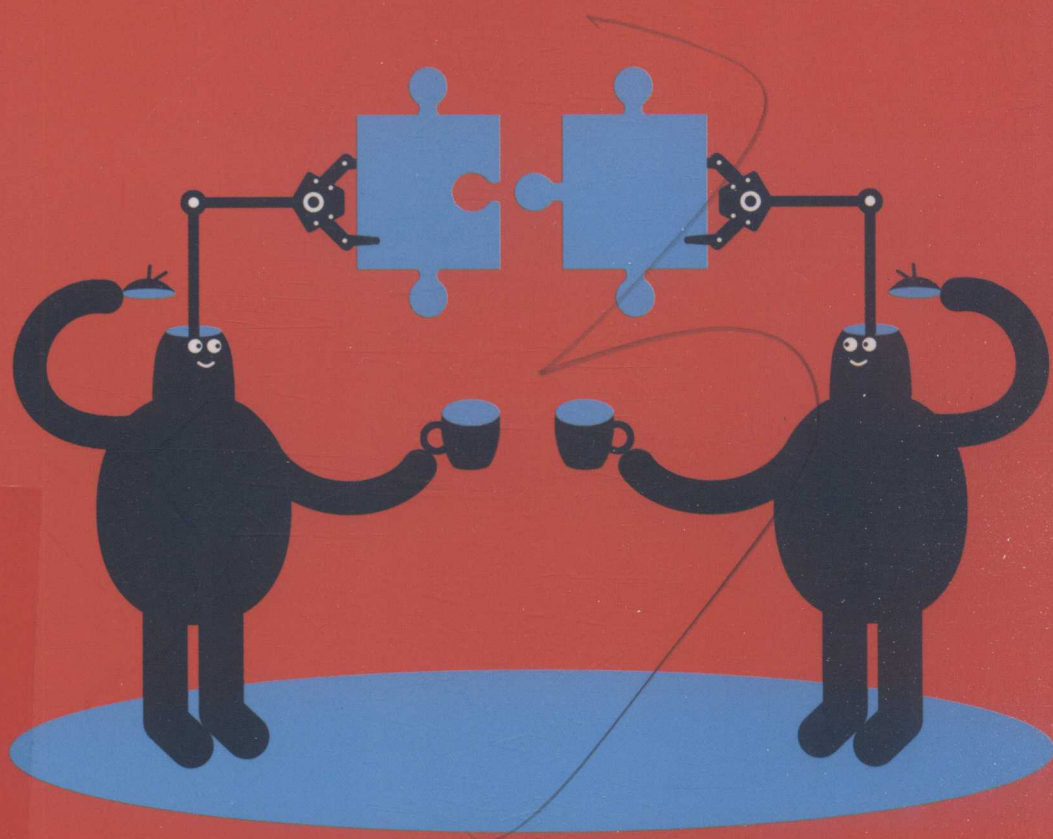


Web开发经典丛书

Get Programming with JavaScript Next: New features of ECMAScript 2015, 2016, and beyond

ES2015/2016编程实战

掌握标准JavaScript应用开发



[美] JD·艾萨克斯(JD Isaacks) 著

林赐 译



MANNING



清华大学出版社

Web 开发经典丛书

ES 2015/2016 编程实战

[美] JD·艾萨克斯(JD Isaacks) 著
林 赐 译

清华大学出版社
北 京

JD Isaacks

Get Programming with JavaScript Next: New features of ECMAScript 2015, 2016, and beyond
EISBN: 978-1-61729-420-4

Original English language edition published by Manning Publications, 178 South Hill Drive,
Westampton, NJ 08060 USA. Copyright©2018 by Manning Publications. Simplified Chinese-
language edition copyright © 2019 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Manning 出版公司授权清华大学出版社独家出版。未经出版者书面许可，
不得以任何方式复制或抄袭本书内容。
版权所有，侵权必究。

