# 为什么要用JavaScript？

有许多编程语言可以使用，为什么要用JavaScript呢？本章关注在选择编程语言时非常重要的七个问题，并给出JavaScript在总体上做得更好的论据。

1． 是否可以自由获取？

2． 语言是否优雅？

3． 实际是否有用？

4．是否有好工具，尤其是好的集成开发环境（Integrated Development Environments，IDEs）？

5．速度是否达到开发者的要求？

6．是否广泛使用？

7．是否有未来？

## JavaScript是否可以自由获取？

JavaScript是现有最开放的编程语言，这点充满争议，它的规范：ECMA-262是ISO（International Standardization Organization，国际标准化组织）标准。许多来自独立团体的实现都紧密遵循了该规范。其中一些实现是开源的。此外，JavaScript语言的进化是由TC39委员会处理的，TC39是由多家公司，包括所有主流浏览器厂商组成的委员会。其中许多公司按理说还是竞争者，但他们为了JavaScript语言的利益而共同工作。

## JavaScript是否优雅？

是，也不是。我用不同范式的多种语言写过大量代码。因此，我很清楚JavaScript语言并非站在优雅性的顶峰。不过，它是一门非常灵活的语言，有相当优雅的内核，能使用面向对象和函数编程的混合方法。

JavaScript引擎之间的语言兼容性曾是个问题，但现在不是了，部分要感谢test262测试套件，它会做引擎的ECMAScript规范一致性检查。相反，浏览器和DOM差异仍然是挑战。这也是为什么通常最好依赖框架来隐藏这些差别。

## JavaScript是否有用？

世界上最漂亮的编程语言如果不能写出所需的程序也是没用的。

## 图形用户界面

在图形用户界面领域，JavaScript得益于是HTML5的一部分。本节用术语HTML5表示“浏览器平台”（HTML，CSS以及浏览器的JavaScript API）。HTML5部署广泛并在平稳进步。慢慢变成了书写全功能、跨平台应用的完整层。类似于，比如说，Java平台，几乎像一个嵌入式操作系统。HTML5的卖点之一在于让人编写跨平台的用户界面。通常需要有的妥协是：放弃一些功能以免被限制在单一的操作系统上。过去，“跨平台”意味着windows，Mac OS和Linux。但现在多了两种交互平台：Web和移动端。有了HTML5，可以通过[PhoeGap](http://phonegap.com/)，[Chrome Apps](http://developer.chrome.com/apps/)和[TideSDK](http://www.tidesdk.org/)等技术将目标定位为所有平台。

此外，一些平台会将web应用作为原生应用，或者可以原生安全，例如Chrome OS,Firefox OS以及Android。

## 其他补充JavaScript的技术

不止HTML5，还有更多技术补充了JavaScript，让JavaScript语言更加有用：

**库：**

JavaScript有丰富的库，可以完成从JavaScript解析（通过[Esprima](http://esprima.org/)）到处理并显示PDF文件（通过[PDF.js](https://github.com/mozilla/pdf.js)）等任务。

**Node.js**

Node.js平台让你可以编写服务器端代码和shell脚本（build工具，测试工具等等）。

**SON（JavaScript Object Notation，JavaScript对象标注，见第22章）**

JSON是扎根在JavaScript的数据格式，因为用于Web数据的交换而流行起来（例如：web服务的结果）。

**NoSQL数据库（例如[CouchDB](http://couchdb.apache.org/)和[MongoDB](http://www.mongodb.org/)）**

这些数据库将JSON和JavaScript紧密结合在了一起。

## JavaScript是否有好工具？

JavaScript有越来越好的build工具（例如：[Grunt](http://gruntjs.com/)）和测试工具（例如：[moncha](http://visionmedia.github.io/mocha/)）。Node.js使得通过shell（而不只是浏览器）运行工具成为可能。该领域的风险在于碎片化，因为我们渐渐地地有了太多这样的工具。

JavaScript IDE领域仍处于初期，但是也在快速发展。web开发的复杂性和动态性让这个领域成为了创新的沃土。两个开源实例为[Brackets](http://brackets.io/)和[Light Table](http://www.lighttable.com/)。

此外，浏览器也变成了越来越有用的开发环境。尤其是Chrome，最近又有一些引人注目的进展。未来IDE和浏览器集成度有多高将是很有趣的事情。

## JavaScript是否够快？

JavaScript引擎已经取得了巨大的进步，从缓慢的解释器进化成了快速的实时（just-in-time）编译器。现在JavaScript引擎的速度已经快到足以满足大部分应用。而且，还有一些新点子正在开发，让JavaScript的速度能够满足剩下的应用：

\* [asm.js](http://asmjs.org/)是JavaScript（非常静态）的子集，在现代引擎上运行迅速，速度大约为编译过的C++代码的70%。asm.js可以用于实现例如web应用中的性能关键算法。Asm.js曾被用于移植基于C++的游戏到Web平台。

\* [ParallelJS](http://www.2ality.com/2013/12/paralleljs.html)并行化了使用新数组方法mapPar，filterPar以及reducePar的JavaScript代码（已有数组方法map、filter和reduce的可并行版本）。为了让并行化工作，必须编写特殊风格的回调；主要的限制在于不能模拟在回调中还没有创建的数据。

## JavaScript是否广泛使用？

一门语言被广泛使用有两个好处。首先，这样的语言有更好的文档和支持。其次，更多的程序员了解它，这对于无论是招聘还是寻找基于该语言的工具的客户都非常重要。

JavaScript广泛使用，具有以下两点优势：

\* 现在有各种类型和大小的JavaScript文档和支持：书、播客、博客文章、电子邮件新闻、论坛等等，第13章会将你指向这些重要的资源。

\* 对JavaScript开发者的需求旺盛，而且级别也在稳步提升。

## JavaScript是否有未来？

几件事说明JavaScript有光明的未来：

\* 语言稳步进化；ECMAScript6看上去不错。

\* 有许多与JavaScript相关的创新（例如之前提到的asm.js和ParallelJS,微软的TypeScript等等）

\* 和JavaScript是一个整体的Web平台正快速成熟起来。

\* JavaScript得到了广泛公司联盟的支持—而非受单个人或者公司控制。

## 总结

考虑上面列出的语言吸引力，JavaScript做得非常不错。当然，JavaScript并非完美，但是目前还难以被击败--而且事情只会变得越来越好。