第一图书网, tushu007.com

<<JavaScript经典实例>>

图书基本信息

书名：<<JavaScript经典实例>>

13位ISBN编号：9787512320581

10位ISBN编号：7512320582

出版时间：2012-3

出版时间：中国电力出版社

作者：Shelley Powers

页数：512

译者：李强

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

内容概要

当你在JavaScript中遇到问题的时候，没有必要再去做一些重复无谓的劳动。
因为《JavaScript经典实例》各节中的完整代码解决了常见的编程问题，并且给出了在任何浏览器中构建Web应用程序的技术。
只需要将这些代码示例复制并粘贴到你自己的项目中就行了，可以快速完成工作，并且在此过程中学习JavaScript的很多知识。

你还将学习如何利用ECMAScript 5和HTML5中的最新功能，包括新的跨域挂件通信技术、HTML5的video和audio元素，以及绘制画布。
书中一些章节介绍了如何将这些技术与JavaScript一起使用，构建高品质的应用程序界面。

作者简介

Shelley Powers，已经从事Web技术工作和写作达15年之久，从JavaScript初次开发到现在最新的图形和设计工具发布。
她最近在O'Reilly出版的图书包括语义网、Ajax、JavaScript和Web图形。

书籍目录

前言
第1章 使用JavaScript字符串
1.0 简介
1.1 连接两个或多个字符串
1.2 连接字符串和另一种数据类型
1.3 条件比较字符串
1.4 在字符串中查找子字符串
1.5 从一个字符串提取子字符串
1.6 检查一个存在的、非空的字符串
1.7 将一个关键字字符串分解为单独的关键字
1.8 插入特殊字符
1.9 处理textarea的单个行
1.10 去除字符串末尾的空白
1.11 左补充或右补充一个字符串
第2章 使用正则表达式
2.0 简介
2.1 测试一个子字符串是否存在
2.2 测试不区分大小写的子字符串匹配
2.3 验证社会安全号码
2.4 找到并突出显示一个模式的所有实例
2.5 使用新字符串替换模式
2.6 使用捕获圆括号交换一个字符串中的单词
2.7 使用正则表达式来去除空白
2.8 使用命名实体来替代HTML标签
2.9 搜索特殊字符
第3章 日期、时间和定时器
3.0 简介
3.1 打印出今天的日期
3.2 打印出UTC日期和时间
3.3 打印出一个ISO 8601格式日期
3.4 把一个ISO 8601格式的日期转换为Date对象可接受的一种格式
3.5 创建一个特定的日期
3.6 规划未来的一个日期
3.7 记录流逝的时间
3.8 创建一个延迟
3.9 创建重复性定时器
3.10 使用带有定时器的函数闭包
第4章 使用Number和Math
4.0 简介
4.1 保持一个递增的计数
4.2 把十进制数转换为一个十六进制值
4.3 创建一个随机数生成器
4.4 随机产生颜色
4.5 把表中的字符串转换为数字
4.6 把表中一列的所有数字加和

4.7 在角度和弧度之间转换
4.8 找到页面元素可容纳的一个圆的半径和圆心
4.9 计算圆弧的长度
第5章 使用数组和循环
5.0 简介
5.1 循环遍历数组
……
第6章 JavaScript函数构建重用性
第7章 处理事件
第8章 浏览器模块
第9章 表单元素和验证
第10章 调试和错误处理
第11章 访问页面元素
第12章 创建和删除元素和属性
第13章 使用Web页面空间
第14章 使用JavaScript、CSS和ARIA创建交互和可访问性效果
第15章 创建富媒体和交互应用程序
第16章 JavaScript对象
第17章 JavaScript库
第18章 通信
第19章 使用结构化数据
第20章 持久化
第21章 JavaScript创新用法

章节摘录

版权页： 插图： 讨论 我们想要尽可能地避免全局变量，因为它和库有潜在的冲突。
然而有时候你需要存储要跨越几个函数调用使用的数据，并且你不想每次都必须重复地给该函数发送这些信息。
维持这些从一个函数到另一个函数的数据的一种方法是，在一个函数内部创建一个函数，让它们都访问该数据，然后从外围函数返回内部函数。
从另一个函数返回一个函数，这叫做函数闭包(function closure)。
在具体介绍函数闭包之前，我想先花几分钟来介绍函数和作用域。
注意：这种类型的函数闭包也叫做局部函数(partial function)，或者叫做科里化(currying)，6.8节将介绍它。
在解决方案中，内部函数addName在外部函数greetingHaker之内定义。
这两个函数都有一个参数。
内部函数访问了自己的参数以及外部函数的参数，但是外部函数没有访问传递给内部函数的参数。
内部函数可以操作外部函数的参数，是因为它在同样的环境（或外部函数的作用域）内运行。
在JavaScript中，有一个为最外围的应用程序创建的作用域。
所有全局变量、函数和对象都包含在这个外围作用域之中。
当创建一个函数的时候，你创建了一个新的作用域，只要这个函数存在，其作用域就存在。
函数可以访问其作用域内的所有变量，以及来自外围作用域的所有变量，但是外围作用域不能访问该函数中的变量。
由于这些作用域规则，我们可以在浏览器应用程序中访问window和document对象，并且解决方案中的内部函数也可以访问传给它的数据，或者最初位于包围它的外围函数中的那些数据。
注意：这也解释了为什么6.6节中的递归函数可以内部地访问那些它们在外部应用程序作用域中赋值的变量。
然而外围函数不能访问内部函数的参数或局部变量，因为它们存在于另一个作用域中。
内部函数不一定必须从外部函数返回。
它可能是外围函数的代码中的一个调用指令。
当返回它的时候，就像在解决方案和如下代码中那样： function outer(X){return function(y){return x*y；)；} var multiThree=outer（3）；alert（multiThree（2））；／／打印出6 alert（multiThree（3））；／／打印出9 返回的函数形成了一个闭包。
JavaScript闭包是对一个函数来说是局部的变量，当该函数返回的时候，这个变量仍然存在。

编辑推荐

《JavaScript经典实例》讲述如何利用最新的网络功能，包括HTML5的持久存储机制和绘图优势，能够帮助读者开发具有可靠性的应用程序，兼容各种浏览器。
更新了ECMAScript5和覆盖范围包括HTML5，解释如何让JavaScript允许开发人员接触到页面和样式的核心，演示如何正则表达式可以简化关键字，储存在浏览器应用程序的信息，演示如何使用共同的JavaScript库，包括jQuery和Prototype，适用于XML和JSON的应用。

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