北京市版权局著作权合同登记号 图字：01-2018-3783

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

ES 2015/2016 编程实战 / (美) JD·艾萨克斯(JD Isaacks) 著；林赐 译. —北京：清华大学出版社，2019

(Web 开发经典丛书)

书名原文：Get Programming with JavaScript Next: New features of ECMAScript 2015, 2016,
and beyond

ISBN 978-7-302-51941-6

I. ①E… II. ①J… ②林… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 288418 号

责任编辑：王 军

封面设计：孔祥峰

版式设计：思创景点

责任校对：牛艳敏

责任印制：沈 露

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座 邮 编：100084

社 总 机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者：三河市国英印务有限公司

经 销：全国新华书店

开 本：170mm×240mm 印 张：21.5 字 数：421 千字

版 次：2019 年 3 月第 1 版 印 次：2019 年 3 月第 1 次印刷

定 价：69.80 元

产品编号：080076-01

以下是第2单元中将学习的一些新语法的预览

新增的解构语法与现有的数据结构化语法的对称对应体。



使用简写属性和方法，可以消除对象字面量中的大量冗余。

```
const verbose = {  
  method: function() {  
    //....  
  },  
  name: name  
}  
  
const concise = {  
  method() {  
    //...  
  },  
  name  
}
```

简写方法
remove:function

简写属性
remove:name

以下是第7单元中的promise和异步函数预览

```
[source,js]
----
async function fetchImage(url){
  const resp = await fetch(url);
  const blob = await resp.blob();
  return createImageBitmap(blob);
};

fetchImage('my-image.png').then(image => {
  // do something with image
});
----
```

如果不使用promise或异步函数，那么要实现这一点，需要编写大量笨重的代码。

```
[source,js]
----
function fetchImage(url, cb){
  fetch(url, function(resp){
    resp.blob(function(blob){
      createImageBitmap(blob, cb);
    })
  })
};

fetchImage('my-image.png', function(image){
  // do something with image
});
----
```

译者序

数月来，我一直在贝尔实验室(加拿大)编写一个项目，使用 React+Redux+Webpack 作为前端，Django 作为后端。在项目开发过程中，JavaScript 优越的语言特性、强大的功能以及简洁的语法给我留下深刻的印象。遥想初识 JavaScript 的时候，我还年少，对 JavaScript 的概念仅停留在给静态页面做一点特效和添加一个按钮上，难登大雅之堂。弹指一挥间，10 年过去了，沧海桑田。现如今，JavaScript 鲤鱼跃龙门，成为排名第一的编程语言，士别三日，当刮目相看。江海之所以能为百谷王者，以其善下之，故能为百谷王。我想，用这句话来形容 JavaScript 是恰如其分的。

既是译员，也是程序员，很多人都认为这两个职业相差这么大，要同时做好，并非易事。其实，译员和程序员的工作性质差不多，都是与语言打交道，不同点在于，是与人说话还是与计算机对话。与人说话，输入和输出是非线性的，不管你讲得明白还是不明白，听者都有两种反应——理解和不理解；而与计算机对话，输入和输出是线性的，讲得明白，计算机就能理解，反之亦然。在这几个月里，我白天写代码，晚上翻译，马不停蹄。繁忙工作之余，我也偶尔与友人一起，去踏青、踩春、逛博物馆等，因此对渥太华的感觉也由最初的陌生变成现在的亲切。

渥太华是个宁静又喧哗的城市，开放中带了点含蓄，活泼不缺端庄，古典不失浪漫，前卫却又自律。作为首都，渥太华横跨了两个省份，因此严谨一点说，应该称其为渥太华-加蒂诺地区。以渥太华河南岸的国会山为中心，跨过 Alexandra 桥就到了魁北克省的加蒂诺地区，加蒂诺公园是渥太华居民经常远足、郊游、赏枫的好去处。国会山向西开车大约 30 分钟，就到了具有北方硅谷之称的 Kanata，这为渥太华成为世界著名的通信研究中心奠定了基础，而距离国会山向南不到 1 公里的地方，就是加拿大最古老的全球最大的英法双语大学：渥太华大学。

