

TURING

图灵程序设计丛书

[PACKT]
PUBLISHING

[印度] Ved Antani 著 门佳 译

JavaScript 编程精粹

Mastering JavaScript

掌握JavaScript基础知识要点及其现代技术和工具，
用正确的编码风格开发Web应用



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

JavaScript 编程精粹



Mastering JavaScript

[印度] Ved Antani 著 门佳 译

人民邮电出版社
北 京

图书在版编目 (C I P) 数据

JavaScript编程精粹 / (印) 韦德·安塔尼
(Ved Antani) 著 ; 门佳译. — 北京 : 人民邮电出版社,
2017.8

(图灵程序设计丛书)
ISBN 978-7-115-46569-6

I. ①J… II. ①韦… ②门… 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181430号

内 容 提 要

本书详细介绍讲述了 JavaScript 的基础知识以及一些现代语言工具和库, 例如 jQuery、Underscore.js 和 Jasmine。主要内容包括: JavaScript 基础知识, 函数、闭包和模块, 数据结构和相关处理, 面向对象的 JavaScript, JavaScript 设计模式, 测试与调试, ECMAScript 6, DOM 事件和操作, 服务器端 JavaScript。

本书适合所有 JavaScript 开发人员阅读。

-
- ◆ 著 [印度] Ved Antani
 - 译 门 佳
 - 责任编辑 岳新欣
 - 执行编辑 赵雪梅
 - 责任印制 彭志环
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 11.5
 - 字数: 272千字 2017年8月第1版
 - 印数: 1—3 500册 2017年8月河北第1次印刷
 - 著作权合同登记号 图字: 01-2016-6690号
-

定价: 39.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

站在巨人的肩上
Standing on Shoulders of Giants



iTuring.cn

版 权 声 明

Copyright © 2016 Packt Publishing. First published in the English language under the title *Mastering JavaScript*.

Simplified Chinese-language edition copyright © 2017 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Packt Publishing授权人民邮电出版社独家出版。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。

致 谢

感谢妻子Meghna对我的支持。她总会在我最需要的时候给予我鼓励和帮助。

前言

看起来，与JavaScript有关的内容该写的都已经写过了。坦白说，很难找出JavaScript中还有什么没被人翻来覆去讨论过的话题。但是，JavaScript的发展速度太快了。ECMAScript 6有可能会给这门语言以及编码方式带来巨大的转变。Node.js已经改变了我们使用JavaScript编写服务器的方法，而像React和Flux这些新理念将会推动JavaScript的再次迭代。我们在学习这些新特性的同时，不应该忽略那些必须掌握的JavaScript基础知识。这些知识是根基，不容忽视。如果你已经是一位JavaScript的开发老手，会意识到现代JavaScript已经与大多数人记忆中的它大相径庭了。现代JavaScript不但要求遵循特定的编码风格以及严格的设计思路，而且编程工具的功能愈发强大，已经逐渐成为了开发流程中不可或缺的组成部分。尽管JavaScript在不断改变，但它是根植于一些非常坚实、稳定的概念之上的。本书强调的正是这些基本概念。

在本书写作之时，JavaScript的变化仍未停歇。幸运的是，所有重要的相关更新都已经包括在了本书中。

本书为你详细讲述了JavaScript的基础知识，以及一些现代的语言工具和库，例如jQuery、Underscore.js和Jasmine。

希望你能够享受阅读本书的过程，就像我们享受写作本书的过程一样。

内容简介

第1章，JavaScript入门。本章的重点在于语言构件（language construct），在基本细节方面不会花费过多篇幅。本章将讲述变量作用域和循环中不太容易掌握的地方，以及类型和数据结构的最佳使用实践。另外，还包括一些编码风格的相关知识，以及推荐的代码组织模式。

第2章，函数、闭包与模块。本章将讲述JavaScript语言错综复杂的核心，探讨在JavaScript中，因闭包的不同用法而造成的复杂性。本章细致详尽的讨论将为你今后深入学习更高级的设计模式奠定基础。

第3章，数据结构及相关操作。本章将详细讲述正则表达式和数组。数组是JavaScript的基本数据类型，本章将帮助你学会有效地使用数组。正则表达式能够使代码更简洁，我们将仔细讲解

