

前言

关于 JavaScript，谷歌公司的一位技术经理曾经跟我分享过一个无法反驳的观点。他说 JavaScript 并不是一门真正有内聚力的编程语言，至少形式上不是。ECMA-262 规范定义了 JavaScript，但 JavaScript 没有唯一的实现。更重要的是，这门语言与其宿主关系密切。实际上宿主为 JavaScript 定义了与外界交互所需的全部 API：DOM、网络请求、系统硬件、存储、事件、文件、加密，还有数以百计的其他 API。各种浏览器及其 JavaScript 引擎都按照自己的理解实现了这些规范。Chrome 有 Blink/V8，Firefox 有 Gecko/SpiderMonkey，Safari 有 WebKit/JavaScriptCore，微软有 Trident/EdgeHTML/Chakra。浏览器以合规的方式运行绝大多数 JavaScript，但 Web 上随处可见迎合各种浏览器偏好的页面。因此，对 JavaScript 更准确的定位应该是一组浏览器实现。

Web 纯化论者可能认为 JavaScript 本身并非网页不可或缺的部分，但他们必须承认，如果没有 JavaScript，那么现代 Web 势必发生严重倒退。毫不夸张地讲，JavaScript 才是真正不可或缺的。如今，手机、计算机、平板设备、电视、游戏机、智能手表、冰箱，甚至连汽车都内置了可以执行 JavaScript 代码的 Web 浏览器。地球上有近 30 亿人在使用安装了 Web 浏览器的智能手机。这门语言迅速发展的社区催生了大量高质量的开源项目。浏览器也已经支持模拟原生移动应用程序的 API。Stack Overflow 2019 年的开发者调查显示，JavaScript 连续七年位于最流行编程语言榜首。

我们正迎来 JavaScript 的文艺复兴。

本书将从 JavaScript 的起源讲起，从最初的 Netscape 浏览器直到今天各家浏览器支持的让人眼花缭乱的技术。全书对大量高级技术进行了鞭辟入里的剖析，以确保读者真正理解这些技术并掌握它们的应用场景。简而言之，通过学习本书，读者可以透彻地理解如何选择恰当的 JavaScript 技术，以解决现实开发中遇到的业务问题。

读者对象

本书适合以下读者阅读。

- ❑ 有经验的开发者，熟悉面向对象编程，因为 JavaScript 与 Java 和 C++ 等传统面向对象（OO，Object Oriented）语言的关系而希望学习 JavaScript。
- ❑ Web 应用程序开发者，希望增强自己的网站或 Web 应用程序的易用性。

❑ 初级 JavaScript 开发者，希望更好地理解这门语言。

此外，熟悉以下相关技术对阅读本书非常有帮助。

- ❑ Java
- ❑ PHP
- ❑ Python
- ❑ Ruby
- ❑ Golang
- ❑ HTML
- ❑ CSS

本书内容

本书第 4 版全面深入地介绍了 JavaScript 开发者必须掌握的前端开发技术，涉及 JavaScript 的基础特性和高级特性。

本书从 JavaScript 的起源开始，逐步讲解到今天的最新技术。书中详尽讨论了 JavaScript 的各个方面，重点介绍 ECMAScript 和 DOM 标准。

在此基础上，接下来的各章揭示了 JavaScript 的基本概念，包括类、期约、迭代器、代理，等等。另外，书中还深入探讨了客户端检测、事件、动画、表单、错误处理及 JSON。

本书最后介绍近几年来涌现的最新和最重要的规范，包括 Fetch API、模块、工作者线程、服务线程以及大量新 API。

组织结构

本书包含如下这些章。多个章节配有免费视频课二维码，扫描即可观看。

第 1 章，介绍 JavaScript 的起源：从哪里来，如何发展，以及现今的状况。这一章会谈到 JavaScript 与 ECMAScript 的关系、DOM、BOM，以及 Ecma 和 W3C 相关的标准。

