

版权注意事项：

- 1、书籍版权归作者和出版社所有
- 2、本PDF仅限用于个人获取知识，进行私底下的知识交流
- 3、PDF获得者不得在互联网上以任何目的进行传播
- 4、如觉得书籍内容很赞，请购买正版实体书，支持作者
- 5、请于下载PDF后24小时内删除本PDF。

ECMAScript 6 Primer

ES6 标准入门

(第3版)

阮一峰 著



中国工信出版集团

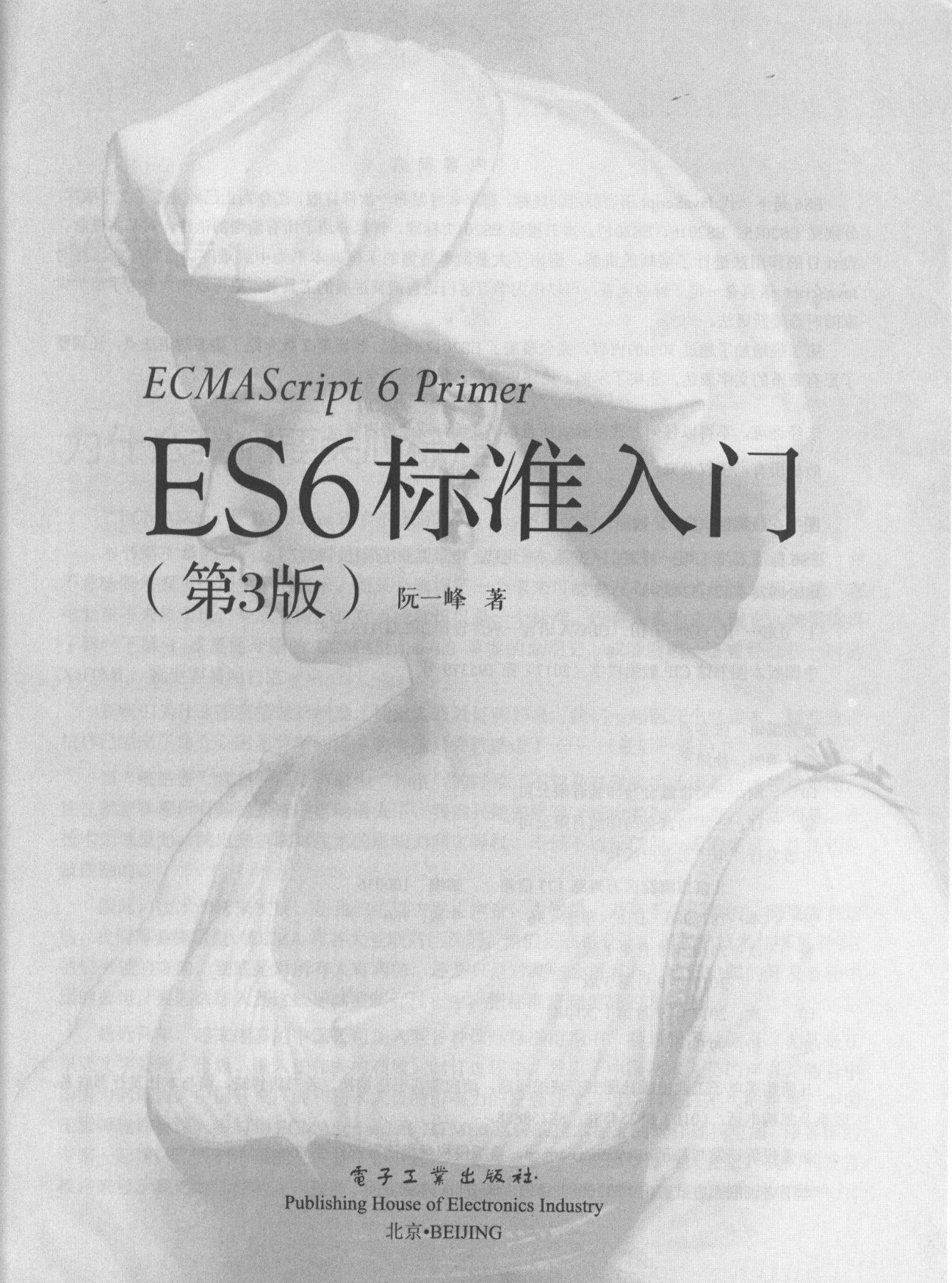


电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn



阮一峰

资深JavaScript语言专家，知名技术博客作者。专注于网站开发技术十余年。畅销书《黑客与画家》《软件随想录》的译者。现就职于蚂蚁金服集团。



ECMAScript 6 Primer

ES6 标准入门

(第3版) 阮一峰 著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

ES6 是下一代 JavaScript 语言标准的统称，每年 6 月发布一次修订版，迄今为止已经发布了 3 个版本，分别是 ES2015、ES2016、ES2017。本书根据 ES2017 标准，详尽介绍了所有新增的语法，对基本概念、设计目的和用法进行了清晰的讲解，给出了大量简单易懂的示例。本书为中级难度，适合那些已经对 JavaScript 语言有一定了解的读者，可以作为学习这门语言最新进展的工具书，也可以作为参考手册供大家随时查阅新语法。

第 3 版增加了超过 30% 的内容，完全覆盖了 ES2017 标准，相比第 2 版介绍了更多的语法点，还调整了原有章节的文字表达，充实了示例，论述更准确，更易懂易学。

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

ES6 标准入门 / 阮一峰著. —3 版. —北京: 电子工业出版社, 2017.9

ISBN 978-7-121-32475-8

I. ①E… II. ①阮… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 195379 号

策划编辑: 张春雨

责任编辑: 徐津平

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

