

# 导论

Source: <https://wangdoc.com/javascript/basic/introduction.html>

网道（WangDoc.com），互联网文档计划

## 什么是 JavaScript 语言？ #

JavaScript 是一种轻量级的脚本语言。所谓“脚本语言”（script language），指的是它不具备开发操作系统的能力，而是只用来编写控制其他大型应用程序（比如浏览器）的“脚本”。

JavaScript 也是一种嵌入式（embedded）语言。它本身提供的核心语法不算很多，只能用来做一些数学和逻辑运算。

JavaScript 本身不提供任何与 I/O（输入/输出）相关的 API，都要靠宿主环境（host）提供，所以 JavaScript 只合适嵌入更大型的应用程序环境，去调用宿主环境提供的底层 API。

目前，已经嵌入 JavaScript 的宿主环境有多种，最常见的环境就是浏览器，另外还有服务器环境，也就是 Node 项目。

从语法角度看，JavaScript 语言是一种“对象模型”语言。各种宿主环境通过这个模型，描述自己的功能和操作接口，从而通过 JavaScript 控制这些功能。但是，JavaScript 并不是纯粹的“面向对象语言”，还支持其他编程范式（比如函数式编程）。这导致几乎任何一个问题，JavaScript 都有多种解决方法。阅读本书的过程中，你会诧异于 JavaScript 语法的灵活性。

JavaScript 的核心语法部分相当精简，只包括两个部分：基本的语法构造（比如操作符、控制结构、语句）和标准库（就是一系列具有各种功能的对象比如 Array、Date、Math 等）。除此之外，各种宿主环境提供额外的 API（即只能在该环境使用的接口），以便 JavaScript 调用。以浏览器为例，它提供的额外 API 可以分成三大类。

- 浏览器控制类：操作浏览器
- DOM 类：操作网页的各种元素
- Web 类：实现互联网的各种功能

如果宿主环境是服务器，则会提供各种操作系统的 API，比如文件操作 API、网络通信 API 等等。这些你都可以在 Node 环境中找到。

本书主要介绍 JavaScript 核心语法和浏览器网页开发的基本知识，不涉及 Node。全书可以分成以下四大部分。

- 基本语法
- 标准库
- 浏览器 API
- DOM

JavaScript 语言有多个版本。本书的内容主要基于 ECMAScript 5.1 版本，这是学习 JavaScript 语法的基础。ES6 和更新的语法请参考我写的 《ECMAScript 6 入门》 → <http://es6.ruanyifeng.com/>。

## 为什么学习 JavaScript? #

JavaScript 语言有一些显著特点，使得它非常值得学习。它既适合作为学习编程的入门语言，也适合当作日常开发的工作语言。它是目前最有希望、前途最光明的计算机语言之一。

## 操控浏览器的能力 #

JavaScript 的发明目的，就是作为浏览器的内置脚本语言，为网页开发者提供操控浏览器的能力。它是目前唯一一种通用的浏览器脚本语言，所有浏览器都支持。它可以让网页呈现各种特殊效果，为用户提供良好的互动体验。

目前，全世界几乎所有网页都使用 JavaScript。如果不用，网站的易用性和使用效率将大打折扣，无法成为操作便利、对用户友好的网站。

对于一个互联网开发者来说，如果你想提供漂亮的网页、令用户满意的上网体验、各种基于浏览器的便捷功能、前后端之间紧密高效的联系，JavaScript 是必不可少的工具。

## 广泛的使用领域 #

近年来，JavaScript 的使用范围，慢慢超越了浏览器，正在向通用的系统语言发展。

### （1）浏览器的平台化

随着 HTML5 的出现，浏览器本身的功能越来越强，不再仅仅能浏览网页，而是越来越像一个平台，JavaScript 因此得以调用许多系统功能，比如操作本地文件、操作图片、调用摄像头和麦克风等等。这使得 JavaScript 可以完成许多以前无法想象的事情。

### （2）Node

Node 项目使得 JavaScript 可以用于开发服务器端的大型项目，网站的前后端都用 JavaScript 开发已经成为了现实。有些嵌入式平台（Raspberry Pi）能够安装 Node，于是 JavaScript 就能为这些平台开发应用程序。