不过，渥太华的这点荣誉比起其左右两边的两个大都市又稍逊风骚。渥太华向东，驱车两小时，行驶 190 公里，就到达仅次于巴黎的第二大法语城市，也就是加拿大的第二大城市蒙特利尔。这里自然风光旖旎，城中到处弥漫着一种罗曼蒂克的情调，风情万种的街道和建筑总是惹得游人流连忘返，有“加拿大哈佛”美誉的麦吉尔大学就坐落在这座古朴、端庄的城市。而渥太华以西 400 公里，经过大约 4 小时的车程，就来到了加拿大的经济中心多伦多。比起古老的蒙特利尔，后起之秀的多伦多就显得比较有活力，灯红酒绿，加拿大排名第一的多伦多大学就在这个城市。

这三所大学在计算机科学方面各领风骚。打个不恰当的比方，如果说麦吉尔大学是 C 语言(以其古老厚重闻名)，多伦多大学是 Java(以规范严谨著称)，那么渥太华大学就是 JavaScript(出生于草根，却以顽强的生命力让人刮目相看)。宝剑锋从磨砺出，梅花香自苦寒来。JavaScript 从诞生之初，在经历了“两军”对峙的最初发展阶段，到现在一统 PC 端、手机端和平板电脑，走过了风雨 20 年。如果说 HTML 是一个温柔端庄的静美人，那么 JavaScript 就是身着铠甲的金装武士，上山捉虎，下海擒龙，披荆斩棘，无往不利。这两种语言一静一动，琴瑟合鸣，珠联璧合，真可谓是天造地设的一双。

虽然十几年来，我都在与语言打交道，但是平心而论，我不算是一个积极的语言学习者，无论是自然语言，还是计算机语言，都是如此。我从不为这一点感到骄傲，读者切莫学我。对于学语言，我有一个偏执的理念，就是“用以致学”。语言是一种实践性很强的知识，只有勤于实践，才能写出好代码，理解语言的精妙之处。如果只是蜻蜓点水似地看看书，抄抄代码，这对提高语言的修养没有太大用处。虽然我不能身体力行，但是还要劝勉读者几句，要力学笃行：打牢基础，勤于实践。想成为一名好的程序员，要经过三个阶段：实践、思考、创新。正如《荀子·儒效》中所说，知之不若行之。学至于行而止矣。但学而不思则罔，思而不学则殆。在实践中，我们也要注意思考，寻找技术背后的原理，才能触类旁通，举一反三。不过，如果要百尺竿头，更进一步，仅仅成为知识的搬运工是不够的。掌握了知识之后，我们还应该学会创新，唯有这样，才能长江后浪推前浪，成为社会的中流砥柱。

在本序即将结束前，我要特别感谢清华大学出版社，感谢他们对我的信任和理解，把这样一本好书交给我翻译；也要感谢他们为本书的翻译投入了巨大的热情，可谓呕心沥血。没有他们的耐心和帮助，本书不可能顺利付梓。同时，在翻译过程中，我也得到了加拿大友人 Jack Liu 和 Connie Wang 指点迷津，才能为读者提供更贴切的译文。

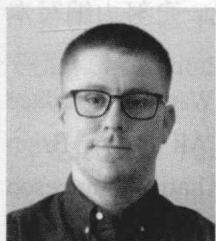
译者才疏学浅，见闻浅薄，言辞多有不足之处，还望谅解并不吝指正。读者如有任何意见和建议，请将反馈信息发送到邮箱 cilin2046@gmail.com，不胜感激。本书主要章节由林赐翻译，参与翻译的还有林凤、王树富、林斌等人。

林 赐

2018 年 08 月 30 日

于渥太华大学

作者简介



JD Isaacks 已经从事编程工作 15 年了，主要使用基于 ECMAScript 的语言。他先前是 The Iron Yard 编码学院的 JavaScript 讲师，喜欢开源，为 React、Backbone 和 D3 等许多热门项目做出了贡献。他也是 Bower.js 和 Moment.js 团队的成员，是最受欢迎的 Sublime Text 软件包之一的 GitGutter 的创建者。