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: 36 字数: 786 千字

版 次: 2014 年 8 月第 1 版

2017 年 9 月第 3 版

印 次: 2017 年 9 月第 1 次印刷

定 价: 99.00 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

推荐序 1

为什么我们要关心标准

“ECMAScript 是 JavaScript 语言的国际标准，JavaScript 是 ECMAScript 的实现。”

本书第 1 章的这句话已经清楚地告诉我们，这是一本不实用的书。我们学习了这本书，并不意味着掌握了一项实用的技术，而只是掌握了一个未来可能会发布的技术标准。而标准，有可能在将来被实现，变成主流，也有可能就仅仅是一个标准，没有人真的去实践它。如果你再了解一下第 1 章里面介绍的 ECMAScript 4.0 草案的血泪史，或者回顾一下曾经红极一时的 XHTML，就更容易明白这一点了。

那我们为什么不直接忽略标准，拥抱实践就好呢？来，我们一起翻开小学课本，跟我念：柏林已经来了命令，阿尔萨斯和洛林的学校只许教德语了……（《最后一课》）

当“统治者”宣布一门语言成为“标准”的时候，不管是在现实生活还是技术领域里面，往往就意味着所有其他的选项自动消失了，我们只能去学习“统治者”的语言。幸运的是，在技术领域里面，跳出来争取对技术的影响力和主导权，不但不违反任何一国的宪法，往往还是被鼓励的。

因此，技术的未来发展，是我们可以去发出声音，去影响，乃至去引领的。而要做到这些，我们需要搞清楚，ECMA 和各大互联网巨头们，他们正在做什么，正在把技术往哪里引领；他们引领的方向，到底是对所有人有利的，还是只是对某些公司有利；我们中国的开发者和中国的公司，要怎么加入到这些标准的制订过程中，把标准带到更好的方向上。

最近几年，越来越多的中国公司加入到各种国际标准组织中，参与到各种标准（尤其是在东亚文字处理、排版、输入法相关的领域）制订过程中，发出了中国技术人员的声音。随着中国国力的增强，中国开发厂商和技术人员的影响力的发展壮大，可以预见，不久的将来，中国工程师也许会深入参与到 ECMAScript 7 和 HTML6 这样的技术标准的制订过程里面，跟各国的专家一起探讨，我们中国的开发者不喜欢这样，更喜欢那样。在那些标准大会上，我们的发言权将来自于我们对标准的深入理解、我们对技术发展的独到眼光和我们建设起来的技术影响力。

