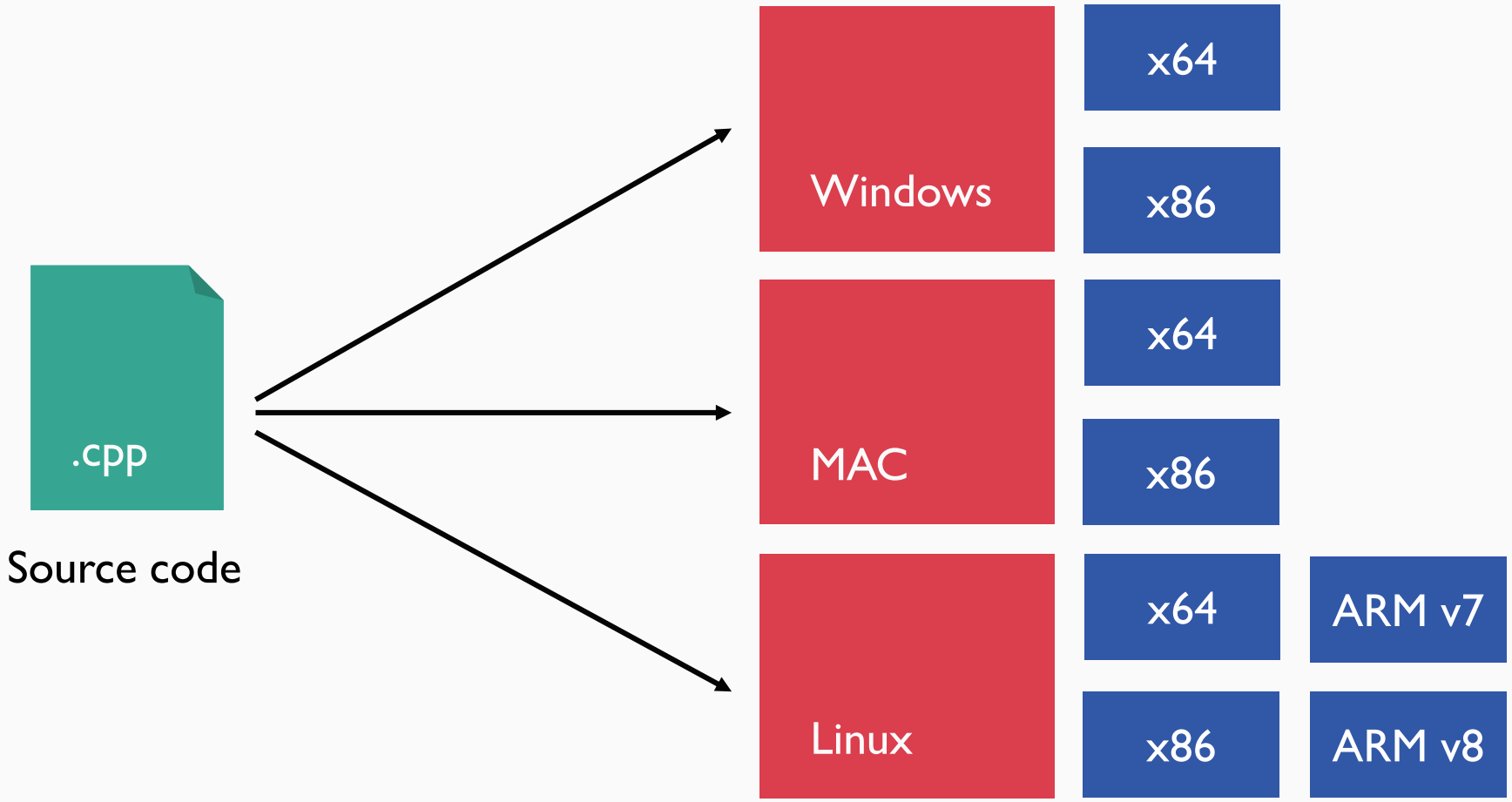
How WebAssembly is changing the Web and what it means for you

Boyan Mihaylov
@boyanio
boyan.io

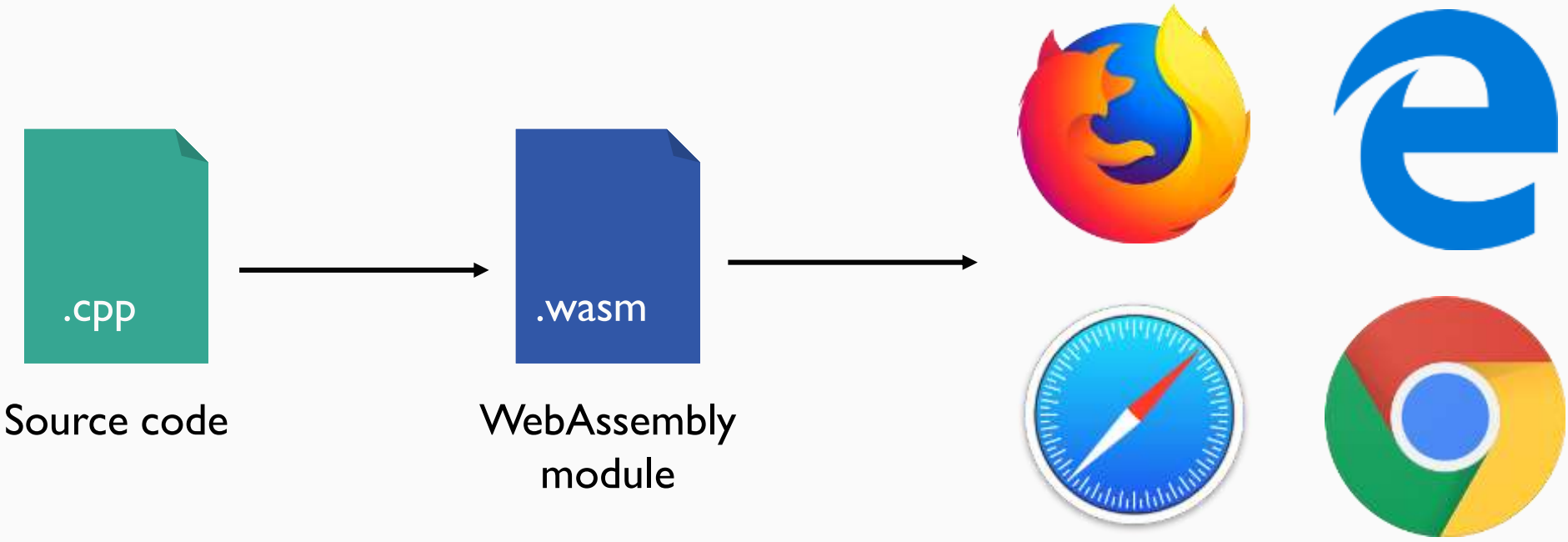
WebAssembly (*WASM*) is compiler target for programs on the Web

```
C:\wasm>type index.c
#include <stdio.h>

int main(void) {
        printf("Hello, cool people!\n");
        return 0;
}
C:\wasm>clang index.c

C:\wasm>a.exe
Hello, cool people!

C:\wasm>emcc -o a.js index.c

C:\wasm>node a.js
Hello, cool people!
```

Traditional multi-target compilation



Multi-target compilation with WebAssembly



```
function add(a, b) {
  return a + b;
}
```

```
> add(2, 3)
<· 5
> add("a", 5)
<· "a5"
> add("a", null)
<· "anull"
> add(5, {})
<· "5[object Object]"
> add({}, "a")
<· "[object Object]a"
> add("a")
<· "aundefined"
```

```
> '7' - 3
< 4
```

weak typing, implicit conversion

```
> '7' + 3
< "73"
```

…not really consistent

```
> '7' - '3'
< 4
```

string – string = number ?

```
> 7 + '3'
< "73"
```

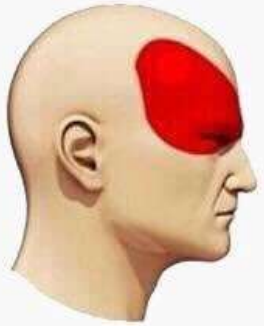
“+” is for concatenation

```
> 7 + + '3'
< 10
```

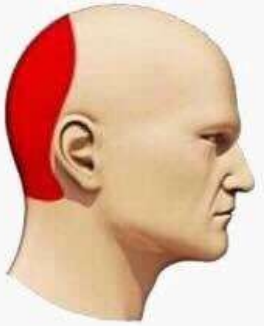
“+ +” is for addition ?

