本书在上一版基础上进行了大幅度更新和修订，融入了近几年来JavaScript应用发展的最新成果，几乎涵盖了所有需要理解的重要概念和最新的JavaScript应用成果。
从颇具深度的JavaScript语言基础到作用域(链)，从引用类型到面向对象编程，从极其灵活的匿名函数到闭包的内部机制，从浏览器对象模型(BOM)、文档对象模型(DOM)到基于事件的Web脚本设计，从XML(E4X)到Ajax及JSON，从高级前端开发技术到前沿的客户端存储，从最佳编程实践到即将成为现实的API，直至JavaScript未来的发展，全景式地展示了JavaScript高级程序设计的方方面面。

本书适合不同层次的JavaScript/Web开发人员阅读参考，也可作为高校相关专业课程的教材。

作者简介

作者：（美国）尼古拉斯·泽卡斯（Nicholas C.Zakas） 译者：李松峰 曹力尼古拉斯·泽卡斯（Nicholas C.Zakas），世界顶级Web技术专家，现为Yahoo！
公司首席前端工程师尼古拉斯拥有丰富的Web开发和界面设计经验.曾经参与许多世界级大公司的Web解决万案开发。

书籍目录

第1章 JavaScript简介
1.1 JavaScript简史
1.2 JavaScript实现
1.2.1 ECMAScript
1.2.2 文档对象模型(DOM)
1.2.3 浏览器对象模型(BOM)
1.3 JavaScript版本
1.4 小结
第2章 在HTML中使用JavaScript
2.1 script元素
2.1.1 标签的位置
2.1.2 延迟脚本
2.1.3 在XHTML中的用法
2.1.4 不推荐使用的语法
2.1.5 嵌入代码与外部文件
2.2 文档模式
2.3 noscript元素
2.4 小结
第3章 基本概念
3.1 语法
3.1.1 区分大小写
3.1.2 标识符
3.1.3 注释
3.1.4 语句
3.2 关键字和保留字
3.3 变量
3.4 数据类型
3.4.1 typeof操作符
3.4.2 Undefined类型
3.4.3 Null类型
3.4.4 Boolean类型
3.4.5 Number类型
3.4.6 String类型
3.4.7 Object类型
3.5 操作符
3.5.1 一元操作符
3.5.2 位操作符
3.5.3 布尔操作符
3.5.4 乘性操作符
3.5.5 加性操作符
3.5.6 关系操作符
3.5.7 相等操作符
3.5.8 条件操作符
3.5.9 赋值操作符
3.5.10 逗号操作符

3.6 语句
3.6.1 if语句
3.6.2 do-while语句
3.6.3 while语句
3.6.4 for语句
3.6.5 for-in语句
3.6.6 label语句
3.6.7 break和continue语句
3.6.8 with语句
3.6.9 switch语句
3.7 函数
3.7.1 理解参数
3.7.2 没有重载
3.8 小结
第4章 变量、作用域和内存问题
4.1 基本类型和引用类型的值
4.1.1 动态属性
4.1.2 复制变量值
4.1.3 传递参数
4.1.4 检测类型
4.2 执行环境及作用域
4.2.1 延长作用域链
4.2.2 没有块级作用域
4.3 垃圾收集
4.3.1 标记清除
4.3.2 引用计数
4.3.3 性能问题
4.3.4 管理内存
4.4 小结
第5章 引用类型
5.1 Object类型
5.2 Array类型
5.2.1 转换方法
5.2.2 栈方法
5.2.3 队列方法
5.2.4 重排序方法
5.2.5 操作方法
5.3 Date类型
5.3.1 继承的方法
5.3.2 日期格式化方法
5.3.3 日期/时间组件方法
5.4 RegExp类型
5.4.1 RegExp实例属性
5.4.2 RegExp实例方法
5.4.3 RegExp构造函数属性
5.4.4 模式的局限性
5.5 Function类型