### （3）数据库操作

JavaScript 甚至也可以用来操作数据库。NoSQL 数据库这个概念，本身就是在 JSON（JavaScript Object Notation）格式的基础上诞生的，大部分 NoSQL 数据库允许 JavaScript 直接操作。基于 SQL 语言的开源数据库 PostgreSQL 支持 JavaScript 作为操作语言，可以部分取代 SQL 查询语言。

### （4）移动平台开发

JavaScript 也正在成为手机应用的开发语言。一般来说，安卓平台使用 Java 语言开发，iOS 平台使用 Objective-C 或 Swift 语言开发。许多人正在努力，让 JavaScript 成为各个平台的通用开发语言。

PhoneGap 项目就是将 JavaScript 和 HTML5 打包在一个容器之中，使得它能同时在 iOS 和安卓上运行。Facebook 公司的 React Native 项目则是将 JavaScript 写的组件，编译成原生组件，从而使它们具备优秀的性能。

Mozilla 基金会的手机操作系统 Firefox OS，更是直接将 JavaScript 作为操作系统的平台语言，但是很可惜这个项目没有成功。

### (5) 内嵌脚本语言

越来越多的应用程序，将 JavaScript 作为内嵌的脚本语言，比如 Adobe 公司的著名 PDF 阅读器 Acrobat、Linux 桌面环境 GNOME 3。

### (6) 跨平台的桌面应用程序

Chromium OS、Windows 8 等操作系统直接支持 JavaScript 编写应用程序。Mozilla 的 Open Web Apps 项目、Google 的 Chrome App 项目 → [http://developer.chrome.com/apps/about\\_apps](http://developer.chrome.com/apps/about_apps)、Github 的 Electron 项目 → <http://electron.atom.io/>、以及 TideSDK 项目 → <http://tidesdk.multipart.net/docs/user-dev/generated/>，都可以用来编写运行于 Windows、Mac OS 和 Android 等多个桌面平台的程序，不依赖浏览器。

### (7) 小结

可以预期，JavaScript 最终将能让你只用一种语言，就开发出适应不同平台（包括桌面端、服务器端、手机端）的程序。早在2013年9月的统计 → <http://adambard.com/blog/top-github-languages-for-2013-so-far/>之中，JavaScript 就是当年 Github 上使用量排名第一的语言。

著名程序员 Jeff Atwood 甚至提出了一条 “Atwood 定律” → <http://www.codinghorror.com/blog/2007/07/the-principle-of-least-power.html>:

“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.)

## 易学性 #

相比学习其他语言，学习 JavaScript 有一些有利条件。

### (1) 学习环境无处不在

只要有浏览器，就能运行 JavaScript 程序；只要有文本编辑器，就能编写 JavaScript 程序。这意味着，几乎所有电脑都原生提供 JavaScript 学习环境，不用另行安装复杂的 IDE（集成开发环境）和编译器。

### (2) 简单性

相比其他脚本语言（比如 Python 或 Ruby），JavaScript 的语法相对简单一些，本身的语法特性并不是特别多。而且，那些语法中的复杂部分，也不是必需要学会。你完全可以只用简单命令，完成大部分的操作。

### (3) 与主流语言的相似性

JavaScript 的语法很类似 C/C++ 和 Java，如果学过这些语言（事实上大多数学校都教），JavaScript 的入门会非常容易。

必须说明的是，虽然核心语法不难，但是 JavaScript 的复杂性体现在另外两个方面。

首先，它涉及大量的外部 API。JavaScript 要发挥作用，必须与其他组件配合，这些外部组件五花八门，数量极其庞大，几乎涉及网络应用的各个方面，掌握它们绝非易事。

其次，JavaScript 语言有一些设计缺陷。某些地方相当不合理，另一些地方则会出现怪异的运行结果。学习 JavaScript，很大一部分时间是用来搞清楚哪些地方有陷阱。Douglas Crockford 写过一本有名的书，名字就叫《JavaScript: The Good Parts》→ <http://javascript.crockford.com/>，言下之意就是这门语言不好的地方很多，必须写一本书才能讲清楚。另外一些程序员则感到，为了更合理地编写 JavaScript 程序，就不能用 JavaScript 来写，而必须发明新的语言，比如 CoffeeScript、TypeScript、Dart 这些新语言的发明目的，多多少少都有这个因素。

