## What is Java Script:

JavaScript is a scripting language that is one of the three core languages used to develop websites. Whereas HTML and CSS give a website structure and style, JavaScript lets you add functionality and behaviors to your website, allowing your website’s visitors to interact with content in many imaginative ways.

JavaScript is primarily a client-side language, meaning it runs on your computer within your browser. However, more recently the introduction of Node.js has allowed JavaScript to also execute code on servers.

Since its release, JavaScript has surpassed Java, Flash, and other languages because it is relatively easy to learn, has a free and open community, and, most importantly, is incredibly useful, allowing developers to quickly create apps with audiences in the millions.

JavaScript Origins

The early to mid-1990s was an important time for the internet. Key players like Netscape and Microsoft were in the midst of browser wars, with Netscape’s Navigator and Microsoft’s Internet Explorer going head to head.

In September 1995, a Netscape programmer named [Brandan Eich](https://en.wikipedia.org/wiki/Brendan_Eich) developed a new scripting language in just 10 days. It was originally named Mocha, but quickly became known as LiveScript and, later, JavaScript.

## ECMAScript Is Born

In 1997, due to JavaScript’s rapid growth, it became clear that the language would need to be properly maintained and managed. Therefore, Netscape handed the job of creating a language specification to the European Computer Manufacturers Association (ECMA), a body founded with the goal of standardizing computing. The ECMA specifications were labeled ECMA-262 and ECMAScript languages included JavaScript, JScript, and ActionScript.

Between 1997 and 1999, ECMA-262 had three revisions, but nearly 10 years later, version 4 was abandoned due to differing opinions on the direction of the language and its proposed features. Interestingly, many of these controversial features, such as generators, iterators, and restructuring assignments, have been included in more recent ECMAScript specifications.

**JavaScript engine**

A **JavaScript engine** is a program or an interpreter which executes JavaScript code. A JavaScript engine can be implemented as a standard interpreter, or just-in-time compiler that compiles JavaScript to bytecode in some form.

They convert the highlevel code into machine-readable code which lets computer to perform some specific tasks. We will understand this using an image.

This is a list of popular projects that are implementing a JavaScript engine:

* [**V8**](https://en.wikipedia.org/wiki/V8_(JavaScript_engine)) — open source, developed by Google, written in C++
* [**Rhin**](https://en.wikipedia.org/wiki/Rhino_(JavaScript_engine))**o** — managed by the Mozilla Foundation, open source, developed entirely in Java
* SpiderMonkey— the first JavaScript engine, which back in the days powered Netscape Navigator, and today powers Firefox
* [**JavaScriptCore**](https://en.wikipedia.org/wiki/JavaScriptCore) — open source, marketed as Nitro and developed by Apple for Safari
* [**KJS**](https://en.wikipedia.org/wiki/KJS_(KDE)) — KDE’s engine originally developed by Harri Porten for the KDE project’s Konqueror web browser
* [**Chakra** (JScript9)](https://en.wikipedia.org/wiki/Chakra_(JScript_engine)) — Internet Explorer
* [**Chakra** (JavaScript)](https://en.wikipedia.org/wiki/Chakra_(JavaScript_engine)) — Microsoft Edge
* [**Nashorn**](https://en.wikipedia.org/wiki/Nashorn_(JavaScript_engine)), open source as part of OpenJDK, written by Oracle Java Languages and Tool Group
* [**JerryScript**](https://en.wikipedia.org/wiki/JerryScript) — is a lightweight engine for the Internet of Things.

The V8 Engine which is built by Google is open source and written in **C++**. This engine is used inside Google Chrome. Unlike the rest of the engines, however, V8 is also used for the popular Node.js runtime.

Javascript engine functionality

Profiler

Interpreter

Compiler

Optimized code

Bytecode

Abstract syntax tree(ABS)

Parser