作为一个 JS 开发者,实话说,对于 ECMAScript 6 里面的很多内容(比如 let 语句),我并不同意。但是很遗憾,这个标准的制定过程没我们什么事。但是如果我们从现在开始关注国际标准,翻译标准文档,让更多人了解标准,更多公司加入标准组织、参与标准制订,也许未来的中国技术圈不但会是很多人的一个圈子,还会是很有影响力的一个圈子。

“我们说的话,让世界都认真听话。”(S.H.E,《中国话》)

腾讯驻 W3C 顾问委员会代表,黄希彤(stone)

黄希彤(网名 emu),Web 性能优化(WPO)领域实践者,信息无障碍领域推动者。腾讯 Web 前端专家,腾讯驻 W3C 顾问委员会代表,腾讯 QQ 空间技术总监。

推荐序 2

因为一件往事，我现在轻易不敢给别人写序或者书评。那天我在想，如果我要给这本书写序，是不是应该先把这本书拿给贺老（hax）看看。后来呢，我到阮一峰老师的 GitHub 上看了一看，发现这本书有 7500 多个 star，若干个已解决和未解决的 issue，所以我就放心了。开源真是好啊！

这本书是关于 ES6 的，我对 ES6 并没有特别系统的研究，但是也在工作中使用了一部分 ES6 的特性，使用得最多的是 Promise，其他的特性只是研究，很少使用，主要是因为本身支持 ES6 的环境和工具有限。浏览器就不说了，现在的前端工程师在一些产品中能够抛弃 IE6 已经是很幸福的事情了，但是即使是 IE8，离真正的 ES6 也还很遥远。在其他领域，比如手机游戏领域，cocos2d-js v3.0 使用的脚本引擎是 SpiderMonkey v28，因此情况要好很多，但是周边的一些工具，比如 closure copiler 不能很好地压缩和优化 ES6，当然你可以采用转换工具先将 ES6 转成 ES5，然后再做压缩和优化，但是这多出来的一步造成更多出错的可能，而且和享受 ES6 的语法糖的快乐相比，开销有点大——如果无论如何需要再转一步，那么为什么我们不干脆考虑 TypeScript 或者其他选择呢？

为什么会选择使用 ES6 的 Promise 呢？那是因为 Promise 算是比较好解决异步嵌套问题的方案，另外 Promise 本身在低版本下也有比较好的 polyfill 实现 (<https://github.com/jakearchibald/es6-promise>)，对于我和一些前端工程师来说，是十分乐意为将来去写一些能够向前兼容的符合标准的代码的。

目前这个阶段，前端学习 ES6 并不意味着能够很快将 ES6 的好处带到工作中，因为我们毕竟还受到现在的浏览器环境的制约。但是，即使单纯从学习一门编程语言的核心 API 的角度来说，ES6 也是值得学习的。它的很多新特性，真正涉及现代编程语言概念中很流行的部分，不管是解构赋值还是迭代器或者 yield，都是超棒超赞的思想，不但易于理解，也能节省很多键盘操作，而另一些诸如 const、作用域之类的设定，则让脚本引擎代替程序员人肉检查做更多的事情，让我们最终上线的代码变得更加安全和更加优美。

不管怎样，ES6 代表着一种前端的未来，这种未来，无疑能让前端工程师们工作得更高效，也更有乐趣。更进一步说，ECMAScript 还是开放的标准，对这门语言的新特性，有什么好的想法，都是有机会提交为标准的，也就是说，前端程序员的未来，是由我们前端程序员自己来创

造的, 还有什么比自由更加美好的呢? 所以, 为了未来, 加油!

