

实战ES2015

深入现代 JavaScript 应用开发



小问 著

前端撷英馆

实战ES2015

深入现代 JavaScript 应用开发



小问 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

ES2015 一直吸引着众多 JavaScript 开发者去积极尝试,如今,使用 ES2015 标准进行工程开发也已深入人心。随着工程师们对 ES2015 的热情日益增长,许多新特性应运而生。本书为读者介绍了 ES2015 的详细特性和意义,以及 JavaScript 在 ES2015 标准中的表现,同时向读者展示了利用 ES2015 中的新特性完成的 JavaScript 应用开发实例,以实际案例说明利用 ES2015 的新特性如何提高 JavaScript 应用前端和后台的开发效率。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

实战 ES2015: 深入现代 JavaScript 应用开发 / 小问著. —北京: 电子工业出版社, 2016.10
ISBN 978-7-121-30018-9

I. ①实… II. ①小… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2016)第 238713 号

策划编辑: 张春雨

责任编辑: 徐津平

印 刷: 北京天宇星印刷厂

装 订: 北京天宇星印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×980 1/16 印张: 18.5 字数: 385 千字

版 次: 2016 年 10 月第 1 版

印 次: 2017 年 1 月第 2 次印刷

印 数: 1500 册 定价: 79.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 010-51260888-819, faq@phei.com.cn。

前言

历时至少 7 年制定的新 ECMAScript 标准——ECMAScript 6（亦称 ECMAScript Harmony，简称 ES6），终于在 2015 年 6 月正式发布。ES6 也被称为 ES2015，自从 2009 年上一个标准版本 ES5 发布以来，ES2015 就一直以其新语法、新特性吸引着众多 JavaScript 开发者，驱使他们积极尝鲜。

ES6 的第一个特性草案发布于 2011 年 7 月。

虽然各大浏览器厂商所开发的 JavaScript 引擎至今都还没有实现对 ES2015 中所有特性的完美支持，但这并不能阻挡工程师们对 ES2015 的热情。于是，Babel、Traceur 等编译器便出现了。在 ES2015 标准正式发布之前，这些编译器便能将尚未得到支持的 ES2015 特性转换为 ES5 标准的代码，使其得到浏览器的支持。其中，Babel 因具有模块化转换器（Transpiler）这一设计特点赢得了绝大多数 JavaScript 开发者的青睐，本文也将以 Babel 为基础工具，向大家展示 ES2015 的神奇魅力。

如今，使用 ES2015 标准进行工程开发已经深入人心，甚至连 ES2016 标准也已经在 2016 年正式发布。在这个如此恰当的时机，我觉得应该写一本通俗易懂的关于 ES2015 标准的书来引导广大 JavaScript 爱好者和工程师向新时代迈进。

本书内容

本书以 JavaScript 作为基本编程语言，并以最新的 ES2015 标准作为 JavaScript 代码编写标准，旨在向读者展示如何以最新的 JavaScript 代码标准编写出更具可读性、更方便、更具工程化优势的代码。

本书首先介绍了 JavaScript 标准版本的发展历史，然后简要概括了 ES2015 的作用和意义。在此基础上，详细讲解了 JavaScript 在 ES2015 标准中新增内容（如箭头函数、模板字符串、新

的数据结构、类语法、生成器等)的基本用法和注意要点。当大部分读者对 ES2015 有了进一步的了解后,本书便会以实际的开发项目向大家展示如何利用 ES2015 标准,较好地实现 JavaScript 应用。

最后,本书还会对最新发布的 ES2016 标准进行研究和探讨,展望未来 JavaScript 的发展方向。

本书读者

本书的目标读者有以下三类:

1. 正在学习 JavaScript 开发,对 JavaScript 语言有基本的了解和熟悉度,且希望能更早地了解 JavaScript 的发展情况的人。
2. 正从事 JavaScript 开发相关工作,熟悉 JavaScript 的基本开发要领,且有意掌握最新的 JavaScript 技术进行自我提升的 Web 工程师(此处不区分前端与后端)。
3. 希望更深入地研究 JavaScript 这门开发语言的 JavaScript 工程师。

