

<<编写可维护的JavaScript>>

图书基本信息

书名：<<编写可维护的JavaScript>>

13位ISBN编号：9787115310088

10位ISBN编号：7115310084

出版时间：2013-4

出版时间：人民邮电出版社

作者：扎卡斯

译者：李晶,郭凯,张散集

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>

<<编写可维护的JavaScript>>

内容概要

《编写可维护的JavaScript》向开发人员阐述了如何在团队开发中编写具备高可维护性的JavaScript代码，书中详细说明了作为团队一分子，应该怎么写JavaScript。

《编写可维护的JavaScript》内容涵盖了编码风格、编程技巧、自动化、测试等几方面，既包括具体风格和原则的介绍，也包括示例和技巧说明，最后还介绍了如何通过自动化的工具和方法来实现一致的编程风格。

《编写可维护的JavaScript》适合前端开发工程师、JavaScript程序员和学习JavaScript编程的读者阅读，也适合开发团队负责人、项目负责人阅读。

<<编写可维护的JavaScript>>

作者简介

作者：（美国）Nicholas C. Zakas 译者：李晶 郭凯 张散集 Nicholas C. Zakas 是一名前端开发顾问、作者、演讲家。

他在Yahoo! 供职超过5年时间。

在这期间，他曾是Yahoo！

首页首席前端工程师和YUI代码贡献者。

他著有《JavaScript高级程序设计》、《Ajax高级程序设计》和《高性能JavaScript》。

Nicholas C. Zakas 是顶级的Web技术专家，也是《JavaScript高级程序设计》一书的作者。

他曾是Yahoo！

的首席前端开发工程师，在完成了从一名“独行侠”到“团队精英”的蜕变后，他站在前端工程师的角度提炼出众多的最佳编程实践，其中包括很多业内权威所推崇的最佳法则，而这些宝贵经验正是《编写可维护的JavaScript》的核心内容。

<<编写可维护的JavaScript>>

书籍目录

第一部分 编程风格 第1章 基本的格式化 1.1 缩进层级 1.2 语句结尾 1.3 行的长度 1.4 换行 1.5 空行 1.6 命名 1.6.1 变量和函数 1.6.2 常量 1.6.3 构造函数 1.7 直接量 1.7.1 字符串 1.7.2 数字 1.7.3 null 1.7.4 undefined 1.7.5 对象直接量 1.7.6 数组直接量 第2章 注释 2.1 单行注释 2.2 多行注释 2.3 使用注释 2.3.1 难于理解的代码 2.3.2 可能被误认为错误的代码 2.3.3 浏览器特性hack 2.4 文档注释 第3章 语句和表达式 3.1 花括号的对齐方式 3.2 块语句间隔 3.3 switch语句 3.3.1 缩进 3.3.2 case语句的“连续执行” 3.3.3 default 3.4 with语句 3.5 for循环 3.6 for-in循环 第4章 变量、函数和运算符 4.1 变量声明 4.2 函数声明 4.3 函数调用间隔 4.4 立即调用的函数 4.5 严格模式 4.6 相等 4.6.1 eval () 4.6.2 原始包装类型 第二部分 编程实践 第5章 UI层的松耦合 5.1 什么是松耦合 5.2 将JavaScript从CSS中抽离 5.3 将CSS从JavaScript中抽离 5.4 将JavaScript从HTML中抽离 5.5 将HTML从JavaScript中抽离 5.5.1 方法1：从服务器加载 5.5.2 方法2：简单客户端模板 5.5.3 方法3：复杂客户端模板 第6章 避免使用全局变量 6.1 全局变量带来的问题 6.1.1 命名冲突 6.1.2 代码的脆弱性 6.1.3 难以测试 6.2 意外的全局变量 避免意外的全局变量 6.3 单全局变量方式 6.3.1 命名空间 6.3.2 模块 6.4 零全局变量 第7章 事件处理 7.1 典型用法 7.2 规则1：隔离应用逻辑 7.3 规则2：不要分发事件对象 第8章 避免“空比较” 8.1 检测原始值 8.2 检测引用值 8.2.1 检测函数 8.2.2 检测数组 8.3 检测属性 第9章 将配置数据从代码中分离出来 9.1 什么是配置数据 9.2 抽离配置数据 9.3 保存配置数据 第10章 抛出自定义错误 10.1 错误的本质 10.2 在JavaScript中抛出错误 10.3 抛出错误的好处 10.4 何时抛出错误 10.5 try-catch语句 10.6 错误类型 第11章 不是你的对象不要动 11.1 什么是你的 11.2 原则 11.2.1 不覆盖方法 11.2.2 不新增方法 11.2.3 不删除方法 11.3 更好的途径 11.3.1 基于对象的继承 11.3.2 基于类型的继承 11.3.3 门面模式 11.4 关于Polyfill的注解 11.5 阻止修改 第12章 浏览器嗅探 12.1 User-Agent检测 12.2 特性检测 12.3 避免特性推断 12.4 避免浏览器推断 12.5 应当如何取舍 第三部分 自动化 第13章 文件和目录结构 13.1 最佳实践 13.2 基本结构 第14章 Ant 14.1 安装 14.2 配置文件 14.3 执行构建 14.4 目标操作的依赖 14.5 属性 14.6 Buildr项目 第15章 校验 15.1 查找文件 15.2 任务 15.3 增强的目标操作 15.4 其他方面的改进 15.5 Buildr任务 第16章 文件合并和加工 16.1 任务 16.2 行尾结束符 16.3 文件头和文件尾 16.4 加工文件 第17章 文件精简和压缩 17.1 文件精简 17.1.1 使用YUI Compressor精简代码 17.1.2 用Closure Compiler精简 17.1.3 使用UglifyJS精简 17.2 压缩 17.2.1 运行时压缩 17.2.2 构建时压缩 第18章 文档化 18.1 JSDoc Toolkit 18.2 YUI Doc 第19章 自动化测试 19.1 YUI Test Selenium引擎 19.1.1 配置一台Selenium服务器 19.1.2 配置YUI Test Selenium引擎 19.1.3 使用YUI Test Selenium引擎 19.1.4 Ant的配置写法 19.2 Yeti 19.3 PhantomJS 19.3.1 安装及使用 19.3.2 Ant的配置写法 19.4 JsTestDriver 19.4.1 安装及使用 19.4.2 Ant的配置写法 第20章 组装到一起 20.1 被忽略的细节 20.2 编制打包计划 20.2.1 开发版本的构建 20.2.2 集成版本的构建 20.2.3 发布版本的构建 20.3 使用CI系统 20.3.1 Jenkins 20.3.2 其他CI系统 附录A JavaScript编码风格指南 附录B JavaScript工具集

