

《实战 ES2015》

图书基本信息

书名：《实战 ES2015》

13位ISBN编号：9787121300184

出版时间：2016-10-1

作者：小问

页数：275

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu111.com

内容概要

在ES2015标准中的表现，以及利用ES2015中新特性在应用开发中的实践。以清晰的思路说明ES2015的详细特性和意义，并以实际案例展示利用ES2015中的特性如何提高应用的前端和后端的开发速度和工程化模式

作者简介

小问，本名甘超阳，数学系理学学士，高级JavaScript工程师，国内最早的Node.js研究者之一，技术布道者、讲师，参与并维护多个优秀JavaScript开源项目。对多方面技术领域都有研究和自己的理解，熟悉各类JavaScript应用的开发和机器学习应用开发。同时也是一个业余摄影师，热爱人像摄影，个人博客<http://lifemap.in>

书籍目录

- 第1章 ECMAScript版本发展历程 1
 - 1.1 ECMAScript的历史更替 2
 - 1.1.1 ECMA-262 / ECMA-262 Edition 2 2
 - 1.1.2 ECMA-262 Edition 3 3
 - 1.1.3 ECMA-262 Edition 5 4
 - 1.1.4 ECMA-262 Edition 4 4
 - 1.2 小结 5
- 第2章 ES2015能为实际开发带来什么 6
 - 2.1 简单概括ES2015 6
 - 2.2 语法糖 7
 - 2.3 工程优势 8
 - 2.3.1 模块化 8
 - 2.3.2 模块化与组件化结合 11
 - 2.3.3 内存安全 13
 - 2.4 小结 14
- 第3章 ES2015新语法详解 15
 - 3.1 let、const和块级作用域 16
 - 3.1.1 块级作用域 16
 - 3.1.2 let定义变量 20
 - 3.1.3 const定义常量 22
 - 3.1.3.1 使用语法 23
 - 3.1.3.2 const与块级作用域 25
 - 3.1.4 变量的生命周期 27
 - 3.1.5 更佳体验 29
 - 3.1.5.1 let VS const 29
 - 3.1.5.2 let、const与循环语句 30
 - 3.2 箭头函数 (Arrow Function) 31
 - 3.2.1 使用语法 31
 - 3.2.1.1 单一参数的单行箭头函数 31
 - 3.2.1.2 多参数的单行箭头函数 31
 - 3.2.1.3 多行箭头函数 32
 - 3.2.1.4 无参数箭头函数 32
 - 3.2.2 this穿透 33
 - 3.2.2.1 程序逻辑注意事项 34
 - 3.2.2.2 编写语法注意事项 36
 - 3.3 模板字符串 (Template String) 37
 - 3.3.1 字符串元素注入 37
 - 3.3.2 多行字符串 37
 - 3.3.3 模板字符串使用语法 38
 - 3.3.3.1 支持字符串元素注入 38
 - 3.3.3.2 支持换行 39
 - 3.3.4 注意事项 41
 - 3.4 对象字面量扩展语法 (Enhanced Object Literals) 41
 - 3.4.1 函数类属性的省略语法 41
 - 3.4.2 支持 __proto__ 注入 42
 - 3.4.3 可动态计算的属性名 43
 - 3.4.4 将属性名定义省略 44

