AMBIENT-IT

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

Updated on 16/05/2023

Sign up

WebAssembly 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. As always, we'll teach you the latest version, WebAssembly 1.1.

Objectives

- Exploring WebAssembly elements
- Creating and loading a WebAssembly module
- Creating an application from scratch
- Create single-page web applications with .Net or C++, without JavaScript code

Target audience

Web Developer

Prerequisites

- Programming skills
- Knowledge of JavaScript, C/C++

Further information

- Discover how WebAssembly and the ultra-fast Rust language work together with our WebAssembly with Rust training course.
- To complete this training, we offer Node.JS training to help you manage integration and transition between the two frameworks.

WebAssembly Training Program

Introduction to WebAssembly

- What is WebAssembly?
- What problems can it solve?
- How does it work?
- Structure of a WebAssembly module
- WebAssembly text format
- How secure is WebAssembly?
- What languages can I use to create a WebAssembly module?
- Where can I use my module?

WebAssembly Performance

- WebAssemmbly vs. JavaScript Performance
- Installing Emscripten toolkit (emcc)

WebAssembly browser support

- Current browser support
- Browser roadmaps for WebAssembly support
- Support for Parralel WASM/JavaScript (Pollyfill)

WebAssembly module

- Create your first module
- The Emscripten toolbox
- Web assembly modules
- Emscripten output options
- Compile C or C++ with Emscripten and use the HTML template
- Generate JavaScript plumbing code by Emscripten
- Make Emscripten generate only the WebAssembly file
- Feature detection: How to test whether WebAssembly is available
- Use C or C++ to create a module with Emscripten plumbing
- Use C or C++ to create a module without Emscripten
- Examples of real-life use cases

Creating your first WebAssembly module

- The Emscripten toolbox
- Web assembly modules

Dynamic link: Implementation

- Dynamic link: Advantages and disadvantages
- Dynamic link options
- Examining dynamic links
- Creating WebAssembly modules
- Adjusting the web page

Threading: Web workers & pthreads

- Advantages of web workers
- Considerations when using web workers
- Prefetching a WebAssembly module using a web worker
- Using pthreads

WebAssembly modules in Node.js

- Server-side validation
- Working with modules built in Emscripten
- Using the WebAssembly JavaScript API

Debugging & testing

- Text format
- Create basic game logic using WebAssembly text format
- Generating a WebAssembly module from text format
- The module generated by Emscripten
- Creating HTML and JavaScript files
- Viewing results

Debugging customization

- Adjust HTML
- Display number of trials
- Increase the number of attempts
- Summary screen update
- Installing the JavaScript test framework
- Test creation and execution

Functions: ccall, cwrap, direct, Emscripten macros

- ccall, cwrap, Direct function calls
- Passage : Passing an array to a module
- emscripten_run_script macros
- EM_JS macros
- EM_ASM macros

WebAssembly modules

- Module declaration
- Exports
- Index and starting method
- Linear memory
- Code insertion
- Debugging information
- Integration with ECMAScript (ES6)
- Compile C or C++ with Emscripten
- Creating an HTML template
- Interop with JavaScript
- Call to JavaScript
- Calling the module from JavaScript

Compilation and abstract syntax trees (AST)

- Why AST?
- Syntax and semantic analysis
- Properties and annotations
- AST Design
- Variable types
- Left and Right components
- Identifiers

- Useful design templates
- Binary serialization of ASTs

WebAssembly text format

- Displaying the source on a WebAssembly module
- S-Expression compilation and online assembly
- Symbol integration debugging
- Profiling

Creating applications from scratch

- Application design
- Server requirements
- Interact with server services

Advanced WebAssembly modules

- Integration with Node.JS
- Thread options
- Using WebWorkers
- Using PThreads
- Features to come

Companies concerned

This course is aimed at companies, large or small, wishing to train their teams in a new, advanced computer technology.

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.

Training organization registered under number 11 75 54743 75. This registration does not imply government approval.
Ambient IT 2015-2023. All rights reserved. Paris, France - Switzerland - Belgium - Luxembourg