

<<高性能JavaScript>>

图书基本信息

书名：<<高性能JavaScript>>

13位ISBN编号：9787121119323

10位ISBN编号：7121119323

出版时间：2010-11

出版时间：电子工业出版社

作者：Nicholas C.Zakas

页数：210

译者：丁琛,赵泽欣

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>

<<高性能JavaScript>>

前言

这是一本关于JavaScript性能的书。

在Web应用日趋丰富的今天，越来越多的JavaScript被运用在我们的网页中。

随着用户体验被日益重视，前端性能对用户体验的影响开始备受关注，而引起性能问题的因素相对复杂，因此它很难得到全面的解决。

这本书是一个契机，它尝试着从多个方面综合分析导致性能问题的原因，并给出适合的解决方案，帮助我们改善Web应用的品质。

这本书页数不多，但它承载着JavaScript性能方面最为宝贵的经验。

不仅从语言特性、数据结构、浏览器机理、网络传输等层面分析导致性能问题的原因，还介绍了多种工具来帮助我们提升开发过程和部署环节的工作效率。

本书作者Nicholas C.Zakas是位经验丰富的前端专家，他的许多研究对前端业界的贡献让我们受益匪浅。

本书的另外五位特约作者均为各自领域的专家，他们的专业技能和知识的融入使得本书内容更为充实，更具实用价值。

特别感谢赵泽欣（小马），他为审阅译文花了大量的时间和精力，他的耐心和细致让我十分敬佩。

感谢朱宁（白鸦）和周筠老师的引荐让我得以参与本书的翻译。

还要感谢博文视点的编辑徐定翔先生和刘唯一女士在本书翻译过程中给予的极大理解和帮助。

<<高性能JavaScript>>

内容概要

如果你使用JavaScript构建交互丰富的Web应用，那么JavaScript代码可能是造成你的Web应用速度变慢的主要原因。

本书揭示的技术和策略能帮助你在开发过程中消除性能瓶颈。

你将会了解如何提升各方面的性能，包括代码的加载、运行、DOM交互、页面生存周期等。

雅虎的前端工程师Nicholas C. Zakas和其他五位JavaScript专家介绍了页面代码加载的最佳方法和编程技巧，来帮助你编写更为高效和快速的代码。

你还会了解到构建和部署文件到生产环境的最佳实践，以及有助于定位线上问题的工具。

<<高性能JavaScript>>

作者简介

Nicholas C.Zakas, 雅虎首页的主要开发者, 雅虎用户界面库 (YUI) 代码贡献者, 擅长利用JavaScript、HTML、CSS、XML、XSLT设计和实现WEB界面的软件工程师。

书籍目录

前言第1章：加载和执行1.1 脚本位置1.2 组织脚本1.3 无阻塞的脚本1.3.1 延迟的脚本1.3.2 动态脚本元素1.3.3 XMLHttpRequest脚本注入1.3.4 推荐的无阻塞模式1.4 小结第2章：数据访问2.1 管理作用域2.1.1 作用域链和标识符解析2.1.2 标识符解析的性能2.1.3 改变作用域链2.1.4 动态作用域2.1.5 闭包，作用域和内存2.2 对象成员2.2.1 原型2.2.2 原型链2.2.3 嵌套成员2.2.4 缓存对象成员值2.3 小结第3章：DOM编程3.1 浏览器中的DOM3.1.1 天生就慢3.2 DOM访问与修改3.2.1 innerHTML对比DOM方法3.2.2 节点克隆3.2.3 HTML集合3.2.4 遍历DOM3.3 重绘与重排3.3.1 重排何时发生？3.3.2 渲染树变化的排队与刷新3.3.3 最小化重绘和重排3.3.4 缓存布局信息3.3.5 让元素脱离动画流3.3.6 IE和：hover3.4 事件委托3.5 小结第4章：算法和流程控制第5章：字符串和正则表达式第6章：快速响应的用户界面第7章：Ajax第8章：编程实践第9章：构建并部署高性能JavaScript应用第10章：工具索引

<<高性能JavaScript>>

章节摘录

在1996年, JavaScript引擎只要能支持页面里数十行的JavaScript代码就好, 而今天, 却运行着成千上万行JavaScript代码的Web应用。

从许多方面来说, 如果不是因为浏览器自身在语言管理和基础设施方面的落后, JavaScript本可能取得更大规模的成功。

IE6就是一个明证, 发布之初, 它的稳定性和性能都被人们称颂, 但后来却因为自身的Bug和反应迟钝而被痛批为令人讨厌的Web应用平台。

事实上, IE6并没有变慢, 它只是被寄予了厚望。

2001年IE 6刚发布时出现的各类早期Web应用比2005年后出现的应用更轻量, JavaScript代码也远没有那么多。

JavaScript代码数量的增长带来的影响变得明显, IE 6的JavaScript引擎吃不消了, 原因在于它的“静态垃圾回收机制”。

该引擎监视内存中固定数量的对象来确定何时进行垃圾回收。

早期的Web应用开发人员很少会遇到这个极限值, 随着更多的JavaScript代码产生越来越多的对象, 复杂的Web应用开始频繁遭遇这个门槛。

问题变得清晰起来: JavaScript开发人员和Web应用都在发展, 而JavaScript引擎却没有。

尽管其他浏览器。

有着更加完善的垃圾回收机制和更好的运行性能, 但大多数仍然使用JavaScript解释器来执行代码。

解释性代码天生就没有编译性代码快, 因为解释性代码必须经历把代码转化成计算机指令的过程。

无论解释器怎样优化和多么智能, 它总是会带来一些性能损耗。

编译器已经有了各种各样的优化, 使得开发人员可以按照他们想要的方式编写代码, 而不需要担心是否是最优。

编译器可以基于词法分析去判断代码想实现什么, 然后产生出能完成任务的运行最快的机器码来进行优化。

解释器很少有这样的优化, 这很大程度上意味着, 代码怎么写, 就被怎么执行。

实际上, 通常在其他语言中由编译器处理的优化, 在JavaScript中却要求开发人员来完成。

.....

<<高性能JavaScript>>

媒体关注与评论

《高性能JavaScript》涵盖了当今JavaScript开发者需要了解的所有性能问题，毫无疑问，它已加入我的性能最佳实践列表。

” ——Steve Souders 《高性能JavaScript》是个让人印象深刻的JavaScript话题、技巧、秘诀的集合。

如果你想编写高质量JavaScript代码，这本书值得一读。

” ——Venkat Udayasankar 雅虎搜索性能专家

<<高性能JavaScript>>

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>