<<编写可维护的JavaScript>>

章节摘录

版权页：插图：方法封装了从DOM中删除一个元素的操作，屏蔽了开发者要访问该元素父节点的需求。

从JavaScript的可维护性而言，门面是非常合适的方式，自己可以完全控制这些接口。你可以允许访问任何底层对象的属性或方法，反之亦然，也就是有效地过滤对该对象的访问。你也可以对已有的方法进行改造，使其更加简单易用（上段示例代码就是一个案例）。底层的对象无论如何改变，只要修改门面，应用程序就能继续正常工作。

门面实现一个特定接口，让一个对象看上去像另一个对象，就称作一个适配器。

门面和适配器唯一的不同是前者创建新接口，后者实现已存在的接口。

11.4关于Polyfill的注解 随着ECMAScript 5和HTML 5的特性开始在各种浏览器中的实现，JavaScript polyfills（也称为 shims）变得流行起来了。

一个polyfill是指一种功能的模拟，这些功能在新版本的浏览器中已经有完备定义并原生实现了。

例如，ECMAScript 5为数组增加了forEach（）函数。

该方法可以在ECMAScript 3中模拟，以便在老版本的浏览器中如同新版本一样使用。

polyfills的关键是它们模拟的原生功能要以完全兼容的方式来实现。

因此在有些浏览器中存在着这些功能，所以有必要检测不同情况下它们的处理是否符合标准的方式。

为了达到目的，polyfills经常会给非自己拥有的对象新增一些方法。

我不是polyfills的粉丝，不过对于别人使用它们，我表示理解。

相比其他的对象修改而言，polyfills是有界限的，是相对安全的。

因为原生实现中是存在这些方法并能工作的，有且仅当原生方法不存在时，polyfills才新增这些方法，并且它们和原生版本方法的行为是完全一致的。

polyfills的优点是，当只支持浏览器的原生功能时，它们非常容易删除。

如果你选择使用某个polyfill，你自己做好严格审查。

要保证它的功能和原生的版本尽可能的近似，多检查一下这种库代码有单元测试并严格验证了这些功能。

polyfills的缺点是，它们可能没有精确地实现它们（原生浏览器环境）所缺失的功能，从而给你带来的麻烦比缺失的功能要多得多。

<<编写可维护的JavaScript>>

编辑推荐

《编写可维护的JavaScript》中强调：任何语言都需要强调编码风格的一致性。只要是团队开发，每个人都以相同方式编写代码就是至关重要的。这样大家才能方便地互相看懂和维护对方的代码。运用《编写可维护的JavaScript》中讲述的技巧和技术，可以使JavaScript团队编程从狭义的个人偏好的阴霾走出来，走向真正的高可维护性、高效能和高水准。它是一本构建编码风格手册，帮助开发团队从“游击队”走向“正规军”。

<<编写可维护的JavaScript>>

名人推荐

“ 本书是一本教你写出具有前瞻性的JavaScript代码的完全手册，在团队作战中特别有用。

”——Ryan Grove, Yahoo! YUI工程师“ 作者将他十多年工作经验的精华部分浓缩至这本通俗易懂的书中。

我建议每个开发工程师和在校学生尽早阅读本书。

不管你有多少经验，本书中的每一页内容都会让你变得更加优秀且倍受大家欢迎。

”——Lea Verou, Web设计师兼Web开发人员“ 这是一本前端工程师的指南，指明了在编码过程需要注意的方方面面。

提高可维护性是一个非常大的话题，而这本书，是一个非常不错的起点。

”——王保平（玉伯），支付宝Web前端工程师

<<编写可维护的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>