如何充分发挥出正则表达式的功能。

第4章，面向对象的JavaScript。本章将讲述JavaScript中的面向对象编程，包括继承和原型，重点将放在理解JavaScript的原型继承模型上。另外，还将讨论该模型与其他面向对象模型之间的差异，以帮助Java或C++程序员熟悉这种变化。

第5章，JavaScript模式。本章将论述常见的设计模式以及如何在JavaScript中实现这些模式。一旦你掌握了JavaScript的面向对象模型，就更容易理解设计和编程模式，从而写出易于维护的模块化代码。

第6章，测试与调试。本章将讲述各种现代化的测试方法以及JavaScript的调试问题。另外，还将探究JavaScript的持续测试和测试驱动方法。本章中将采用Jasmine作为测试框架。

第7章，ECMAScript 6。本章将重点放在ECMAScript 6（ES6）所引入的新语言特性上。它们使得JavaScript的功能更加强大，本章将帮助你理解并使用这些新特性。

第8章，DOM操作与事件。本章将JavaScript作为一种浏览器语言进行了详尽的描述，讨论了DOM操作以及浏览器事件。

第9章，服务器端JavaScript。本章将讲解如何使用Node.js编写可伸缩的服务器系统，探讨Node.js的架构和一些实用技术。

阅读前提

本书中所有的示例都可以运行在任何现代浏览器中。学习最后一章时，你需要安装Node.js。运行书中的示例需要满足以下前提条件。

- ❑ 一台安装了Windows 7（或更高版本）、Linux或Mac OS X操作系统的计算机。
- ❑ 最新版的Google Chrome或Mozilla Firefox浏览器。
- ❑ 选择一款文本编辑器。Sublime Text、vi、Atom或者Notepad++都挺不错。选择权完全在你。

目标读者

本书旨在教授精通JavaScript必须掌握的基础知识，适合以下读者。

- ❑ 具备其他面向对象语言经验的开发人员。书中的内容能够帮助他们利用已有的经验转换到JavaScript。
- ❑ 比较熟悉JavaScript的Web开发人员。本书将帮助他们学习JavaScript的高级概念，改善他们的编程风格。

□ 希望了解并最终成为JavaScript高手的初学者。本书包含了必不可少的入门内容。

排版约定

在本书中，你会发现针对不同信息类型的文本样式。下面是这些样式的示例和解释。

正文中的代码、数据库表名、用户输入的样式如下：“第一种方式是通过<head>中的<script>标签导入JavaScript，第二种方式是利用<script>标签嵌入内联JavaScript。”

代码块的版式如下：

```
function sayHello(what) {  
    return "Hello " + what;  
}  
console.log(sayHello("world"));
```

当需要读者特别注意代码块中的某一部分时，相关的代码行和项将以粗体显示：

```
<head>  
  <script type="text/javascript" src="script.js"></script>  
  <script type="text/javascript">  
    var x = "Hello World";  
    console.log(x);  
  </script>  
</head>
```

命令行输入或输出形式如下：

```
EN-VedA:~$ node  
> 0.1+0.2  
0.30000000000000004  
> (0.1+0.2)===0.3  
false
```

新术语或重要词汇会以黑体显示。



此图标表示警告或重要的注意事项。



此图标表示提示和技巧。

读者反馈

欢迎读者反馈意见。我们想知道读者对于本书的看法——喜欢哪些内容或不喜欢哪些内容。读者反馈对于我们出版读者真正需要的图书至关重要。

如有反馈意见, 请将电子邮件发送到feedback@packtpub.com, 不要忘记在邮件标题中注明你要反馈的书名。

客户支持

现在你已经成为Packt图书的主人了, 为了使此书尽可能物有所值, 我们还提供了其他服务。

下载示例代码

你可以用你的账户从<http://www.packtpub.com>中下载所购买的所有Packt图书的示例代码。如果你是从其他地方购买的本书(英文版), 可以访问<http://www.packtpub.com/support>并注册, 以便通过电子邮件取得示例代码。

下载彩色图片

我们为你提供了一份PDF文件, 其中包含了书中出现的所有截屏和图示的彩色图片。这些彩色图片有助于你更好地理解输出内容的变化。你可以从这里下载到该文件: https://www.packtpub.com/sites/default/files/downloads/MasteringJavaScript_ColorImages.pdf。

