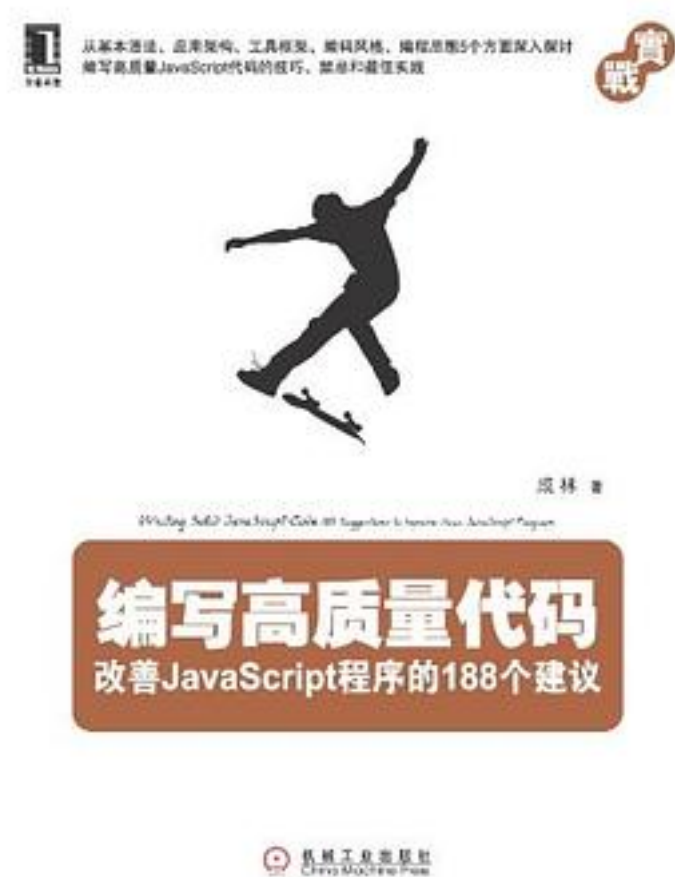


编写高质量代码



[编写高质量代码_下载链接1](#)

著者:成林

出版者:机械工业出版社

出版时间:2012-11

装帧:

isbn:9787111399056

《编写高质量代码:改善JavaScript程序的188个建议》是Web前端工程师进阶修炼的必读之作，将为你通往“JavaScript技术殿堂”指点迷津！内容全部由编写高质量的JavaScript代码的最佳实践组成，从基本语法、应用架构、工具框架、编码风格、编程思想

等5大方面对Web前端工程师遇到的疑难问题给出了经验性的解决方案，为Web前端工程师如何编写更高质量的JavaScript代码提供了188条极为宝贵的建议。对于每一个问题，不仅以建议的方式给出了被实践证明为十分优秀的解决方案，而且还给出了经常被误用或被错误理解的不好的解决方案，从正反两个方面进行了分析和对比，犹如醍醐灌顶，让人豁然开朗。

《编写高质量代码:改善JavaScript程序的188个建议》针对每个问题所设计的应用场景都非常典型，给出的建议也都与实践紧密结合。每一条建议都可能在你的下一行代码、下一个应用或下一个项目中被用到，建议你将《编写高质量代码:改善JavaScript程序的188个建议》放置在手边，随时查阅，一定能使你的学习和开发工作事半功倍。

作者介绍:

