

# Java vs. JavaScript: Everything You Need to Know

By Joel Lee

Read the original article here: <http://www.makeuseof.com/tag/java-vs-javascript/>



If GitHub is anything to go by, we can see that Java and JavaScript are the two **most popular programming languages in the world**, followed by **Python**, PHP, and Ruby. **Yet despite their similar names, Java and JavaScript are unrelated.**

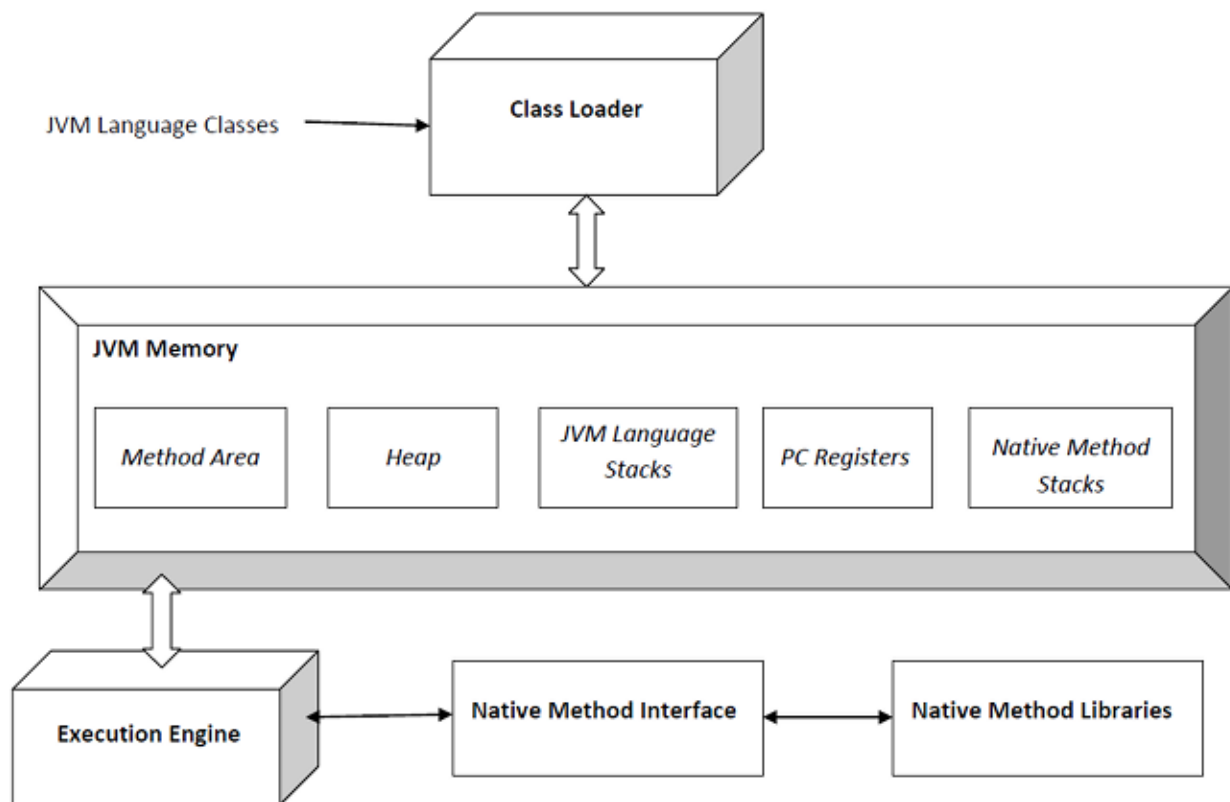
Common sense tells you that JavaScript should be a lighter, simpler, easier version of Java. But common sense is wrong. The two languages share some things in common, but they're more different from each other than they are alike.

In this post, we'll explore some of these differences and help you figure out which one to learn depending on your needs. As you'll see, the two languages rarely serve the same purpose.

## What Is Java?

Java, which was code named "Oak" and "Green" during initial development, came on the scene in May 1995. Many saw it as a successor to or evolution of C/C++ since it borrowed much of the same syntax while introducing new concepts that made coding safer and easier.

