# Why JavaScript？

# 为什么要用JavaScript？

There are many programming languages out there. Why should you use JavaScript?

编程语言有很多，为什么要使用JavaScript呢？

This chapter looks at seven aspects that are important when you are choosing a programming language and argues that JavaScript does well overall:

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. Is it freely available?

1． 是否可以自由获取？

2. Is it an elegant programming language?

2． 语言是否优雅？

3. Is it useful in practice?

3． 实际是否有用？

4. Does it have good tools, especially good *integrated development environments*

(IDEs)?

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

5. Is it fast enough for what you want to do?

5．速度是否达到开发者的要求？

6. Is it widely used?

6．是否广泛使用？

7. Does it have a future?

7．是否有未来？

## Is JavaScript Freely Available?

## JavaScript是否可以自由获取？

JavaScript is arguably the most open programming language there is: ECMA-262, its

specification, is an ISO standard.

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。

That specification is closely followed by many implementations

from independent parties. Some of those implementations are open source.

Furthermore, the evolution of the language is handled by TC39, a committee comprising several companies, including all major browser vendors. Many of those companies are normally competitors, but they work together for the benefit of the language.

许多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。虽然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。

## Is JavaScript Elegant?

## JavaScript是否优雅？

Yes and no. I’ve written fair amounts of code in several programming languages from different paradigms.

既可以说是，也可以说不是。我用不同范式的不同语言写过大量代码。

Therefore, I’m well aware that JavaScript isn’t the pinnacle of elegance. However, it is a very flexible language, has a reasonably elegant core, and

enables you to use a mixture of object-oriented programming and functional

programming.

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。

Language compatibility between JavaScript engines used to be a problem, but isn’t anymore, partly thanks to the test262 suite that checks engines for conformance to the ECMAScript specification. In contrast, browser and DOM differences are still a challenge. That’s why it is normally best to rely on frameworks for hiding those differences.

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。

## Is JavaScript Useful?

## JavaScript是否有用？

The most beautiful programming language in the world is useless unless it allows you to write the program that you need.

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。

## Graphical User Interfaces

## 图形用户界面

In the area of graphical user interfaces, JavaScript benefits from being part of *HTML5*. In this section, I use the term HTML5 for “the browser platform” (HTML, CSS, and browser JavaScript APIs). HTML5 is deployed widely and making constant progress. It is slowly becoming a complete layer for writing full-featured, cross-platform applications; similar to, say, the Java platform, it’s almost like an embedded operating system.

One of HTML5’s selling points is that it lets you write cross-platform graphical user interfaces. Those are always a compromise: you give up some quality in exchange for not being limited to a single operating system. In the past, “cross-platform” meant Windows,

Mac OS, or Linux. But we now have two additional interactive platforms: web

and mobile. With HTML5, you can target all of these platforms via technologies such as PhoneGap, Chrome Apps, and TideSDK

在图形用户界面领域，JavaScript得益于它是HTML5的一部分。本节中术语HTML5表示“浏览器平台”（HTML，CSS以及浏览器的JavaScript API）。HTML5被广泛使用并在平稳进步。它慢慢变成了书写全功能、跨平台应用的完整的一层。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/)等技术来定位到所有平台。

Additionally, several platforms have web apps as native apps or let you install them natively—for example, Chrome OS, Firefox OS, and Android.

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。

## Other Technologies Complementing JavaScript

## 其他补充JavaScript的技术

There are more technologies than just HTML5 that complement JavaScript and make

the language more useful:

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

*Libraries*

**库：**

JavaScript has an abundance of libraries, which enable you to complete tasks ranging from parsing JavaScript (via Esprima) to processing and displaying PDF files (via PDF.js).

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**

The Node.js platform lets you write server-side code and shell scripts (build tools, test runners, etc.).

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）。

*JSON (JavaScript Object Notation, covered in Chapter 22)*

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

JSON is a data format rooted in JavaScript that has become popular for exchanging

data on the Web (e.g., the results of web services).

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 databases (such as CouchDB and MongoDB)*

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

These databases tightly integrate JSON and JavaScript

这些数据库将JSON和JavaScript紧密结合在了一起。

## Does JavaScript Have Good Tools?

## JavaScript是否有好工具？

JavaScript is getting better build tools (e.g., Grunt) and test tools (e.g., mocha). Node.js makes it possible to run these kinds of tools via a shell (and not only in the browser). One risk in this area is fragmentation, as we are progressively getting too many of these

tools.

JavaScript有越来越好的build工具（例如：[Grunt](http://gruntjs.com/)）和测试工具（例如：[mocha](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。

The JavaScript IDE space is still nascent, but it’s quickly growing up. The complexity and dynamism of web development make this space a fertile ground for innovation. Two open source examples are Brackets and Light Table.

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/)。

Additionally, browsers are becoming increasingly capable development environments.

Chrome, in particular, has made impressive progress recently. It will be interesting to see how much more IDEs and browsers will be integrated in the future.

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和浏览器之间的集成度有多高将是很有趣的事情。

## Is JavaScript Fast Enough?

## JavaScript是否够快？

JavaScript engines have made tremendous progress, evolving from slow interpreters to fast just-in-time compilers. They are now fast enough for most applications. Furthermore, new ideas are already in development to make JavaScript fast enough for the remaining applications:

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，还有一些新特性正在开发，让JavaScript的速度能够满足剩下的应用：

asm.js is a (very static) subset of JavaScript that runs fast on current engines, approximately 70% as fast as compiled C++. It can, for example, be used to implement performance-critical algorithmic parts of web applications. It has also been used to port C++-based games to the web platform.

\* [asm.js](http://asmjs.org/)是JavaScript（纯静态）的子集，在目前的引擎上运行迅速，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 parallelizes JavaScript code that uses the new array methods mapPar,

filterPar, and reducePar (parallelizable versions of the existing array methods

map, filter, and reduce). In order for parallelization to work, callbacks must be

written in a special style; the main restriction is that you can’t mutate data that hasn’t been created inside the callbacks.

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

## Is JavaScript Widely Used?

## JavaScript是否广泛使用？

A language that is widely used normally has two benefits. First, such a language is better documented and supported. Second, more programmers know it, which is important whenever you need to hire someone or are looking for customers for a tool based on the language.

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 is widely used and reaps both of the aforementioned benefits:

JavaScript的广泛使用收获了以下两点优势：

These days, documentation and support for JavaScript comes in all shapes and sizes:

books, podcasts, blog posts, email newsletters, forums, and more. Chapter 33 points you toward important resources.

\* 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。

JavaScript developers are in great demand, but their ranks are also constantly

increasing.

\* 对JavaScript开发者的需求旺盛，而且级别也在稳步提升。

## Does JavaScript Have a Future?

## JavaScript是否有未来？

Several things indicate that JavaScript has a bright future:

几件事说明JavaScript有光明的未来：

The language is evolving steadily; ECMAScript 6 looks good.

\* 语言稳步进化；ECMAScript6看上去不错。

There is much JavaScript-related innovation (e.g., the aforementioned asm.js and

ParallelJS, Microsoft’s TypeScript, etc.).

\* 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）

The web platform of which JavaScript is an integral part is maturing rapidly.

\* 和JavaScript是一个整体的Web平台正快速成熟起来。

JavaScript is supported by a broad coalition of companies—no single person or

company controls it.

\* JavaScript得到了广泛的公司联盟支持，而不是受单个人或单个公司控制。

## Conclusion

## 总结

Considering the preceding list of what makes a language attractive, JavaScript is doing remarkably well. It certainly is not perfect, but at the moment, it is hard to beat—and things are only getting better.

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。