

# NodeJs 简介 安装 开发工具配置

主讲教师：（大地）

合作网站：[www.itying.com](http://www.itying.com)

## 目录

一、Nodejs 介绍.....	1
二、NodeJs 的优势.....	3
三、NodeJs 适合做什么？ .....	4
四、NodeJs 环境搭建。 .....	5
五、开发工具配置。 .....	6

## 一、Nodejs 介绍

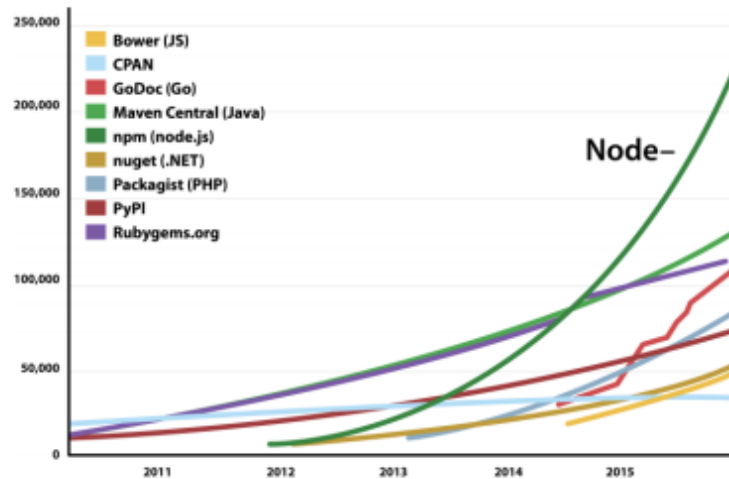
Node.js 是一个 Javascript 运行环境(runtime)。它让 JavaScript 可以开发后端程序，实现几乎其他后端语言实现的所有功能，可以与 PHP、JSP、Python、Ruby 等后端语言平起平坐。

Nodejs 是基于 V8 引擎，V8 是 Google 发布的开源 JavaScript 引擎，本身就是用于 Chrome 浏览器的 JS 解释部分，但是 Ryan Dahl 这哥们，鬼才般的，把这个 V8 搬到了服务器上，用于做服务器的软件。

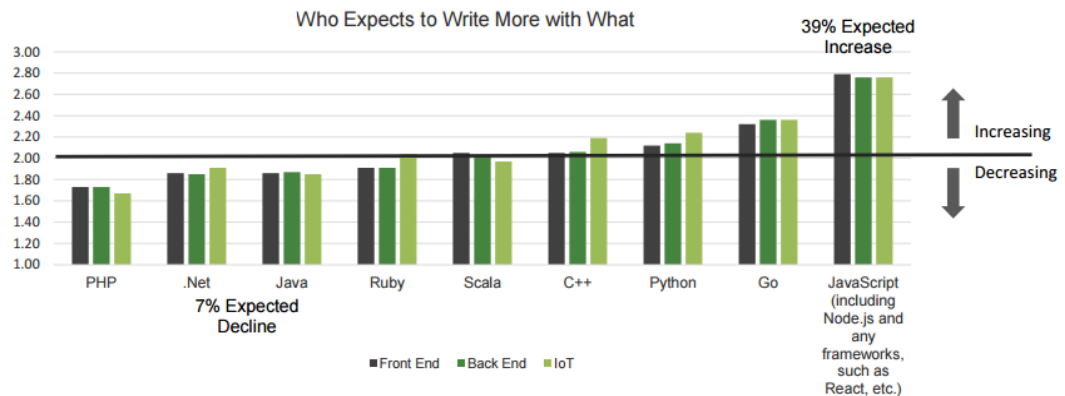


短短几年的时间，Node 取得了巨大的成功。在企业界，Node 的应用也越来越广泛，2016 年 nodeJS 官

方的调查报告。2016 年全球有 350 万开发者使用 nodeJS, 相比去年保持了 100%的增长率。像 Yahoo、Microsoft 这样的大公司, 有好多应用已经迁移到 Node 了。国内的阿里巴巴、网易、腾讯、新浪、百度等公司的很多线上产品也纷纷改用 Node 开发, 并取得了很好的效果。据统计很多 A 轮、B 轮的创业公司更喜欢使用 NodeJs 开发。



前端, 后端, 物联网 语言使用情况 (预测)