5.5.1 没有重载(深入理解)
5.5.2 函数声明与函数表达式
5.5.3 作为值的函数
5.5.4 函数内部属性
5.5.5 函数属性和方法
5.6 基本包装类型
5.6.1 Boolean类型
5.6.2 Number类型
5.6.3 String类型
5.7 内置对象
5.7.1 Global对象
5.7.2 Math对象
5.8 小结
第6章 面向对象的程序设计
6.1 创建对象
6.1.1 工厂模式
6.1.2 构造函数模式
6.1.3 原型模式
6.1.4 组合使用构造函数模式和原型模式
6.1.5 动态原型模式
6.1.6 寄生构造函数模式
6.1.7 稳妥构造函数模式
6.2 继承
6.2.1 原型链
6.2.2 借用构造函数
6.2.3 组合继承
6.2.4 原型式继承
6.2.5 寄生式继承
6.2.6 寄生组合式继承
6.3 小结
第7章 匿名函数
7.1 递归
7.2 闭包
7.2.1 闭包与变量
7.2.2 关于this对象
7.2.3 内存泄漏
7.3 模仿块级作用域
7.4 私有变量
7.4.1 静态私有变量
7.4.2 模块模式
7.4.3 增强的模块模式
7.5 小结
第8章 BOM
8.1 window对象
8.1.1 全局作用域
8.1.2 窗口关系及框架
8.1.3 窗口位置

8.1.4 窗口大小
8.1.5 导航和打开窗口
8.1.6 间歇调用和超时调用
8.1.7 系统对话框
8.2 location对象
8.2.1 查询字符串参数
8.2.2 位置操作
8.3 navigator对象
8.3.1 检测插件
8.3.2 注册处理程序
8.4 screen对象
8.5 history对象
8.6 小结
第9章 客户端检测
9.1 能力检测
9.2 怪癖检测
9.3 用户代理检测
9.3.1 用户代理字符串的历史
9.3.2 用户代理字符串检测技术
9.3.3 完整的代码
9.3.4 使用方法
9.4 小结
第10章 DOM
10.1 节点层次
10.1.1 Node类型
10.1.2 Document类型
10.1.3 Element类型
10.1.4 Text类型
10.1.5 Comment类型
10.1.6 CDATASection类型
10.1.7 DocumentType类型
10.1.8 DocumentFragment类型
10.1.9 Attr类型
10.2 DOM扩展
10.2.1 呈现模式
10.2.2 滚动
10.2.3 children属性
10.2.4 contains()方法
10.2.5 操作内容
10.3 DOM操作技术
10.3.1 动态脚本
10.3.2 动态样式
10.3.3 操作表格
10.3.4 使用NodeList
10.4 小结
第11章 DOM2和DOM3
11.1 DOM变化

11.1.1 针对XML命名空间的变化
11.1.2 其他方面的变化
11.2 样式
11.2.1 访问元素的样式
11.2.2 操作样式表
11.2.3 元素大小
11.3 遍历
11.3.1 NodeIterator
11.3.2 TreeWalker
11.4 范围
11.4.1 DOM中的范围
11.4.2 IE中的范围
11.5 小结
第12章 事件
12.1 事件流
12.1.1 事件冒泡
12.1.2 事件捕获
12.1.3 DOM事件流
12.2 事件处理程序(或事件侦听器)
12.2.1 HTML事件处理程序
12.2.2 DOM0级事件处理程序
12.2.3 DOM2级事件处理程序
12.2.4 IE事件处理程序
12.2.5 跨浏览器的事件处理程序
12.3 事件对象
12.3.1 DOM中的事件对象
12.3.2 IE中的事件对象
12.3.3 跨浏览器的事件对象
12.4 事件类型
12.4.1 UI事件
12.4.2 鼠标事件
12.4.3 键盘事件
12.4.4 HTML事件
12.4.5 变动事件
12.4.6 专有事件
12.4.7 移动Safari支持的事件
12.5 内存和性能
12.5.1 事件委托
12.5.2 移除事件处理程序
12.6 模拟事件
12.6.1 DOM中的事件模拟
12.6.2 IE中的事件模拟
12.7 小结
第13章 表单脚本
13.1 表单
13.1.1 提交表单
13.1.2 重置表单

13.1.3 表单字段
13.2 文本框脚本
13.2.1 选择文本
13.2.2 过滤输入
13.2.3 自动切换焦点
13.3 选择框脚本
13.3.1 选择选项
13.3.2 添加选项
13.3.3 移除选项
13.3.4 移动和重排选项
13.4 表单序列化
13.5 富文本编辑
13.5.1 操作富文本
13.5.2 富文本选区
13.5.3 表单与富文本
13.6 小结
第14章 错误处理与调试
14.1 浏览器错误报告
14.1.1 Internet Explorer
14.1.2 Firefox
14.1.3 Safari
14.1.4 Opera
14.1.5 Chrome
14.2 错误处理
14.2.1 try-catch语句
14.2.2 抛出错误
14.2.3 错误(error)事件
14.3 错误处理策略
14.3.1 常见的错误类型
14.3.2 区分致命错误和非致命错误
14.3.3 把错误记录到服务器
14.4 调试技术
14.4.1 将消息记录到控制台
14.4.2 将消息记录到当前页面
14.4.3 抛出错误
14.5 常用的IE错误
14.5.1 操作终止
14.5.2 无效字符
14.5.3 未找到成员
14.5.4 未知运行时错误
14.5.5 语法错误
14.5.6 系统无法找到指定资源
14.6 调试工具
14.6.1 IE中的调试器
14.6.2 Firebug
14.6.3 Drosera
14.6.4 Opera中的JavaScript调试器

