





Matthias Endler

...works at trivago
...mostly Backend
...afraid of Frontend
...Python/Golang/Rust
...runs Hello Rust!
...subscribe to my
Newsletter!
...too many dots



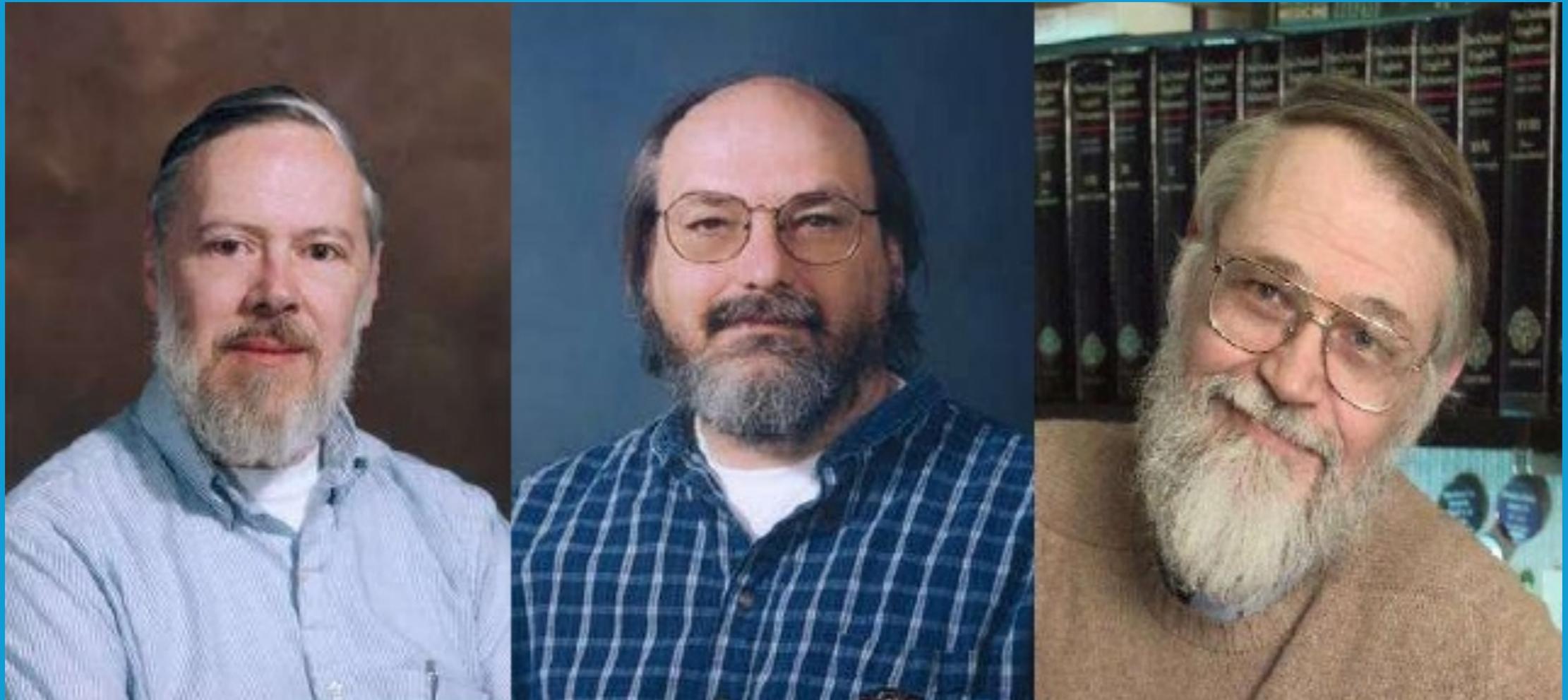
WTF is WASM?











Ritchie

Thompson

Kernighan





SECOND EDITION

THE



PROGRAMMING
LANGUAGE

BRIAN W. KERNIGHAN
DENNIS M. RITCHIE

PRENTICE HALL SOFTWARE SERIES

"UNIX is very simple.
It just takes a genius
to understand its simplicity."