Types of Headaches

Migraine



Hypertension



Stress

JavaScript

@boyanio

WebAssembly is a typed language

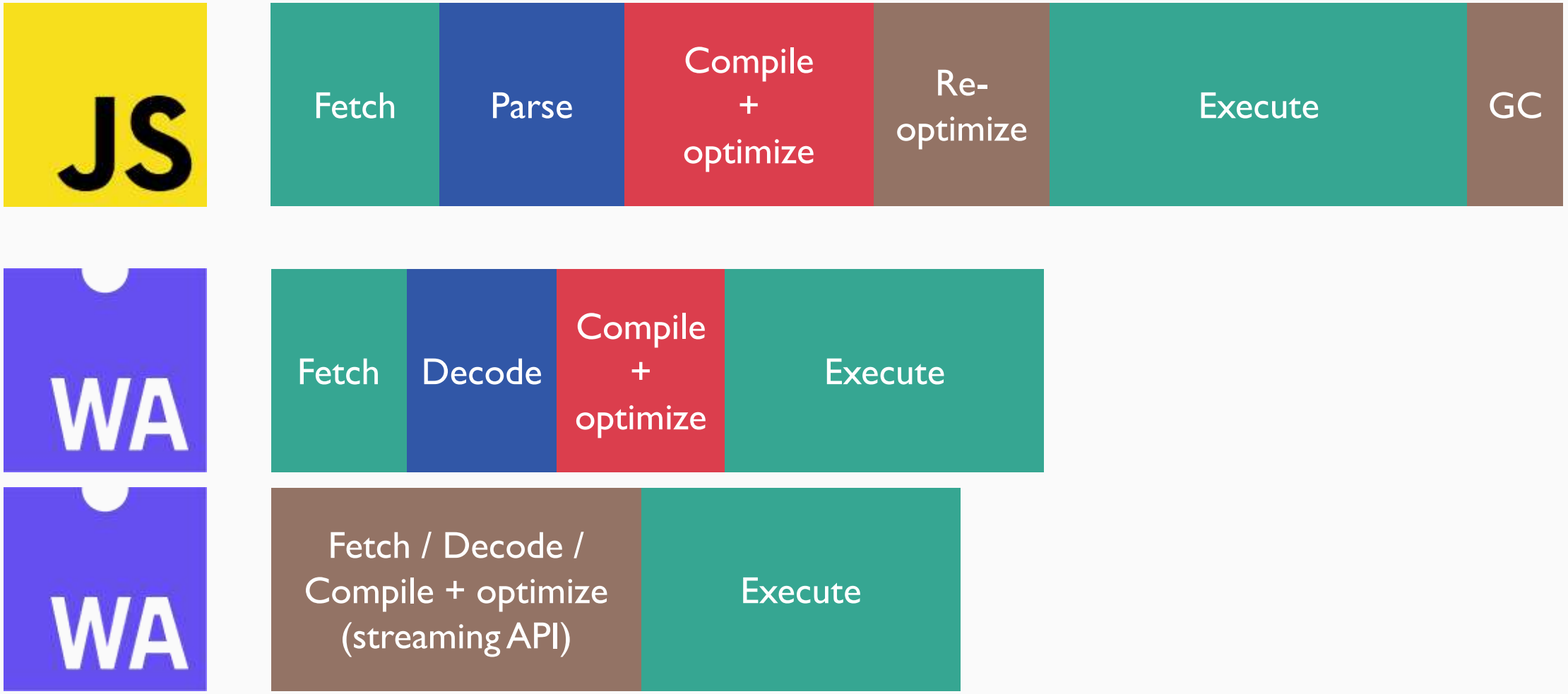
It supports 32 and 64-bit integers (i32, i64) and floating points (f32, f64)

Binary representation (.wasm)

```
0061 736d 0100 0000 0187 8080 8000 0160
027f 7f01 7f03 8280 8080 0001 0004 8480
8080 0001 7000 0005 8380 8080 0001 0001
0681 8080 8000 0007 9080 8080 0002 066d
656d 6f72 7902 0003 6164 6400 000a 8d80
8080 0001 8780 8080 0000 2001 2000 6a0b
```

Textual representation (.wat)

```
(module
  (table 0 anyfunc)
  (memory $0 1)
  (export “memory” (memory $0))
  (export “add” (func $add))
  (func $add (; 0 ;) (param $0 i32) (param $1 i32) (result i32)
    (i32.add
      (get_local $1)
      (get_local $0)
    )
  )
)
```


JS
Fetch
Parse
Compile + optimize
Re-optimize
Execute
GC
WA
Fetch
Decode
Compile + optimize
Execute
WA
Fetch / Decode / Compile + optimize (streaming API)
Execute

