+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 05/09/2023

Sign up

WebAssembly with Rust training

3 days (21 hours)

Presentation

WebAssembly is essentially the spiritual successor to ASM.js, and is developed by Google, Microsoft, Mozilla and others. Its main benefits are faster loading times for users and code compatibility (WebAssembly will support older platforms by translating the wasm binary into asm.js code.

WebAssembly represents a fundamental advance in the web platform. It enables code from high-level languages such as C/C++/Rust to be executed on the Web with performance similar to that of native applications.

WebAssembly is designed to be used in conjunction with JavaScript. Thanks to the WebAssembly JavaScript API, WebAssembly modules can be loaded into a JavaScript application and functionality shared between the two. This allows you to take advantage of WebAssembly's performance and JavaScript's flexibility, even if you don't know how to write WebAssembly code.

This course will show you how to use this technology to write high-performance applications that run in the browser.

You'll be introduced to powerful WebAssembly concepts that will help you write lightweight, powerful Web applications with native performance. Learning WASM starts by familiarizing you with the evolution of Web programming and what can be done with this tool. You'll then see how to move from JavaScript to asm.js via WebAssembly.

As you progress, you'll analyze the anatomy of a WebAssembly module and the relationship between binary and text formats, as well as the corresponding JavaScript API.

Objectives

- Rust language basics
- Understanding WASM concepts
- Creating a Web application in Rust from scratch
- Going further with Rust and WASM

Target audience

Web Developer

Prerequisites

Knowledge of JavaScript, C/C++

Further information

To complete this training, we offer Node.JS training to help you manage the integration and transition between the two frameworks.

Program of our WebAssembly with Rust training course

Introduction to WebAssembly

- WebAssembly history
- How WebAssembly works
- Safety at the heart of WASM
- Module format
- Browser communication and Javascript API
- WASM support in browsers
- The future of WebAssembly and its impact on Web development
- WABT: understanding the binary format
- WABT: understanding the text format
- Emscripten: compile C/C++ in WASM

Introduction to Rust

- The history of Rust
- Downloading and installing the Rust environment: rustup, cargo, rustc, crates.io
- My first program
- Writing unit tests: unit tests
- Dependencies: Using a crate

Rust and WebAssembly

- Project organization and tools
- Basic types
- Conditions and loops
- Pattern matching
- Structures and enumerations
- Error management
- Options
- Ownership
- References in Rust
- Lifetime
- Dynamic allocation
- Functional programming
- Features in Rust
- Server-side application with NodeJS
- Using WebGL

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.