14.6.5 其他调试工具
14.7 小结
第15章 JavaScript与XML
15.1 浏览器对XML DOM的支持
15.1.1 DOM2级核心
15.1.2 DOMParser类型
15.1.3 XMLSerializer类型
15.1.4 DOM3级加载和保存
15.1.5 IE对XML的支持
15.1.6 跨浏览器处理XML
15.2 浏览器对XPath的支持
15.2.1 DOM3级XPath
15.2.2 IE中的XPath
15.2.3 跨浏览器使用XPath
15.3 浏览器对XSLT的支持
15.3.1 IE中的XSLT
15.3.2 XSLTProcessor类型
15.3.3 跨浏览器使用XSLT
15.4 小结
第16章 E4X
16.1 E4X的类型
16.1.1 XML类型
16.1.2 XMLList类型
16.1.3 命名空间类型
16.1.4 QName类型
16.2 一般用法
16.2.1 访问特性
16.2.2 其他节点类型
16.2.3 查询
16.2.4 构建和操作XML
16.2.5 解析和序列化
16.2.6 命名空间
16.3 其他变化
16.4 全面启用E4X
16.5 小结
第17章 Ajax与JSON
17.1 XHR对象
17.1.1 XHR的用法
17.1.2 HTTP头部信息
17.1.3 GET请求
17.1.4 POST请求
17.1.5 浏览器差异
17.1.6 安全
17.2 跨域请求
17.2.1 XDomainRequest对象
17.2.2 跨域XHR
17.3 JSON

17.3.1 在Ajax中使用JSON
17.3.2 安全
17.4 小结
第18章 高级技巧
18.1 高级函数
18.1.1 作用域安全的构造函数
18.1.2 惰性载入函数
18.1.3 函数绑定
18.1.4 函数柯里化
18.2 高级定时器
18.2.1 重复的定时器
18.2.2 Yielding Processes
18.2.3 函数节流
18.3 自定义事件
18.4 拖放
18.4.1 修缮拖动功能
18.4.2 添加自定义事件
18.5 小结
第19章 客户端存储
19.1 cookie
19.1.1 限制
19.1.2 cookie的成分
19.1.3 JavaScript中的cookie
19.1.4 子cookie
19.1.5 关于cookie的思考
19.2 IE用户数据
19.3 DOM存储机制
19.3.1 存储类型
19.3.2 sessionStorage对象
19.3.3 globalStorage对象
19.3.4 localStorage对象
19.3.5 StorageItem类型
19.3.6 storage事件
19.3.7 限制
19.4 总结
第20章 最佳实践
20.1 可维护性
20.1.1 什么是可维护的代码
20.1.2 代码约定
20.2 松散耦合
20.3 性能
20.3.1 注意作用域
20.3.2 选择正确方法
20.3.3 最小化语句数
20.3.4 优化DOM交互
20.4 部署
20.4.1 构建过程

20.4.2 验证
20.4.3 压缩
20.5 小结
第21章 未来的API
21.1 选择器API
21.1.1 querySelector()方法
21.1.2 querySelectorAll()方法
21.1.3 现今和未来的支持情况
21.2 HTML5
21.2.1 字符集属性
21.2.2 类相关的增加
21.2.3 自定义数据特性
21.2.4 跨文档消息传递
21.2.5 媒体元素
21.2.6 canvas元素
21.2.7 离线支持
21.2.8 历史的改变
21.2.9 数据库存储
21.2.10 拖放操作
21.2.11 WebSocket类型
21.2.12 HTML5的未来
21.3 小结
第22章 JavaScript的未来
22.1 ECMAScript4/JavaScript2
22.1.1 JavaScript1.5
22.1.2 JavaScript1.6
22.1.3 JavaScript1.7
22.1.4 JavaScript1.8
22.1.5 JavaScript1.9
22.1.6 ECMAScript4提案
22.1.7 变量类型
22.1.8 函数
22.1.9 类型定义
22.1.10 类和接口
22.1.11 接口
22.1.12 继承
22.1.13 命名空间
22.1.14 包
22.1.15 语言上的其他变更
22.1.16 ECMAScript4的未来
22.2 ECMAScript3.1
22.2.1 对对象内部实现的变更
22.2.2 静态对象方法
22.2.3 本地的JSON支持
22.2.4 Decimal
22.2.5 用法子集
22.2.6 ECMAScript3.1的未来