余 简 香 拜

致 谢

我要感谢我的妻子 Christina 和孩子 Talan 和 Jonathan。在我一直努力编写本书时，他们做出了牺牲，我无法抽出时间陪伴他们。我爱他们。

我还要感谢 Manning 出版社，特别是编辑 Dan Maharry、Nick Watts 和 Candace West。我要特别感谢让这本书变得更优秀的审阅者：Aïmen Saïhi、Ali Naqvi、Brian Norquist、Casey Childers、Ethien Daniel Salinas Domínguez、Fasih Khatib、Francesco Strazzullo、Giancarlo Massari、Laurence Giglio、Matteo Gildone、Michael J. Haller、Michael Jensen、Miguel Paraz、Pierfrancesco D'Orsogna、Richard Ward、Sean Lindsay 和 Ticean Bennett。

在本书即将结束前，我要特别感谢清华大学出版社，感谢他们对我的信任和理解，把这样一本好书交给我翻译，也要感谢他们为本书的翻译投入了巨大的热情、时间和精力。没有他们的耐心和帮助，本书不可能顺利付梓。同时，在翻译过程中，我也得到了新家大友友 Jack Lin 和 Camille Wang 点点滴滴才能为我提供更加帮助的帮助。

本书才疏学浅，难免有错，自然会有不足之处，其解释权并不在我。如有任何意见和建议，请发反馈邮件至 china@china.com，不胜感激。本书主要章节由其他翻译，如有错漏，还请各位，不吝教，并致感谢。

2015 年 10 月 30 日

于清华大学

序 言

我使用基于 ECMAScript 的语言大约有 15 年了。事实上，我学习的第一种编程语言是 ActionScript(基于 ECMAScript)。当时，我有点沉迷于编程，本来我想成为一名平面设计师。现在，我很高兴我做到了。在初中之前，我一直在画画，使用 Microsoft Paint 逐个像素制作复杂的图画。在高中时，我学习了互动多媒体课程，掌握了 Adobe Photoshop 和 Macromedia Flash。在我发现了 Photoshop 的强大功能后，我再也不想回过头来逐个像素地画画了。使用 Flash，我更进了一步，不再局限于创建静态图像，而是可以创建丰富的动画。

这是 eBaum's World 和 Newgrounds(展示社区 Flash 游戏的网站)的时代。我访问了这些网站，想知道这些游戏是如何创建的。我尝试通过一次次的试验来学会 ActionScript(Flash 的内部语言)，但是我一直没学会，直到我购买了第一本编程书籍。在此之后，我就被迷住了。能够添加交互是一个在创建动画之外的全新维度，就像创建动画是创建静态图像之外的全新维度一样。但这只是我编程旅程的开始。

在职业生涯中，当我学到了比正在使用的技术强大得多的技术后，一般就不会回头。最近，在了解了 ES2015 及更高版本中包含的所有新特性的强大功能之后，我再也不想使用以前的技术了，我希望读者能够理解并支持我的做法。

关于代码

本书以放在正文中的代码清单的形式包含了许多源代码示例。源代码的格式为等宽字体，以便与正文分开。有些时候，代码以粗体显示，以突出显示此代码与本章前面步骤中的代码不同，例如要新的特性添加到现有的代码行中。

在许多情况下，重新格式化了原始的源代码，添加换行符，重新缩进以适应书中狭小的页面空间。在极少数情况下，空间仍然不够用，代码清单将使用行继续标记(//...)

