

Level up with WebAssembly

The Book:



Chapter 1. Introduction

- What is WebAssembly?
- Who uses WebAssembly?
- Hello World example
- Under the hood (advanced)



Chapter 2. Setup + Hello World

- Setup Emscripten container
- Run the Hello World example



Chapter 3. Wait but why?

- Why Use WebAssembly?
- When to use WebAssembly
- When to avoid WebAssembly



Chapter 4. Hit the ground running

- Hello World
- Call main() with function arguments
- Calling custom C functions from JavaScript
- Calling JavaScript functions from C
- A note about C++
- Calling main() with cwrap() (advanced)



Chapter 5. The Module Object

- What is the Module object?
- Configure initial conditions
- Custom stdout/stderr handling
- Commonly-used Module parameters



Chapter 6. File Management

- Enable the virtual file system
- How to mount files
- Commonly-used FS functions
- Enable gzip support



Chapter 7. Compiling Existing Tools

- A simple example
- Using Makefiles
- Compiling the jq command line tool
- Building an interactive jq app



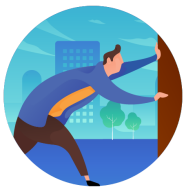
Chapter 8. Graphics

- A simple example
- Compile Pong to WebAssembly
- Compile Pacman to WebAssembly
- What's Next?



Chapter 9. Persisting the File System

- Create a persistent file system
- Reload the file system after page refresh
- Pre-loading files at compile-time
- Mounting File objects instead of strings



Chapter 10. WebAssembly + WebWorkers

- Introduction
- WebWorkers Recap
- A simple example
- Mount a File object



Chapter 11. Conclusion

- Summary
- Resources

The Starter Kit:

Include **The Book** plus:



Cheat Sheets

- Emscripten flags
- Module Object
- File System
- FileReader API
- WebWorkers
- Serverless WebAssembly



Guide: WebWorkers and FileReader APIs

WebWorkers

- What are WebWorkers?
- How to send and transfer data with `postMessage()`
- Typed Arrays as Views (advanced)

FileReader API

- Key Concepts
- A simple example



Guide: Serverless WebAssembly

- Introduction
- When to use Serverless WebAssembly
- Serverless Wasm with AWS, GCP, and Azure (Node.js)
- Serverless Wasm with Cloudflare (V8 isolates)
- Serverless Wasm with Fastly (Lucet runtime)
- Overview of Cloud providers, features, and pricing

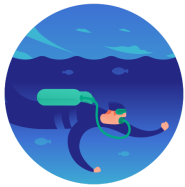
The Complete Edition:

Include **The Starter Kit** plus:



Command-Line Screencasts

- Setup + Hello World
- Call `main()` with custom arguments
- Call C and C++ functions from JavaScript
- Module Object: custom initialization
- Module Object: custom `stdout/stderr`
- File system: Mount files to the virtual file system
- File system: Read files from the file system
- File system: Enable `.gz` support
- Compile an off-the-shelf library
- Leverage Makefiles with `emmake`
- How to compile `jq` to WebAssembly



Guide: Porting UNIX tools to the web

- Introduction
- Port `awk`
- Port `diffutils` + build a sample application
- Port `coreutils` (`cut`, `sort`, `uniq`, `wc`, etc.)



Capstone Project

- Goals: K-Means Clustering
- Sample solution