第 2 章，了解 JavaScript 如何与 HTML 结合起来创建动态网页，主要介绍在网页中嵌入 JavaScript 的不同方式，还有 JavaScript 的内容类型及其与<script>元素的关系。

第 3 章，介绍语言的基本概念，包括语法和流控制语句；解释 JavaScript 与其他类 C 语言在语法上的异同点。在讨论内置操作符时也会谈到强制类型转换。此外还将介绍所有的原始类型，包括 Symbol。

第 4 章，探索 JavaScript 松散类型下的变量处理。这一章将涉及原始类型与引用类型的不同，以及与变量有关的执行上下文。此外，这一章也会讨论 JavaScript 中的垃圾回收，涉及在变量超出作用域时如何回收内存。

第 5 章，讨论 JavaScript 所有内置的引用类型，如 Date、Regexp、原始类型及其包装类型。每种

引用类型既有理论上的讲解，也有相关浏览器实现的剖析。

第 6 章，继续讨论内置引用类型，包括 `Object`、`Array`、`Map`、`WeakMap`、`Set` 和 `WeakSet` 等。

第 7 章，介绍 ECMAScript 新版中引入的两个基本概念：迭代器和生成器，并分别讨论它们最基本的行为和在当前语言环境下的应用。

第 8 章，解释如何在 JavaScript 中使用类和面向对象编程。首先会深入讨论 JavaScript 的 `Object` 类型，进而探讨原型式继承，接下来全面介绍 ES6 类及其与原型式继承的紧密关系。

第 9 章，介绍两个紧密相关的概念：`Proxy`（代理）和 `Reflect`（反射）API。代理和反射用于拦截和修改这门语言的基本操作。

第 10 章，探索 JavaScript 最强大的一个特性：函数表达式，主要涉及闭包、`this` 对象、模块模式、创建私有对象成员、箭头函数、默认参数和扩展操作符。

第 11 章，介绍两个紧密相关的异步编程构造：`Promise` 类型和 `async/await`。这一章讨论 JavaScript 的异步编程范式，进而介绍期约（`promise`）与异步函数的关系。

第 12 章，介绍 BOM，即浏览器对象模型，跟与浏览器本身交互的 API 相关。所有 BOM 对象都会涉及，包括 `window`、`document`、`location`、`navigator` 和 `screen` 等。

第 13 章，解释检测客户端机器及其能力的不同手段，包括能力检测 and 用户代理字符串检测。这一章讨论每种手段的优缺点，以及适用的场景。

第 14 章，介绍 DOM，即文档对象模型，主要是 DOM Level 1 定义的 API。这一章将简单讨论 XML 及其与 DOM 的关系，进而全面探索 DOM 以及如何利用它操作网页。

第 15 章，解释其他 DOM API，包括浏览器本身对 DOM 的扩展，主要涉及 `Selectors API`、`Element Traversal API` 和 HTML5 扩展。

第 16 章，在之前两章的基础上，解释 DOM Level 2 和 Level 3 对 DOM 的扩展，包括新增的属性、方法和对象。这一章还会介绍 DOM4 的相关内容，比如 `Mutation Observer`。

第 17 章，解释事件在 JavaScript 中的本质，以及事件的起源及其在 DOM 中的运行方式。

第 18 章，围绕 `<canvas>` 标签讨论如何创建动态图形，包括 2D 和 3D 上下文（`WebGL`）等动画和游戏开发所需的基础。这一章还会讨论 `WebGL1` 和 `WebGL2`。

第 19 章，探索使用 JavaScript 增强表单交互及突破浏览器限制，主要讨论文本框、选择框等表单元素及数据验证和操作。

第 20 章，介绍各种 JavaScript API，包括 `Atoms`、`Encoding`、`File`、`Blob`、`Notifications`、`Streams`、`Timing`、`Web Components` 和 `Web Cryptography`。