勘误

尽管我们已竭尽全力确保本书内容的准确性, 但错误终难避免。如果你发现了书中的错误(不管是文字错误还是代码错误)并愿意告知我们, 我们将非常感激。这样不仅可以减少其他读者的疑惑, 也有助于本书后续版本的改进。要提交所发现的错误, 请访问<http://www.packtpub.com/submit-errata>, 选择书名, 点击Errata Submission Form(勘误提交表单)链接, 输入详细的错误信息。^①一旦勘误得到核实, 我们将接受你的提交, 同时勘误内容也会被上传到我们的网站, 或是添加到对应书目勘误区的现有勘误表中。

要想查看之前提交的勘误, 进入<https://www.packtpub.com/books/content/support>, 在搜索框中

^① 本书中文版勘误可到www.it-ebooks.com.cn/book/2069查看和提交。——编者注

输入书名即可。你所需的信息会出现在Errata（勘误）下方。

举报盗版

各类媒体在网络上一一直饱受版权侵害的困扰。Packt坚持不懈地严格保护版权和授权。如果你在網上发现了我社图书的任何形式的盗版，请立即为我们提供地址或网站名称，以便我们采取措施。

请将疑似侵权的网站链接发送至copyright@packtpub.com。

衷心感谢你为保护作者的知识产权以及我们的劳动成果所做的工作。

疑难解答

如果你对本书的任何方面有疑问，请通过questions@packtpub.com联系我们，我们将尽力为你解决。

电子书

扫描如下二维码，即可购买本书电子版。



目 录

第 1 章 JavaScript 入门	1	2.12 小结	55
1.1 JavaScript 极简史	2	第 3 章 数据结构及相关操作	56
1.2 如何阅读本书	2	3.1 正则表达式	56
1.3 Hello World	4	3.2 严格匹配	57
1.3.1 JavaScript 概览	5	3.3 匹配字符组	58
1.3.2 JavaScript 类型	21	3.4 重复出现	61
1.3.3 自动插入分号	23	3.5 首部与尾部	63
1.3.4 JavaScript 代码风格指南	25	3.6 向后引用	63
1.4 小结	33	3.7 贪婪限定符与惰性限定符	64
第 2 章 函数、闭包与模块	34	3.8 数组	65
2.1 函数的字面形式	34	3.9 map	71
2.2 函数作为数据	36	3.10 set	72
2.3 作用域	38	3.11 编码风格	73
2.3.1 全局作用域	38	3.12 小结	74
2.3.2 局部作用域	39	第 4 章 面向对象的 JavaScript	75
2.3.3 函数作用域与块作用域	39	4.1 理解对象	75
2.3.4 行内函数表达式	42	4.1.1 JavaScript 对象的行为	77
2.3.5 块作用域	42	4.1.2 原型	78
2.4 函数声明与函数表达式	44	4.2 实例属性与原型属性	79
2.5 arguments 参数	45	4.3 继承	83
2.6 匿名函数	48	4.4 接收器与设置器	89
2.6.1 对象创建过程中的匿名函数	48	4.5 小结	91
2.6.2 列表创建过程中的匿名函数	49	第 5 章 JavaScript 模式	92
2.6.3 作为函数参数的匿名函数	49	5.1 设计模式	92
2.6.4 出现在条件逻辑中的匿名函数	49	5.2 命名空间模式	94
2.7 闭包	50	5.3 模块模式	95
2.8 计时器和回调函数	52	5.4 工厂模式	100
2.9 私有变量	53	5.5 mixin 模式	101
2.10 循环与闭包	53	5.6 装饰器模式	102
2.11 模块	54		