-- Dennis Ritchie

"C is what made
Unix portable"

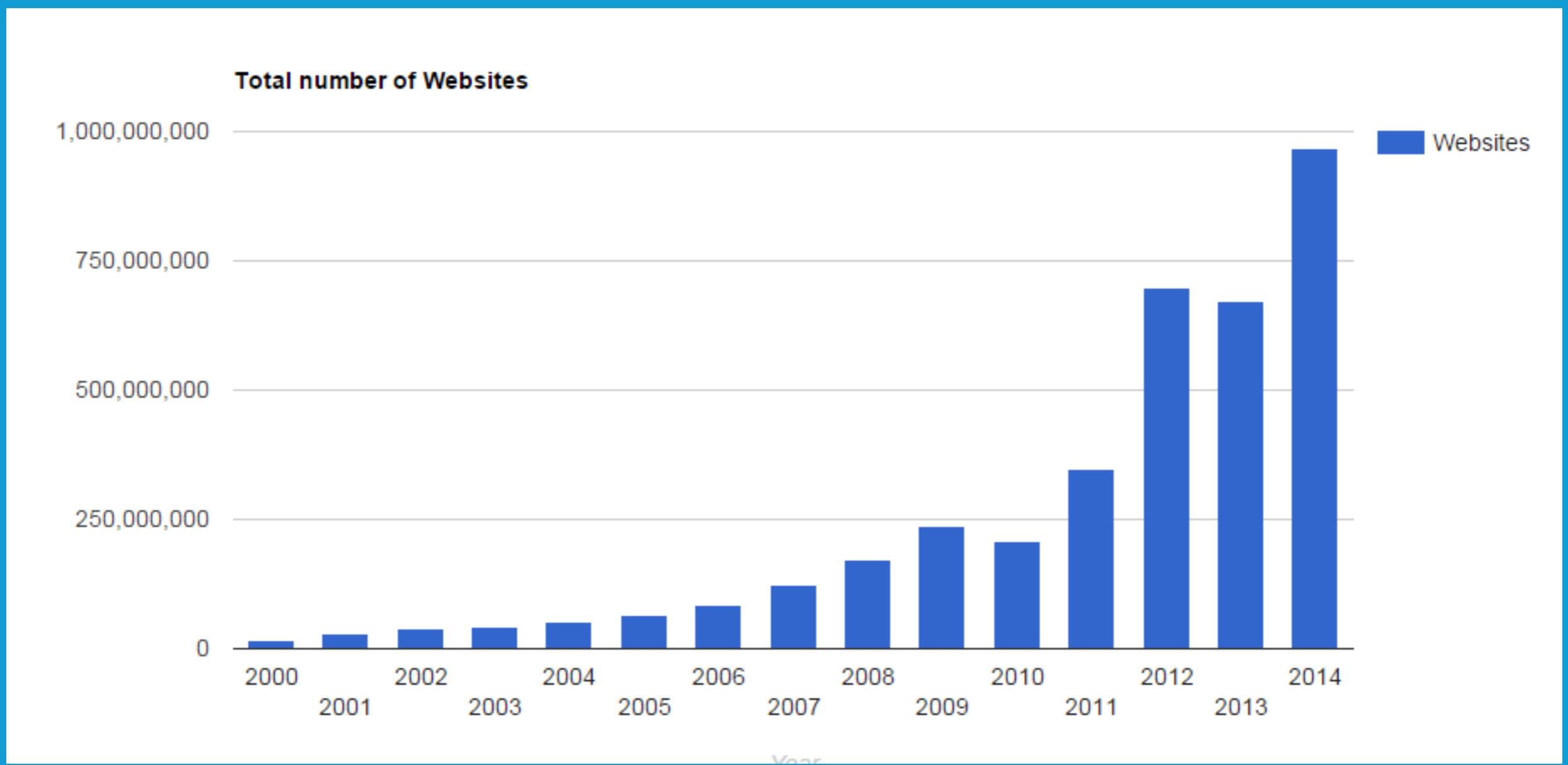
-- Dennis Ritchie

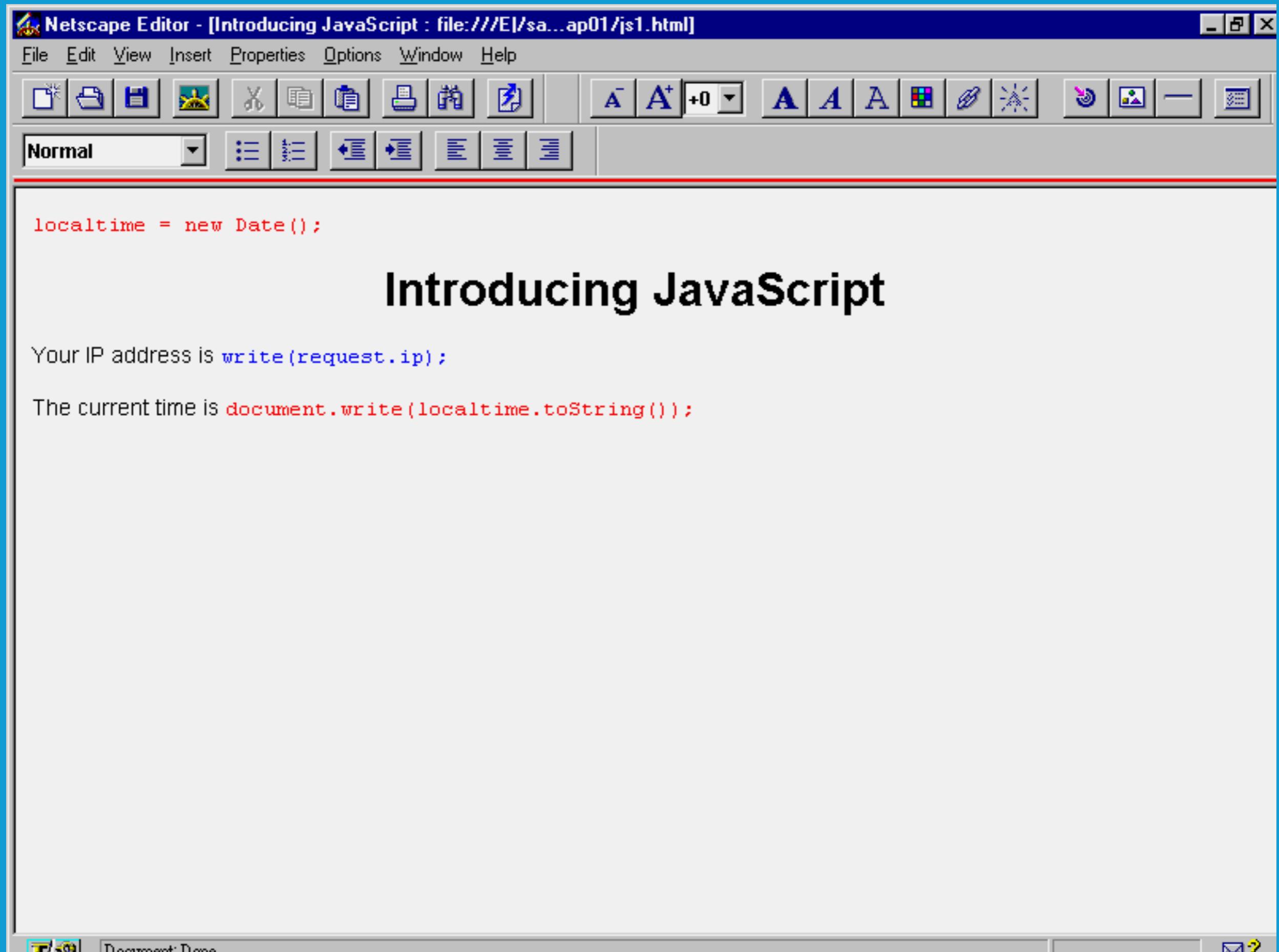




```
<HTML>
  <HEAD>
    <TITLE>WWW</TITLE>
  </HEAD>
  <BODY>
    <BLINK>Hello, World!</BLINK>
  </BODY>
</HTML>
```