第 21 章，讨论浏览器如何处理 JavaScript 代码中的错误及几种错误处理方式。这一章同时介绍了每种浏览器的调试工具和技术，包括简化调试过程的建议。

第 22 章, 介绍通过 JavaScript 读取和操作 XML 数据的特性, 解释了不同浏览器支持特性和对象的差异, 提供了简化跨浏览器编码的建议。这一章也讨论了使用 XSLT 在客户端转换 XML 数据。

第 23 章, 介绍作为 XML 替代的 JSON 数据格式, 还讨论了浏览器原生解析和序列化 JSON, 以及使用 JSON 时要注意的安全问题。

第 24 章, 探讨浏览器请求数据和资源的常用方式, 包括早期的 XMLHttpRequest 和现代的 Fetch API。

第 25 章, 讨论应用程序离线时在客户端机器上存储数据的各种技术。先从 cookie 谈起, 然后讨论 Web Storage 和 IndexedDB。

第 26 章, 介绍模块模式在编码中的应用, 进而讨论 ES6 模块之前的模块加载方式, 包括 CommonJS、AMD 和 UMD。最后介绍新的 ES6 模块及其正确用法。

第 27 章, 深入介绍专用工作者线程、共享工作者线程和服务工作者线程。其中包括工作者线程在操作系统和浏览器层面的实现, 以及使用各种工作者线程的最佳策略。

第 28 章, 探讨在企业级开发中进行 JavaScript 编码的最佳实践。其中提到了提升代码可维护性的编码惯例, 包括编码技巧、格式化及通用编码建议。深入讨论应用性能和提升速度的技术。最后介绍与上线部署相关的话题, 包括项目构建流程。

预备条件

要运行本书示例代码, 需要如下条件。

- ❑ 现代操作系统, 包括 Windows、Linux、Mac OS、Android 或 iOS。
- ❑ 现代浏览器, 如 IE11+、Edge 12+、Firefox 26+、Chrome 39+、Safari 10+、Opera 26+ 或 iOS Safari 10+。

扫描封底二维码, 可以下载本书源代码, 并加入图灵前端研发小组。^①

电子书及附录

扫描下方二维码, 即可购买本书中文版电子书, 并从“随书下载”处获取本书电子版附录。



^① 读者也可访问本书图灵社区页面 (<https://www.ituring.com.cn/book/2472>) 下载本书配套学习资源, 并提交中文版勘误。

——编者注

致 谢

感谢 Wiley 出版社让我接手这本书。编写本书第 4 版对我来说是前所未有的挑战，也让我收获非常大。来自 Wiley 的包容和支持是本书得以完成的前提。感谢 Wiley 的工作人员，特别是把这本书交到我手上并紧盯着整个流程的 Jim Minatel。

感谢本书前 3 版的作者 Nicholas C. Zakas，感谢他在我接手之前所做的一切。没有他之前打下的良好基础，就不会有本书今天的成就。衷心祝愿他早日康复。

特别感谢 Adaobi Obi Tulton 的指导。如果没有她对整个流程的把控，以及她的耐心和专业水准，我不可能写完这一版。

还要感谢对本书草稿给出反馈意见的所有人：Samuel Kallner、Chaim Krause、Marcia Wilbur、Nancy Rapoport、Athyappan Lalith Kumar，还有 Evelyn Wellborn。这样一本书，少了你们任何人的帮助，都不会像现在这么完善。

最后，我想感谢 Zach Tratar 为本书作序。我非常幸运地在搬到旧金山的头一天就认识了 Zach Tratar。几年来，作为良师益友，他的求知若渴和博学多才一直感染着我，何况他还是一位杰出的软件工程师。他同意为本书作序是我的荣幸。

