

Compilation on the web

Matyáš Racek, xracek11

The native language of the web platform is JavaScript. Web developers today use it as a source to write their apps, but also as a compilation target. Libraries such as Webpack or Babel provide tools to transform source code in various ways before it is deployed. This is mostly done to ensure backward compatibility and to optimize for runtime speed and loading time. While all of these tools are good, they mostly compensate for limitations of the platform and JavaScript itself.

JavaScript has some features which make it very hard to optimize. Most notably it is not strictly typed. Therefore it is challenging to write performant code for use cases like audio/video manipulation or game engine. To solve this problem, ASM.js and Emscripten projects were created. ASM.js is a highly optimizable subset of Javascript and Emscripten is compiler from LLVM to ASM.js, which makes it possible to run programs written in languages like C or C++ in a web browser.

Inspired by ASM.js success, W3C group took this idea even further and started developing a binary format for the web called WebAssembly aka WASM. WASM recently reached version 1.0 and is now supported in all major browsers. Developers can use it to write fast code for web apps in statically typed languages like C, C++, Rust and more.

The presentation focuses mainly on WebAssembly, current state, future, when to use it and how to use it.