目录: 前言

第1章 JavaScript语言基础 / 1

建议1: 警惕Unicode乱码 / 1

建议2: 正确辨析JavaScript句法中的词、句和段 / 2

建议3: 减少全局变量污染 / 4

建议4: 注意JavaScript数据类型的特殊性 / 6

建议5: 防止JavaScript自动插入分号 / 11

建议6: 正确处理JavaScript特殊值 / 12

建议7: 小心保留字的误用 / 15

建议8: 谨慎使用运算符 / 16

建议9: 不要信任hasOwnProperty / 20

建议10: 谨记对象非空特性 / 20

建议11: 慎重使用伪数组 / 21

建议12: 避免使用with / 22

建议13: 养成优化表达式的思维方式 / 23

建议14: 不要滥用eval / 26

建议15: 避免使用continue / 27

建议16: 防止switch贯穿 / 28

建议17: 块标志并非多余 / 29

建议18: 比较function语句和function表达式 / 29

建议19: 不要使用类型构造器 / 30

建议20: 不要使用new / 31

建议21: 推荐提高循环性能的策略 / 31

建议22: 少用函数迭代 / 35

建议23: 推荐提高条件性能的策略 / 35

建议24: 优化if逻辑 / 36

建议25: 恰当选用if和switch / 39

建议26: 小心if嵌套的思维陷阱 / 40

建议27: 小心if隐藏的Bug / 42

建议28: 使用查表法提高条件检测的性能 / 43

建议29: 准确使用循环体 / 44

建议30: 使用递归模式 / 48

建议31: 使用迭代 / 49

建议32: 使用制表 / 50

建议33: 优化循环结构 / 51

第2章 字符串、正则表达式和数组 / 53

建议34: 字符串是非值操作 / 53

建议35: 获取字节长度 / 55

建议36: 警惕字符串连接操作 / 56

建议37: 推荐使用replace / 59
建议38: 正确认识正则表达式工作机制 / 62
建议39: 正确理解正则表达式回溯 / 63
建议40: 正确使用正则表达式分组 / 65
建议41: 正确使用正则表达式引用 / 68
建议42: 用好正则表达式静态值 / 69
建议43: 使用exec增强正则表达式功能 / 71
建议44: 正确使用原子组 / 72
建议45: 警惕嵌套量词和回溯失控 / 73
建议46: 提高正则表达式执行效率 / 74
建议47: 避免使用正则表达式的场景 / 76
建议48: 慎用正则表达式修剪字符串 / 77
建议49: 比较数组与对象同源特性 / 80
建议50: 正确检测数组类型 / 81
建议51: 理解数组长度的有限性和无限性 / 82
建议52: 建议使用splice删除数组 / 83
建议53: 小心使用数组维度 / 84
建议54: 增强数组排序的sort功能 / 85
建议55: 不要拘泥于数字下标 / 87
建议56: 使用arguments模拟重载 / 89
第3章 函数式编程 / 91
建议57: 禁用Function构造函数 / 91
建议58: 灵活使用Arguments / 94
建议59: 推荐动态调用函数 / 96
建议60: 比较函数调用模式 / 99
建议61: 使用闭包跨域开发 / 101
建议62: 在循环体和异步回调中慎重使用闭包 / 104
建议63: 比较函数调用和引用本质 / 106
建议64: 建议通过Function扩展类型 / 108
建议65: 比较函数的惰性求值与非惰性求值 / 109
建议66: 使用函数实现历史记录 / 111
建议67: 套用函数 / 113
建议68: 推荐使用链式语法 / 114
建议69: 使用模块化规避缺陷 / 115
建议70: 惰性实例化 / 117
建议71: 推荐分支函数 / 118
建议72: 惰性载入函数 / 119
建议73: 函数绑定有价值 / 121
建议74: 使用高阶函数 / 123
建议75: 函数柯里化 / 125
建议76: 要重视函数节流 / 126
建议77: 推荐作用域安全的构造函数 / 127
建议78: 正确理解执行上下文和作用域链 / 129
第4章 面向对象编程 / 133
建议79: 参照Object构造体系分析prototype机制 / 133
建议80: 合理使用原型 / 137
建议81: 原型域链不是作用域链 / 140
建议82: 不要直接检索对象属性值 / 142
建议83: 使用原型委托 / 143
建议84: 防止原型反射 / 144
建议85: 谨慎处理对象的Scope / 145
建议86: 使用面向对象模拟继承 / 149
建议87: 分辨this和function调用关系 / 152
建议88: this是动态指针, 不是静态引用 / 153

