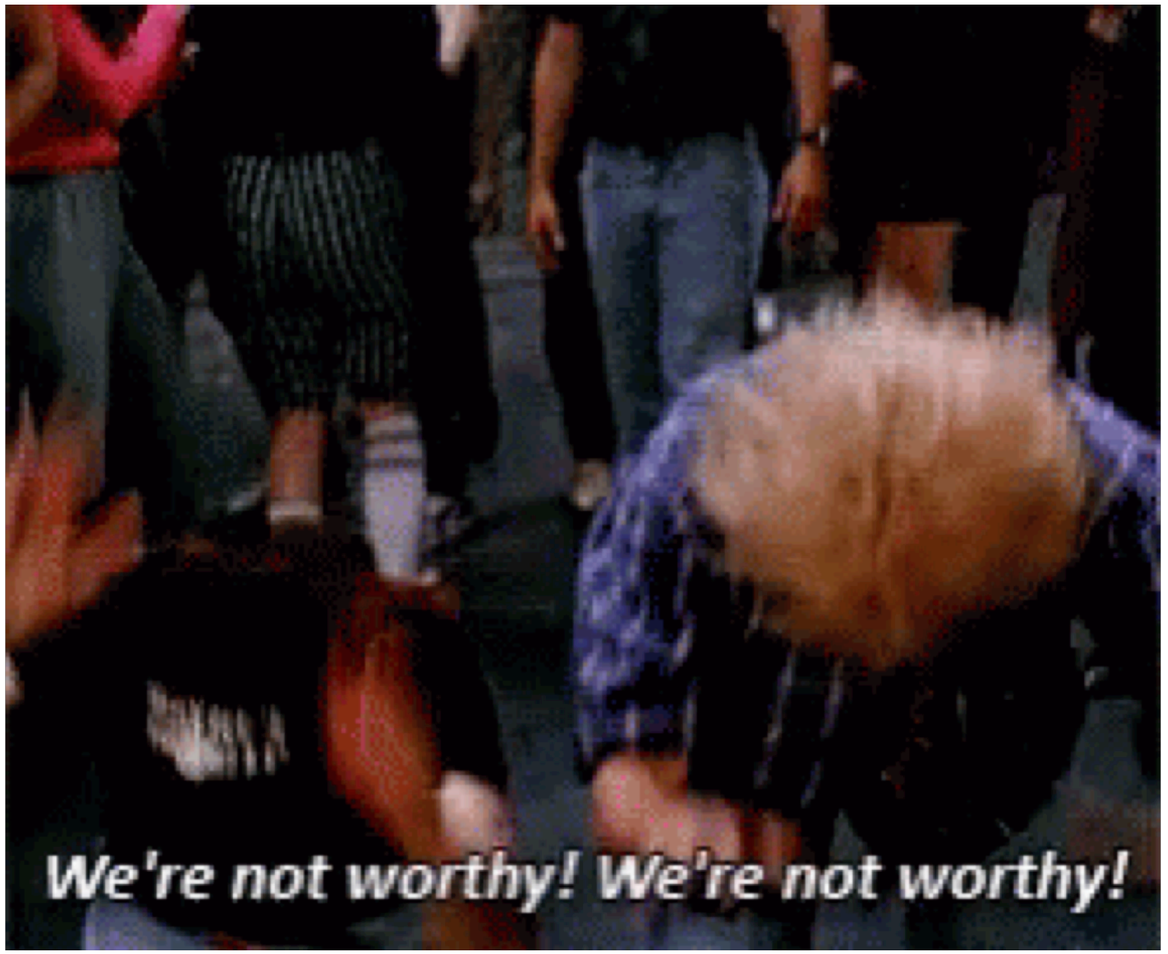
How did I get here?



WebAssembly Dance Party

Learn a Language: Web Assembly

Who is this guy? (@seanhelvey)

- M.S. Computer Science at New York University
- Web Development Instructor at Galvanize Boulder
- Teacher of JavaScript, Node.js, React, Redux
- Co-organizer of Front Range Elm Meetup

I do not work for a browser vendor!