@boyanio

WebAssembly provides consistent, predictable performance

16 Anim (2-214)

System WebAssembly

NumRunners 15

Close Controls

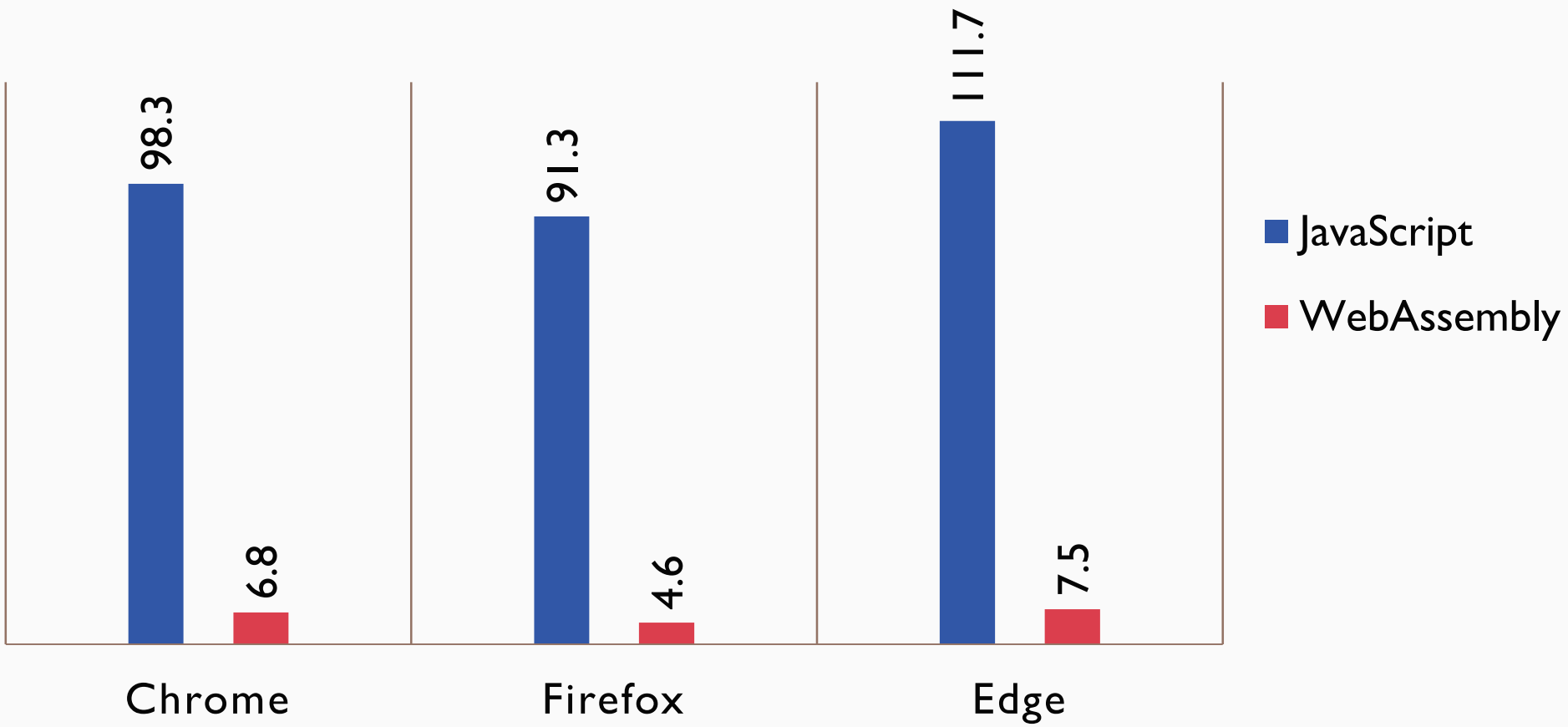
3D animation performance

https://github.com/sessamekesh/wasm-3d-animation-demo

@boyanio

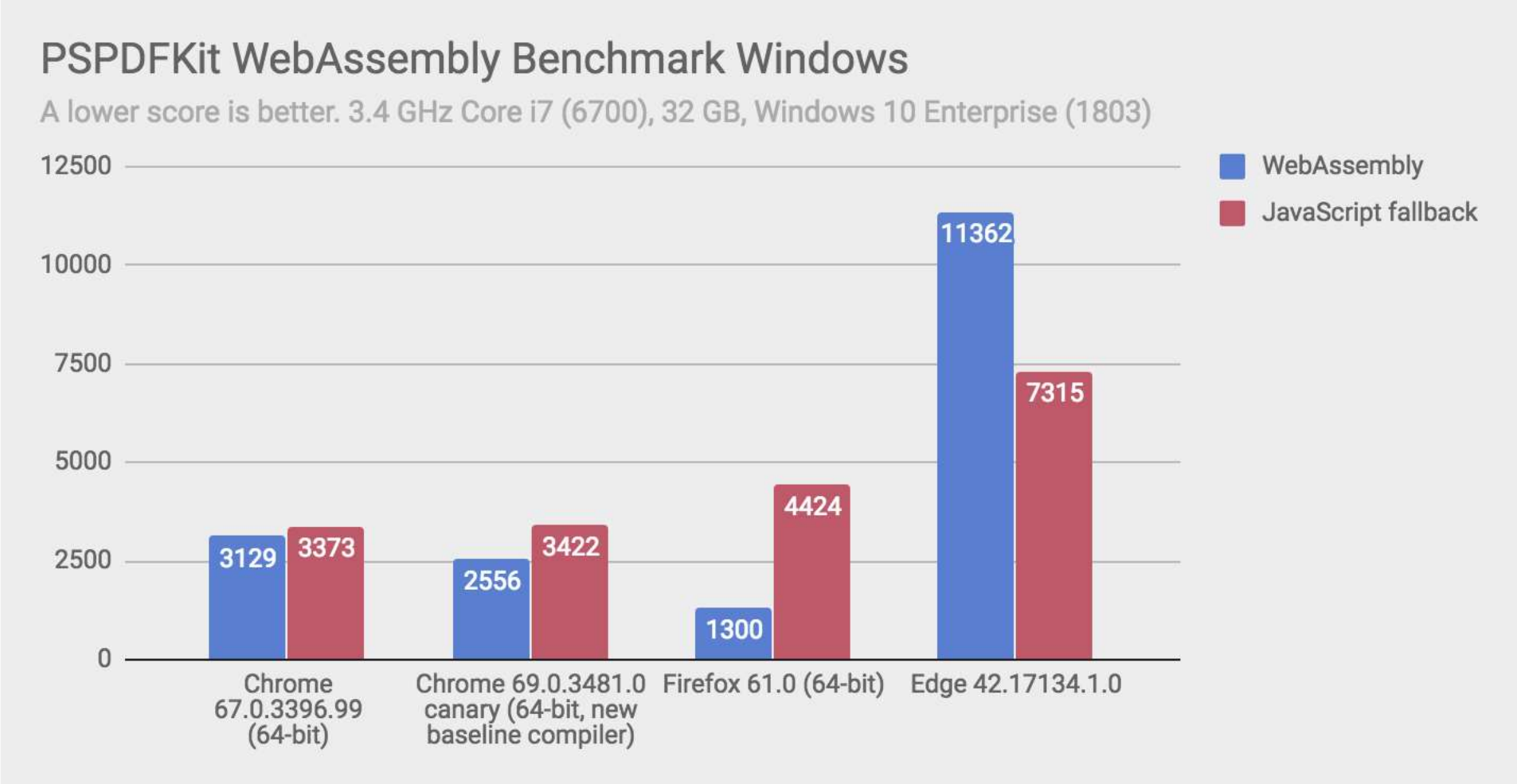
Performance comparison

Average animation time (ms)



https://www.lucidchart.com/techblog/2017/05/16/webassembly-overview-so-fast-so-fun-sorta-difficult/

A real-world WebAssembly benchmark



https://pspdfkit.com/blog/2018/a-real-world-webassembly-benchmark/

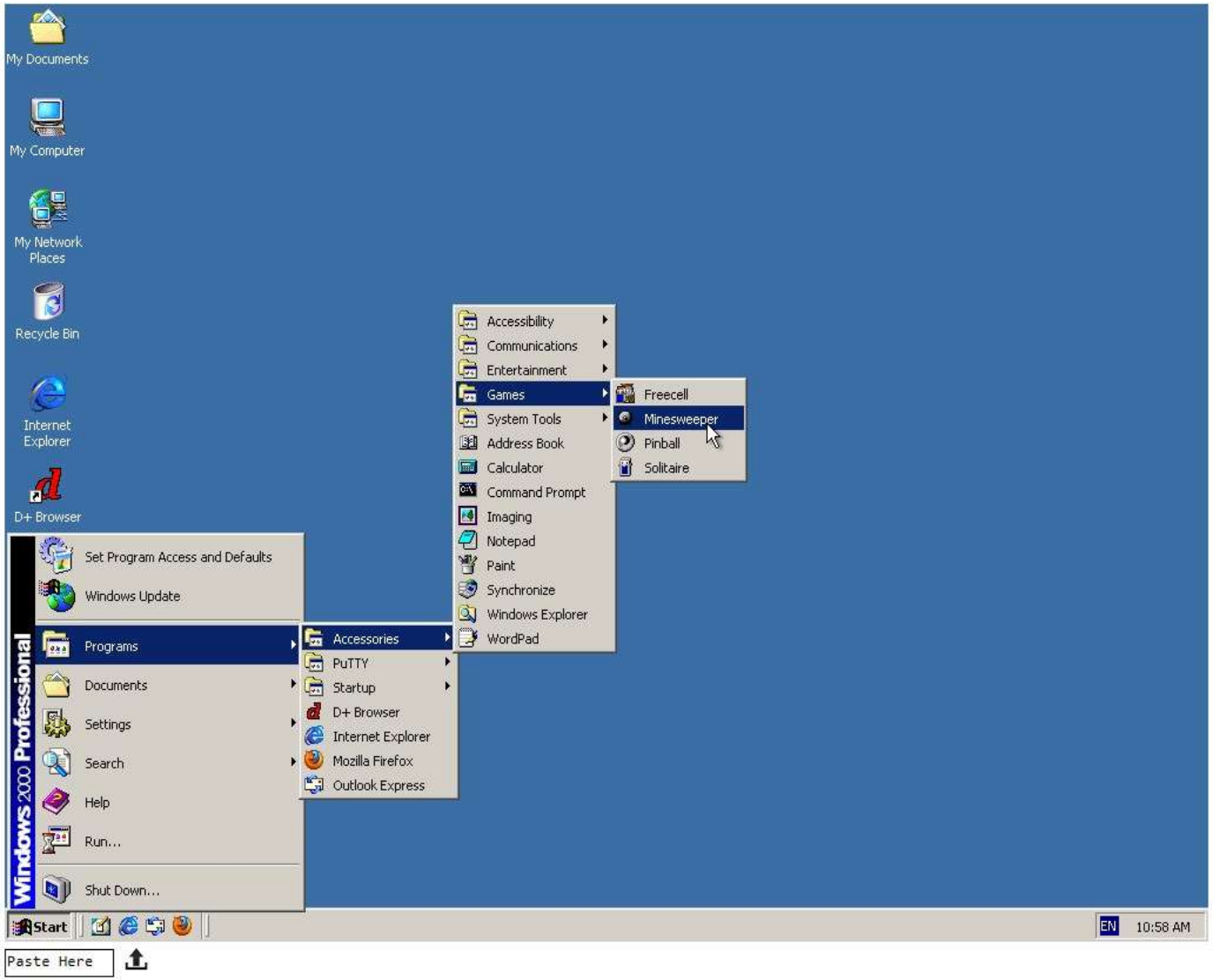
The Adobe Flash plugin has crashed.
Send crash report

Reusing code on the Web

@boyanio

WebAssembly enables code reusability between native and Web

What can we do with WebAssembly?



