

编写可维护的JavaScript



[编写可维护的JavaScript_下载链接1](#)

著者:扎卡斯

出版者:人民邮电出版社

出版时间:2013-4

装帧:

isbn:9787115310088

《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。

作者介绍:

Nicholas C.

Zakas是一名前端开发顾问、作者和演讲家。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库代码贡献者。他著有《JavaScript高级程序设计》、《Ajax高级程序设计》和《高性能JavaScript》等书籍。Zakas倡导了很多最佳实践，包括渐进增强、可访问性、性能、扩展性和可维护性等。他的博客地址是：<http://www.nczonline.net/>。他的Twitter是：@slicknet。

李晶，花名拔赤，淘宝前端工程师，具有多年前端开发经验，在团队协作、组件开发、移动Web

App等方面有深入研究，曾经参与淘宝首页、KISSY等项目开发。他翻译过《JavaScript Web富应用开发》、《JavaScript权威指南（第六版）》、《What is Node?》等书籍，热爱分享，喜欢折腾。微博<http://weibo.com/jayli>。

郭凯，花名流火，淘宝前端工程师，喜欢登山，也喜欢夜深人静时一人静静地写代码，折腾过许多小站均未能持久，有In、Juicer等开源项目。钟爱JavaScript，也喜欢Python，自嘲所学杂而不精。博客<http://benben.cc>。

张散集，花名一舟，淘宝前端工程师。他主要从事技术管理，负责淘宝网（北京）的新业务技术和前端团队，热爱前端新技术的推广与应用。翻译作品有《JavaScript Web富应用开发》和《JavaScript权威指南（第六版）》。

目录: 目录

第一部分 编程风格 1

第1章 基本的格式化 4

1.1 缩进层级 4

1.2 语句结尾 7

1.3 行的长度 8

1.4 换行 9

1.5 空行 10

1.6 命名 11

1.6.1 变量和函数 12

1.6.2 常量 13

1.6.3 构造函数 14

1.7 直接量 15

1.7.1 字符串 15

1.7.2 数字 16

1.7.3 null 17

1.7.4 undefined 18

1.7.5 对象直接量 19

1.7.6 数组直接量 20

第2章 注释 21

2.1 单行注释 21

2.2 多行注释 23

2.3 使用注释 24

2.3.1 难于理解的代码	25
2.3.2 可能被误认为错误的代码	26
2.3.3 浏览器特性hack	26
2.4 文档注释	27
第3章 语句和表达式	30
3.1 花括号的对齐方式	31
3.2 块语句间隔	32
3.3 switch语句	33
3.3.1 缩进	33
3.3.2 case语句的“连续执行”	35
3.3.3 default	36
3.4 with语句	37
3.5 for循环	37
3.6 for-in循环	39
第4章 变量、函数和运算符	41
4.1 变量声明	41
4.2 函数声明	44
4.3 函数调用间隔	45
4.4 立即调用的函数	46
4.5 严格模式	47
4.6 相等	49
4.6.1 eval()	51
4.6.2 原始包装类型	52
第二部分 编程实践	54
第5章 UI层的松耦合	55
5.1 什么是松耦合	56
5.2 将JavaScript从CSS中抽离	57
5.3 将CSS从JavaScript中抽离	58
5.4 将JavaScript从HTML中抽离	60
5.5 将HTML从JavaScript中抽离	62
5.5.1 方法1：从服务器加载	63
5.5.2 方法2：简单客户端模板	64
5.5.3 方法3：复杂客户端模板	67
第6章 避免使用全局变量	70
6.1 全局变量带来的问题	70
6.1.1 命名冲突	71
6.1.2 代码的脆弱性	71
6.1.3 难以测试	72
6.2 意外的全局变量	72
避免意外的全局变量	73
6.3 单全局变量方式	74
6.3.1 命名空间	76
6.3.2 模块	78
6.4 零全局变量	81
第7章 事件处理	83
7.1 典型用法	83
7.2 规则1：隔离应用逻辑	84
7.3 规则2：不要分发事件对象	85
第8章 避免“空比较”	88
8.1 检测原始值	88
8.2 检测引用值	90
8.2.1 检测函数	92
8.2.2 检测数组	94
8.3 检测属性	95