CoffeeScript

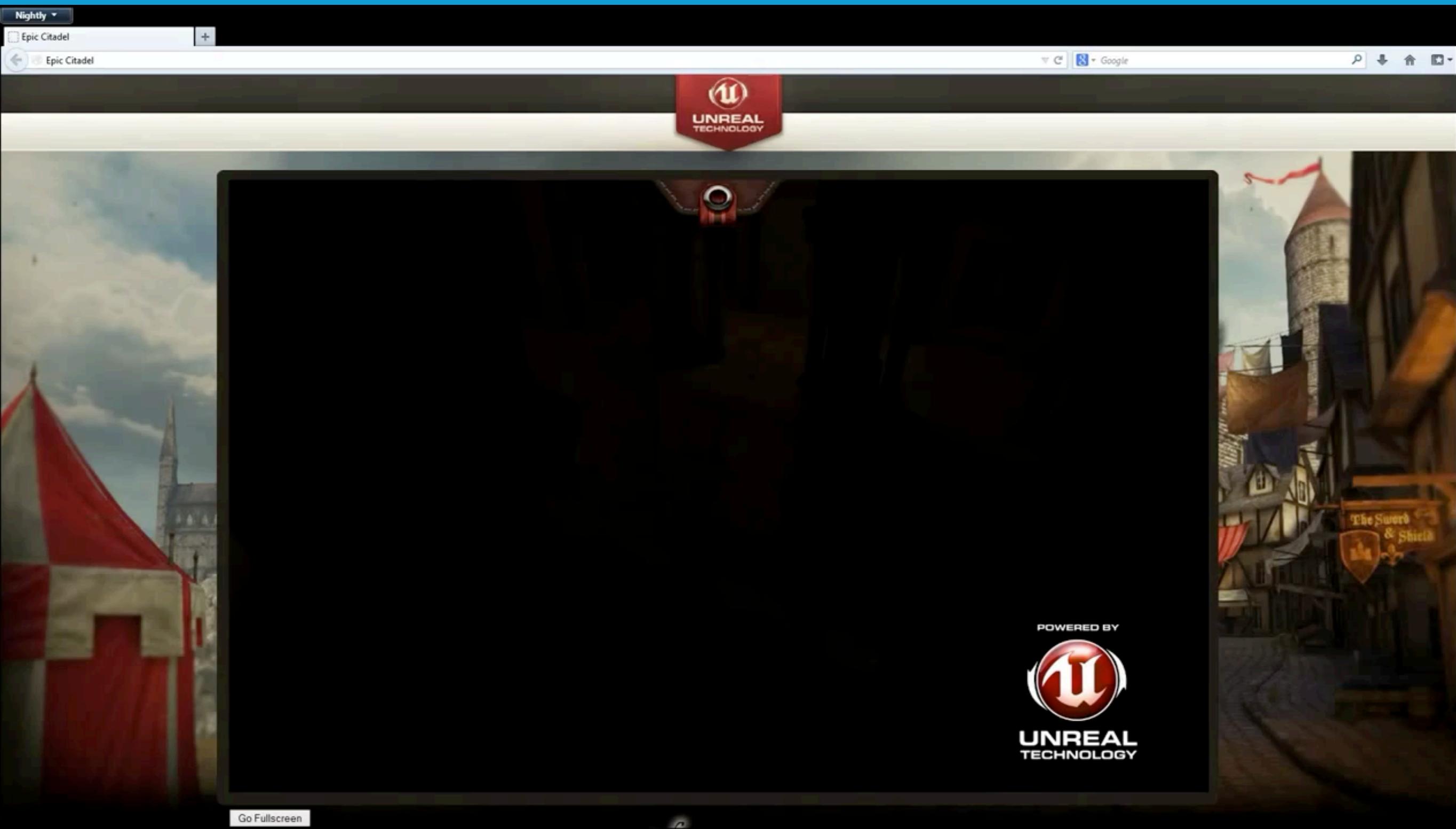
```
# Assignment:  
number: 42  
opposite_day: true  
  
# Conditions:  
number: -42 if opposite_day  
  
# Functions:  
square: x => x * x.  
  
# Arrays:  
list: [1, 2, 3, 4, 5]  
  
# Objects:  
math: {  
    root: Math.sqrt  
    square: square  
    cube: x => x * square(x).  
}  
  
# Array comprehensions:  
cubed_list: math.cube(num) for num in list.
```

```
var __a, __b, __c, __d, cubed_list, list, math, num,  
number, opposite_day, square;  
// Assignment:  
number = 42;  
opposite_day = true;  
// Conditions:  
if (opposite_day) {  
    number = -42;  
}  
// Functions:  
square = function(x) {  
    return x * x;  
};  
// Arrays:  
list = [1, 2, 3, 4, 5];  
// Objects:  
math = {  
    root: Math.sqrt,  
    square: square,  
    cube: function(x) {  
        return x * square(x);  
    }  
};  
// Array comprehensions:  
__a = list;  
__d = [];  
for (__b=0, __c=__a.length; __b<__c; __b++) {  
    num = __a[__b];  
    __d[__b] = math.cube(num);  
}  
cubed_list = __d;
```