- 3.5 表达式结构 (Destructuring) 45
 - 3.5.1 使用语法 47
 - 3.5.1.1 使用对象作为返回载体 (带有标签的多返回值) 47
 - 3.5.1.2 使用数组作为返回载体 47
 - 3.5.2 使用场景 48
 - 3.5.2.1 Promise与模式匹配 48
 - 3.5.2.2 Swap (变量值交换) 49
 - 3.5.3 用法 49
 - 3.5.3.1 解构别名 50
 - 3.5.3.2 无法匹配的缺省值 50
 - 3.5.3.3 深层匹配 50
 - 3.5.3.4 配合其他新特性 52
- 3.6 函数参数表达、传参 53
 - 3.6.1 默认参数值 54
 - 3.6.1.1 使用语法 54
 - 3.6.1.2 使用场景 54
 - 3.6.2 剩余参数 55
 - 3.6.2.1 使用语法 56
 - 3.6.2.2 使用场景 57
 - 3.6.2.3 注意事项 57
 - 3.6.3 解构传参 58
- 3.7 新的数据结构 59
 - 3.7.1 Set有序集合 59
 - 3.7.1.1 使用语法 60
 - 3.7.1.2 增减元素 61
 - 3.7.1.3 检查元素 61
 - 3.7.1.4 历遍元素 61
 - 3.7.2 WeakSet 62
 - 3.7.3 Map映射类型 64
 - 3.7.3.1 使用语法 64
 - 3.7.3.2 增减键值对 65
 - 3.7.3.3 获取键值对 66
 - 3.7.3.4 检查映射对象中的键值对 66
 - 3.7.3.5 历遍映射对象中的键值对 66
 - 3.7.3.6 映射对象与Object的区别 67
 - 3.7.4 WeakMap 67
- 3.8 类语法 (Classes) 68
 - 3.8.1 使用语法 69
 - 3.8.1.1 基本定义语法 69
 - 3.8.1.2 继承语法 70
 - 3.8.1.3 Getter/Setter 72
 - 3.8.1.4 静态方法 73
 - 3.8.1.5 技巧 77
 - 3.8.2 注意事项 78
 - 3.8.3 遗憾与期望 79
- 3.9 生成器 (Generator) 80
 - 3.9.1 由来 80
 - 3.9.2 基本概念 81
 - 3.9.2.1 生成器函数 (Generator Function) 81

- 3.9.2.2 生成器 (Generator) 82
- 3.9.3 使用方法 83
 - 3.9.3.1 构建生成器函数 83
 - 3.9.3.2 启动生成器 83
 - 3.9.3.3 运行生成器内容 84
- 3.9.4 深入理解 85
 - 3.9.4.1 运行模式 85
 - 3.9.4.2 生成器函数以及生成器对象的检测 85
 - 3.9.4.3 生成器嵌套 88
 - 3.9.4.4 生成器与协程 90
- 3.10 Promise 93
 - 3.10.1 基本语法 94
 - 3.10.1.1 创建Promise对象 94
 - 3.10.1.2 进行异步操作 94
 - 3.10.1.3 处理Promise的状态 95
 - 3.10.2 使用方法 97
 - 3.10.2.1 Promise.all(iterable) 97
 - 3.10.2.2 Promise.race(iterable) 97
- 3.11 代码模块化 98
 - 3.11.1 引入模块 99
 - 3.11.1.1 引入默认模块 99
 - 3.11.1.2 引入模块部分接口 100
 - 3.11.1.3 引入全部局部接口到指定命名空间 101
 - 3.11.1.4 混入引入默认接口和命名接口 101
 - 3.11.1.5 不引入接口，仅运行模块代码 102
 - 3.11.2 定义模块 102
 - 3.11.3 暴露模块 103
 - 3.11.3.1 暴露单一接口 103
 - 3.11.3.2 暴露模块默认接口 104
 - 3.11.3.3 混合使用暴露接口语句 104
 - 3.11.3.4 从其他模块暴露接口 105
 - 3.11.3.5 暴露一个模块的所有接口 105
 - 3.11.3.6 暴露一个模块的部分接口 106
 - 3.11.3.7 暴露一个模块的默认接口 106
- 3.12 Symbol 106
 - 3.12.1 基本语法 107
 - 3.12.1.1 生成的Symbol值 107
 - 3.12.1.2 注册全局可重用 Symbol 108
 - 3.12.1.3 获取全局 Symbol的key 109
 - 3.12.2 常用Symbol值 109
 - 3.12.3 Symbol.iterator 110
 - 3.12.3.1 for-of循环语句与可迭代对象 111
 - 3.12.3.2 使用Symbol.iterator定义一个可迭代对象 111
 - 3.12.4 Symbol.hasInstance 113
 - 3.12.5 Symbol.match 113
 - 3.12.6 Symbol.unscopables 114
 - 3.12.7 Symbol.toPrimitive 115
 - 3.12.8 Symbol.toStringTag 116
- 3.13 Proxy 117

