

开篇介绍：Node 10 年大跃进与当下在互联网研发中的地位



大家好，我是四两哥，很荣幸受掘金邀请，在 Node 十年之际，来为大家编写 Node 的学习小册，这是 Node 学习系列的第一本，我们选择性的学习 Node 里面核心且基础的模块，整体学习难度较小。另外这本册子精心准备了可以练手的小案例，在后面章节中，[Nodejs \(https://nodejs.org/en/\)](https://nodejs.org/en/) 将本色出演，下文一概使用 Node 来作为 Nodejs 的简称。

截止今天，像谷歌、亚马逊、Facebook 等全球的 Top 10 互联网大公司，甚至 [沃尔玛](http://www.electrode.io/site/web.html) (<http://www.electrode.io/site/web.html>)、[IBM/Intel](https://foundation.nodejs.org/about/members) (<https://foundation.nodejs.org/about/members>) 这样的非互联网公司都早已入坑 Node，光 2017 年 Node 就有 [2500 万的下载量](https://nodesource.com/node-by-numbers) (<https://nodesource.com/node-by-numbers>)，越来越多 [App Case](https://foundation.nodejs.org/resources/app-showcase) (<https://foundation.nodejs.org/resources/app-showcase>) 站在了 Node 的基石之上，作为一名工程师，对 Node 感兴趣但却不太熟悉的话，该怎么权衡是否要学习 Node 呢，以及学习 Node 的需要掌握哪些知识，在深度和广度上面可以沉淀到什么程度呢？本册会帮助大家梳理 Node 的学习路径，以及通过配套代码案例，来掌握基础的核心知识。

关于 Node 和 Javascript 的几个常识

- Javascript 是一门编程语言，通常在浏览器的宿主环境中运行，也即我们熟知的网页是它的主战场，用户点击反馈/数据异步加载/动画控制等等需要借助它跟浏览器进行对话
- Node 是一个基于 Javascript 运行时（Runtime），它不是一个 Javascript 框架，也不是一门编程语言

大白话翻译一下：

- 从前你可以在网页中使用 Javascript，它的角色是作为脚本代码被浏览器执行
- 现在你可以在服务器上使用 Javascript，它的角色是作为程序代码被 Node 运行

有了 Node，Javascript 的角色就从浏览器这样的客户端上，扩展到了传统意义上被 Java/PHP 等后端语言占据的服务器领域，你可以像写 PHP 一样写 Javascript，为用户提供无二的后端服务。

一门编程语言，吃遍前后端，在 Node 这里，有了可能性，只要熟悉 Javascript，都可以比较容易的上手 Node。

Github 项目排行榜与 Google Node 趋势

截止 2017 年，Github 用户超过 100 万，其中 70 万来自中国，虽然 Java/PHP/Python/Ruby 等依然是主流开发语言，但在整个 Github 的前 15 最受欢迎语言排名中，Javascript 以 230 万次的 Pull Requests (PR) 稳坐头把交椅，甚至跟 Java/Ruby/PHP 的 PR 总和 240 万次也相差不多，而推动 Javascript 车轮滚滚向前的一大引擎，便是 Node。

在开始之前，我们先把 Github 上 Node 有关的事情说完，这是上文所说 Github 最受欢迎语言排行榜的截图，可以窥见 Javascript 社区庞大的群众基础和参与密度。