emscripten

```
function vb(d) {
    d = d | 0;
    var e = 0, f = 0, h = 0, j = 0, k = 0, l = 0, m = 0, n = 0,
        o = 0, p = 0, q = 0, r = 0, s = 0;
    e = i;
    i = i + 12 | 0;
    f = e | 0;
    [l + 4 >> 2] = 0;
    l = (c[1384465] | 0) + 3 | 0;
    do {
        if (l >>> 0 < 26) {
            if (((c[1356579] | 0) > 0) {
                m = d + 4 | 0;
                n = 0;
                while (1) {
                    o = c[(c[1356577] | 0) + (n << 2) >> 2] | 0;
                    do {
                        if (a[o + 22 | 0] << 24 >> 24 == 24) {
                            if (!(vp(d, o | 0) | 0))) {
                                break
                            }
                        }
                    }
                    p = (c[m >> 2] | 0) + (((c[h >> 2] | 0) - 1 | 0) * 40 & -1) + 12 | 0;
                    q = o + 28 | 0;
                    c[p >> 2] = c[q >> 2] | 0;
                    c[p + 4 >> 2] = c[q + 4 >> 2] | 0;
                }
            }
        }
    }
}
```



[\[Source\]](#)

Disadvantages of asm.js

- Performance differences between browsers
- Still requires parsing, compiling, optimising
- No specification

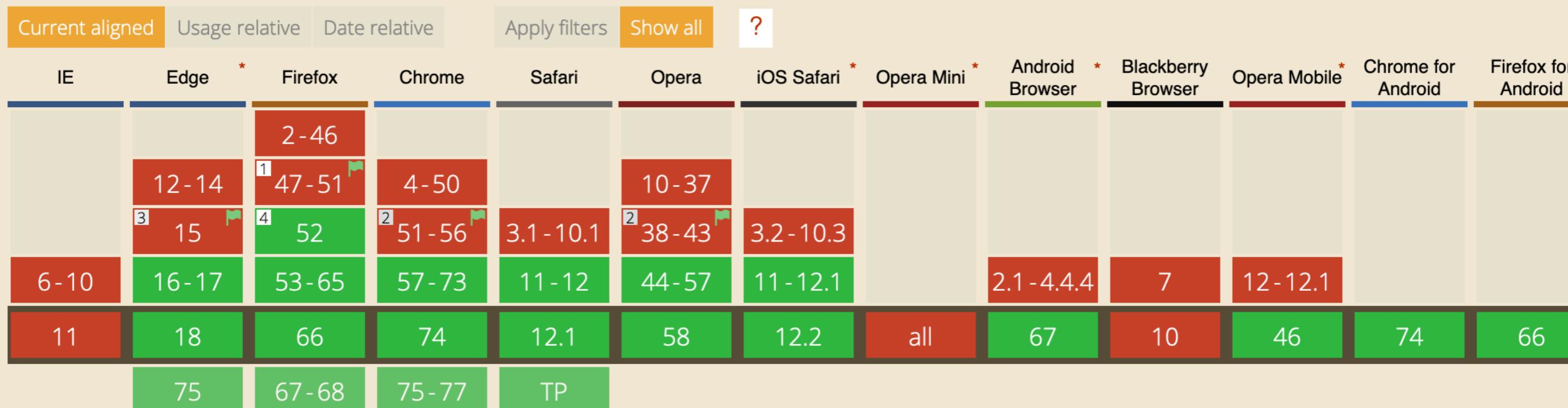
Enter WASM

(Webassembly)

WebAssembly

Usage % of all users ▾ ?