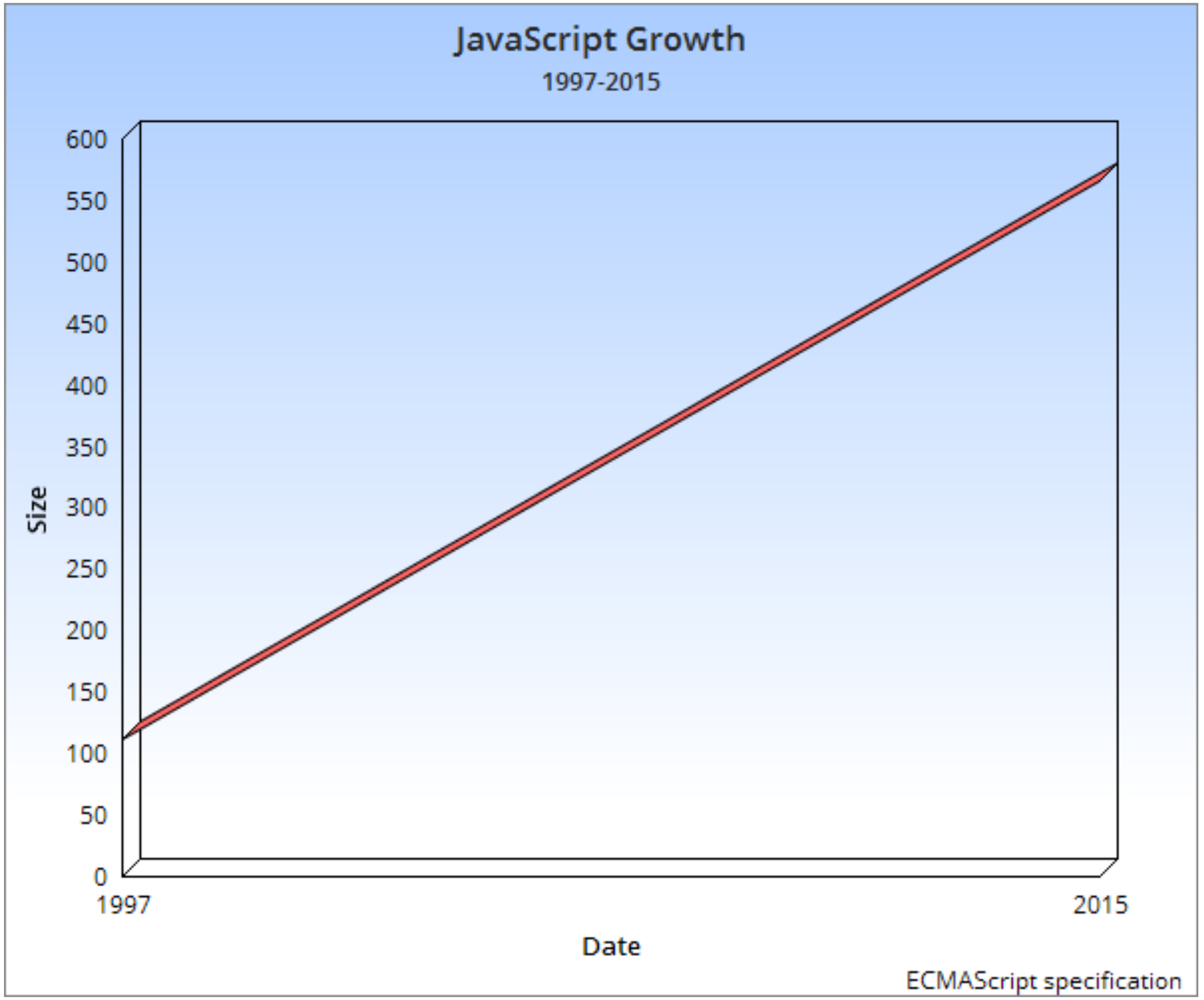
- Most talks you will find on WebAssembly are biased
- Not saying this is a bad thing, but let's be honest
- Also not trying to compile my C++ game to the web
- I'm an average web developer interested in the future

Objectives

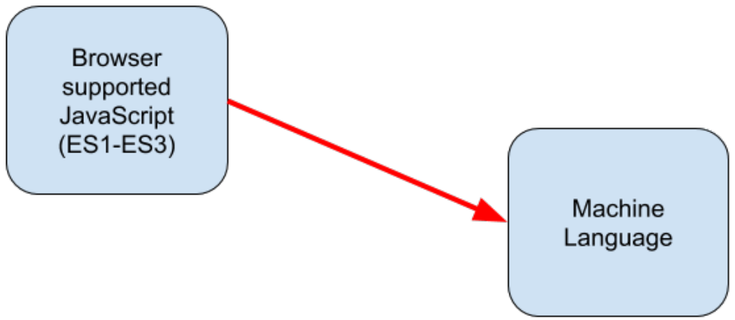
- Illustrate growth of ECMAScript
- Introduce WebAssembly (Wasm)
- Understand how to use Wasm
- Outline future vision