Windows 2000 in the browser

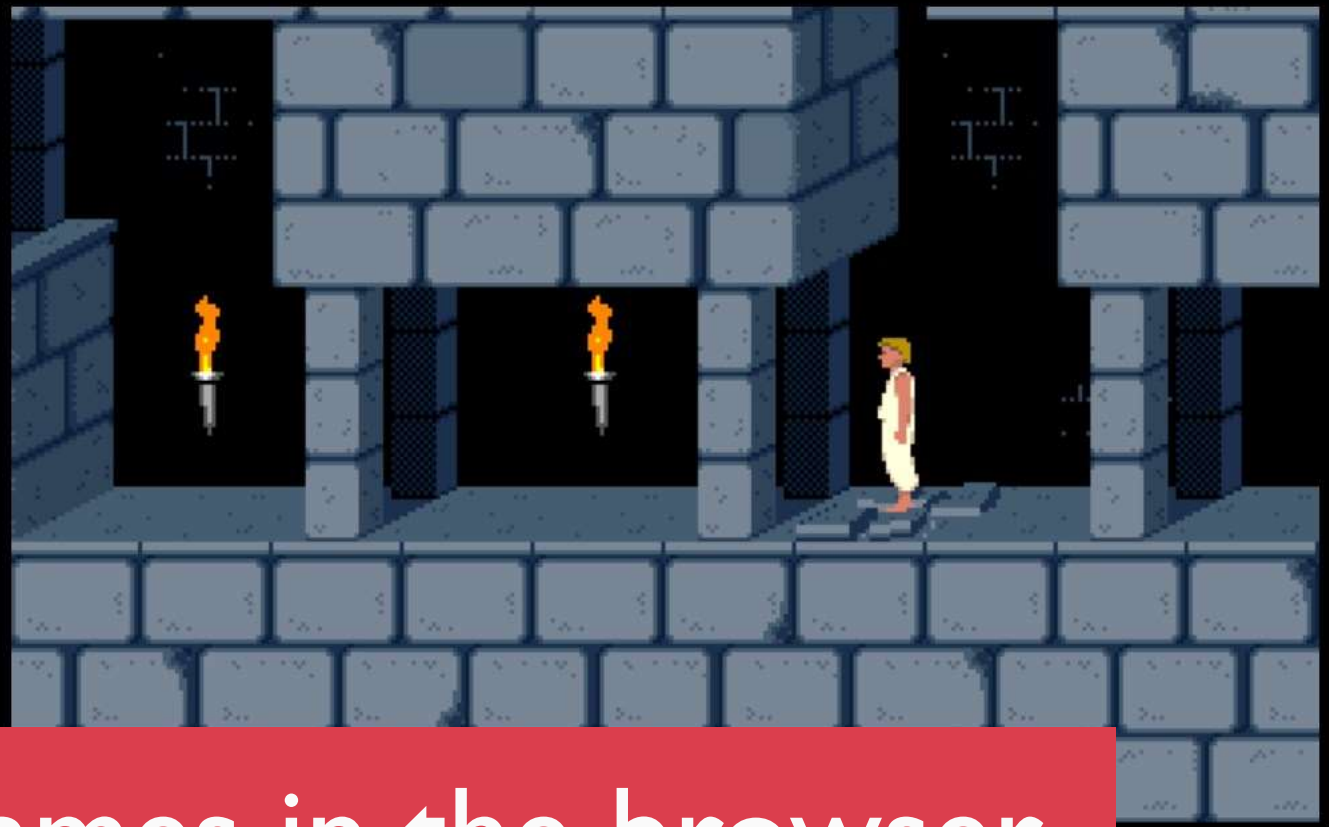
https://bellard.org/jslinux/vm.html?url=https://bellard.org/jslinux/win2k.cfg&mem=192&graphic=1&w=1024&h=768

DOSBox SVN, CPU spee

https://archive.org/details/msdos_Prince_of_Persia_1990#

SIGN IN

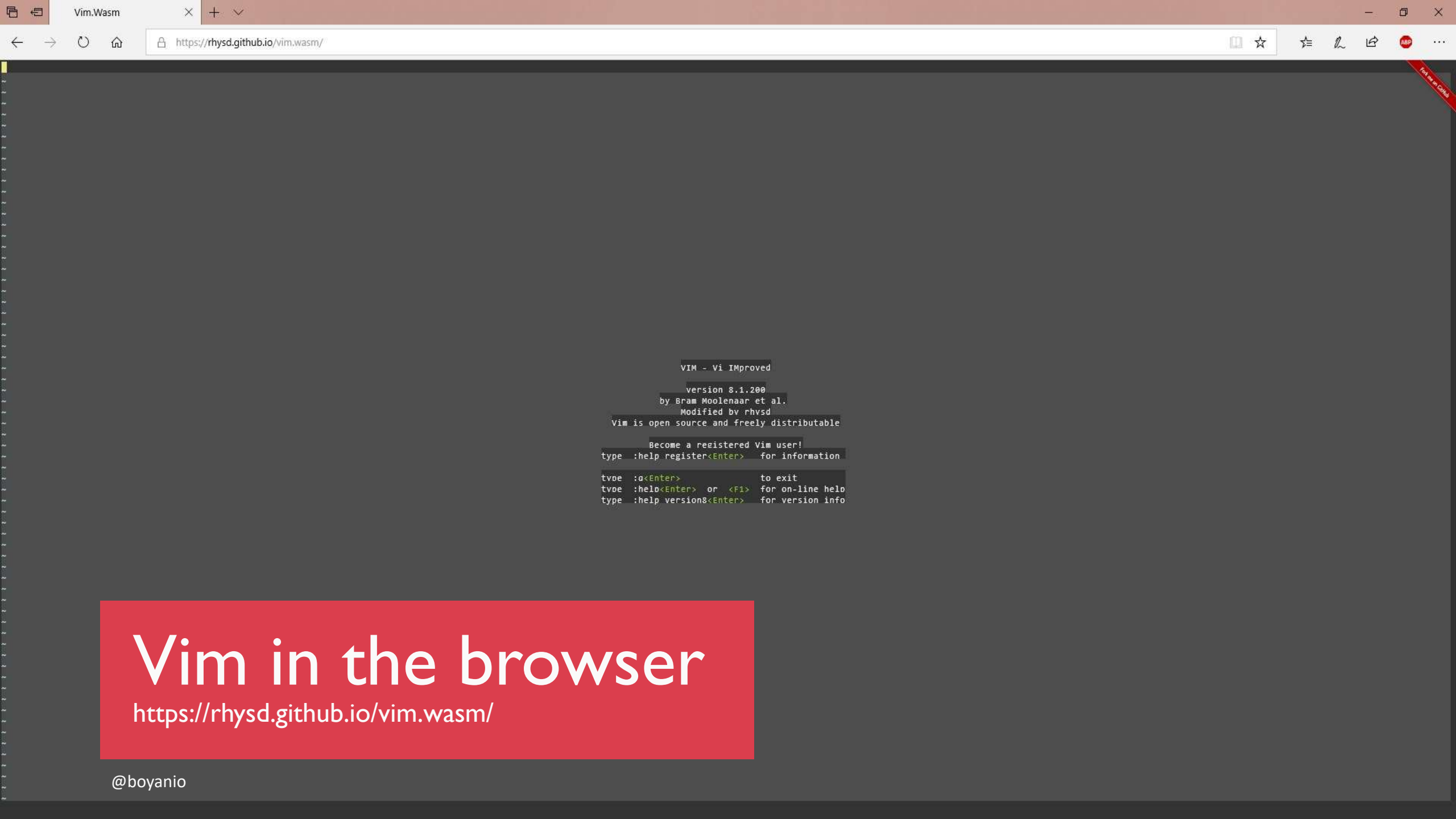
ABOUT CONTACT BLOG PROJECTS HELP DONATE JOBS VOLUNTEER PEOPLE



DOS games in the browser

https://archive.org/details/msdos_Prince_of_Persia_1990

by Jul 29, 2014


Vim.Wasm
https://rhysd.github.io/vim.wasm/
VIM - Vi IMproved
version 8.1.200
by Bram Moolenaar et al.
Modified by rhysd
Vim is open source and freely distributable
Become a registered Vim user!
type :help register<Enter> for information
type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version8<Enter> for version info
Vim in the browser
https://rhysd.github.io/vim.wasm/
@boyanio