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

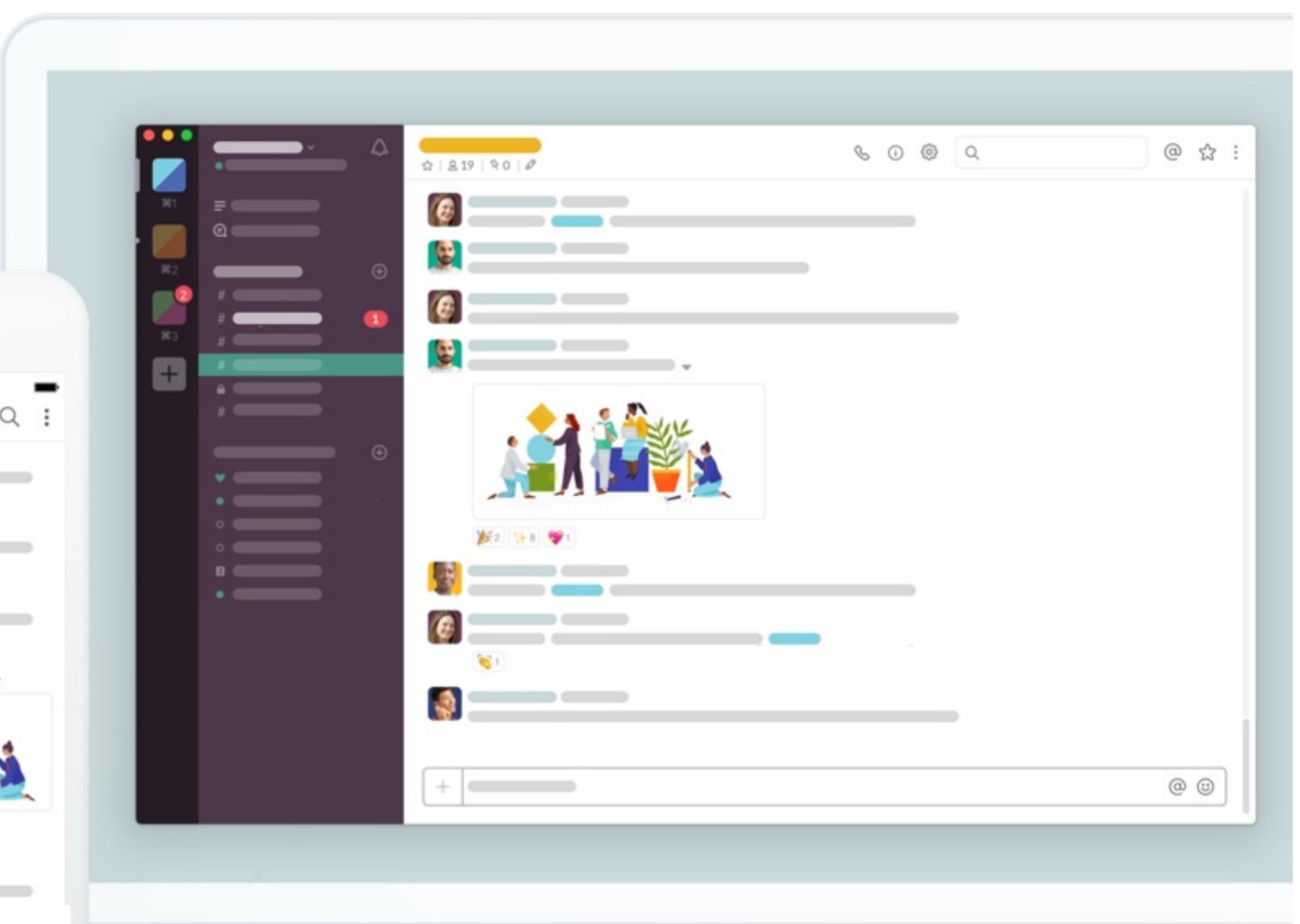
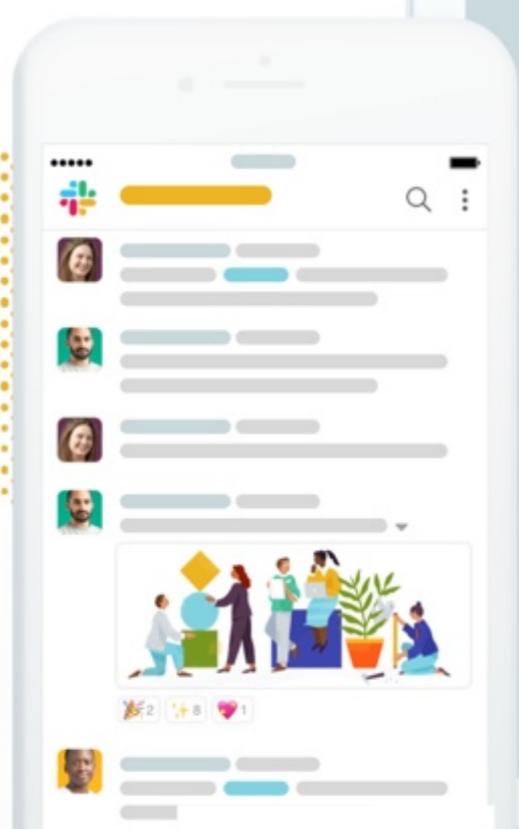






xenitaph





Wasm is . . .

