# Documentation

V8 is Google’s open source high-performance JavaScript and WebAssembly engine, written in C++. It is used in Chrome and in Node.js, among others.

This documentation is aimed at C++ developers who want to use V8 in their applications, as well as anyone interested in V8’s design and performance. This document introduces you to V8, while the remaining documentation shows you how to use V8 in your code and describes some of its design details, as well as providing a set of JavaScript benchmarks for measuring V8’s performance.

## About V8

V8 implements [ECMAScript](https://tc39.es/ecma262/) and [WebAssembly](https://webassembly.github.io/spec/core/), and runs on Windows 7 or later, macOS 10.12+, and Linux systems that use x64, IA-32, or ARM processors. Additional systems (IBM i, AIX) and processors (MIPS, ppcle64, s390x) are externally maintained, see [ports](https://v8.dev/docs/ports). V8 can run standalone, or can be embedded into any C++ application.

V8 compiles and executes JavaScript source code, handles memory allocation for objects, and garbage collects objects it no longer needs. V8’s stop-the-world, generational, accurate garbage collector is one of the keys to V8’s performance.

JavaScript is commonly used for client-side scripting in a browser, being used to manipulate Document Object Model (DOM) objects for example. The DOM is not, however, typically provided by the JavaScript engine but instead by a browser. The same is true of V8 — Google Chrome provides the DOM. V8 does however provide all the data types, operators, objects and functions specified in the ECMA standard.

V8 enables any C++ application to expose its own objects and functions to JavaScript code. It’s up to you to decide on the objects and functions you would like to expose to JavaScript.

## Documentation overview

* [Building V8 from source](https://v8.dev/docs/build)
  + [Checking out the V8 source code](https://v8.dev/docs/source-code)
  + [Building with GN](https://v8.dev/docs/build-gn)
  + [Cross-compiling and debugging for ARM/Android](https://v8.dev/docs/cross-compile-arm)
  + [Cross-compiling for iOS](https://v8.dev/docs/cross-compile-ios)
  + [GUI and IDE setup](https://v8.dev/docs/ide-setup)
  + [Compiling on Arm64](https://v8.dev/docs/compile-arm64)
* [Contributing](https://v8.dev/docs/contribute)
  + [Respectful code](https://v8.dev/docs/respectful-code)
  + [V8’s public API and its stability](https://v8.dev/docs/api)
  + [Becoming a V8 committer](https://v8.dev/docs/become-committer)
  + [Committer’s responsibility](https://v8.dev/docs/committer-responsibility)
  + [Blink web tests (a.k.a. layout tests)](https://v8.dev/docs/blink-layout-tests)
  + [Evaluating code coverage](https://v8.dev/docs/evaluate-code-coverage)
  + [Release process](https://v8.dev/docs/release-process)
  + [Design review guidelines](https://v8.dev/docs/design-review-guidelines)
  + [Implementing and shipping JavaScript/WebAssembly language features](https://v8.dev/docs/feature-launch-process)
  + [Checklist for staging and shipping of WebAssembly features](https://v8.dev/docs/wasm-shipping-checklist)
  + [Flake bisect](https://v8.dev/docs/flake-bisect)
  + [Handling of ports](https://v8.dev/docs/ports)
  + [Official support](https://v8.dev/docs/official-support)
  + [Merging & patching](https://v8.dev/docs/merge-patch)
  + [Node.js integration build](https://v8.dev/docs/node-integration)
  + [Reporting security bugs](https://v8.dev/docs/security-bugs)
  + [Running benchmarks locally](https://v8.dev/docs/benchmarks)
  + [Testing](https://v8.dev/docs/test)
  + [Triaging issues](https://v8.dev/docs/triage-issues)
* Debugging
  + [Arm debugging with the simulator](https://v8.dev/docs/debug-arm)
  + [Cross-compiling and debugging for ARM/Android](https://v8.dev/docs/cross-compile-arm)
  + [Debugging builtins with GDB](https://v8.dev/docs/gdb)
  + [Debugging over the V8 Inspector Protocol](https://v8.dev/docs/inspector)
  + [GDB JIT Compilation Interface integration](https://v8.dev/docs/gdb-jit)
  + [Investigating memory leaks](https://v8.dev/docs/memory-leaks)
  + [Stack trace API](https://v8.dev/docs/stack-trace-api)
  + [Using D8](https://v8.dev/docs/d8)
  + [V8 Tools](https://v8.dev/tools)
* Embedding V8
  + [Guide to embedding V8](https://v8.dev/docs/embed)
  + [Version numbers](https://v8.dev/docs/version-numbers)
  + [Built-in functions](https://v8.dev/docs/builtin-functions)
  + [i18n support](https://v8.dev/docs/i18n)
  + [Untrusted code mitigations](https://v8.dev/docs/untrusted-code-mitigations)
* Under the hood
  + [Ignition](https://v8.dev/docs/ignition)
  + [TurboFan](https://v8.dev/docs/turbofan)
  + [Torque user manual](https://v8.dev/docs/torque)
  + [Writing Torque built-ins](https://v8.dev/docs/torque-builtins)
  + [Writing CSA built-ins](https://v8.dev/docs/csa-builtins)
  + [Adding a new WebAssembly opcode](https://v8.dev/docs/webassembly-opcode)
  + [Maps, aka "Hidden Classes"](https://v8.dev/docs/hidden-classes)
  + [Slack Tracking - what is it?](https://v8.dev/blog/slack-tracking)
  + [WebAssembly compilation pipeline](https://v8.dev/docs/wasm-compilation-pipeline)
* Writing optimizable JavaScript
  + [Using V8’s sample-based profiler](https://v8.dev/docs/profile)
  + [Profiling Chromium with V8](https://v8.dev/docs/profile-chromium)
  + [Using Linux perf with V8](https://v8.dev/docs/linux-perf)
  + [Tracing V8](https://v8.dev/docs/trace)
  + [Using Runtime Call Stats](https://v8.dev/docs/rcs)