How to get started with WebAssembly

Wasm Fiddle

https://wasdk.github.io/WasmFiddle/

@boyanio

Open Source LLVM to JavaScript compiler

emcc index.c –o index.js

The distributable, loadable, and executable unit of code in WebAssembly is called a **module**.

https://github.com/WebAssembly/design/blob/master/Modules.md

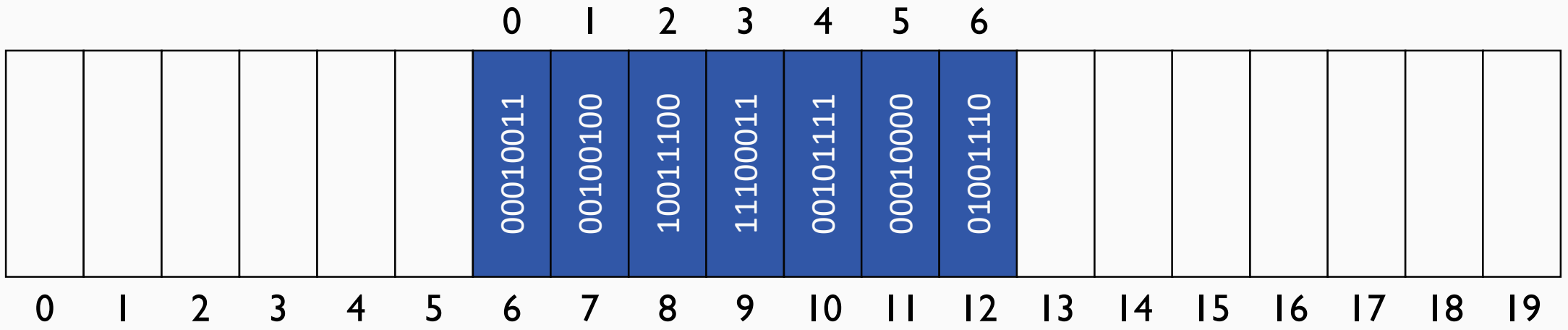
Module imports & exports

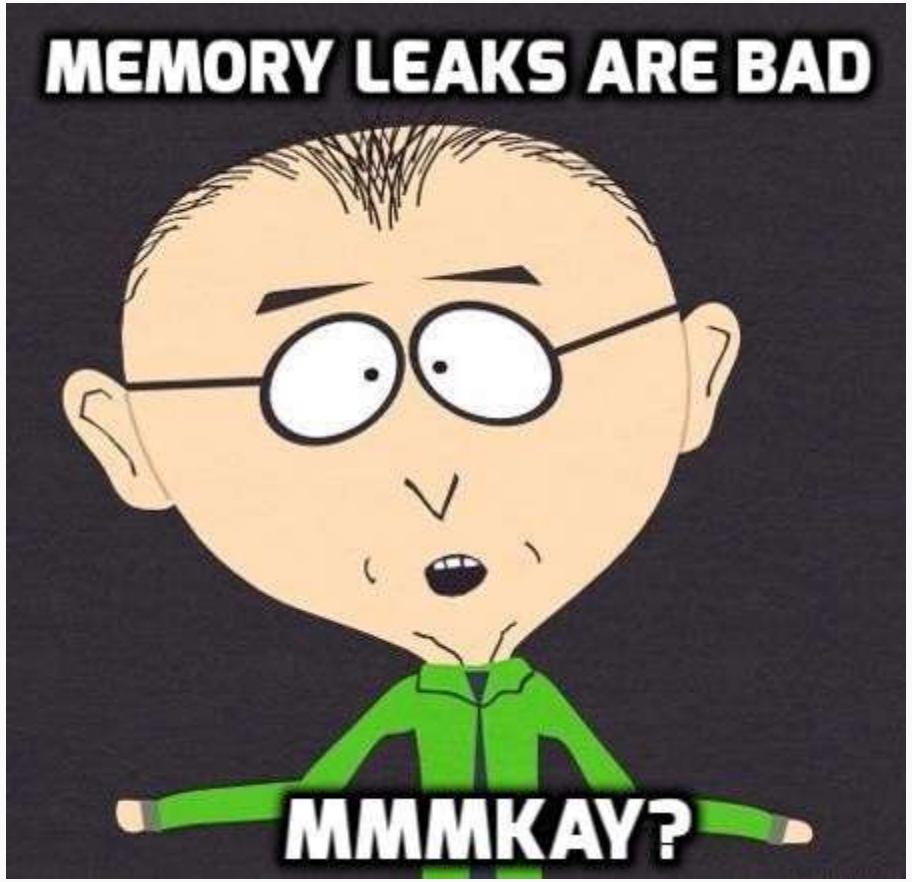
```
const imports = {
  “name”: {
    “first”: “Anna”,
    “last”: “Nanna”
  },
  “print”: function (what) {
    console.log(what);
  }
};
```

```
const exports = module.exports;
exports.printName();
exports.reverseName();
```

Linear memory

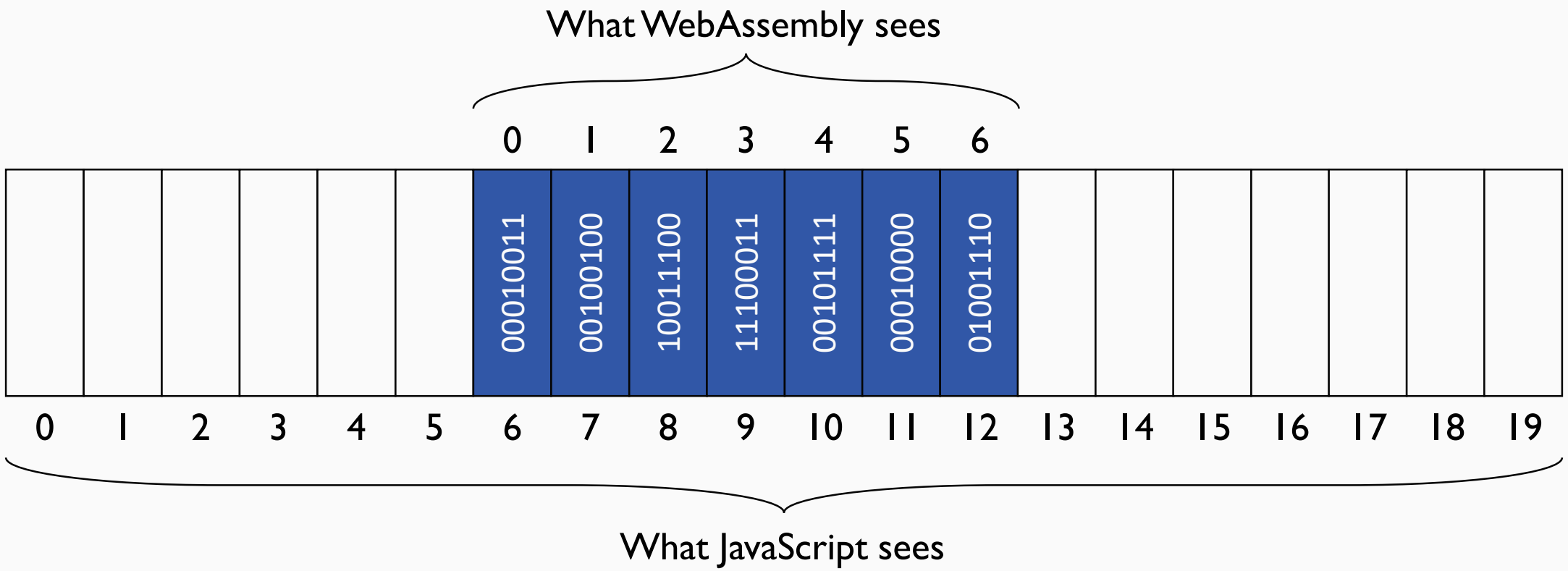
```
const imports = {
  “env”: {
    “memory”: new WebAssembly.Memory({ initial: 10, maximum: 100}),
    …
  }
};
```