前言

《ES 2015/2016 编程实战》的目标受众是那些希望学习 2015 及后续版本中所引入的现代特性的 JavaScript 编程人员。本书并未专注于 ES2015 或 ES2016 的特定版本，而是专注于开发人员将会遇到的最佳新特性，这是开发人员投身到现代 JavaScript 开发环境中应该理解的特性。

本书目标读者

任何程序员，无论其技术水平如何，都可以从本书中获益良多。本书没有讲授“如何编程”。只要读者能够自如地使用经典的 JavaScript 编程，无须成为 JavaScript 专家，就可以阅读并理解此书的内容。

本书的组织方式

本书被分解为内聚性强的几个单元。每个单元围绕特定主题(如函数或异步编码)展开。这些单元都会分成特定主题的课堂，在开始上课之前，为了以正确的方式打开读者的思维，每节课都会从启发式问题开始。在每节课中，都会有快速测试，在读者继续学习之前，帮助确定读者是否理解了本节的核心概念。在每节课结束时，都会有练习帮助读者理解和应用所学习的内容。在每个单元结束时，读者将使用本单元中所学的所有内容构建顶点项目。

关于代码

本书以嵌在正文中的代码清单的形式包含了许多源代码示例。源代码的格式为特殊字体，以便与正文分开。有些时候，代码以粗体显示，以突出显示此代码与本章前面步骤中的代码不同，例如将新的特性添加到现有的代码行中。

在许多情况下，重新格式化了原始的源代码。添加换行符、重新缩进以适应书中狭小的页面空间。在极少数情况下，空间依然不够用，代码清单将使用行继续标记(➡)。

此外，在文本中描述代码时，源代码中的注释通常已移除。在许多代码清单中，都使用代码注释突出重要概念。

本书中使用的代码可以在出版商的网站(<https://www.manning.com/books/get-programming-with-javascript-next>)或 GitHub(<https://github.com/jisaacks/get-programming-jsnext>)上获得，也可扫描封底二维码获得。

书籍论坛

本书的购买者可以免费访问由 Manning Publications 运营的私有网络论坛，在这个论坛中，购买者可以对该书发表评论、提出技术问题，从作者和其他用户那里获得帮助。要访问论坛，请访问 <https://forums.manning.com/forums/get-programming-with-javascript-next>。购买者还可以访问 <https://forums.manning.com/forums/about>，了解有关 Manning 论坛和行为规则的更多信息。

Manning 承诺为读者提供一个交流的场所，让读者之间以及读者和作者之间能够进行有意义的对话。这不是作者做出的任何具体参与论坛的承诺，作者在论坛上贡献自己的时间和知识仍然是自愿无偿的行为。我们建议读者尝试向作者提出一些具有挑战性的问题，保持作者的兴趣。只要本书还在出版，就可以从出版商的网站上访问论坛和之前讨论的档案。

在线资源

在 <https://github.com/tc39/ecma262> 上，读者可以与时俱进，了解 JavaScript 的最新特性。在此处，读者可以得知何种特性处在提案流程中的哪个阶段。稍后在本书中，我们将谈论这些阶段和提案的工作机制。