- 3.13.1 元编程 117
- 3.13.2 使用语法 118
- 3.13.3 handler.has 119
- 3.13.4 handler.get 120
- 3.13.5 handler.set 121
- 3.13.6 handler.apply 122
- 3.13.7 handler.construct 122
- 3.13.8 创建可解除Proxy对象 123
- 3.13.9 使用场景 123
 - 3.13.9.1 看似“不可能”的自动填充 123
 - 3.13.9.2 只读视图 124
 - 3.13.9.3 入侵式测试框架 125
- 3.14 小结 127
- 第4章 ES2015的前端开发实战 129
 - 4.1 Filmy的功能规划 129
 - 4.1.1 数据分级 130
 - 4.1.2 数据结构 130
 - 4.1.2.1 核心数据 130
 - 4.1.2.2 分类数据 131
 - 4.1.2.3 相册数据 131
 - 4.1.3 数据搜索 132
 - 4.1.3.1 搜索分类 132
 - 4.1.3.2 搜索相册 132
 - 4.1.4 界面原型规划 133
 - 4.1.4.1 着陆页面 133
 - 4.1.4.2 分类目录页面 133
 - 4.1.4.3 分类内容页面 134
 - 4.1.4.4 相册页面 134
 - 4.2 功能组件分割 135
 - 4.2.1 根组件分割 135
 - 4.2.2 着陆页面 136
 - 4.2.3 目录页面 136
 - 4.2.4 分类页面 137
 - 4.2.5 相册页面 137
 - 4.3 技术选型 139
 - 4.3.1 整体架构 139
 - 4.3.2 数据层 139
 - 4.3.3 逻辑层及UI层 140
 - 4.3.3.1 AngularJS 141
 - 4.3.3.2 React.js 141
 - 4.3.3.3 Vue.js 142
 - 4.3.4 程序架构 143
 - 4.3.4.1 路由组件 143
 - 4.3.4.2 数据组件 144
 - 4.3.4.3 视图组件 145
 - 4.4 数据层开发 146
 - 4.4.1 安装依赖 147
 - 4.4.2 配置七牛 SDK 147
 - 4.4.3 核心配置数据 150

- 4.4.3.1 获取核心配置数据 150
- 4.4.3.2 更新配置数据 151
- 4.4.4 分类数据 154
 - 4.4.4.1 数据结构 155
 - 4.4.4.2 数据索引 157
 - 4.4.4.3 更新分类数据 159
- 4.4.5 相册数据 159
 - 4.4.5.1 数据加载 160
 - 4.4.5.2 数据更新 161
 - 4.4.5.3 数据检索 161
- 4.5 入口文件与路由组件开发 165
 - 4.5.1 路由基础组件 166
 - 4.5.2 入口文件 166
 - 4.5.2.1 简单的字符串处理 167
 - 4.5.2.2 多国语言处理 168
- 4.6 着陆页面开发 170
 - 4.6.1 路由组件开发 171
 - 4.6.2 着陆页视图 174
 - 4.6.2.1 引入数据 174
 - 4.6.2.2 绑定视图 174
 - 4.6.3 分类目录视图 177
 - 4.6.3.1 分类元素视图组件 177
 - 4.6.3.2 渲染分类目录 178
 - 4.6.4 路由组件、视图组件与数据组件的联系 180
 - 4.6.4.1 在逻辑控制器中进行数据操作 180
 - 4.6.4.2 在视图中进行数据操作 181
 - 4.6.4.3 组织方式的区别与项目应用 182
- 4.7 分类页面开发 182
 - 4.7.1 路由组件开发 183
 - 4.7.2 分类元素视图组件 185
 - 4.7.3 相册列表视图组件 186
 - 4.7.4 相册页面开发 188
 - 4.7.4.1 相册页面的路由组件 188
 - 4.7.4.2 相册信息视图组件 189
 - 4.7.4.3 图片列表视图组件 189
- 4.8 管理工具开发 190
- 4.9 初始化Filmy实例 191
 - 4.9.1 基本元素 192
 - 4.9.2 基本逻辑 194
 - 4.9.2.1 获取七牛云的上传凭证 195
 - 4.9.2.2 检查并处理管理员对背景图片的填写方式 196
 - 4.9.2.3 将核心数据部署到七牛云上 197
- 4.10 管理工具的总体组织 197
 - 4.10.1 管理页面的总体架构 198
 - 4.10.2 侧边栏 198
 - 4.10.3 路由配置 200
- 4.11 相册发布页面 202
 - 4.11.1 基本逻辑 202
 - 4.11.1.1 绑定数据 202