360 奇舞团团长, 月影

吴亮(网名月影), 先后在微软亚洲研究院做过访问学生, 在金蝶软件有限公司担任过核心开发工程师、设计师和项目经理, 在百度电子商务事业部担任 Web 开发项目经理。现任奇虎 360 高级技术经理, 360 前端团队奇舞团负责人。多年来致力于 JavaScript 技术和 Web 标准的推广, 活跃于国内各技术社区, 现为 w3ctech 顾问。

推荐序 3

同大多数读者一样，我最早看到阮一峰先生的文字是在其博客上。他的第一篇博文于 2003 年写就，迄今已有 1500 多篇文章，可谓高产。阮先生并非计算机相关专业，但这一点并没有妨碍他从事技术写作，其文字朴实，思路清晰，所有人都能看懂，更能感受到他写文章的用心程度，而这本书完美地体现了他的一贯风格。另外，这本书是开源作品，也很好地践行了他一贯的贡献原则。

自我写下第一行前端代码到现在已经十来年了，前端的基础设施也发生了巨大的变化。变化最大的还是浏览器环境，从原来烂熟 IE6 的各种 bug 和 hack，到现在 IE6 已经完全不在我的考虑范围内。其次是前端的工程化程度，2011 年，我做 FIS (<http://fis.baidu.com>) 时，完全没想到前端的工程化进展会如此之快。而变化最慢的，要数语言本身了，1999 年发布的 ECMAScript 3.0 其实相当于第 1 版；10 年后的 2009 年发布了第 2 版：ECMAScript 5.0；ECMAScript 6 则是 2015 年发布的。

我的一贯主张是，要学好 JavaScript，ECMAScript 标准比什么书都强。ECMAScript 标准已经用最严谨的语言和最完美的角度展现了语言的实质和特性。理解语言的本质后，你已经从沙堆里挑出了珍珠，能经受得起时光的磨砺。

我从 2009 年开始正式接触 ECMAScript 规范，当时我在写百度的 JavaScript 基础库 Tangram 1.0，ECMAScript 5 还处于草案状态。我自己打印了一本小册子，上下班时在地铁上慢慢看。那时才知道，有很多问题在网络上被包装了太多次，解释得千奇百怪，但用规范的语言来描述竟是如此简单。

ECMAScript 标准经历了很多变故——尤其是 ECMAScript 4 那次——也从语言的角度反映了各大厂商之间的立场差异。不过，ECMAScript 5 的正式发布和发展，为所有 Web 开发者奠定了稳定的基础，尽管浏览器之间存在大量差异，尤其是 DOM，但在 JavaScript 语言层面，都相对严格地遵循着 ECMAScript 5 的规范。

JavaScript 遵守“一个 JavaScript”的原则，所有版本都需要向后兼容。Web 语言的解释器版本不是由开发者而是由用户决定的，所以 JavaScript 无法像 Python、Ruby、Perl 那样，发布一个不向下兼容的大版本，这也就是 ECMAScript 4 失败的根源，由于它会导致大量已有网页

的“bug”，浏览器厂商会强烈反对。当然，ECMAScript 6 的 strict mode 也在尝试逐步淘汰一些不良实践。

ECMAScript 6 相比 5 有了很大的进步。经过这次改进，JavaScript 语法更精简，变得更有表现力了；在严格模式下，开发者受到了适当而必要的约束；新增了几种数据类型（map、set）和函数能力（Generator、迭代器）；进一步强化了 JavaScript 的特点（promise、proxy）；并且让 JavaScript 能适用于更大型的程序开发（modules、class）。更重要的是，这个规范会被浏览器厂商、不同的平台广泛支持。

实际上，所有的语言改进都是从使用者的最佳实践中提炼出来的。JavaScript 的约束一直很少，这一灵活性让开发者能相当自由地积累形形色色的使用经验和实践，也就是说，我们所有 ECMAScript 的使用者，也是其标准的间接贡献者。

百度高级工程师、前端通用组技术负责人，雷志兴

雷志兴（网名 berg），资深工程师，2007 年加入百度工作至今，负责过多项前端基础技术、架构的设计和搭建；骑行爱好者，行程万余公里；微信公众号“行云出岫”（DevLife）的维护者。

第3版前言

4年前，当我开始写这本书的时候，ECMAScript 5.1 版刚刚开始普及，最流行的框架还是 jQuery。ES6 看上去就像一个遥远的蓝图，无人知道何时会实现。