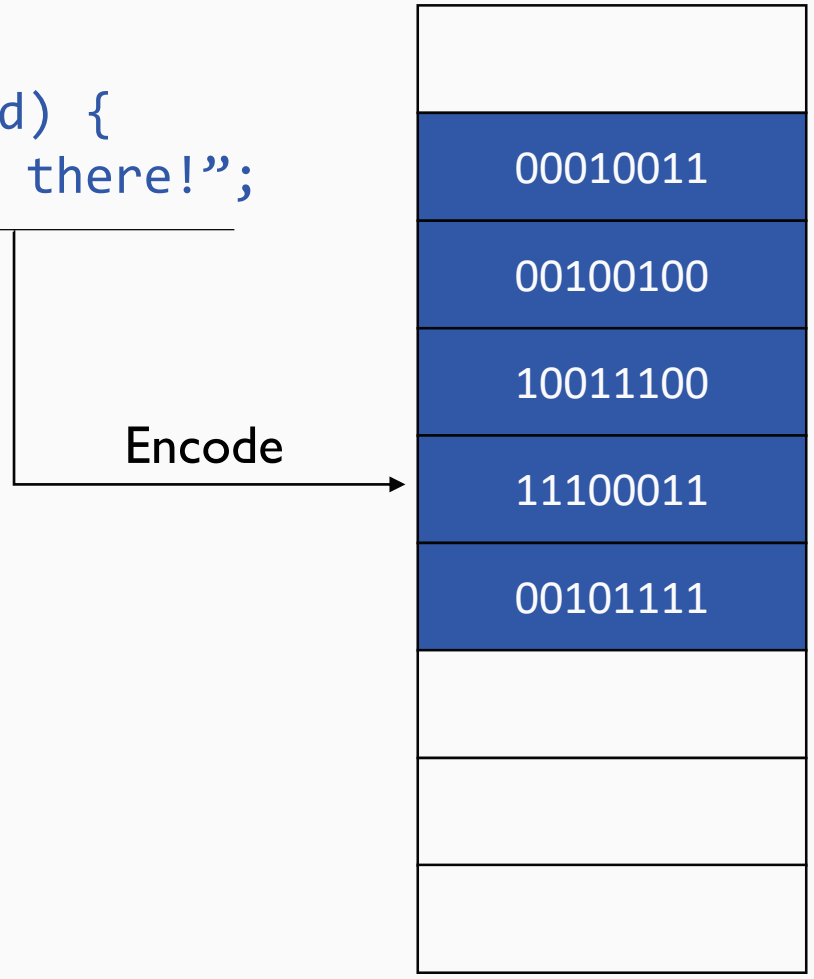
@boyanio

Linear memory



Working with strings

```
// app.c
char * hello(void) {
  return “Hello, there!”;
}
```




```
// index.js
let exp = wasmInst.exports;
let result = exp.hello();

console.log(result);
// 12

console.log(decode(result));
// Hello, there!
```

Decode

Loading WebAssembly

```
// Traditional approach
fetch('app.wasm')
  .then(result => result.arrayBuffer())
  .then(buffer => WebAssembly.instantiate(buffer, imports))
  .then(({ module, instance }) => {
    instance.exports.main();
  });

// Using the streaming API
WebAssembly.instantiateStreaming(fetch('app.wasm'), imports)
  .then(({ module, instance }) => {
    instance.exports.main();
  });
```

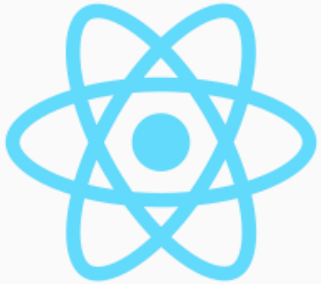
@boyanio

https://www.washingtonexaminer.com/cnn-nyt-reporters-aggressively-miss-the-point-with-nikki-haleys-reaction-to-the-grammys-stupid-fire-and-fury-reading


“WebAssembly fills in the gaps that would be awkward to fill with JavaScript.”
Eric Elliott
@boyanio
https://jeremybutterfield.files.wordpress.com/2014/12/conclusion.jpg

WebAssembly will break the JavaScript monopoly

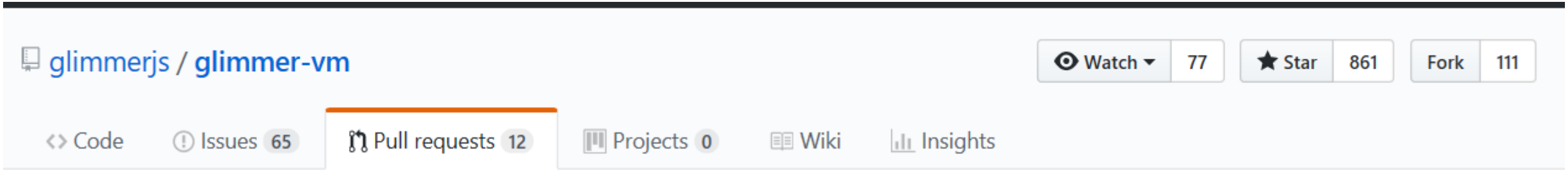
https://maas.museum/inside-the-collection/2010/03/17/see-you-round-like-a-monopoly-board/


The Web of JavaScript frameworks

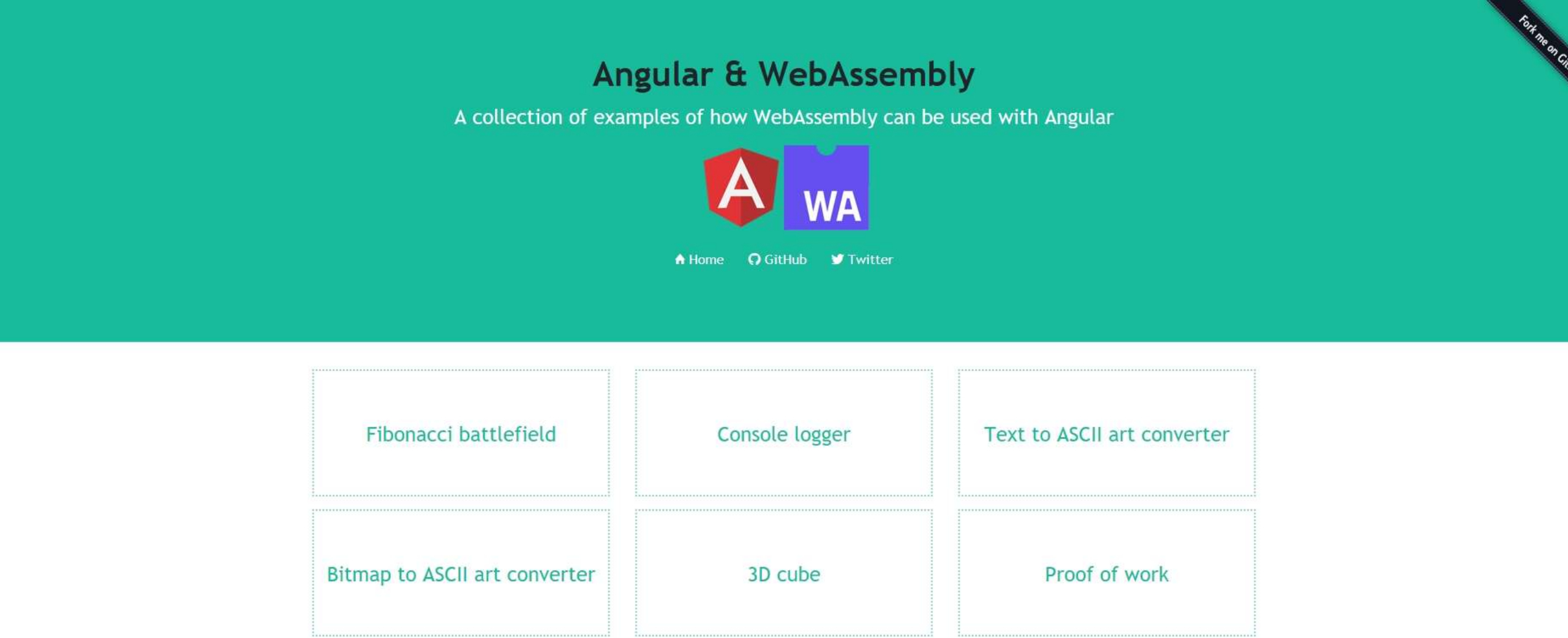

Webassembly integration. Split the core into two parts. #8193

https://github.com/vuejs/vue/issues/8193



Initial stab at porting `asm/stack.ts` to Rust #752

https://github.com/glimmerjs/glimmer-vm/pull/752



Angular & WebAssembly

https://boyan.io/angular-wasm/

@boyanio

The rise of non-JavaScript frameworks

B Docs API Community Blog

Blazor

Full-stack web development with C# and WebAssembly

Get Started

Build a Web UI with C#

Blazor is an experimental .NET web framework using C# and HTML that runs in the browser.