5.7 观察者模式	104	7.3.7 Map 与 Set	131
5.8 JavaScript 的 Model-View-* 模式	106	7.3.8 Symbol	133
5.8.1 模型-视图-控制器	106	7.3.9 迭代器	134
5.8.2 模型	106	7.3.10 for..of 循环	134
5.8.3 视图	107	7.3.11 箭头函数	134
5.8.4 控制器	107	7.4 小结	137
5.9 模型-视图-表现器	107		
5.10 模型-视图-视图模型	108	第 8 章 DOM 操作与事件	138
5.11 小结	109	8.1 DOM	138
第 6 章 测试与调试	110	8.1.1 访问 DOM 元素	138
6.1 单元测试	111	8.1.2 访问特定的节点	140
6.1.1 测试驱动开发	111	8.2 链式方法	145
6.1.2 行为驱动开发	112	8.3 遍历与操作	146
6.2 JavaScript 调试	117	8.4 处理浏览器事件	147
6.2.1 语法错误	117	8.5 事件传播	148
6.2.2 使用严格模式	117	8.6 jQuery 事件处理及传播	149
6.2.3 运行时异常	118	8.7 事件委托	151
6.3 小结	123	8.8 事件对象	152
第 7 章 ECMAScript 6	124	8.9 小结	153
7.1 shim/polyfill	124	第 9 章 服务器端 JavaScript	154
7.2 转换编译器	125	9.1 浏览器的异步事件模型	154
7.3 ES6 语法上的变化	125	9.2 回调函数	158
7.3.1 块级作用域	125	9.3 计时器	160
7.3.2 默认参数	127	9.4 EventEmitter	161
7.3.3 spread 与 rest	127	9.5 模块	162
7.3.4 解构	128	9.6 npm	165
7.3.5 对象字面量	129	9.7 JavaScript 性能	166
7.3.6 模板字面量	130	9.8 小结	169

第 1 章

JavaScript入门



万事开篇难，尤其是对于像JavaScript这种主题。为难之处主要在于这门语言已经被讨论得太多了。JavaScript已经成为Web世界的语言——在Netscape Navigator出现之初，它就作为一种通用语言存在了。从业余爱好者的一件工具摇身变为行家手中的利器，JavaScript发展之快令人瞠目结舌。

JavaScript是Web上最流行的语言和开源生态环境。<http://github.info/>跟踪记录了这门语言过去几年中在GitHub上活跃仓库的数量以及整体受欢迎程度。JavaScript如此流行且重要，归功于它同浏览器的紧密结合。作为经过极度优化的JavaScript引擎，Google的V8和Mozilla的SpiderMonkey分别为Google Chrome和Mozilla Firefox浏览器的壮大贡献了一臂之力。

尽管Web浏览器是JavaScript应用最广泛的平台，但如今像MongoDB和CouchDB这类现代数据库，也开始使用JavaScript作为其脚本和查询语言。JavaScript已经成为浏览器之外的一个重要平台。例如，Node.js和io.js项目就为使用JavaScript开发可伸缩的服务器环境提供了强有力的平台；另外一些引入瞩目的项目更是将这门语言的性能推向了极限，例如Emscripten（<http://kripken.github.io/emscripten-site/>），这是一个基于低层虚拟机（Low-Level Virtual Machine, LLVM）的项目，它能够将C和C++代码编译成高度优化的asm.js格式的JavaScript代码。这使得你可以在Web上以接近于原生速度运行C/C++代码。

JavaScript有着坚实的构建基础，其中包括函数、动态对象、松散类型、原型继承以及强大的对象字面记法。

尽管JavaScript的设计理念非常不错，但遗憾的是，它只能跟随浏览器的发展。在支持各种特性和标准方面，Web浏览器声名狼藉。JavaScript试图去适应浏览器中各种奇思妙想的设计，结果作出了一些非常糟糕的设计决策。多数人让这些不良成分（bad parts，借由Douglas Crockford火起来的一个术语）蒙蔽了双眼，忽视了语言中那些优秀的部分。一些程序员写出的差劲代码成为调试这些代码的另一批程序员的噩梦，而JavaScript则背负了骂名。因此，JavaScript不幸地成为被误解最多的编程语言之一（<http://javascript.crockford.com/javascript.html>）。

对JavaScript的另一种批评是，你不用真正掌握这门语言就可以把问题搞定。我见过一些程序

员写出了糟糕得令人发指的JavaScript代码，仅仅就是因为他们想尽快完成任务，而JavaScript偏偏也允许他们这么干。我曾花费大量时间来调试一个压根就算不上程序员的家伙编写的JavaScript代码，质量真是差劲到家了。但是你不能因为自己的编程水平不到家，就把怨气撒在语言身上，毕竟它只是一种工具而已。就像所有的手艺一样，编程也需要全身心的投入和自律。

1.1 JavaScript 极简史

1993年，美国国家超级计算机应用中心（NCSA）的Mosaic浏览器是当时流行的首批Web浏览器之一。一年后，网景公司推出了其专有的Web浏览器：Netscape Navigator。多名Mosaic的早期设计人员参与了Navigator的开发。