2017 Github 图鉴之最流行语言 PR 排名

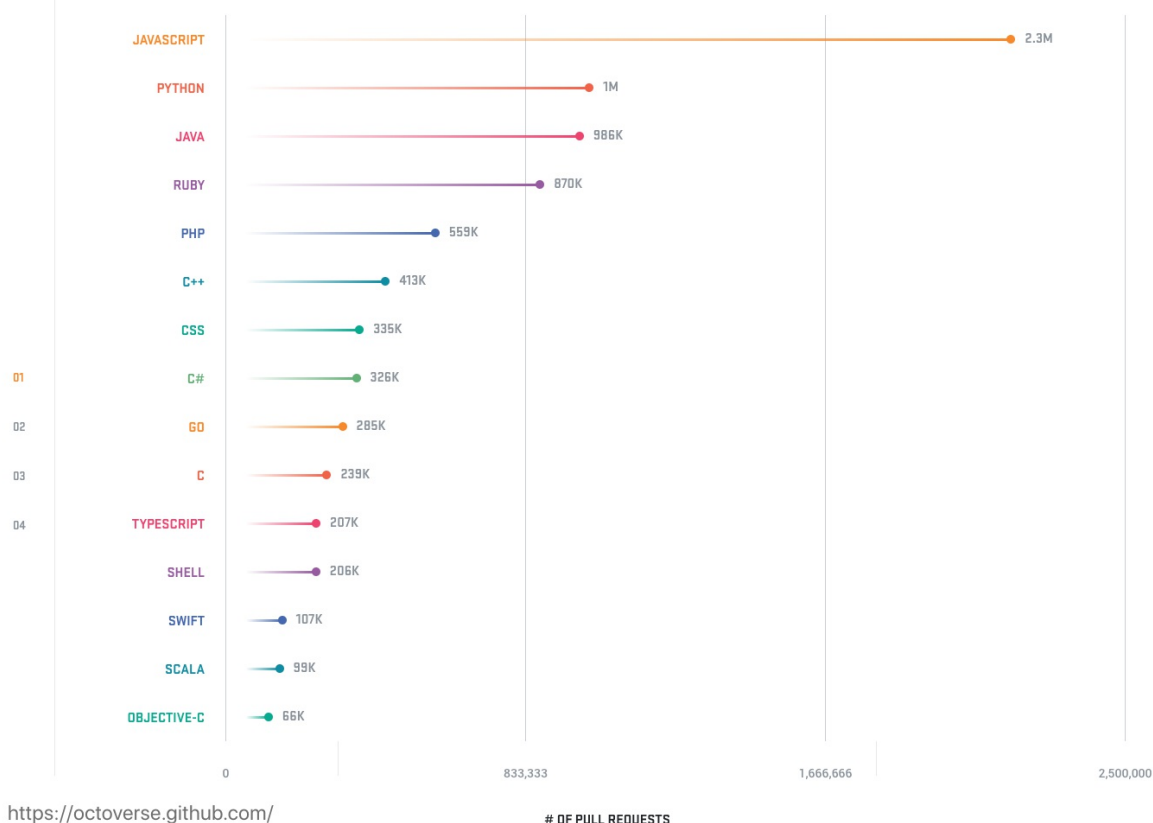


The fifteen most popular languages on GitHub

GitHub

by opened pull request

GitHub is home to open source projects written in 337 unique programming languages—but especially JavaScript.



Github 把 Node 相关的项目也归属到了 Javascript 里面，所以上图并没有出现 Node，但是在 Github 上被 Review 最多和被讨论最多的项目中，都有 Node 的身影，同时被 Fork 最多和贡献者数量最多的项目中，至少有 30% 的项目，都是强依赖 Node 的，比如 React、Vue、ReactNative 等，Javascript 生态的繁荣是建立在 Node 这一革命性工具上的：

2017 Github 图鉴之项目热度排名



换句话说讲，没有 Node，就没有 Javascript 各种全家桶的今天，当然没有 Javascript（ECMA）标准规范的支撑，同样没有 Node 今天的生态蓬勃，作为佐证，可以看下 Google 的搜索趋势（Trends），也就是全球开发者关心某项技术的程度，曲线反映了这项技术的热度，越接近 100，热度越高，可以发现 Node 自 2009 年面世以来，热度整体在持续上升，只不过在中国，晚了两三年时间。

谷歌 Nodejs - 美国 9 年热度变化趋势图





对比下图 Javascript（红色）和 Node（蓝色），也可以发现两者在热度上也基本保持一样的趋势。



还有更多有趣的数字，大家可以点击查看 [2016 Github Octoverse \(https://octoverse.github.com/2017/\)](https://octoverse.github.com/2017/)、[2017 Github Octoverse \(https://octoverse.github.com/2016/\)](https://octoverse.github.com/2016/) 和 搜索 [Google Trends \(https://trends.google.com/trends/explore?date=2009-10-01%202018-10-01&geo=US&q=nodejs\)](https://trends.google.com/trends/explore?date=2009-10-01%202018-10-01&geo=US&q=nodejs)，那么 [2018 Github Octoverse 最新公布的项目情况 \(https://octoverse.github.com/\)](https://octoverse.github.com/)，也建议大家特别关注下，来跟 2017 作对比，这些靠前的项目，哪些是离不开 Node 的支撑的：

Top open source projects

VS Code, React, and Tensorflow once again top our list of open source projects by contributor count. New to the list are projects that manage containerized applications, share Azure documentation, and consolidate TypeScript type definitions: Kubernetes, Azure Docs, and DefinitelyTyped. *

Contributors		
1	Microsoft/vscode	19k
2	facebook/react-native	10k
3	tensorflow/tensorflow	9.3k
4	angular/angular-cli	8.8k
5	MicrosoftDocs/azure-docs	7.8k
6	angular/angular	7.6k
7	ansible/ansible	7.5k
8	kubernetes/kubernetes	6.5k
9	npm/npm	6.1k
10	DefinitelyTyped/DefinitelyTyped	6.0k

Fastest growing open source projects

Overall, we're seeing trends in growth of projects related to machine learning, gaming, 3D printing, home automation, scientific programming, data analysis, and full stack JavaScript development. *

Change		
1	MicrosoftDocs/azure-docs	4.7x
2	pytorch/pytorch	2.8x
3	godotengine/godot	2.2x
4	nuxt/nuxt.js	2.1x
5	ethereum/go-ethereum	2.0x
6	wix/react-native-navigation	1.9x
7	spyder-ide/spyder	1.8x
8	tensorflow/models	1.8x
9	home-assistant/home-assistant	1.6x
10	MarlinFirmware/Marlin	1.6x

Node 的行业前景与就业优势

通过 Github 与 Google 上比较客观的数据，我们大概了解到 Javascript 以及 Node 的共荣生态，以及这样一门技术（语言）它的受欢迎程度，在 2018 年来看，这算是它的语言优势、生态优势，但它和我们的工作又有什么关系呢？我们来探讨一下。