来自: <https://nodejs.org/static/documents/2016-survey-report.pdf>

## 二、NodeJs 的优势

### 1. NodeJs 语法完全是 js 语法，只要你懂 JS 基础就可以学会 Nodejs 后端开发。

Node 打破了过去 JavaScript 只能在浏览器中运行的局面。前后端编程环境统一，可以大大降低开发成本。

### 2. NodeJs 超强的并发能力。

Node.js 的首要目标是提供一种简单的、用于创建高性能服务器及可在该服务器中运行的各种应用程序的开发工具。

首先让我们来看一下现在的服务器端语言中存在着什么问题。在 Java、PHP 或者 .net 等服务器端语言中，会为每一个客户端连接创建一个新的线程。而每个线程需要耗费大约 2MB 内存。也就是说，理论上，一个 8GB 内存的服务器可以同时连接的最大用户数为 4000 个左右。要让 Web 应用程序支持更多的用户，就需要增加服务器的数量，而 Web 应用程序的硬件成本当然就上升了。

Node.js 不为每个客户连接创建一个新的线程，而仅仅使用一个线程。当有用户连接了，就触发一个内部事件，通过非阻塞 I/O、事件驱动机制，让 Node.js 程序宏观上也是并行的。使用 Node.js，一个 8GB 内存的服务器，可以同时处理超过 4 万用户的连接。

### 3. 实现高性能服务器

严格地说，Node.js 是一个用于开发各种 Web 服务器的开发工具。在 Node.js 服务器中，运行的是高性能 V8 JavaScript 脚本语言，该语言是一种可以运行在服务器端的 JavaScript 脚本语言。

那么，什么是 V8 JavaScript 脚本语言呢？该语言是一种被 V8 JavaScript 引擎所解析并执行的脚本语言。V8 JavaScript 引擎是由 Google 公司使用 C++ 语言开发的一种高性能 JavaScript 引擎，该引擎并不局限于在浏览器中运行。Node.js 将其转用在了服务器中，并且为其提供了许多附加的具有各种不同用途的 API。例如，在一个服务器中，经常需要处理各种二进制数据。在 JavaScript 脚本语言中，只具有非常有限的对二进制数据的处理能力，而 Node.js 所提供的 Buffer 类则提供了丰富的对二进制数据的处理能力。

另外，在 V8 JavaScript 引擎内部使用一种全新的编译技术。这意味着开发者编写的高端的 JavaScript 脚本代码与开发者编写的低端的 C 语言具有非常相近的执行效率，这也是 Node.js 服务器可以提供的的一个重要特性。

### 4. 开发周期短、开发成本低、学习成本低。

Node.js 自身哲学，是花最小的硬件成本，追求更高的并发，更高的处理性能。

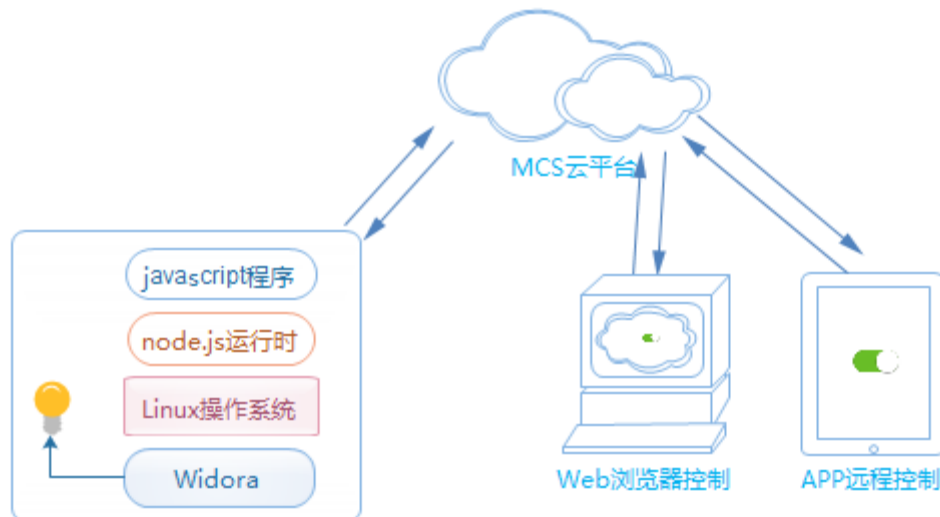
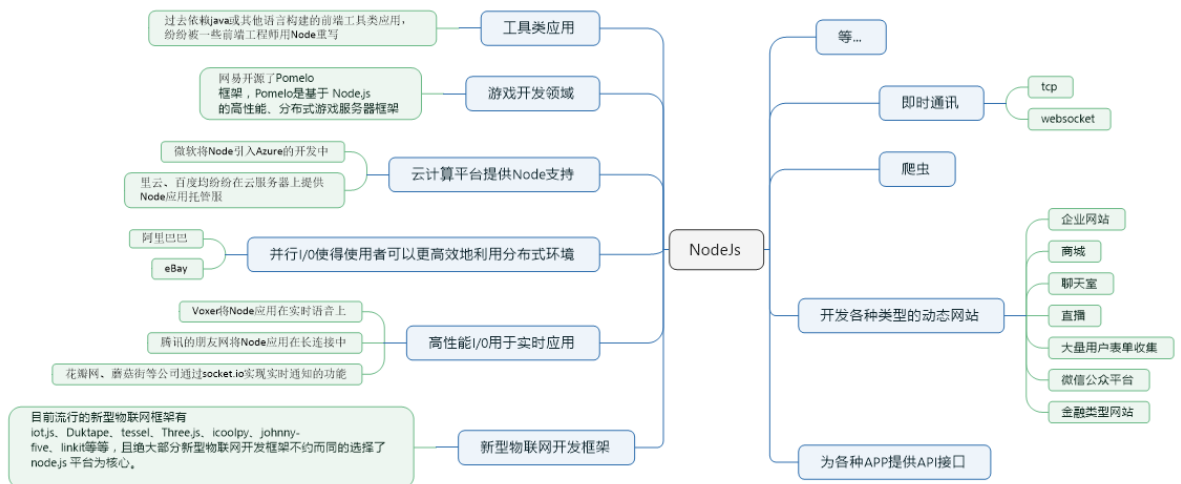
官网: <https://nodejs.org/en/>