目 录

第 1 章 什么是 JavaScript	1	3.3 变量	24
1.1 简短的历史回顾	1	3.3.1 var 关键字	24
1.2 JavaScript 实现	2	3.3.2 let 声明	25
1.2.1 ECMAScript	2	3.3.3 const 声明	28
1.2.2 DOM	6	3.3.4 声明风格及最佳实践	29
1.2.3 BOM	8	3.4 数据类型	30
1.3 JavaScript 版本	9	3.4.1 typeof 操作符	30
1.4 小结	10	3.4.2 Undefined 类型	30
第 2 章 HTML 中的 JavaScript	11	3.4.3 Null 类型	32
2.1 <script>元素	11	3.4.4 Boolean 类型	33
2.1.1 标签位置	13	3.4.5 Number 类型	33
2.1.2 推迟执行脚本	14	3.4.6 String 类型	38
2.1.3 异步执行脚本	14	3.4.7 Symbol 类型	44
2.1.4 动态加载脚本	15	3.4.8 Object 类型	56
2.1.5 XHTML 中的变化	16	3.5 操作符	56
2.1.6 废弃的语法	17	3.5.1 一元操作符	56
2.2 行内代码与外部文件	18	3.5.2 位操作符	59
2.3 文档模式	18	3.5.3 布尔操作符	64
2.4 <noscript>元素	19	3.5.4 乘性操作符	66
2.5 小结	20	3.5.5 指数操作符	67
第 3 章 语言基础	21	3.5.6 加性操作符	68
3.1 语法	21	3.5.7 关系操作符	69
3.1.1 区分大小写	21	3.5.8 相等操作符	70
3.1.2 标识符	21	3.5.9 条件操作符	72
3.1.3 注释	22	3.5.10 赋值操作符	72
3.1.4 严格模式	22	3.5.11 逗号操作符	73
3.1.5 语句	22	3.6 语句	73
3.2 关键字与保留字	23	3.6.1 if 语句	73
		3.6.2 do-while 语句	74
		3.6.3 while 语句	74

3.6.4	for 语句	74	5.3.2	Number	115
3.6.5	for-in 语句	75	5.3.3	String	117
3.6.6	for-of 语句	76	5.4	单例内置对象	128
3.6.7	标签语句	76	5.4.1	Global	128
3.6.8	break 和 continue 语句	76	5.4.2	Math	132
3.6.9	with 语句	78	5.5	小结	135
3.6.10	switch 语句	78	第 6 章	集合引用类型	136
3.7	函数	80	6.1	Object	136
3.8	小结	82	6.2	Array	138
第 4 章	变量、作用域与内存	83	6.2.1	创建数组	138
4.1	原始值与引用值	83	6.2.2	数组空位	140
4.1.1	动态属性	83	6.2.3	数组索引	141
4.1.2	复制值	84	6.2.4	检测数组	142
4.1.3	传递参数	85	6.2.5	迭代器方法	143
4.1.4	确定类型	86	6.2.6	复制和填充方法	143
4.2	执行上下文与作用域	87	6.2.7	转换方法	145
4.2.1	作用域链增强	89	6.2.8	栈方法	147
4.2.2	变量声明	90	6.2.9	队列方法	147
4.3	垃圾回收	94	6.2.10	排序方法	148
4.3.1	标记清理	95	6.2.11	操作方法	150
4.3.2	引用计数	95	6.2.12	搜索和位置方法	151
4.3.3	性能	96	6.2.13	迭代方法	153
4.3.4	内存管理	97	6.2.14	归并方法	154
4.4	小结	101	6.3	定型数组	155
第 5 章	基本引用类型	103	6.3.1	历史	155
5.1	Date	103	6.3.2	ArrayBuffer	155
5.1.1	继承的方法	105	6.3.3	DataView	156
5.1.2	日期格式化方法	106	6.3.4	定型数组	159
5.1.3	日期/时间组件方法	106	6.4	Map	163
5.2	RegExp	107	6.4.1	基本 API	164
5.2.1	RegExp 实例属性	109	6.4.2	顺序与迭代	166
5.2.2	RegExp 实例方法	109	6.4.3	选择 Object 还是 Map	168
5.2.3	RegExp 构造函数属性	111	6.5	WeakMap	168
5.2.4	模式局限	113	6.5.1	基本 API	168
5.3	原始值包装类型	113	6.5.2	弱键	170
5.3.1	Boolean	114	6.5.3	不可迭代键	170
			6.5.4	使用弱映射	171