目前市面上跟 Node 打交道的主要是两个群体，一个是前端工程师，一个是 Node 服务端工程师，至于其他群体比如做嵌入式软硬件、桌面应用开发等小方向我们暂不做讨论。

我 2010 年入行迄今 8 年了，从我的视角看过去，前端工程师始终是稀缺工种，整个市场的缺口并没有因为各种培训机构，高校的课程设计，包括整个行业的人才沉淀而逐年大幅缩小，几乎市面上每个在扩张中的互联网公司都急需前端工程师，从春天招聘到冬天，从校招社招到猎头挖掘，只要不倒闭几乎不休止，大家可以到 [拉勾 - 前端工程师](#)

(https://www.lagou.com/jobs/list_%E5%89%8D%E7%AB%AF%E5%B0%B9%E5%8F%91)、[Boss 直聘前端](#) (https://www.zhipin.com/job_detail/?query=%E5%89%8D%E7%AB%AF%E5%BC%80%E5%8F%91)、[拉勾 - Node](#) (https://www.zhipin.com/job_detail/?query=node) 等招聘网站去查看行业需求，如果你想要换工作，完全不缺少市场机会，大把的公司可以挑挑拣拣，完全是卖方市场，即便是放眼在未来，前端依然是稀缺岗位，这里是有客观原因的，列举几个：

- 近 10 年的互联网创业浪潮一波又一波，整个行业的互联网公司的大盘数量在快速增长，研发人才的紧俏成为必然
- 后端工程师的领域有着一二十年的行业积淀和高校对口专业培养，而前端工程师工种太年轻，Ajax 异步交互才流行多少年，jQuery 才风头过去多少年，整个人才基础在中国也就只有大几千的 Top 资深老人而已（对标阿里的 P7/P8/P9）
- 只要是服务于产品线上化的公司，必然有内部的各种 ERP/CRM 系统，外部各种服务型展示型网站甚至 App，前端越来越重越来越大，代码搭建越来越系统化工程化，老人不够用新人跟不上，不对等的供需关系导致前端始终坑多人少，这个行业薪资也是一年年水涨船高，2010 年校招实习生转正 6~8k 就非常不错了，2018 年能开出 13~15k 的公司比比皆是。

新兴公司大盘基数变大，前端工种过于年轻，前端产品愈加多样交互趋于复杂这三个客观现实，再加上智能手机的普及，微信生态的崛起，整个移动浪潮的滚滚向前，一个前端工程师的技术栈需要不断更新才能来适应各种物理设备，各种屏幕尺寸，各种客户端操作系统和浏览器（宿主环境）版本，包括来适应门槛更高的资深前端竞争环境，同时 Javascript 的语言版本也在不断进化，框架和编程理念也在进化，新老技术共存，于是有写不完的页面学不完的框架，兼容不完的无穷无尽的终端，老人需要不断更新技术栈，新人需要不断积淀更系统化的知识来夯实基础，可恰恰是新人缺少系统化的学习体系，以及必要的实战经验，也就是所谓的经验，知识结构过于碎片化，这就是当下的前端开发现状，这个新生行业面临着巨大的机会同时也面临着巨大的挑战，对我对你对所有人都是如此。

花这么大篇幅讲前端，是因为 Node 工程师天然就需要学习和掌握 Node，这个很好理解，但是对于一个前端，如果你说你不会使用 React/Vue/Angular 中的任何一个，且你本地没有安装过 Node，这样的事情放到 5 年前还好，放到今天则明显处于弱势了，今天一个前端工程师，框架的选型会有所不同，但是构建的工具环境选择 Node 几乎是标配了。当然我身边的人，的确还有极少极少数人是完全 jQuery 一招走天下的，到今天亦是如此，这需要有稳定的业务环境和忠诚的客户群体来保证，但是也不会太久了，比如小程序是不支持 jQuery 的，当客户有这样的需求时，技术栈仍然需要更新，仍然依赖于 Node 的工具和构建能力。

总结

好了，让我们简单总结一下：

- 第一：前端工程师的必备技能之一就是 Javascript，无论是做网页开发，还是移动端如 ReactNative 开发，还是小程序开发，掌握 Javascript 就掌握了关键技能。

- 第二：Javascript 对于学习 Node 也是必要技能，Node 之于前端，在如今的职业技能栈变得至关重要，尝试将它作为一个工具环境，做一些构建工作是好事，但仅局限于此，不在前端工程体系中深入学习则竞争力大打折扣，甚至跳槽面试都会失去很多加分机会点。

会用 Node 的一些工具套件而不了解 Node 系统化知识，这也是我最近 4 年作为面试官遇到候选人普遍存在的一个问题，想想还是比较可惜的，已经迈出第一步了，缺乏后面的进一步深入体验和总结，那么这一本小册子，就是帮大家打开 Node 学习的窗户，推动你再迈出去一步，通过理论加案例，会发现 Node 比想象中要简单的多，不需要大而全的训练，小而精的练习同样可以，那我们就从这本小册开始升级打怪吧。