建议89: 正确应用this / 157
建议90: 预防this误用的策略 / 161
建议91: 推荐使用构造函数原型模式定义类 / 164
建议92: 不建议使用原型继承 / 166
建议93: 推荐使用类继承 / 168
建议94: 建议使用封装类继承 / 171
建议95: 慎重使用实例继承 / 172
建议96: 避免使用复制继承 / 174
建议97: 推荐使用混合继承 / 175
建议98: 比较使用JavaScript多态、重载和覆盖 / 176
建议99: 建议主动封装类 / 179
建议100: 谨慎使用类的静态成员 / 181
建议101: 比较类的构造和析构特性 / 183
建议102: 使用享元类 / 186
建议103: 使用掺元类 / 188
建议104: 谨慎使用伪类 / 190
建议105: 比较单例的两种模式 / 192
第5章 DOM编程 / 195
建议106: 建议先检测浏览器对DOM支持程度 / 195
建议107: 应理清HTML DOM加载流程 / 198
建议108: 谨慎访问DOM / 200
建议109: 比较innerHTML与标准DOM方法 / 200
建议110: 警惕文档遍历中的空格Bug / 202
建议111: 克隆节点比创建节点更好 / 203
建议112: 谨慎使用HTML集合 / 204
建议113: 用局部变量访问集合元素 / 206
建议114: 使用nextSibling抓取DOM / 207
建议115: 实现DOM原型继承机制 / 207
建议116: 推荐使用CSS选择器 / 210
建议117: 减少DOM重绘和重排版次数 / 211
建议118: 使用DOM树结构托管事件 / 216
建议119: 使用定时器优化UI 队列 / 217
建议120: 使用定时器分解任务 / 220
建议121: 使用定时器限时运行代码 / 221
建议122: 推荐网页工人线程 / 222
第6章 客户端编程 / 226
建议123: 比较IE和W3C事件流 / 226
建议124: 设计鼠标拖放方案 / 229
建议125: 设计鼠标指针定位方案 / 231
建议126: 小心在元素内定位鼠标指针 / 233
建议127: 妥善使用DOMContentLoaded事件 / 234
建议128: 推荐使用beforeunload事件 / 236
建议129: 自定义事件 / 236
建议130: 从CSS样式表中抽取元素尺寸 / 238
建议131: 慎重使用offsetWidth和offsetHeight / 241
建议132: 正确计算区域大小 / 244
建议133: 谨慎计算滚动区域大小 / 247
建议134: 避免计算窗口大小 / 248
建议135: 正确获取绝对位置 / 249
建议136: 正确获取相对位置 / 251
第7章 数据交互和存储 / 254
建议137: 使用隐藏框架实现异步通信 / 254
建议138: 使用iframe实现异步通信 / 257
建议139: 使用script实现异步通信 / 259