Discuss growth of ECMAScript specification

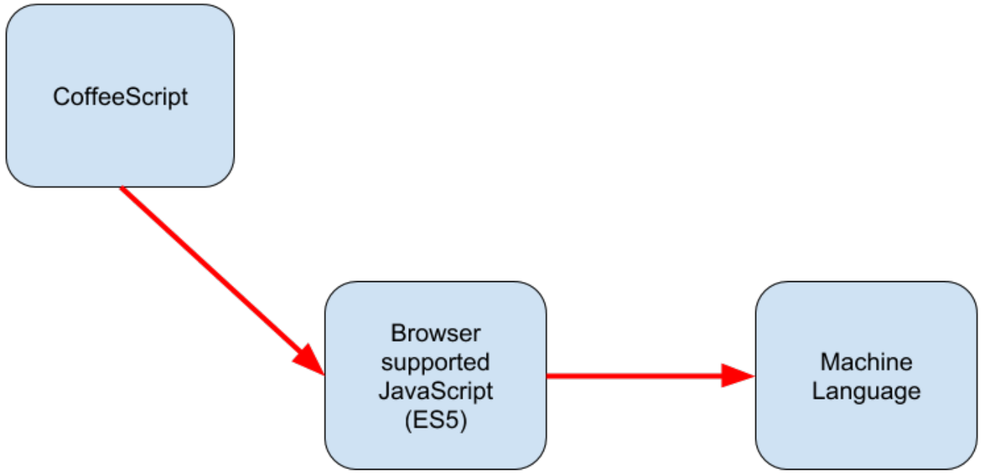
- 1995: JavaScript created
- 1997: *110* page ECMAScript specification
- 2015: *566* page ECMAScript specification


JavaScript Growth
1997-2015
600
550
500
450
400
350
300
250
200
150
100
50
0
Size
1997
2015
Date
ECMAScript specification

