

WebAssembly becoming the biggest platform

Sven Sauleau

2018

Sven Sauleau @svensauleau



“The Web is already the biggest platform and WebAssembly will expand it to rest of the world.”

— Sven Sauleau, 5min ago

An issue with JavaScript ?



It's not a good compilation target.

WebAssembly has different use cases¹

¹VPN, databases, games, platform emulation, VM, ...

- Deterministic and easier to reason about
- Designed to be **Safe** to execute

“[...] memory is disjoint from code space, the execution stack, and the engine's data structures; **therefore compiled programs cannot corrupt their execution environment**, jump to arbitrary locations, or perform other undefined behavior”

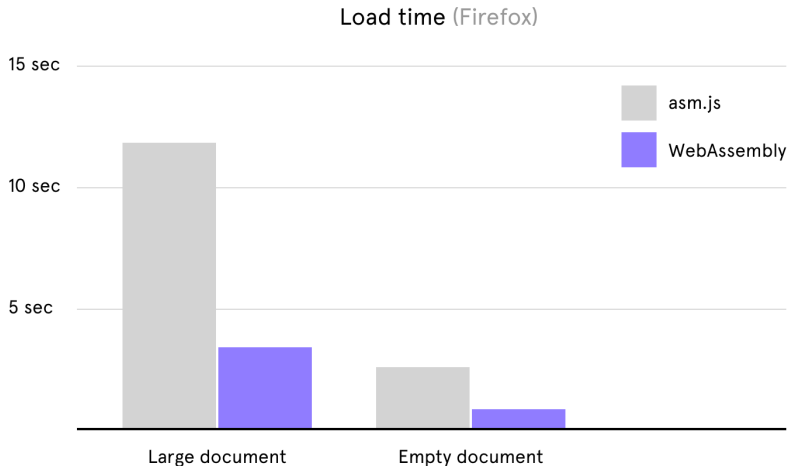
— Bringing the Web up to Speed with WebAssembly

Executes fast

- Near native speed (and JIT soon)
- Compilers optimize the code Ahead Of Time

Note: interop with JS can be slow

Compact and easy to decode



“Our load time improved by more than 3x [...]”

— Figma, medium

Safe and efficient representation

- Compact and easy to decode
- Streamable and parallelizable

How to use it?

Usage

```
1 $ my-dopi-compiler --target=wasm32 file
```

Languages ²

- .Net
- Astro
- Brainfuck
- C / C# / C++
- Elixir
- Faust
- Forest
- Forth
- Haskell
- Golang

- Java
- Kotlin/Native
- Kou
- Lua
- OCaml
- Plorth
- Rust
- Turboscript
- Wah
- Wracket
- Xlang

²<https://github.com/appcypher/awesome-wasm-langs>

Browser support

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android
			49				
			63		10.3		
		58	64	11	11.2		
11	16	59	65	11.1	11.3	all	64
	17	60	66	TP			
	18	61	67				
			68				



Is it going to kill
JavaScript ?

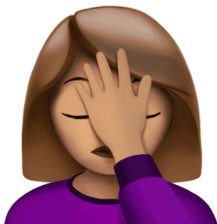
Ben Hartman
© 2010

No.

- WebAssembly will not replace JavaScript
- Other languages will target WebAssembly instead

A hand holds a torn piece of paper against a background of a cityscape. On the paper is a line drawing of a T-Rex holding a handgun. The text 'Will we compile JavaScript to WebAssembly?' is written in large, bold, black letters across the center of the paper. A thin orange horizontal line is drawn below the text.

**Will we compile
JavaScript to
WebAssembly?**



Please no.

- JavaScript is very **dynamic**
- **it lacks of static typing**
- **Lots of check** at runtime

AssemblyScript: A TypeScript to WebAssembly compiler ³

```
1 export function add(a: i32, b: i32): i32 {  
2     return a + b;  
3 }
```

³[AssemblyScript.org](https://assemblyscript.org)

Walt: is an alternative syntax for WebAssembly ⁴

```
1 export function fibonacci(n: i32): i32 {  
2     if (n <= 0) return 0;  
3     if (n == 1) return 1;  
4  
5     return fibonacci(n - 1) + fibonacci(n - 2);  
6 }
```

⁴<https://github.com/ballercat/walt>

The incoming parts

Builtin garbage collection

- You can ship your own
- Will track JavaScript refs as well as WebAssembly

Native threads with concurrency primitives ⁵.

⁵Locks, Atomics, etc.

Direct access to browser APIs

But you can import any JavaScript function

Write to the console

```
1  const importObject = {  
2    env: {  
3      log: msg => console.log(msg),  
4    }  
5  };
```

WASM or WAST

WASM: WebAssembly Binary Format ⁶

⁶.wasm

WAST: WebAssembly "Script" Format ⁷

⁷.wast

Tooling

Toolchain for WebAssembly (in JavaScript)⁸:

- WAST/WASM parser/printer
- Interpreter
- AST introspection/manipulation
- WAST/WASM optimizations

⁸<https://github.com/xtuc/webassemblyjs>

DCE in Webpack

```
1 bin = editWithAST(ast, bin, {
2
3   ModuleExport(path) {
4     const usedName = module.isUsed(name);
5
6     if (!usedName) {
7       path.remove();
8     }
9   }
10
11 });
```

Webpack integration demo

Slides on reacteu-2018.ralf.cc

Pictures by Ben Heine (Pencil Vs Camera)