目 录

第 1 课 ECMAScript 规范和提案

流程 1

1.1 ECMAScript 简史 1

1.2 ES2015 增加这么多特性的
原因 2

1.3 谁决定添加何种特性 2

1.3.1 规范阶段 2

1.3.2 选择某个阶段 3

1.4 本书所讨论的内容 4

本课小结 4

第 2 课 使用 Babel 转编译 5

2.1 什么是转编译 5

2.1.1 编译成 JavaScript 语言 5

2.1.2 Babel 的适用场合 6

2.2 配置 Babel 6 6

2.3 本书所需的 Babel 配置 8

2.3.1 源代码映射的注意事项 9

2.3.2 将 Babel 配置为 NPM
脚本 9

本课小结 10

第 3 课 使用 Browserify 捆绑模块 11

3.1 什么是模块 11

3.2 模块在 Node.js 中如何
工作 11

3.3 什么是 Browserify 12

3.4 Browserify 如何协助 ES6
模块 12

3.5 让 Browserify 与 Babel 一起
工作 13

3.5.1 安装 Browserify 13

3.5.2 使用 babelify 创建项目 13

3.6 Browserify 的替代选择 16

本课小结 16

单元 1 变量和字符串

第 4 课 使用 let 声明变量 19

4.1 let 的作用域 20

4.2 let 提升的工作原理 24

4.3 使用 let 还是使用 var 26

本课小结 27

第 5 课 使用 const 声明常量 29

5.1 常量的工作机制 30

5.2 何时使用常量 33

本课小结 35

第 6 课 新字符串方法 37

6.1 搜索字符串 38

6.2 填充字符串 40

本课小结 42

第 7 课 模板字面量 45

7.1 什么是模板字面量 46

7.1.1 模板字面量的字符串
插值 47

7.1.2 模板字面量的多行
字符串 49

7.2 模板字面量是不可重用的
模板 50

7.3 使用标记模板字面量进行
自定义处理 51

本课小结	54	11.2 解构数组	88
第 8 课 顶点项目：构建领域特定语言	57	11.3 结合数组解构和对象解构	89
8.1 创建一些辅助函数	58	11.4 可以解构的类型	91
8.2 创建一个 HTML 转义的 DSL	59	本课小结	92
8.3 创建一个将数组转换为 HTML 的 DSL	60	第 12 课 新对象字面量语法	93
本课小结	61	12.1 简写属性名称	94
单元 2 对象和数组		12.2 简写方法名称	96
第 9 课 新的数组方法	65	12.3 计算属性名称	99
9.1 使用 Array.from 构建数组	66	本课小结	100
9.2 使用 Array.of 构建数组	68	第 13 课 符号	101
9.3 使用 Array.prototype.fill 构建数组	69	13.1 使用符号作为常量	102
9.4 使用 Array.prototype.includes 搜索数组	71	13.2 使用符号作为对象键	104
9.5 使用 Array.prototype.find 搜索数组	72	13.3 使用全局符号创建行为钩子	105
本课小结	73	13.4 使用公知符号修改对象行为	107
第 10 课 Object.assign	75	13.5 符号的陷阱	109
10.1 使用 Object.assign 设置默认值	76	本课小结	110
10.2 使用 Object.assign 扩展对象	78	第 14 课 顶点项目：模拟锁和钥匙	111
10.3 在使用 Object.assign 时防止对象改变	80	14.1 创建锁和钥匙系统	111
10.4 如何使用 Object.assign 赋值	82	14.2 创建 Choose the Door 游戏	113
本课小结	84	本课小结	117
第 11 课 解构	85	单元 3 函数	
11.1 解构对象	86	第 15 课 默认参数和 rest	121
		15.1 默认参数	122
		15.2 使用默认参数避免重新计算值	125
		15.3 使用 rest 操作符收集参数	127

15.4 使用 rest 在函数之间传递参数	129
本课小结	130
第 16 课 解构参数	131
16.1 解构数组参数	132
16.2 解构对象参数	134
16.3 模拟命名参数	134
16.4 创建别名参数	137
本课小结	138
第 17 课 箭头函数	139
17.1 使用箭头函数使代码简洁	140
17.2 使用箭头函数维护上下文	143
17.3 箭头函数的陷阱	145
本课小结	147
第 18 课 生成器函数	149
18.1 定义生成器函数	150
18.2 使用生成器函数	152
18.3 使用生成器函数创建无限列表	154
本课小结	156
第 19 课 顶点项目：囚徒困境	157
19.1 生成囚徒	158
19.2 让囚徒互动	160
19.3 获取和存储结果	161
19.4 将模拟程序结合在一起	162
19.5 哪种囚徒表现最好	164
本课小结	166

单元 4 模块

第 20 课 创建模块	169
20.1 模块规则	170

20.2 如何创建模块	171
20.3 JavaScript 文件何时成为模块	175
本课小结	176
第 21 课 使用模块	177
21.1 指定模块的位置	178
21.2 从模块中导入值	179
21.3 如何绑定导入值	182
21.4 导入副作用	183
21.5 对模块进行分解和组织	184
本课小结	187
第 22 课 顶点项目：猜单词游戏	189
22.1 规划	190
22.2 单词模块	190
22.3 状态模块	191
22.4 游戏界面模块	192
22.5 index	194
本课小结	195