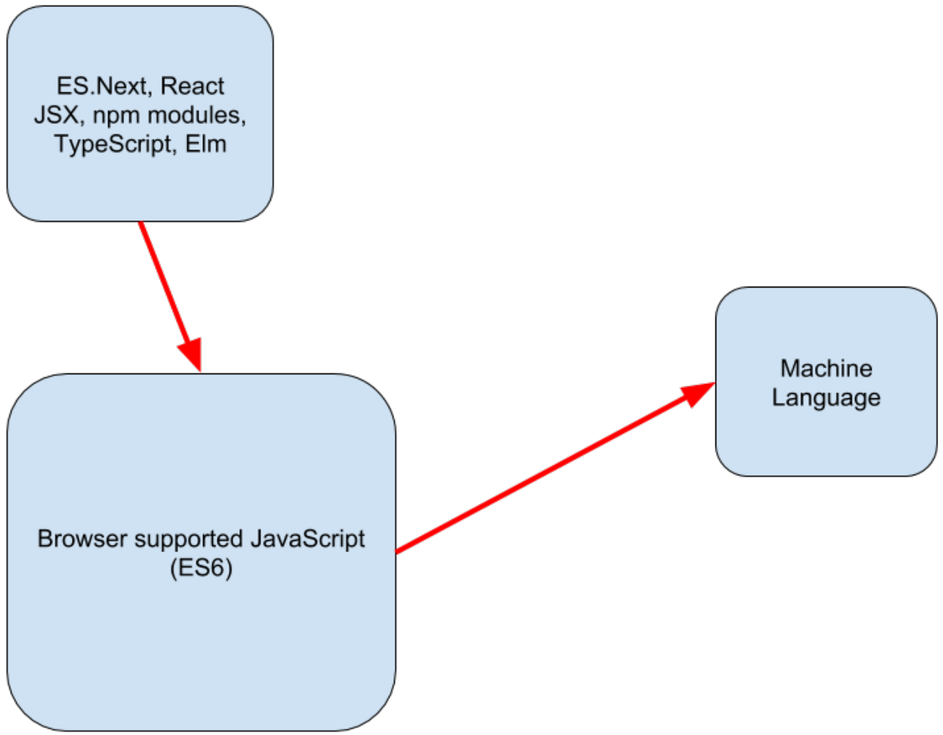
Back in the day



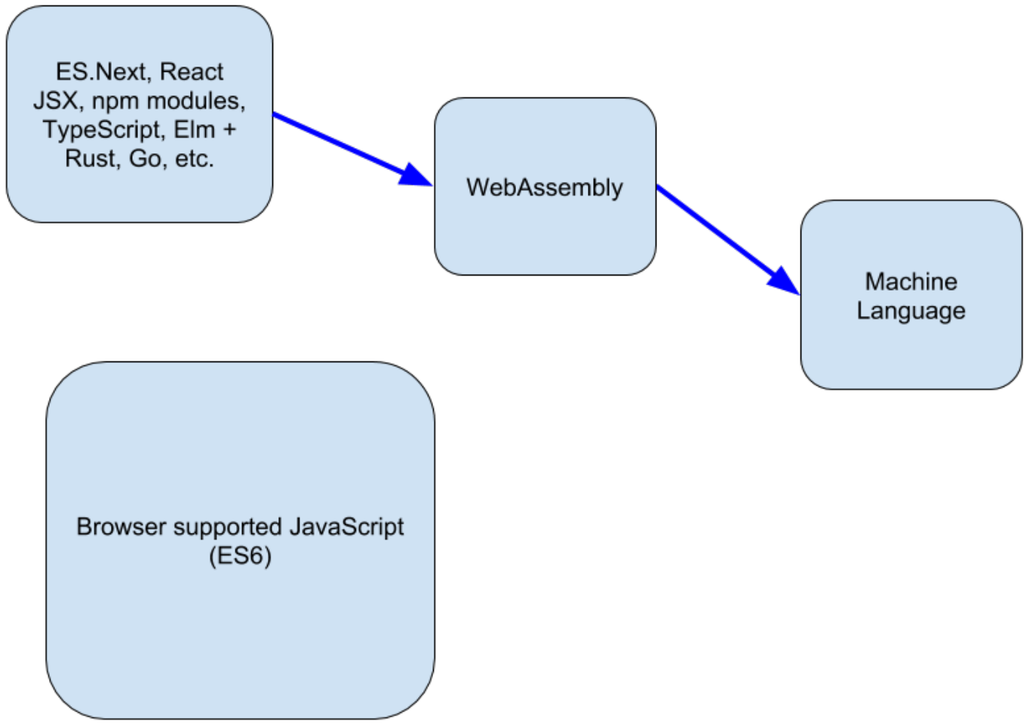
5-10 years ago



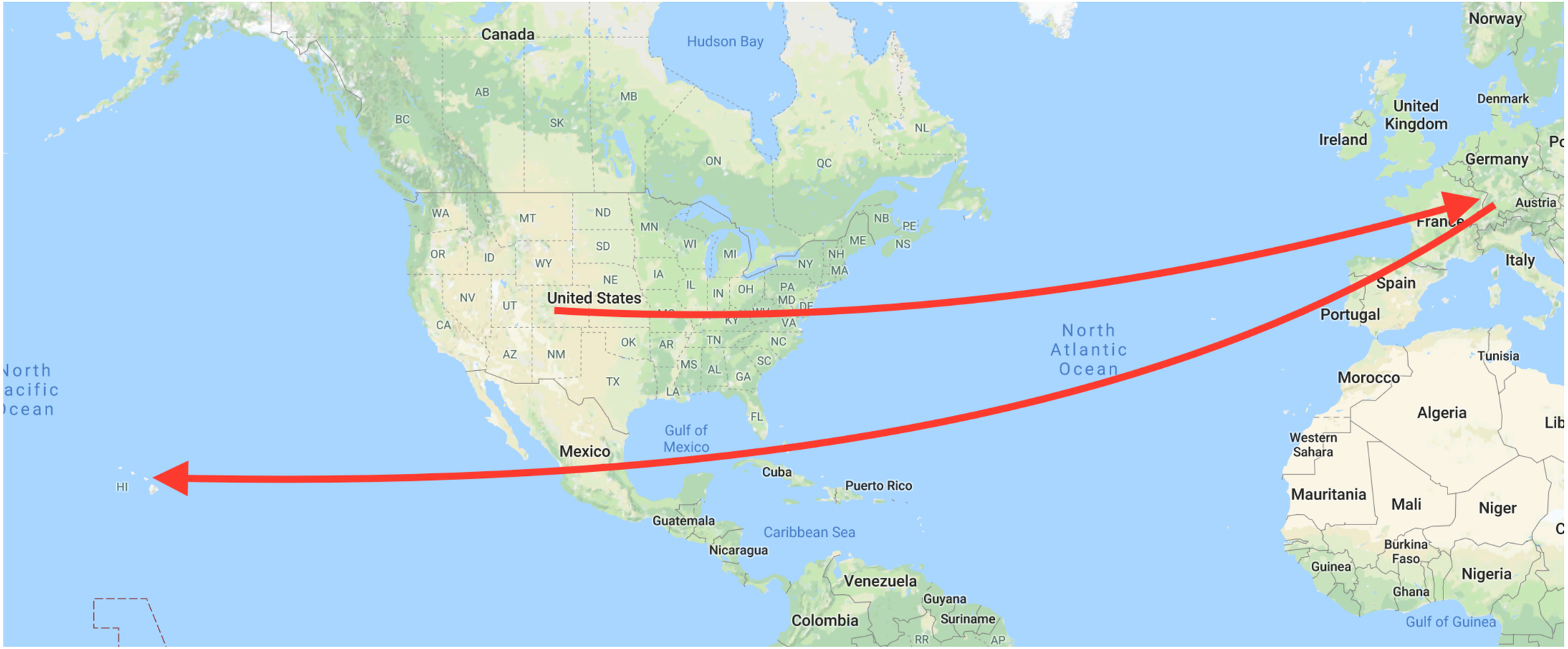
Yesterday