What is Blazor?

Blazor

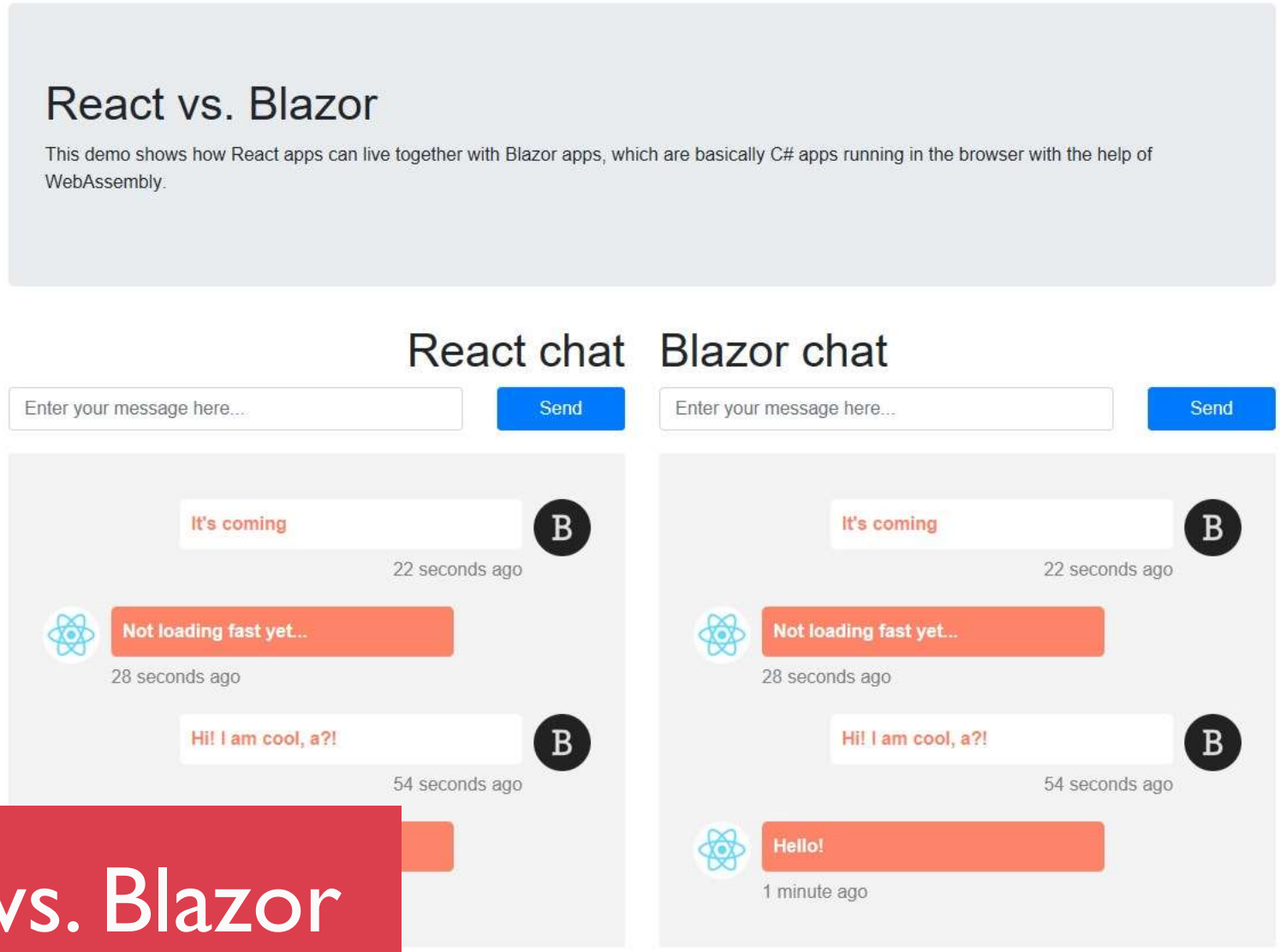
https://blazor.net

Full-stack .NET

Do full-stack .NET development using stable and consistent tools, languages, and APIs both in the browser and on the server.