与此同时,本书也适合正使用其他编程语言(如 Python、Ruby、Java 等)进行 Web 开发的工程师来学习现代前端开发的知识。

使用示例

要运行本书中的示例,需要安装以下系统及软件。

- 操作系统: Mac OS X 10.9 或以上版本、Windows 7 或以上版本、Linux。
- 浏览器: Google Chrome、Safari、Firefox、Internet Explorer 11、Windows Edge。
- 运行环境: Node.js 4.0 或以上版本。

本书结构

第 1 章 回顾 ECMAScript 版本发展历程

讲述了历代 ECMAScript 标准版本的发展历程以及对前一个版本的新增和修正。对于不了

解 ECMAScript 或 JavaScript 的初学者来说,了解 ECMAScript 的发展历程有助于更好地理解它的现状,同时也对往后的学习有更好的帮助。

第 2 章 ES2015 能为 JavaScript 的开发带来什么

对于企业来说,一项技术最重要的是它能为企业带来的效益,ES2015 中集成了不少从前需要开发者自行完成的特性或工具,能够加强 ECMAScript 所属语言的工程化属性。

第 3 章 ES2015 新语法详解

详细介绍 ES2015 中比较重要几种特性,并以较为常见的例子说明它们的作用方法,使读者能更好地理解。从诞生缘由、使用方法以及相关事例几个方面来介绍每一个新的特性,与其他介绍 ES2015 标准的文章或书籍不一样的是,本书将以一线工程师在实际开发经验中总结出来的注意事项为例,将一些已知的、容易出现的错误点讲明白。

第 4 章 ES2015 的前端开发实战

以 Filmy 为背景,利用 ES2015 标准开发该项目的前端 JavaScript 应用部分,其中涉及 JavaScript 的模块化开发、第三方库的使用、前端 JavaScript 应用工程化等内容。开发该项目本着能够真正投入使用原则,旨在让读者明白如何将 ES2015 这一新标准运用到开发当中来优化和提升开发体验,同时积累实际经验。

第 5 章 ES2015 的 Node.js 开发实战

以竞技比赛直播系统为开发内容,讲述 ES2015 在 Node.js 中的开发体验以及相对于 ES5 时代中 Node.js 开发的改进之处。另外,这个项目还会使用到一些新的 Web 技术,如使用 WebSocket、WebRTC (P2P) 等通信 API 来替代一些较为老旧的通信技术,以体现在众多新潮技术的帮助下,目前最为火热的直播类应用如何进行开发。

第 6 章 ES2016 标准

在 ES2015 标准之后,ECMA 标准委员会在 2016 年发布了 ES2016 标准。本章介绍 ES2016 相对于 ES2015 有了哪些改进,以及为开发者带来了哪些更引人注目的特性。

第 7 章 展望更远的未来

ECMAScript 一直处在快速发展的过程中,除了 ES2015 和 ES2016 中所包含的新特性以外,更有很多让人兴奋的新特性处在实验开发阶段,这些功能很可能在不久的将来进入 ECMAScript 标准。

写作感言

这本书是由一篇较长的关于 ES2015 标准介绍的文章所发展出来的，历时 6 个月写作完成。在这 6 个月中，ECMA 委员会还发布了最新的 ES2016 标准，而国内外不同的团队对 ES2015 甚至 ES2016 的使用率也渐渐达到了一个前所未有的高度，新语法、新特性的使用也开始成为 JavaScript 开发团队中的标配。

另外我还得特别感谢一些人。首先要感谢我的家人，在我成为“全职作者”的这段时间内所给予我的支持和鼓励。感谢这本书的策划编辑张春雨老师，是他给予了我这个机会将一篇文章写成了一本完整的书籍，也是他让我重新捡起了搁置多年的出版计划。同时还要感谢贺师俊老师（hax）、程邵非老师（winter）等前辈给予本书的大力支持和宝贵建议，使得这本书的内容不至于空洞，也让我受益良多。

联系作者

我由衷地感谢你购买了此书，希望你能喜欢它，也希望它能够为你带来你所希望获得的知识。虽然我已经非常细心地检查书中所提到的所有内容，但仍有可能存在疏漏，若你在阅读过程中发现错误，在此我先表示歉意。同时欢迎你对本书的内容和相关源代码发表意见和评论。你可以通过我的私人邮箱 willwengunn@gmail.com 与我取得联系，清楚地说明来意，我会一一解答你的疑惑。

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	ES2015 概述	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	高级用法.....	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	高级技巧	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	高级使用方法	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 生成唯一的 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 JavaScript 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	匹配最相近的客户端	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