ES.Next, React
JSX, npm modules,
TypeScript, Elm
Browser supported JavaScript
(ES6)
Machine
Language

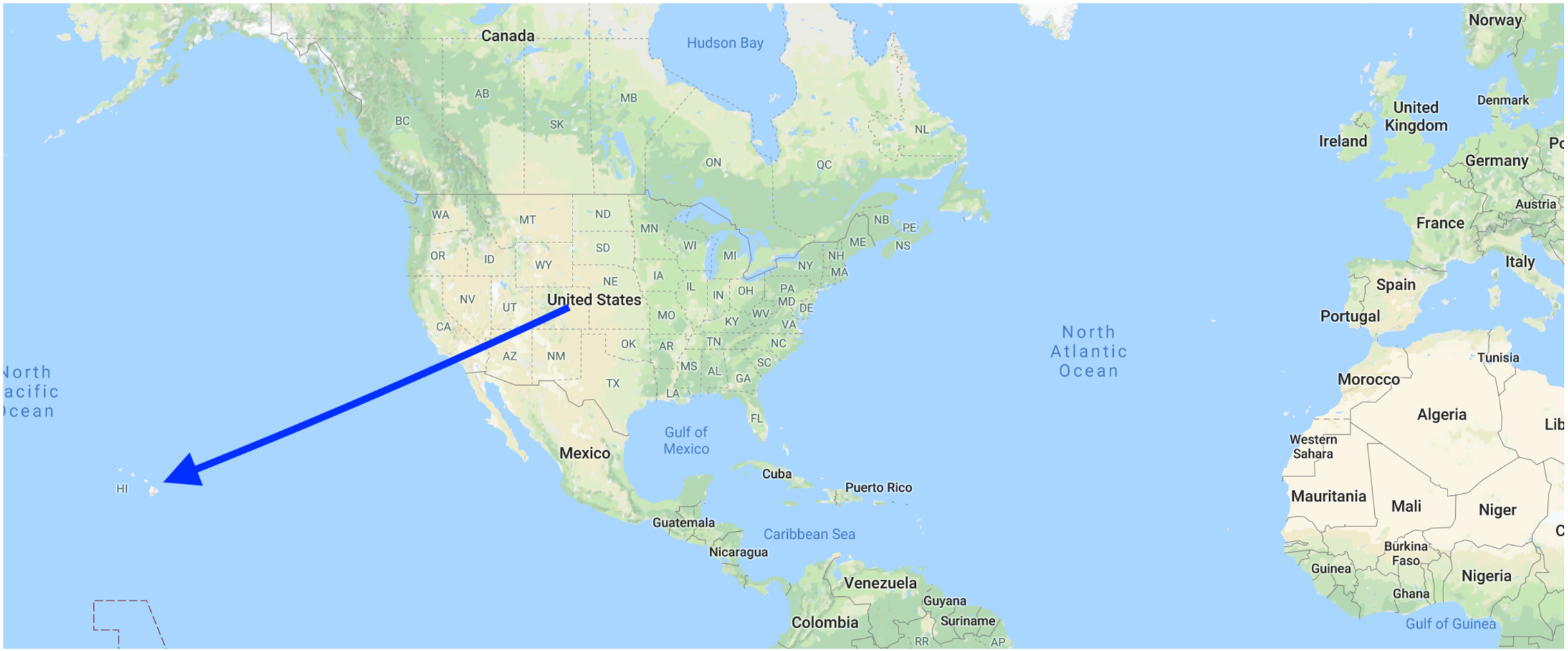
THE FUTURE (Today)