Learn more about the .NET platform

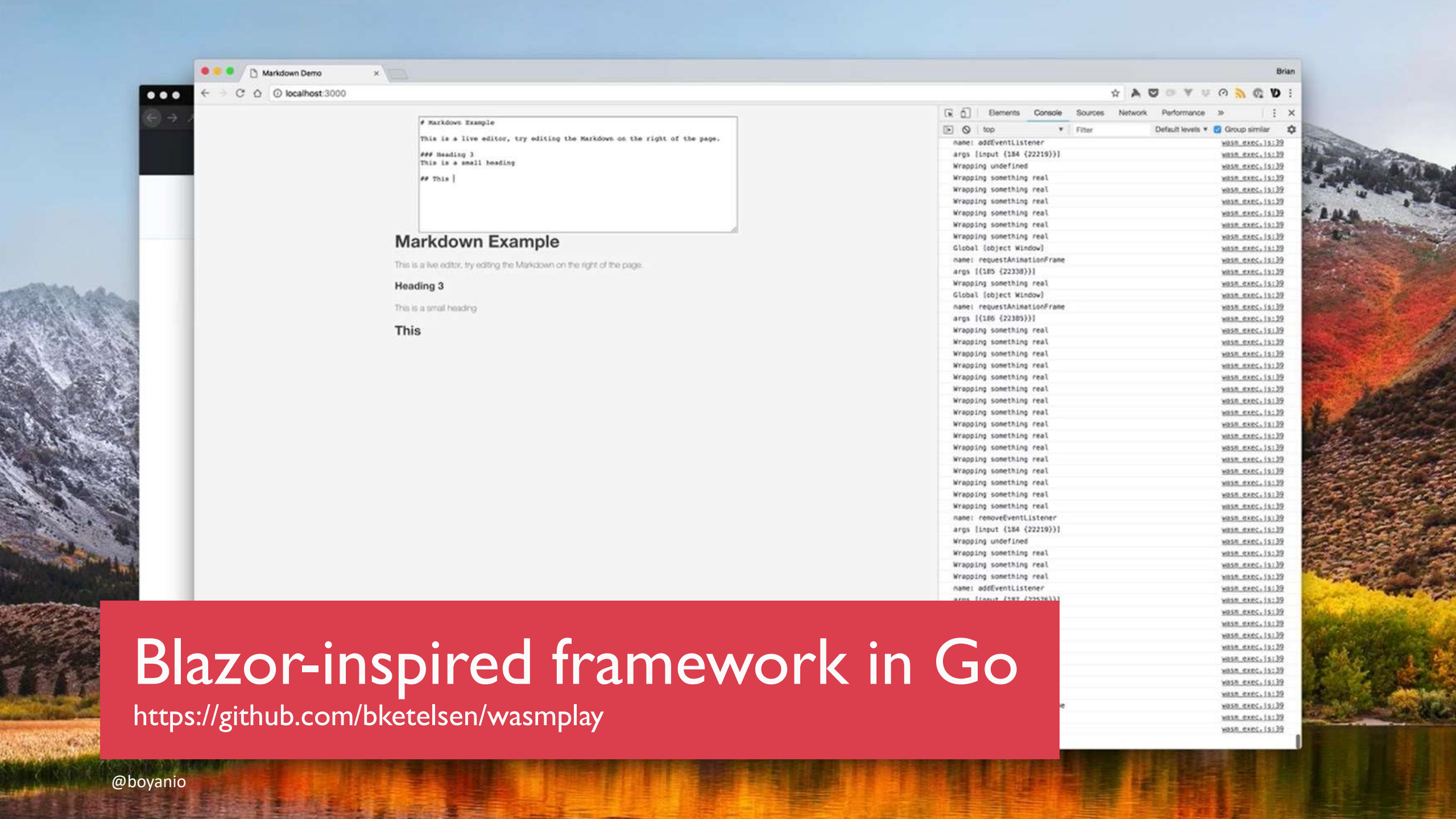
@boyanio



React vs. Blazor

https://boyan.io/react-blazor/

@boyanio


Markdown Demo
localhost:3000
Brian
Markdown Example
This is a live editor, try editing the Markdown on the right of the page.
Heading 3
This is a small heading
This
Markdown Example
This is a live editor, try editing the Markdown on the right of the page.
Heading 3
This is a small heading
This
Elements
Console
Sources
Network
Performance
top
Filter
Default levels
Group similar
Blazor-inspired framework in Go
https://github.com/bketelsen/wasmplay
@boyanio

blog about portfolio

Introducing Ruukh Framework

October 3, 2018 — 12:38 UTC

Rust has its goals set on to be a primary WASM language and it would be awesome to use it both in backend and frontend web. Ruukh is one of such efforts to realise that dream. Ruukh, a frontend web framework, is inspired by both VueJS and ReactJS.

So, what does it look like?

```
#![feature(proc_macro_gen, proc_macro_non_items, decl_macro)]

struct MyApp;
```

React-inspired framework in Rust

https://github.com/csharad/ruukh

README.md

Spasm

Spasm is a libary to develop single page applications in D that compile to webassembly.

It uses D's compile time feature to generate optimized rendering code specific for your application.

Not only are your applications fast, they are also small. The todo-mvc example project is only 5995 (wasm) + 2199 (html+js) bytes when gzipped.

How to start

- run `dub init` in a fresh folder
- add spasm to your dub dependencies

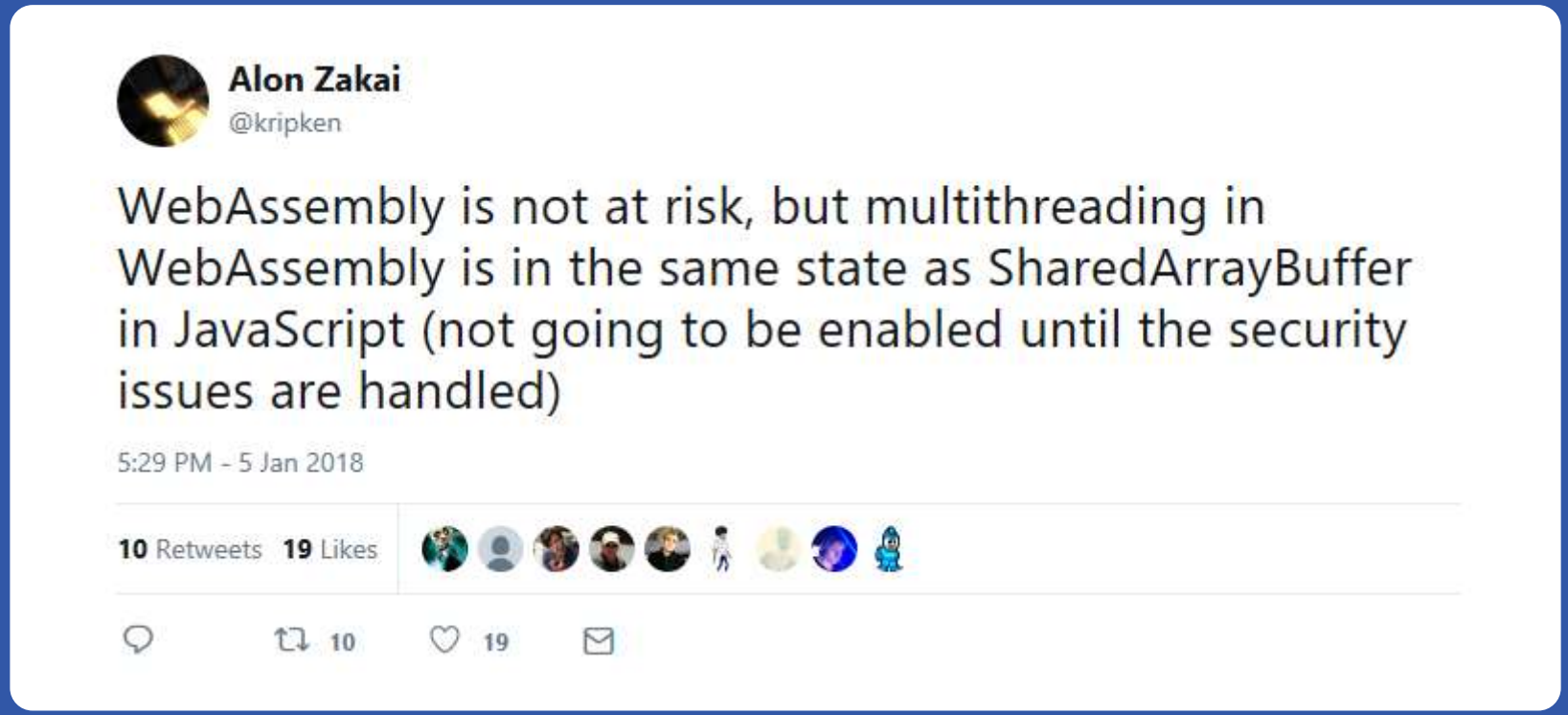
oilerplate

iad features of webpack to include what you

Create SPAs in D language

https://github.com/skoppe/spasm

How secure is WebAssembly?


Alon Zakai
@kripken
WebAssembly is not at risk, but multithreading in WebAssembly is in the same state as SharedArrayBuffer in JavaScript (not going to be enabled until the security issues are handled)
5:29 PM - 5 Jan 2018
10 Retweets 19 Likes
10
19

WebAssembly runs in a memory-safe sandboxed environment

Can I use WebAssembly ?

WebAssembly - OTHER

Usage % of all users
Global 71%

WebAssembly or "wasm" is a new portable, size- and load-time-efficient format suitable for compilation to the web.

Current aligned | Usage relative | Date relative | Show all

IE	Edge *	Firefox	Chrome	Safari	iOS Safari *	Opera Mini *	Chrome for Android	UC Browser for Android	Samsung Internet
			49						
			62		10.2				
		57	63		10.3				4
11	16	58	64	11	11.2	all	64	11.8	6.2
	17	59	65	11.1	11.3				
		60	66	TP					
		61	67						

Notes | Known issues (0) | Resources (7) | Feedback

MS Edge status: Preview Release

WebAssembly enables different languages to work together on the Web

https://boyan.io/wasm-wheel/

RUN

```php
<?php

phpinfo();

```

Fork me on GitHub

PHP Version 7.3.0beta2

System	Emscripten emscripten 1.0 #1 x86-JS
Build Date	Aug 20 2018 11:24:51
Configure Command	'./configure' '--disable-all' '--disable-cgi' '--disable-cli' '--disable-rpath' '--disable-phpdbg' '--without-pear' '--without-pcre-jit' '--with-layout=GNU' '--enable-embed=static' '--enable-bcmath' '--enable-json' '--enable-ctype' '--enable-tokenizer' 'CFLAGS='
Server API	PHP Embedded Library
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc
Loaded Configuration File	(none)
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731,NTS
PHP Extension Build	API20180731,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	disabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp
Registered Stream Socket Transports	tcp, udp, unix, udg
Registered Stream Filters	string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

gram makes use of the Zend Scripting Language Engine:
gine v3.3.0-dev, Copyright (c) 1998-2018 Zend Technologies

Configuration

bcmath

support	enabled

Directive	Local Value	Master Value
bcmath.scale	0	0

PHP in the browser

https://oraoto.github.io/pib/

@boyanio

The future of Web belongs to those, who compile

Boyan Mihaylov / @boyanio / boyan.io