1995年，网景公司雇用了Brendan Eich，允诺他在浏览器中实现Scheme（一种Lisp方言）。在此之前，网景公司与Sun公司（已被Oracle公司收购）进行过接触，试图将Java纳入到Navigator浏览器中。

由于Java的流行性和易用性，网景公司决定自家浏览器中的脚本语言的语法必须和Java类似。这直接就把已有的那些语言，如Python、TCL（Tool Command Language，工具命令语言）或Scheme排除在外了。1995年5月，Eich只花了10天时间就写出了最初的原型（<http://www.computer.org/csdl/mags/co/2012/02/mco2012020007.pdf>）。JavaScript一开始的代号是Mocha，是由Marc Andreessen命名的。由于注册商标方面的原因，网景公司随后将其更名为LiveScript。1995年12月初，Sun公司将注册商标Java授权给了网景公司，于是这门语言终于拥有了最终的名字：JavaScript。

1.2 如何阅读本书

这不是一本快餐类读物。本书关注的是正确的JavaScript编程方式。我们会用大量的篇幅来学习如何避开该门语言中的不良成分，构建可靠、可读的JavaScript代码。为了避免对那些糟糕的语言特性形成依赖，我们选择对其敬而远之；如果已经养成了不好的编码习惯，本书将尽力帮你摆脱这种问题。本书会重点强调使用正确的风格和工具来改善代码。

本书中的大部分概念都反映了现实世界中的问题和模式。我坚持认为读者应该自己动手输入每一段代码，以确保彻底理解这些概念。相信我，要学编程，没有什么方法比动手编写大量代码更好的了。

通常需要创建一个HTML页面来运行嵌入的JavaScript代码：

```
<!DOCTYPE html>
<html>
<head>
  <script type="text/javascript" src="script.js"></script>
```

```
<script type="text/javascript">
  var x = "Hello World";
  console.log(x);
</script>
</head>
<body>
</body>
</html>
```

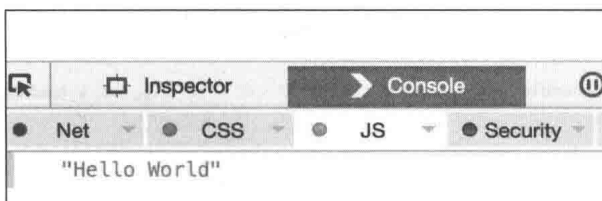
这段示例代码展示了在HTML页面嵌入JavaScript代码的两种方式。第一种方式是通过<head>中的<script>标签导入JavaScript，第二种方式是利用<script>标签嵌入内联JavaScript。

下载示例代码

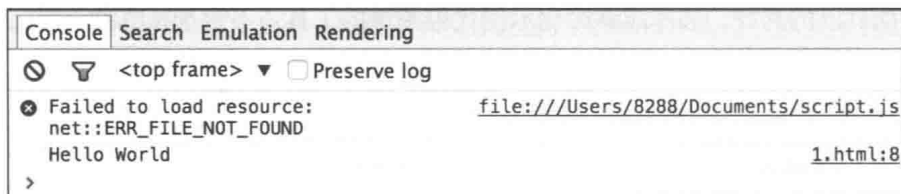


可以用你的账户从<http://www.packtpub.com>中下载所购买的所有Packt图书的示例代码。如果你是从其他地方买的(英文版),可以访问<http://www.packtpub.com/support>并注册,通过电子邮件取得示例代码。

可以把HTML页面保存在本地,然后在浏览器中打开。在Firefox中,打开Developer控制台(FireFox菜单 | Developer | Web Console),此时会看到在Console标签下出现了"Hello World"文本。在你用的操作系统和浏览器上,屏幕显示可能会不太一样:



打开页面,然后使用Chrome的Developer Tool来进行检查:



这里我们注意到一个很有意思的现象:在控制台中出现了一个丢失.js文件的错误信息,这个文件是通过下面这行代码导入的:

```
<script type="text/javascript" src="script.js"></script>
```

使用浏览器的开发者控制台或是扩展(如Firebug),可以非常方便地调试代码中出现的错误。在随后的章节中,将详细讨论相关的调试技巧。