Indirect flight: CO -> Europe -> Hawaii



Direct flight: CO -> Hawaii



A student / teacher perspective on WebAssembly

- I’ve used over 20 different languages and trust me
- Teaching JavaScript is the hardest by far
- ES5 was hard to begin with
- ES6, 7, 8, etc. making it even more difficult

Programming Wisdom @CodeWisdom · Apr 28

"Walking on water and developing software from a specification are easy if both are frozen." - Edward V. Berard

11 850 1.9K

A student / teacher perspective on WebAssembly

- If you can learn JavaScript, you can learn anything!
- Learn to learn, unfamiliar environments, problem solving
- Many students get jobs working with other languages
- Must be able to talk about programming in general

Timeline of WebAssembly developments

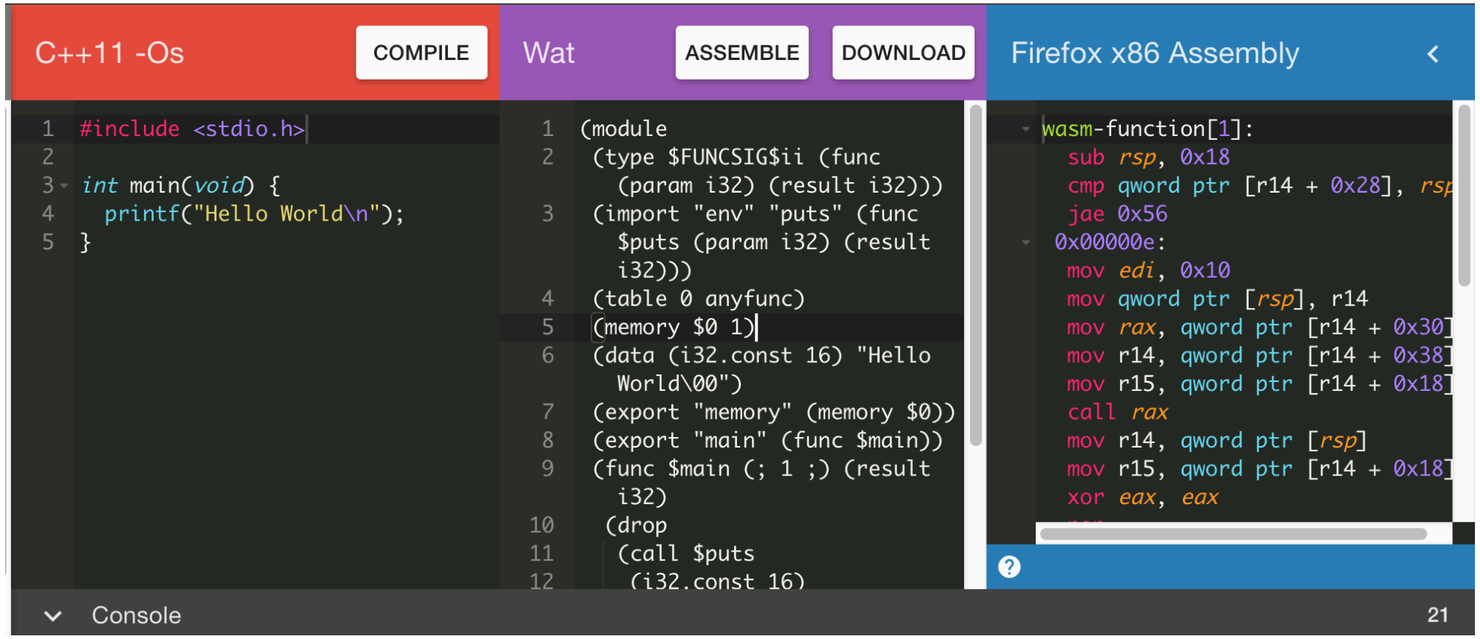
- March 2013 - Predecessor asm.js released
- June 2015 - WebAssembly working group formed
- November 2017 - WebAssembly MVP released
- February 2018 - W3C public draft released

.wat is WebAssembly or Wasm?