尽管如此，目前看来，JavaScript 的地位还是无法动摇。加之，语言标准的快速进化，使得 JavaScript 功能日益增强，而语法缺陷和怪异之处得到了弥补。所以，JavaScript 还是值得学习，况且它的入门真的不难。

## 强大的性能 #

JavaScript 的性能优势体现在以下方面。

### (1) 灵活的语法，表达力强。

JavaScript 既支持类似 C 语言清晰的过程式编程，也支持灵活的函数式编程，可以用来写并发处理（concurrent）。这些语法特性已经被证明非常强大，可以用于许多场合，尤其适用异步编程。

JavaScript 的所有值都是对象，这为程序员提供了灵活性和便利性。因为你可以很方便地、按照需要随时创造数据结构，不用进行麻烦的预定义。

JavaScript 的标准还在快速进化中，并不断合理化，添加更适用的语法特性。

### (2) 支持编译运行。

JavaScript 语言本身，虽然是一种解释型语言，但是在现代浏览器中，JavaScript 都是编译后运行。程序会被高度优化，运行效率接近二进制程序。而且，JavaScript 引擎正在快速发展，性能将越来越好。

此外，还有一种 WebAssembly 格式，它是 JavaScript 引擎的中间码格式，全部都是二进制代码。由于跳过了编译步骤，可以达到接近原生二进制代码的运行速度。各种语言（主要是 C 和 C++）通过编译成 WebAssembly，就可以在浏览器里面运行。

### （3）事件驱动和非阻塞式设计。

JavaScript 程序可以采用事件驱动（event-driven）和非阻塞式（non-blocking）设计，在服务器端适合高并发环境，普通的硬件就可以承受很大的访问量。

## 开放性 #

JavaScript 是一种开放的语言。它的标准 ECMA-262 是 ISO 国际标准，写得非常详尽明确；该标准的主要实现（比如 V8 和 SpiderMonkey 引擎）都是开放的，而且质量很高。这保证了这门语言不属于任何公司或个人，不存在版权和专利的问题。

语言标准由 TC39 委员会负责制定，该委员会的运作是透明的，所有讨论都是开放的，会议记录都会对外公布。

不同公司的 JavaScript 运行环境，兼容性很好，程序不做调整或只做很小的调整，就能在所有浏览器上运行。

## 社区支持和就业机会 #

全世界程序员都在使用 JavaScript，它有着极大的社区、广泛的文献和图书、丰富的代码资源。绝大部分你需要用到的功能，都有多个开源函数库可供选用。

作为项目负责人，你不难招聘到数量众多的 JavaScript 程序员；作为开发者，你也不难找到一份 JavaScript 的工作。

## 实验环境 #

本教程包含大量的示例代码，只要电脑安装了浏览器，就可以用来实验了。读者可以一边读一边运行示例，加深理解。

推荐安装 Chrome 浏览器，它的“开发者工具”（Developer Tools）里面的“控制台”（console），就是运行 JavaScript 代码的理想环境。

进入 Chrome 浏览器的“控制台”，有两种方法。

- 直接进入：按下 Option + Command + J（Mac）或者 Ctrl + Shift + J（Windows / Linux）
- 开发者工具进入：开发者工具的快捷键是 F12，或者 Option + Command + I（Mac）以及 Ctrl + Shift + I（Windows / Linux），然后选择 Console 面板

进入控制台以后，就可以在提示符后输入代码，然后按 Enter 键，代码就会执行。如果按 Shift + Enter 键，就是代码换行，不会触发执行。建议阅读本教程时，将代码复制到控制台进行实验。

作为尝试，你可以将下面的程序复制到“控制台”，按下回车后，就可以看到运行结果。

```
function greetMe(yourName) {  
  console.log('Hello ' + yourName);  
}  
  
greetMe('World')
```