6.6 Set	173	8.2.3 构造函数模式	221
6.6.1 基本 API	173	8.2.4 原型模式	224
6.6.2 顺序与迭代	175	8.2.5 对象迭代	233
6.6.3 定义正式集合操作	176	8.3 继承	238
6.7 WeakSet	178	8.3.1 原型链	238
6.7.1 基本 API	178	8.3.2 盗用构造函数	243
6.7.2 弱值	179	8.3.3 组合继承	244
6.7.3 不可迭代值	180	8.3.4 原型式继承	245
6.7.4 使用弱集合	180	8.3.5 寄生式继承	246
6.8 迭代与扩展操作	180	8.3.6 寄生式组合继承	247
6.9 小结	182	8.4 类	249
第 7 章 迭代器与生成器	183	8.4.1 类定义	249
7.1 理解迭代	183	8.4.2 类构造函数	250
7.2 迭代器模式	184	8.4.3 实例、原型和类成员	254
7.2.1 可迭代协议	184	8.4.4 继承	258
7.2.2 迭代器协议	186	8.5 小结	265
7.2.3 自定义迭代器	188	第 9 章 代理与反射	266
7.2.4 提前终止迭代器	190	9.1 代理基础	266
7.3 生成器	192	9.1.1 创建空代理	266
7.3.1 生成器基础	192	9.1.2 定义捕获器	267
7.3.2 通过 yield 中断执行	194	9.1.3 捕获器参数和反射 API	268
7.3.3 生成器作为默认迭代器	201	9.1.4 捕获器不变式	270
7.3.4 提前终止生成器	202	9.1.5 可撤销代理	271
7.4 小结	204	9.1.6 实用反射 API	271
第 8 章 对象、类与面向对象编程	205	9.1.7 代理另一个代理	273
8.1 理解对象	205	9.1.8 代理的问题与不足	273
8.1.1 属性的类型	206	9.2 代理捕获器与反射方法	274
8.1.2 定义多个属性	208	9.2.1 get()	275
8.1.3 读取属性的特性	209	9.2.2 set()	275
8.1.4 合并对象	210	9.2.3 has()	276
8.1.5 对象标识及相等判定	213	9.2.4 defineProperty()	277
8.1.6 增强的对象语法	213	9.2.5 getOwnProperty-	
8.1.7 对象解构	216	Descriptor()	277
8.2 创建对象	220	9.2.6 deleteProperty()	278
8.2.1 概述	220	9.2.7 ownKeys()	279
8.2.2 工厂模式	220	9.2.8 getPrototypeOf()	279
		9.2.9 setPrototypeOf()	280