仅仅4年，ES6 已经经历了 ES2015、ES2016、ES2017 这3个版本的迭代，各种实现的支持度已经超过90%，不仅可以实现网页的编写，还可以实现服务器脚本、手机 App 和桌面应用的编写。程序员们完全接受了这个标准，甚至大量使用尚未标准化的新语法。JavaScript 语言就像一列高铁，以令人“眩晕”的速度向前冲刺。

互联网行业的蓬勃兴旺造就了 ES6 的成功，也使得这本教程不断更新，越写越厚。第2版问世18个月之后，不得不推出第3版。

第3版新增了超过30%的内容，完全覆盖了 ES2017 标准（第2版只做到覆盖 ES2015 标准），并且对所有章节都进行了修订，文字表达更准确易懂，示例更丰富。对读者来说，这个版本更容易学习，更有参考价值。

这4年来，我对 ES6 的理解和所有的学习笔记，都浓缩在这本教程里面。那些我自己感到最困难的地方，书中都做出了详细讲解，给出了细致的示例，我相信这也是其他国内学习者所需要的。

这本教程当然也包含了些许局限，以及细致检查之后仍然疏漏的各种错误。一旦发现，我会第一时间更正。读者可以到官方仓库 github.com/ruanyf/es6tutorial 中查看勘误。

我在微博上曾经说过一段话，就把它放在这里作为结束吧。

“我水平其实不高，只是好奇心重，从没想到这么多人会关注。希望不要让大家失望，未来做一块垫脚石，为需要的朋友提供帮助，为技术的推广和发展做出力所能及的贡献。”

阮一峰

2017年8月1日，写于杭州

第1版前言

2012年年底，我开始动手做一个开源项目《JavaScript 标准参考教程》(github.com/ruanyf/jstutorial)。原来的设想是将自己的学习笔记整理成一本书，哪里料到，这个项目不断膨胀，最后变成了 ECMAScript 5 及其外围 API 的全面解读和参考手册，写了一年多还没写完。

那个项目的最后一章就是 ECMAScript 6 的语法简介。那一章也是越写越长，最后我不得不决定，把它独立出来，作为一个新项目，也就是您现在看到的这本书。

JavaScript 已经是互联网开发的第一大语言，而且正在变成一种全领域的语言。著名程序员 Jeff Atwood 甚至提出了一条“**Atwood 定律**”：“所有可以用 JavaScript 编写的程序，最终都会出现 JavaScript 的版本。”(Any application that can be written in JavaScript will eventually be written in JavaScript.)

ECMAScript 正是 JavaScript 的国际标准，这就决定了该标准的重要性。而 ECMAScript 6 是 ECMAScript 历史上最大的一次版本升级，在语言的各个方面都有极大的变化，即使是熟练的 JavaScript 程序员，也需要重新学习。由于 ES6 的设计目标是企业级开发和大型项目，所以可以预料，除了互联网开发者，将来还会有大量应用程序开发者（甚至操作系统开发者）成为 ES6 的学习者。

我写作这本书的目标，就是想为上面这些学习者提供一本篇幅较短、简明易懂、符合中文表达习惯的 ES6 教程。它由浅入深、循序渐进，既有重要概念的讲解，又有 API 接口的罗列，便于日后当作参考手册查阅，还提供大量示例代码，让读者不仅一看就懂，还能举一反三，直接复制用于实际项目之中。

需要声明的是，为了突出重点，本书只涉及 ES6 与 ES5 的不同之处，不对 JavaScript 已有的语法进行全面讲解，毕竟市面上这样的教程已经有很多了。因此，本书不是 JavaScript 入门教材，不适合初学者。阅读本书之前，需要对 JavaScript 的基本语法有所了解。

我本人也是一个 ES6 的学习者，不敢说自己有多高的水平，只是较早地接触了这个主题，持续地读了许多资料，追踪标准的进展，做了详细的笔记而已。虽然我尽了最大努力，并且原稿在 GitHub 上公开后已经得到了大量的勘误，但是本书的不尽如人意之处恐怕还有不少。