特点: Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient.

## 三、NodeJs 适合做什么?

在短短几年多的时间里, Node 变得非常热门, 使用者也非常多。这些使用者对于 Node 的各自倚重点也各部相同, 经过整理,主要有下几类

- **前后端编程语言环境统一。**这类倚重点的代表是雅虎。雅虎开放了 Cocktai 框架, 利用自己深厚的前端沉淀, 将 YUI3 这个前端框架的能力借助 Node 延伸到服务器端, 使得使用者摆脱了日常工作中一边写 JavaScript 一边写 PHP 所带来的上下文交换负担。
- **Node 带来的高性能 I/O 用于实时应用。**Voxer 将 Node 应用在实时语音上。国内腾讯的朋友网将 Node 应用在长连接中, 以提供实时功能, 花瓣网、蘑菇街等公司通过 socket.io 实现实时通知的功能。
- **并行 I/O 使得使用者可以更高效地利用分布式环境。**阿里巴巴 eBay 是这方面的典型。阿里巴巴的 NodeFox 和 eBay 的 ql.io 都是借用 Node 并行 I/O 的能力, 更高效地使用已有的数据。
- **并行 I/O •有效利用稳定接口提升 Web 渲染能力。**雪球财经和 LinkedIn 的移动版网站均是这种案例, 撇弃 同步等待式的顺序请求, 大胆采用并行 I/O, 加速数据的获取进而提升 Web 的渲染速度。
- **云计算平台提供 Node 支持。**微软将 Node 引入 Azure 的开发中, 阿里云、百度均纷纷在云服务器上提供 Node 应用托管服务, Joyent 更是云计算中提供 Node 支持的代表。这类平台看重 JavaScript 带来的开发上的优势, 以及低资源占用、高性能的特点。
- **游戏开发领域。**游戏领域对实时和并发有很高的要求, 网易开源了 pomelo 实时框架, 可以应用在游戏和高实时应用中。
- **工具类应用。**过去依赖 java 或其他语言构建的前端工具类应用, 纷纷被一些前端工程师用 Node 重写, 用前端熟悉的语言为前端构建熟悉的工具。



## 四、NodeJs 环境搭建。

Node.js 和 Java 非常像，跨平台的。不管是 Linux 还是 windows 编程是完全一致的（有一些不一样，比如路径的表述）。Linux 版本的 Node.js 环境和 windows 环境是不一样的，但是编程语言一样。

官网：<https://nodejs.org/en/>

下载稳定版本、双击下一步下一步安装。

安装完成打开 CMD。

```
C:\Users\Administrator>node -v  
v6.9.5
```

## 五、开发工具配置。