9.2.10	isExtensible()	280	10.16	私有变量	316
9.2.11	preventExtensions()	281	10.16.1	静态私有变量	317
9.2.12	apply()	281	10.16.2	模块模式	318
9.2.13	construct()	282	10.16.3	模块增强模式	320
9.3	代理模式	283	10.17	小结	321
9.3.1	跟踪属性访问	283	第 11 章	期约与异步函数	322
9.3.2	隐藏属性	283	11.1	异步编程	322
9.3.3	属性验证	284	11.1.1	同步与异步	322
9.3.4	函数与构造函数参数验证	284	11.1.2	以往的异步编程模式	323
9.3.5	数据绑定与可观察对象	285	11.2	期约	325
9.4	小结	286	11.2.1	Promises/A+规范	325
第 10 章	函数	287	11.2.2	期约基础	325
10.1	箭头函数	288	11.2.3	期约的实例方法	329
10.2	函数名	289	11.2.4	期约连锁与期约合成	338
10.3	理解参数	290	11.2.5	期约扩展	345
10.4	没有重载	292	11.3	异步函数	347
10.5	默认参数值	293	11.3.1	异步函数	348
10.6	参数扩展与收集	295	11.3.2	停止和恢复执行	353
10.6.1	扩展参数	295	11.3.3	异步函数策略	356
10.6.2	收集参数	296	11.4	小结	360
10.7	函数声明与函数表达式	297	第 12 章	BOM	361
10.8	函数作为值	297	12.1	window 对象	361
10.9	函数内部	299	12.1.1	Global 作用域	361
10.9.1	arguments	299	12.1.2	窗口关系	362
10.9.2	this	300	12.1.3	窗口位置与像素比	362
10.9.3	caller	301	12.1.4	窗口大小	363
10.9.4	new.target	301	12.1.5	视口位置	364
10.10	函数属性与方法	302	12.1.6	导航与打开新窗口	365
10.11	函数表达式	304	12.1.7	定时器	368
10.12	递归	306	12.1.8	系统对话框	370
10.13	尾调用优化	307	12.2	location 对象	372
10.13.1	尾调用优化的条件	307	12.2.1	查询字符串	372
10.13.2	尾调用优化的代码	309	12.2.2	操作地址	373
10.14	闭包	309	12.3	navigator 对象	375
10.14.1	this 对象	312	12.3.1	检测插件	376
10.14.2	内存泄漏	314	12.3.2	注册处理程序	378
10.15	立即调用的函数表达式	314			

12.4	screen 对象	379	14.3.2	MutationObserverInit 与观察范围	437
12.5	history 对象	379	14.3.3	异步回调与记录队列	442
12.5.1	导航	379	14.3.4	性能、内存与垃圾回收	443
12.5.2	历史状态管理	380	14.4	小结	444
12.6	小结	381	第 15 章	DOM 扩展	445
第 13 章	客户端检测	382	15.1	Selectors API	445
13.1	能力检测	382	15.1.1	querySelector()	445
13.1.1	安全能力检测	383	15.1.2	querySelectorAll()	446
13.1.2	基于能力检测进行浏览器 分析	384	15.1.3	matches()	447
13.2	用户代理检测	386	15.2	元素遍历	447
13.2.1	用户代理的历史	386	15.3	HTML5	448
13.2.2	浏览器分析	392	15.3.1	CSS 类扩展	448
13.3	软件与硬件检测	394	15.3.2	焦点管理	450
13.3.1	识别浏览器与操作系统	394	15.3.3	HTMLDocument 扩展	450
13.3.2	浏览器元数据	395	15.3.4	字符集属性	451
13.3.3	硬件	400	15.3.5	自定义数据属性	451
13.4	小结	400	15.3.6	插入标记	452
第 14 章	DOM	401	15.3.7	scrollIntoView()	456
14.1	节点层级	401	15.4	专有扩展	456
14.1.1	Node 类型	402	15.4.1	children 属性	456
14.1.2	Document 类型	407	15.4.2	contains() 方法	457
14.1.3	Element 类型	414	15.4.3	插入标记	457
14.1.4	Text 类型	420	15.4.4	滚动	459
14.1.5	Comment 类型	423	15.5	小结	459
14.1.6	CDATASection 类型	423	第 16 章	DOM2 和 DOM3	460
14.1.7	DocumentType 类型	424	16.1	DOM 的演进	460
14.1.8	DocumentFragment 类型	424	16.1.1	XML 命名空间	461
14.1.9	Attr 类型	425	16.1.2	其他变化	464
14.2	DOM 编程	426	16.2	样式	467
14.2.1	动态脚本	426	16.2.1	存取元素样式	467
14.2.2	动态样式	428	16.2.2	操作样式表	470
14.2.3	操作表格	429	16.2.3	元素尺寸	472
14.2.4	使用 NodeList	431	16.3	遍历	476
14.3	MutationObserver 接口	432	16.3.1	NodeIterator	478
14.3.1	基本用法	433	16.3.2	TreeWalker	480