**One of the primary tenets of Java was, and still is, the idea that programmers should be able to "write once, run anywhere."** Instead of compiling to separate executables for different target platforms, Java code is compiled to a single JAR file that runs on any Java-supporting system.



This is made possible by the **Java Virtual Machine (JVM)**. The JVM reads the JAR file (which is a collection of Java-specific instructions), compiles it to platform-specific instructions at runtime, then executes the app. This process is aptly named **just-in-time compilation**. Learn more in [our overview of the JVM](#).

So how is Java used in the real world?

- **Android apps** – While other languages can be used, Java is the **official language of Android app development**, which means fastest performance and most native experience. If you're interested in this, check out [these resources for developing Android apps](#).
- **Desktop apps** – Java is perhaps the easiest way to create cross-platform software these days. Swing used to be the UI toolkit of choice for Java developers, but if you're just starting out, skip it and learn JavaFX or SWT instead. You'll thank yourself later.
- **Enterprise systems** – Industries with high-volume data processing (e.g. banking, financial trading, etc.) tend to use Java for non-legacy systems because it's fast, portable, easier to maintain, and less prone to the kinds of catastrophic bugs that are common in lower-level languages.
- **Embedded systems** – Some areas of the embedded space now rely on Java. Examples of embedded systems include digital watches, factory controllers, traffic lights, microcontrollers, hybrid vehicles, and more recently, **Internet of Things devices**.
- **Scientific research** – While data processing, computing, modeling, and simulations are often done in languages like MATLAB and Python, Java is commonly used for more complex tasks like natural language processing and artificial intelligence.

# What Is JavaScript?

JavaScript was born out of HTML's vast limitations. When Netscape Communications foresaw the need for a more dynamic web, they created a new language that could be written right within HTML. This language, which launched under the name LiveScript, was prototyped in just 10 days.

**LiveScript and Java were meant to complement each other**, hence the rename to JavaScript and why the two languages have similar syntax. But that's as far as the similarities go. Java never took off on the web and the two languages parted ways shortly thereafter. Alongside HTML and CSS, JavaScript is one of the **three main pillars of web development**.



Unlike Java, which is compiled, JavaScript is interpreted. When you visit a site that uses JavaScript, your browser receives the full JavaScript source code as-is and interprets it on-the-fly using a **JavaScript engine**. Different browsers use different engines: V8 (Chrome), SpiderMonkey (Firefox), Chakra (Edge), etc.

That being said, JavaScript is no longer just a browser-side web language. Over the past decade, it has become one of the most flexible languages in the world, which explains why it's also the most popular. So how is JavaScript used in the real world?

- **Web apps** — JavaScript can be used for both front-end and back-end web development. Front-end entails browser-side DOM manipulation (e.g. animations, data insertions, asynchronous updates) while back-end entails server-side logic (e.g. routing, data handling, database interactions). Some JavaScript frameworks do both, which is called full-stack web development.
- **Desktop apps** — JavaScript can be taken off the web and packaged into standalone desktop software using frameworks like Electron and NW.js (formerly Node-Webkit). This is done by bundling a JavaScript engine inside the executable, which interprets the HTML/CSS/JS within a standalone desktop window.
- **Mobile apps** — Frameworks like React Native and PhoneGap let you create mobile apps using JavaScript. These frameworks tend to fall into two types: those that transpile the JavaScript into native code and those that use a bundled JavaScript engine to render as an app.

## When to Use Which Language?

The practical differences between Java and JavaScript boil down to two crucial metrics: app performance and development time.

Java is a verbose language with lots of restraint. In general, Java apps are less prone to critical errors and perform much better than JavaScript apps, but take a significantly longer time to develop. JavaScript allows for more freedom and shortcuts, which is great for fast development, but is susceptible to slow runtime speed, messy code, and debugging difficulties.

**For web development, go with JavaScript. For computation-heavy software, go with Java. For desktop and mobile software, both are viable but I recommend Java for large, complex apps due to cleaner debugging and codebase management. Otherwise, JavaScript. What will you go with? Let us know **in the comment section** below!**

Image Credit: Melpomene via [shutterstock.com](https://www.shutterstock.com)

Read more stories like this at