建议140: 正确理解JSONP异步通信协议 / 264
建议141: 比较常用的服务器请求方法 / 267
建议142: 比较常用的服务器发送数据方法 / 271
建议143: 避免使用XML格式进行通信 / 273
建议144: 推荐使用JSON格式进行通信 / 275
建议145: 慎重使用HTML格式进行通信 / 278
建议146: 使用自定义格式进行通信 / 279
建议147: Ajax性能向导 / 280
建议148: 使用本地存储数据 / 281
建议149: 警惕基于DOM的跨域侵入 / 283
建议150: 优化Ajax开发的最佳实践 / 286
建议151: 数据存储要考虑访问速度 / 290
建议152: 使用局部变量存储数据 / 291
建议153: 警惕人为改变作用域链 / 293
建议154: 慎重使用动态作用域 / 294
建议155: 小心闭包导致内存泄漏 / 295
建议156: 灵活使用Cookie存储长信息 / 296
建议157: 推荐封装Cookie应用接口 / 298
第8章 JavaScript引擎与兼容性 / 300
建议158: 比较主流浏览器内核解析 / 300
建议159: 推荐根据浏览器特性进行检测 / 302
建议160: 关注各种引擎对ECMAScript v3的分歧 / 305
建议161: 关注各种引擎对ECMAScript v3的补充 / 316
建议162: 关注各种引擎对Event解析的分歧 / 327
建议163: 关注各种引擎对DOM解析的分歧 / 330
建议164: 关注各种引擎对CSS渲染的分歧 / 335
第9章 JavaScript编程规范和应用 / 339
建议165: 不要混淆JavaScript与浏览器 / 339
建议166: 掌握JavaScript预编译过程 / 340
建议167: 准确分析JavaScript执行顺序 / 344
建议168: 避免二次评估 / 350
建议169: 建议使用直接量 / 351
建议170: 不要让JavaScript引擎重复工作 / 351
建议171: 使用位操作符执行逻辑运算 / 353
建议172: 推荐使用原生方法 / 355
建议173: 编写无阻塞JavaScript脚本 / 356
建议174: 使脚本延迟执行 / 358
建议175: 使用XHR脚本注入 / 362
建议176: 推荐最优化非阻塞模式 / 362
建议177: 避免深陷作用域访问 / 363
建议178: 推荐的JavaScript性能调优 / 365
建议179: 减少DOM操作中的Repaint和Reflow / 368
建议180: 提高DOM访问效率 / 370
建议181: 使用 setTimeout实现工作线程 / 372
建议182: 使用 Web Worker / 375
建议183: 避免内存泄漏 / 377
建议184: 使用SVG创建动态图形 / 380
建议185: 减少对象成员访问 / 385
建议186: 推荐100 ms用户体验 / 388
建议187: 使用接口解决JavaScript文件冲突 / 390
建议188: 避免JavaScript与CSS冲突 / 392

• • • • • ([收起](#))

标签

JavaScript

前端开发

JavaScript

javascript

高质量代码

编程

编写高质量代码

计算机

评论

挺好的一本书，很多技巧，很实用也是我未曾考虑过的，适合熟悉 JavaScript，想进一步深入的读者，不过我读起来还有很多地方比较费力，有空还得读。

确实有些说的很牵强，抄袭也好严重呐，但是有些总结的不错。

以后少读这样的碎片书，读也是快速的读。看的迷迷糊糊，原型、多种继承等等，或许自己JavaScript的基本功还是不行，只会一些基本的运用，一些语法习惯没有去深究背后的原理。

写过一年多的js，看了这本书才知道自己多弱

有些比较牵强，有些说得挺好的。我还发现有一章节是直接翻译一篇外文的。真是伤不起。还是去买图灵引进的外文书，质量有保证些。

“实在”得像是某个前端er的记事本儿,且包括肉眼可见的bug

很不错

系列丛书

挺不错的书，值得多看几遍

抄袭严重

吐槽一万遍，买的最后悔的一本，没有之一，大量篇幅抄袭《高性能javascript》，我记得还有部分抄袭《javascript精粹》，《javascript模式》有没有抄我就不记得了，内容章节安排也很不合理。

先看的《javascript精粹》，后来发现这本书和那本有惊人的相似，不过涵盖得更多。

抄袭多本外国技术书籍,如 javascript语言精粹、编写可维护的 javascript等，而且是大段大段地粘贴复制，搞不懂为什么有 4 星的评分。

帮朋友找学习资料，翻阅了本书大概一半，不建议同学们去读这本书，很多建议很牵强或有错误，印象最深的是建议80:合理使用原型中的四个例子，其中例1例2例4都是很牵强，例2乱改原型是个大坑，这些都将一些从已入门想进阶的同学们误导。

例如 小心if隐藏的Bug 都不是bug根本只是马虎造成的小细节而已 也不详细讲解为什么就是不行不要用就完事了

[编写高质量代码_下载链接1](#)

书评

看完javascript语言精粹之后，看得这本书，刚开始翻几页发现很熟悉，后来把两本书对比一下，几乎完全是抄袭，只是有些地方措辞稍微变了一下，于是我就呵呵了，网上搜了一下作者，呵呵。。。找不到人，不是著名的web前端工程师，你就不怕别人告你侵权么，就是因为存在这种人，不...

[编写高质量代码_下载链接1](#)