- A binary instruction format for a virtual machine
- A portable target for high-level languages
- Java / JVM solved similar portability issue

.wat is WebAssembly or Wasm (Cont…)

- 32 and 64 bit integer and floating point types
- File formats



.wat is WebAssembly or Wasm (Cont...)

- Modules can be loaded in either the UI thread or in a Web Worker
- Run in a safe, sandboxed execution environment & enforces browser's policies and permissions

Memory

- Wasm Memory is represented as a contiguous range of untyped bytes
- The memory accessible by a particular WebAssembly Instance is confined to a range
- Libraries have separate memories that are fully isolated from each other

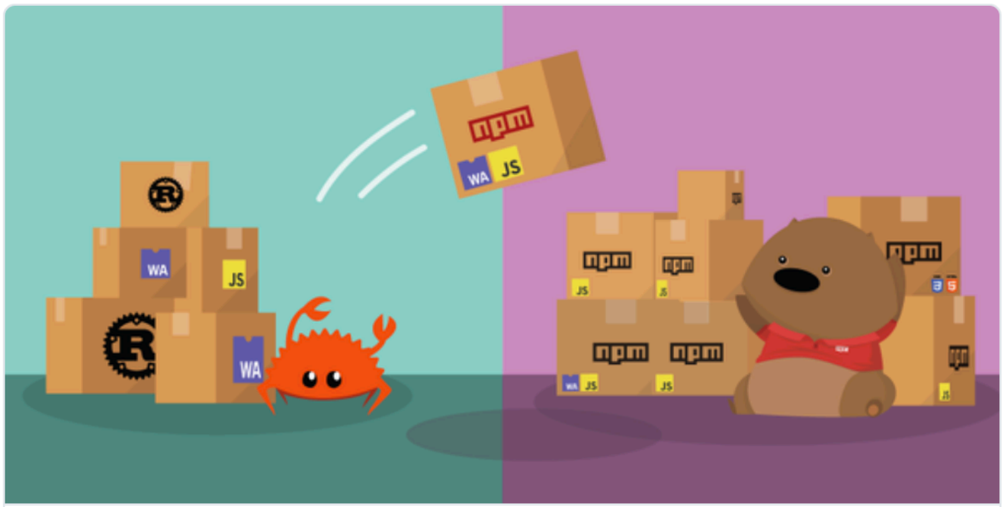
Tables

- Wasm Tables are resizable typed arrays of references
- While Memory provides a resizable typed array of raw bytes, it is unsafe for references
- In the current iteration, functions are the only valid reference type

How can I use it?

Laurie Voss @seldo · May 3

We are EXTREMELY excited to see the amazing work being done by @ag_dubs, @linclark and friends in getting Rust modules transformed into WASM and published to the npm registry, opening up a whole new world of possibilities for highly performant web apps.



Hello wasm-pack! – Mozilla Hacks - the Web developer blog

Introducing wasm-pack, a new tool for assembling and packaging Rust crates that target WebAssembly. These packages can be published to th...

hacks.mozilla.org

wasm-bindgen

```rust
// Called by our JS entry point to run the example
#[wasm_bindgen]
pub fn run() {
    let val = document.createElement("p");
    val.set_inner_html("Hello from Rust!");
    document.body().append_child(val);
}
```

Emscripten

- When you've written a new code module in a language like C/C++, you can compile it into WebAssembly using a tool like Emscripten

```
emcc hello.c -s WASM=1 -o hello.html
```

Future vision

- Is it really just for C/C++/Rust?
- Language quality race to the top
- More modularity and portability
- JavaScript (maybe) as glue code
- Choose the right tool for the job!

Objectives

- Illustrate growth of ECMAScript
- Introduce WebAssembly (Wasm)
- Understand how to use Wasm
- Outline future vision

Thank you!