Introduction

26 March 2024 09:31

Ecma International

In Nov 1996, Netscape submitted JavaScript to Ecma International

It is an industry association dedicated to the standardization of information and communication systems

Netscape wanted a standard specification that all browser vendors could conform to as it would help keep other implementations consistent across browsers

For each new specification Ecma provides a standard specification and a committee

In case of JavaScript, the standard is called ECMA-262 and the committee that works on ECMA-262 is called Technical Committee 39 (TC39)

Ecma International contd.

Ecma however decided to use the term "ECMAScript" to talk about the official language

The reason for this is because Oracle (who acquired Microsystems) owns the trademark for the term “JavaScript”

ECMAScript refers to the standard language whereas JavaScript is what we use in practice and builds on top of ECMAScript

ECMAScript versions

1997 - ECMAScript 1

1998 - ECMAScript 2

1999 - ECMAScript 3

ECMAScript 4 never released

2009 - ECMAScript 5

2015 - ECMAScript 2015 (ES6)

One version every year since 2015

ECMAScript Summary

ECMA-262 is the language specification

ECMAScript is the language that implements ECMA-262

JavaScript is basically ECMAScript at its core but builds on top of that

JavaScript Engine

JavaScript code we write cannot be understood by the computer

A JavaScript engine is a program that converts javascript code that developers write into machine code that allows a computer to perform specific tasks

JavaScript engines are typically developed by web browser vendors

- V8 – Open-source JavaScript Engine developed by Google for Chrome
- SpiderMonkey – The JavaScript Engine powering Mozilla Firefox
- JavaScriptCore – Open-source JavaScript Engine developed by Apple for Safari
- Chakra – A JavaScript Engine for the original Microsoft Edge (The latest version of edge uses V8)

JavaScript Engine

JavaScript code we write cannot be understood by the computer

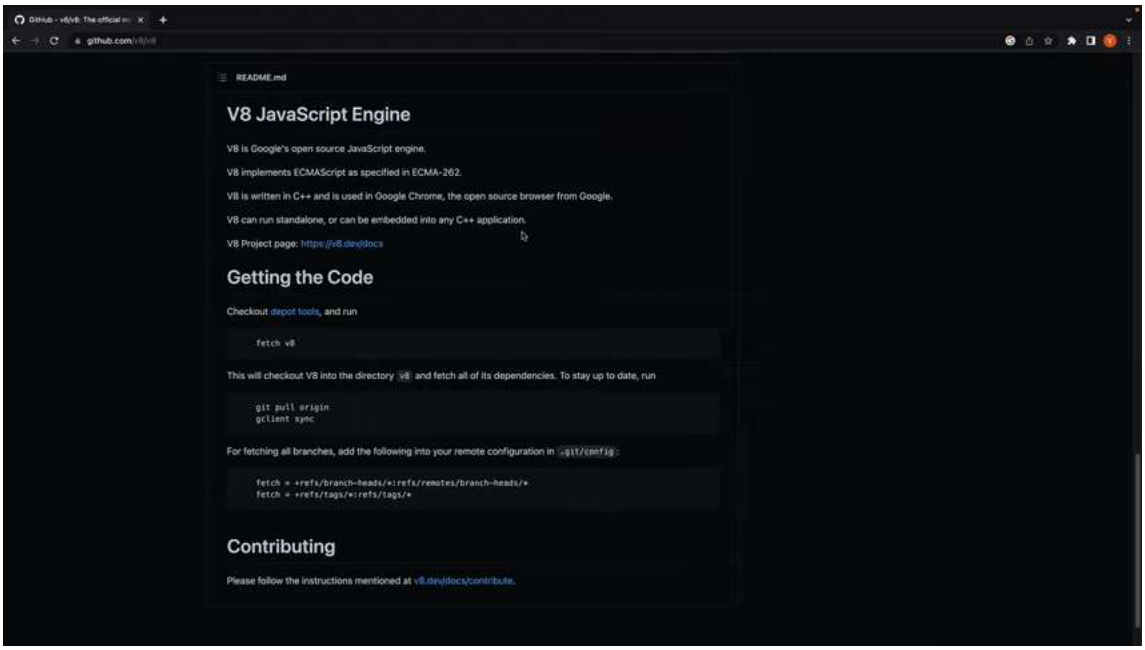
A JavaScript engine is a program that converts javascript code that developers write into machine code that allows a computer to perform specific tasks