单元 5 迭代子

第 23 课 迭代子概述	199
23.1 迭代子的定义	200
23.2 for.of 语句	200
23.3 spread	202
23.4 迭代器——观察藏在迭代子下的机制	206
本课小结	210
第 24 课 集合	211
24.1 创建集合	212
24.2 使用集合	213
24.3 WeakSet 简介	216
本课小结	217

第 25 课	映射	219
25.1	创建映射	220
25.2	使用映射	221
25.3	何时使用映射	223
25.4	WeakMap 简介	227
	本课小结	227

第 26 课	顶点练习: 21 点游戏	229
26.1	若干张纸牌和一副牌	230
26.2	让 CPU 缓慢运行, 便于玩家 观察	234
26.3	将各部分代码组合 在一起	235
	本课小结	239

单元 6 类

第 27 课	类概述	243
27.1	类的声明	244
27.2	实例化类	246
27.3	导出类	247
27.4	类方法不绑定	248
27.5	在类定义中设置实例 属性	249
27.6	静态属性	251
	本课小结	252

第 28 课	扩展类	255
28.1	继承	256
28.2	super	257
28.3	继承类时常见的错误	259
	本课小结	260

第 29 课	顶点项目: 彗星游戏 (Comets)	263
29.1	创建可控的精灵	264
29.2	添加彗星	265

29.3	发射火箭	266
29.4	物体碰撞时	268
29.5	添加爆炸效果	270
	本课小结	271

单元 7 异步工作

第 30 课	promise	275
30.1	使用 promise	276
30.2	错误处理	278
30.3	promise 辅助函数	279
	本课小结	282

第 31 课	高级 promise	283
31.1	创建 promise	284
31.2	嵌套的 promise	286
31.3	捕获错误	288
	本课小结	291

第 32 课	异步函数	293
32.1	使用生成器的异步代码	294
32.2	异步函数概述	296
32.3	异步函数中的错误处理	297
	本课小结	299

第 33 课	observable	301
33.1	创建 observable	302
33.2	组合(编写)observable	304
33.3	创建 observable 组合器	305
	本课小结	306

第 34 课	顶点项目: 画布画廊	307
34.1	获取图片	308
34.2	在画布上绘制图片	309
34.3	重复过程	311
	本课小结	314

附录	习题答案	315
----	------------	-----

第 1 课

ECMAScript 规范和提案流程

在本课中，你将了解到 JavaScript 的起源以及 JavaScript 和 ECMAScript 之间的区别。由于本书会介绍 ES2015 以及后续版本中所引入的新特性，因此在本课中，你将了解到人们如何提出 JavaScript 的这些新特性以及这些提案如何成为该语言规范的一部分。

1.1 ECMAScript 简史

在 1995 年，Netscape 创建了最初的 JavaScript。此后，JavaScript 被提交给 Ecma 国际进行标准化，在 1997 年，发布了 ECMAScript 的第一个版本。Ecma 国际曾经被称为欧洲计算机制造商协会(The European Computer Manufacturers Association, ECMA)，但是后来更名为“Ecma 国际”，以彰显其全球地位。即使 Ecma 已不再是一个缩写，ECMAScript 仍然使用大写的 ECMA。ECMAScript 发布后，在接下来的两年内，每一年都会发布一个更新的版本。在 1999 年 12 月，发布了 ECMAScript 的第 3 版，通常被称为 ES3。

ECMAScript 的第 4 版(ES4)有了一个根本性的转变，它引入了很多新的概念(如类和接口)，并且是静态类型化的。它没有向后兼容 ES3。这意味着，如果实现 ES4，那么这就可能打破市面上现有的 JavaScript 应用程序。毋庸置疑，大家对 ES4 议论纷纷，Ecma 技术委员会最终分裂，形成一个小委员会，致力于一个小得