1 WebAssembly for Web Developers

GOOGLE I/O 2019

⇒ Emscripten → initially meant for asm.js

↳ quickly accomodated webassembly

Also does POSIX emulations

① Compiling the library.

↳ No support for threads and SIMD.

② Write "Bridge code" ⇒ Construction a function to call from Javascript.

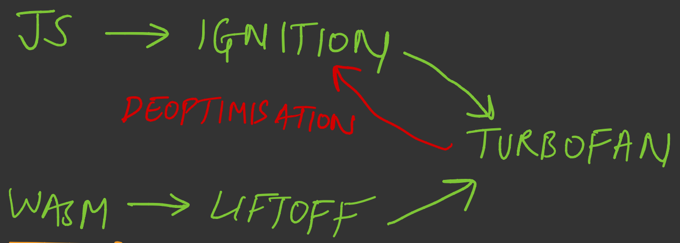
WASM-PACK → Turns RUST code to WebAssembly modules

↳ has smaller glue code as compared to C++.

JS vs WASM

Both have same PEAK PERFORMANCE

↳ will increase for webAssembly after threads and SIMD support.



↳ More predictable performance.

* WebAssembly threads :-

- → Porting multithreaded applications
- → SHARED LINEAR MEMORY as SharedArrayBuffers → For webworkers

  Disabled in some browsers
- → Atomic operations.

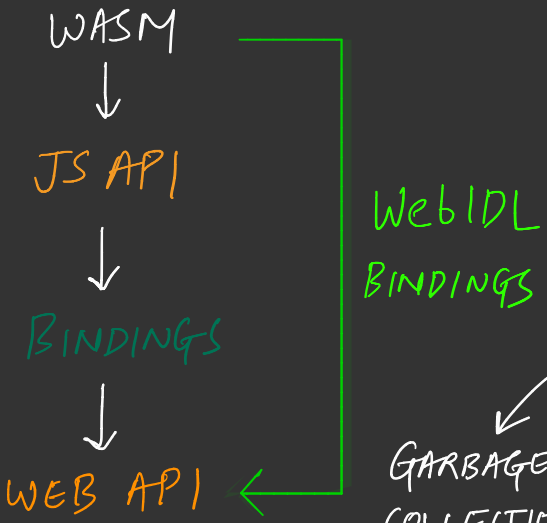
* SIMD Extension

* REFERENCE TYPES

→ Any ref type so that it can access arbitrary Javascript values

* WebIDL BINDINGS

→ Optimise calls from WASM to existing Web APIs in the browser.



* OTHER FEATURES

- GARBAGE COLLECTION

  High level languages
- TAIL CALL OPTIMIZATION

  Functional languages
- EXCEPTION HANDLING

  C++ style exceptions.