- 4.11.1.2 绑定元素以接收文件上传 203
- 4.11.2 上传数据 204
 - 4.11.2.1 图片上传逻辑 204
 - 4.11.2.2 数据提交 205
- 4.12 打包发布 206
 - 4.12.1 准备工作 206
 - 4.12.2 配置 webpack 207
 - 4.12.2.1 安装依赖 207
 - 4.12.2.2 编写配置文件 207
 - 4.12.3 发布到云端 209
- 4.13 小结 210
- 第5章 ES2015的Node.js开发实战 211
 - 5.1 Duel Living的功能规划 211
 - 5.1.1 基本产品组织 211
 - 5.1.2 数据结构 213
 - 5.1.2.1 赛事 (Duel) 213
 - 5.1.2.2 消息 (Message) 214
 - 5.1.2.3 参赛方 (Player) 和主持人 (Host) 216
 - 5.1.3 数据结构的关系 216
 - 5.2 数据层开发 217
 - 5.2.1 文件结构 217
 - 5.2.2 安装依赖 217
 - 5.2.3 主持人数据和参赛方数据 218
 - 5.2.4 赛事数据 223
 - 5.2.5 消息数据 225
 - 5.3 服务端基本架构开发 227
 - 5.3.1 安装依赖 227
 - 5.3.2 程序入口 229
 - 5.3.3 路由表 229
 - 5.4 API开发 230
 - 5.4.1 API安全 230
 - 5.4.2 赛事API 231
 - 5.4.2.1 获取当前可用的所有赛事信息 232
 - 5.4.2.2 获取指定赛事数据 232
 - 5.4.2.3 创建新的赛事 233
 - 5.4.3 消息API 236
 - 5.4.3.1 获取指定赛事中的若干消息 236
 - 5.4.3.2 发布消息到指定赛事 237
 - 5.5 直播网络 237
 - 5.5.1 网络架构 238
 - 5.5.1.1 集中架构 238
 - 5.5.1.2 分布式架构 239
 - 5.5.1.3 P2P 网络 239
 - 5.5.2 技术实现 240
 - 5.5.3 WebSocket 服务端 241
 - 5.5.3.1 建立WebSocket服务端实例 242
 - 5.5.3.2 建立WebSocket通讯连接 242
 - 5.5.3.3 广播消息 244
 - 5.5.4 P2P 协调服务端 245

- 5.5.4.1 建立P2P协调连接 245
- 5.5.4.2 存储客户端地理信息 246
- 5.5.4.3 匹配相近的客户端 248
- 5.6 直播间客户端 249
 - 5.6.1 准备工作 249
 - 5.6.2 建立直播通信 250
 - 5.6.2.1 建立PeerJS客户端实例 251
 - 5.6.2.2 建立WebSocket通信连接 251
 - 5.6.2.3 建立P2P通信连接 253
 - 5.6.3 处理消息 253
- 5.7 部署应用 255
 - 5.7.1 编译代码 255
 - 5.7.2 运行程序 256
 - 5.7.3 发布部署 257
- 5.8 小结 258
- 第6章 ES2016标准 259
 - 6.1 Array.prototype.includes 259
 - 6.2 幂运算符 260
 - 6.3 小结 261
- 第7章 展望更远的未来 262
 - 7.1 async/await 262
 - 7.2 Decorators 264
 - 7.2.1 简单实例 264
 - 7.2.2 黑科技 265
 - 7.3 函数绑定 266
 - 7.4 小结 267
- 附录 其他ES2015新特性 268

《实战 ES2015》

精彩短评

- 1、错误蛮多，代码质量不怎么样。weakset和weakmap的示例代码明显错的啊.....
- 2、2016.56 最近挺火的脸谱书，算是另外一个ES6教程吧，还是不错的，前端学ES6又多了本不错的书值得一看
- 3、后面的实践项目太发散了，重点的新特性讲的然而浅了点
- 4、前半部分介绍es6中新增特性 后半部分有以结合vue的实战 以及nodejs的es实战 总的来说 还不错

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:www.tushu111.com