欢迎大家访问本书的项目主页 (github.com/ruanyf/es6tutorial), 提出意见, 提交 pull request。这些都会包括在本书的下一个版本中。

阮一峰

2014 年 6 月 4 日, 写于上海

读者服务

轻松注册成为博文视点社区用户 (www.broadview.com.cn), 扫码直达本书页面。

- **提交勘误:** 您对书中内容的修改意见可在 [提交勘误](#) 处提交, 若被采纳, 将获赠博文视点社区积分 (在您购买电子书时, 积分可用来抵扣相应金额)。
- **交流互动:** 在页面下方 [读者评论](#) 处留下您的疑问或观点, 与我们和其他读者一同学习交流。

页面入口: <http://www.broadview.com.cn/32475>



目录

第 1 章	ECMAScript 6 简介	1
1.1	ECMAScript 和 JavaScript 的关系	1
1.2	ES6 与 ECMAScript 2015 的关系	1
1.3	语法提案的批准流程	2
1.4	ECMAScript 的历史	3
1.5	部署进度	4
1.6	Babel 转码器	4
1.6.1	配置文件 .babelrc	5
1.6.2	命令行转码 babel-cli	6
1.6.3	babel-node	7
1.6.4	babel-register	8
1.6.5	babel-core	8
1.6.6	babel-polyfill	9
1.6.7	浏览器环境	10
1.6.8	在线转换	10
1.6.9	与其他工具的配合	11
1.7	Traceur 转码器	11
1.7.1	直接插入网页	12
1.7.2	在线转换	13
1.7.3	命令行转换	14
1.7.4	Node 环境的用法	15
第 2 章	let 和 const 命令	17
2.1	let 命令	17

2.1.1	基本用法	17
2.1.2	不存在变量提升	19
2.1.3	暂时性死区	19
2.1.4	不允许重复声明	21
2.2	块级作用域	22
2.2.1	为什么需要块级作用域	22
2.2.2	ES6 的块级作用域	23
2.2.3	块级作用域与函数声明	24
2.2.4	do 表达式	27
2.3	const 命令	28
2.3.1	基本用法	28
2.3.2	本质	29
2.3.3	ES6 声明变量的 6 种方法	30
2.4	顶层对象的属性	30
2.5	global 对象	31
第 3 章	变量的解构赋值	33
3.1	数组的解构赋值	33
3.1.1	基本用法	33
3.1.2	默认值	35
3.2	对象的解构赋值	37
3.3	字符串的解构赋值	41
3.4	数值和布尔值的解构赋值	41
3.5	函数参数的解构赋值	42
3.6	圆括号问题	43
3.6.1	不能使用圆括号的情况	43
3.6.2	可以使用圆括号的情况	44
3.7	用途	44
第 4 章	字符串的扩展	49
4.1	字符的 Unicode 表示法	49
4.2	codePointAt()	50

4.3	String.fromCharCode()	52
4.4	字符串的遍历器接口	52
4.5	at()	53
4.6	normalize()	53
4.7	includes()、startsWith()、endsWith()	54
4.8	repeat()	55
4.9	padStart()、padEnd()	56
4.10	模板字符串	57
4.11	实例：模板编译	60
4.12	标签模板	62
4.13	String.raw()	67
4.14	模板字符串的限制	68
第 5 章	正则的扩展	71
5.1	RegExp 构造函数	71
5.2	字符串的正则方法	72
5.3	u 修饰符	72
5.4	y 修饰符	74
5.5	sticky 属性	77
5.6	flags 属性	77
5.7	s 修饰符：dotAll 模式	78
5.8	后行断言	79
5.9	Unicode 属性类	80
5.10	具名组匹配	81
5.10.1	简介	81
5.10.2	解构赋值和替换	82
5.10.3	引用	83
第 6 章	数值的扩展	85
6.1	二进制和八进制表示法	85
6.2	Number.isFinite()、Number.isNaN()	86
6.3	Number.parseInt()、Number.parseFloat()	87