22.3 小结
附录A JavaScript库
附录B JavaScript工具

章节摘录

插图：ECMAScript变量可能包含两种不同数据类型的值：基本类型值和引用类型值。
基本类型值指的是那些保存在栈内存中的简单数据段，即这种值完全保存在内存中的一个位置，而引用类型值则是指那些保存在堆内存中的对象，意思是变量中保存的实际上只是一个指针，这个指针指向内存中的另一个位置，该位置保存对象。
在将一个值赋给变量时，解析器必须确定这个值是基本类型值，还是引用类型值。
第3章讨论了5种基本数据类型：Undefined、Null、Boolean、Number和String，这5种基本数据类型的值在内存中分别占有固定大小的空间，因此可以把它们的值保存在栈内存中，而且，这样也可以提高查询变量的速度。
对于保存基本类型值的变量，我们说它们是按值访问的，因为我们操作的是它们实际保存的值。
在某些语言中，字符串以对象的形式来表示，因此被认为是引用类型的，ECMAScript放弃丁这一传统。
如果赋给变量的是一个引用类型的值，则必须在堆内存中为这个值分配空间。
由于这种值的大小不固定，因此不能把它们保存到栈内存中。
但内存地址的大小是固定的，因此可以将内存地址保存在栈内存中。
这样，当查询引用类型的变量时，就可以首先从栈中读取内存地址，然后再“顺藤摸瓜”地找到保存在堆中的值，对于这种查询变量值的方式，我们把它叫做按引用访问，因为我们操作的不是实际的值，而是被那个值所引用的对象。
图4.1形象地说明了如何在内存中保存这两种不同数据类型的值。
图4.1中展示了一些保存在栈内存中的基本类型值。
保存在栈内存中的每个值，分别占据着固定大小的空间，可以按照顺序来访问它们。
如果栈内存中保存的是一块内存的地址，则这个值就像是一个指向对象在堆内存中位置的指针。
保存在堆内存中的数据不是按照顺序访问的，因为每个对象所需要的内存空间并不相等。

媒体关注与评论

“如果你像我一样，想学习或者熟练掌握今天最热门的Web开发技术，本书是一个绝佳的起点，适合在所有Ajax图书之前阅读。
”　　——J.Ambrose Little Microsoft MVP“本书作者显然非常了解读者的需要，落笔切中要害，行文信息密集.单单对客户端通信、Web服务、正则表达式、DOM、XML处理等现代JavaScrpt技术的详细讲解，就已经物超所值。
”　　——JavaScriptkit.com

编辑推荐

《JavaScript高级程序设计(第2版)》：JavaScript经典教程Amazon超级畅销书前端开发人员必备JavaScrpt的应用在广度和深度上日益扩大和加深，前端开发亟待掌握的JavaScrpt技能也越来越具有挑战性。
这个新版本几乎全部更新、重写了上一版的内容，融入了作者近几年来奋战在前端开发一线的宝贵经验，是学习和提高JavaScrpt技能的必读经典。
《JavaScript高级程序设计(第2版)》不仅全面深入地讲述了，JavaScript的基本概念，阐释了它特有的面向对象和继承的机制.还详尽讨论了JavaScrpt实现的各个组成部分。
在以大量篇幅全景式剖析ECMAScript和DOM的过程中，各个级别的DOM规范在作者笔下纷至沓来，诸如事件模拟、XML解析、XPath查询等高级主题也讲得分外清楚。
此外，错误处理与调试、Ajax与JSON，乃至客户端存储、未来的API等章节也都条理清晰、异彩纷呈。
《JavaScript高级程序设计(第2版)》适合有一定编程经验的前端开发人员阅读，也可作为高校相关专业课程的教材。
一幅浓墨重彩的语言画卷，一部推陈出新的技术名著

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