- 第9章 将配置数据从代码中分离出来 98
 - 9.1 什么是配置数据 98
 - 9.2 抽离配置数据 99
 - 9.3 保存配置数据 100
- 第10章 抛出自定义错误 103
 - 10.1 错误的本质 103
 - 10.2 在JavaScript中抛出错误 104
 - 10.3 抛出错误的好处 105
 - 10.4 何时抛出错误 106
 - 10.5 try-catch语句 107
 - 10.6 错误类型 109
- 第11章 不是你的对象不要动 112
 - 11.1 什么是你的 112
 - 11.2 原则 113
 - 11.2.1 不覆盖方法 113
 - 11.2.2 不新增方法 114
 - 11.2.3 不删除方法 116
 - 11.3 更好的途径 117
 - 11.3.1 基于对象的继承 118
 - 11.3.2 基于类型的继承 119
 - 11.3.3 门面模式 120
 - 11.4 关于Polyfill的注解 121
 - 11.5 阻止修改 122
- 第12章 浏览器嗅探 125
 - 12.1 User-Agent检测 125
 - 12.2 特性检测 127
 - 12.3 避免特性推断 129
 - 12.4 避免浏览器推断 130
 - 12.5 应当如何取舍 134
- 第三部分 自动化 135
- 第13章 文件和目录结构 137
 - 13.1 最佳实践 137
 - 13.2 基本结构 138
- 第14章 Ant 143
 - 14.1 安装 143
 - 14.2 配置文件 143
 - 14.3 执行构建 145
 - 14.4 目标操作的依赖 145
 - 14.5 属性 146
 - 14.6 Buildr项目 148
- 第15章 校验 149
 - 15.1 查找文件 149
 - 15.2 任务 150
 - 15.3 增强的目标操作 152
 - 15.4 其他方面的改进 153
 - 15.5 Buildr任务 154
- 第16章 文件合并和加工 156
 - 16.1 任务 156
 - 16.2 行尾结束符 157
 - 16.3 文件头和文件尾 158
 - 16.4 加工文件 159
- 第17章 文件精简和压缩 163
 - 17.1 文件精简 163
 - 17.1.1 使用YUI Compressor精简代码 165

- 17.1.2 用Closure Compiler精简 167
- 17.1.3 使用UglifyJS精简 169
- 17.2 压缩 170
 - 17.2.1 运行时压缩 171
 - 17.2.2 构建时压缩 171
- 第18章 文档化 175
 - 18.1 JSDoc Toolkit 175
 - 18.2 YUI Doc 177
- 第19章 自动化测试 180
 - 19.1 YUI Test Selenium引擎 180
 - 19.1.1 配置一台Selenium服务器 181
 - 19.1.2 配置YUI Test Selenium引擎 181
 - 19.1.3 使用YUI Test Selenium引擎 181
 - 19.1.4 Ant的配置写法 183
 - 19.2 Yeti 184
 - 19.3 PhantomJS 186
 - 19.3.1 安装及使用 186
 - 19.3.2 Ant的配置写法 187
 - 19.4 JsTestDriver 188
 - 19.4.1 安装及使用 188
 - 19.4.2 Ant的配置写法 189
- 第20章 组装到一起 191
 - 20.1 被忽略的细节 191
 - 20.2 编制打包计划 192
 - 20.2.1 开发版本的构建 193
 - 20.2.2 集成版本的构建 194
 - 20.2.3 发布版本的构建 195
 - 20.3 使用CI系统 196
 - 20.3.1 Jenkins 196
 - 20.3.2 其他CI系统 199
- 附录A JavaScript编码风格指南 200
- 附录B JavaScript工具集 223
 - • • • • (收起)

[编写可维护的JavaScript 下载链接1](#)

标签

JavaScript

前端

Web开发

Web前端开发

前端开发

javascript

编程

计算机

评论

我相信对于大部分Web开发人员来说，JS写的比后端代码都多，至少我是这样的，但是讲JS规范的书却少了又少，以至于我大部分都是跟着自己的感觉去写，也不知道怎么好怎么不好。这本书算是我这种野生程序员的一个福音吧，当然如果真的想写好JS个人奉劝再读读Javascript设计模式。

防御式编程javascript版...当然，其实还是工程化javascript编程实践版...不过...这封面好山寨...

适合初级向中级进阶。书中前两部分都非常不错，值得每一个业余JavaScript程序员学习规范自己的开发。第三部分略有过时，读读也不错。目前用的比较多的应该是Grunt、gulp和webpack这些吧。不过类似的前端工具发展太快了。

如果你不经常Code Review那就应该看看了.

精炼

#纸质书# 3.7星，各种标准这种之下矫正 JavaScript 书写坏习惯。

内容比较浅显，但都是金玉良言。不是大而全的书，所以适合打好基础，不是用来攻城略地的。

不够深入啊亲。。

薄薄的一本，可以作为一个最佳实践的参考手册来用。

厕所必备读物啊。。。

高估了这本书，大概是因为平时被规范洗脑惯了，才出现了这种想法吧。
仔细想想，对于从未有过此类约束的人来说，看到这本书的内容也许会喘不过气来。
PS， 我老喜欢eval 老喜欢修改原有的函数和属性，这些毛病是得改改了

NCZ的作品好多～

书内容不错，翻译的一般。代码错误、语言不通等。

额。。。干货略少

编程实践部分不错

强推JavaScript高级程序设计一书，作者都是Zakas，这本书更像是高级程序设计的一个小的子集，主要内容是编码规范等，收获不大。

内容实用，对编写可维护的代码很有帮助

看了一下目录 D

第一部分不超过《the good part》的范围，全书最有价值是第二部分，自动化那部分我基本草草翻过，全是基于Java环境的……

蝴蝶书的延续，更富实践性的编写指南

[编写可维护的JavaScript 下载链接1](#)

书评

我是一个野生JavaScript程序员，我相信大多数JavaScript程序员都是野蛮生长，靠的是自身的好奇和勤勉。
什么是野生JavaScript程序员？我是这样定义的：基本靠自学；通过不停的`console.log`和对比来写出凑合能运行的页面；页面上充斥着全局变量；命名风格随心所欲，受到自学的...