JavaScript engines are typically developed by web browser vendors

- V8 – Open-source JavaScript Engine developed by Google for Chrome
- SpiderMonkey – The JavaScript Engine powering Mozilla Firefox
- JavaScriptCore – Open-source JavaScript Engine developed by Apple for Safari
- Chakra – A JavaScript Engine for the original Microsoft Edge (The latest version of edge uses V8)

Chrome's V8 Engine & Node.js

Chrome's V8 engine by Google sits at the core of Node.js



Chrome's V8 Engine & Node.js

Chrome's V8 engine by Google sits at the core of Node.js

By embedding V8 into your own C++ application, you can write C++ code that gets executed when a user writes JavaScript code

You can add new features to JavaScript itself

Since C++ is great for lower level operations like file handling, database connections and network operations, by embedding V8 into your own C++ program, you have the power to add all of that functionality in JavaScript

The C++ program we're talking about is Node.js*

(* Node.js is a lot more than just a C++ program)

Chrome's V8 Engine Summary

A JavaScript engine is a program that executes JavaScript code

In 2008, Google created its own JavaScript engine called V8

V8 is written in C++ and can be used independently or can be embedded into other C++ programs

That allows you to write your own C++ programs which can do everything that V8 can do and more

Your C++ program can run ECMAScript and additional features that you choose to incorporate

For example, features that are available in C++ but not available with JavaScript

JavaScript Runtime

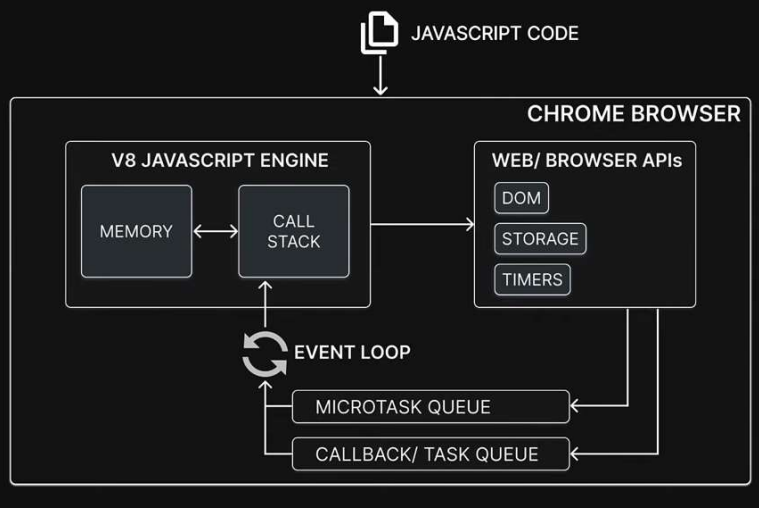
JavaScript runtime is an environment which provides all the necessary components in order to use and run a JavaScript program

Every browser has a JavaScript Engine

A JavaScript Engine is one component in the JavaScript Runtime

What else does the JavaScript Runtime consist of?

Chrome Browser JavaScript Runtime



What is Node.js?

Node.js is an open-source, cross-platform JavaScript runtime environment

Open source - source code is publicly available for sharing and modification

Cross platform - available on Mac, Windows and Linux

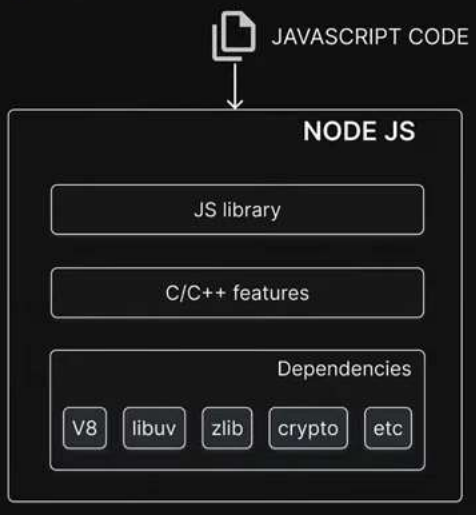
JavaScript runtime environment - provides all the necessary components in order to use and run a JavaScript program *outside the browser*

What can you build with Node.js?

- Traditional websites
- Backend services like APIs
- Real-time applications
- Streaming services
- CLI tools
- Multiplayer games

Node.js allows you to build complex and powerful applications

Node.js JavaScript Runtime



Node.js Summary

Node.js is an open-source, cross-platform JavaScript runtime environment

It is not a language, it is not a framework

Capable of executing JavaScript code outside a browser

It can execute not only the standard ECMAScript language but also new features that are made available through C++ bindings using the V8 engine

It consists of C++ files which form the core features and JavaScript files which expose common utilities and some of the C++ features for easier consumption