如此之薄的一本书，第一版次整本读下来，如果很细心的话你会发现有n多个错误，责任编辑，难道你是边打飞机边工作的么？我只能说这尼玛太没有职业道德了！这实在太符合国产的现状了！
关于翻译，有兴趣的可以看看《编码的奥秘》，呵呵俩字已经代表我的态度！
挣钱——>不择手...

zakas推荐给大家的一个javascript开发规范文档。
从代码风格到一些开发技巧，如何避免一些糟糕的情况，到目录管理，还有持续集成部署。一本全面的规范类书籍！
当然zakas也有讲到一些深层的知识，还有ECMAScript5的一些新api，要搞html5开发的朋友更需要看这本书。你能从...

本书的第二部分是精华。倒没太多新东西，之前的《高性能JS》《JS高级程序设计》以及《JS模式》大部分都提到过，这里算是个总结。
让人很不爽的一点是：原书中的“提示建议”，是以“爪印”图标和不同的样式跟正文区分开的；中文版中图标和样式都被干掉了，跟正文混在一起。奇...

“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。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
“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。” ...

13年的技术 现在来看稍微有点out，不过查缺补漏吧，有点用处。
不过现在有eslint,jslint等检查工具，可以很完美的同一代码规范，看得多了就知道什么格式了。
可以肯一看了解一下前端需要使用那些技术，之后再去看现在比较流行的框架就好了。
茶余饭后看一看就好了。当时只顾得...

Nicholas的每一本书都对得起深入浅出四个字，前面的语法部分可说是一气呵成，例释并举，让人印象深刻。后面的工具部分，稍微有点拖沓和杂乱，不过也指明了解决问题的实用方案。从页数和价格上看，有点小贵，不过还是值了。附录A是干货，可保留电子版随时备查。

书可以给4星。主要分为3部分，分别是：编程风格，编程实践，自动化。
编程风格,基本语法的规范，这里如果你有系统的学习，应该都不会有大问题。
编程实践，这里学到了很多的东西，虽然每一个章节都比较简略，但是很多js的关键点都

有涉及。 自动化，因为本书是2013年出的，大量的...

书主要在三方面给读者写JS程序的指导，一是代码的格式和风格指导，二是代码的实现方式手段的指导，三是项目自动化管理的指导。
有很多内容都是之前看过的了，所以花了几个小时的时间过了一遍。总体上书的废话不多，如果以前没读过此类的指导书籍，相信收获会不少。另外读这本...

让我从浑浑噩噩的js代码规范中拨开云雾见青天。里面的各种js代码规范不仅仅让我对js这一门语言有了清醒的认识，对其他语言亦是如此。
本身虽然简短，但是确实是浓缩的精华。 好书，推荐！

写程序最重要的是思路清晰。
HTML+CSS+Javascript跟其他面向对象开发的语言区别很大，灵活 所以容易 凌乱
本书第二部分讲了很多基本原则，实现解耦。
看完本书，再看一些模块化开发的资料，就开始具备JS的内功。

翻译的挺不错的，我是自己买纸质书的，虽然用了很少时间看完了，但是这本书对于刚刚入门js的童鞋还是不错的，对于规范代码和编写高质量的代码还是特别有建议。但是在生产实践中我还是喜欢用模块的方式来管理代码，不过这本书里没有提及。是一个遗憾哟。

买了绝对后悔的书 漏洞百出 狗屁不通 越看越生气 没有用心翻译和审查
(这评论还短.....我觉得这就够了 翻译团队实在对不住原作
请高抬贵手 金盆洗手)

在学校学的是 C 和 Java，09年低开始在一家小公司做金融行业 J2EE 开发，2011年5月份跳槽到一家互联网公司的一个部门里。那年7月，开始转职做前端（原来那个前端兄弟跑了，我的黑色7、8月啊），然后一直到现在。
从开始的一个小白，到后来培养出几个 Java 实习生做前端，这两...

前几天同事刚推荐的这本“乌龟书”说是牛逼人物写的，大致翻了一遍，如中国足球一样粗糙的翻译是这本书中文版最大的败笔，翻译的太随便了，很多词不达意或不到位的地方，也不知道是译者英文水平的问题还是。。。

[编写可维护的JavaScript_下载链接1](#)