- Fast
- Safe
- Well defined
- Language-independent
- Streamable
- Parallelizable

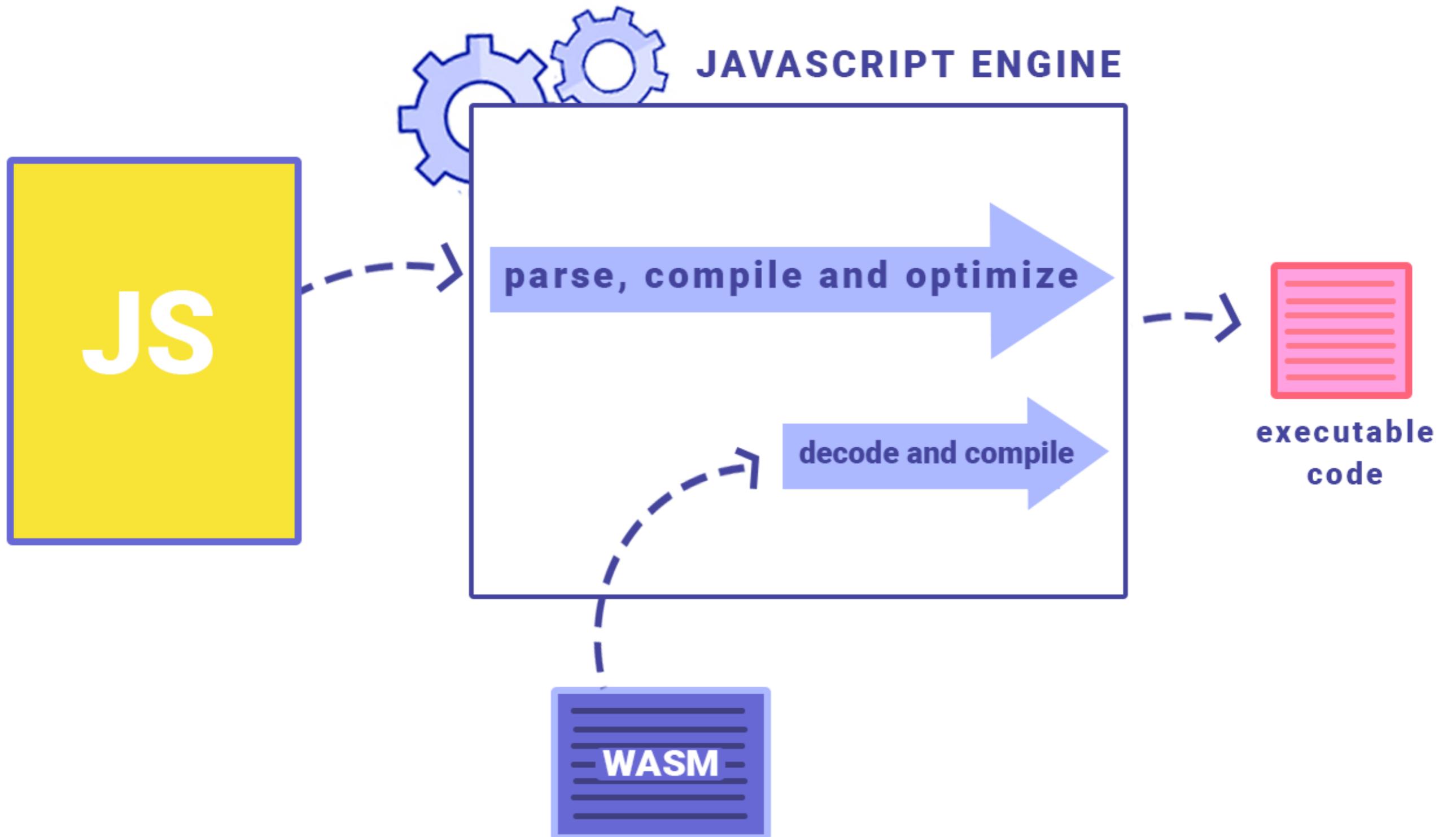
Decoding



Validation

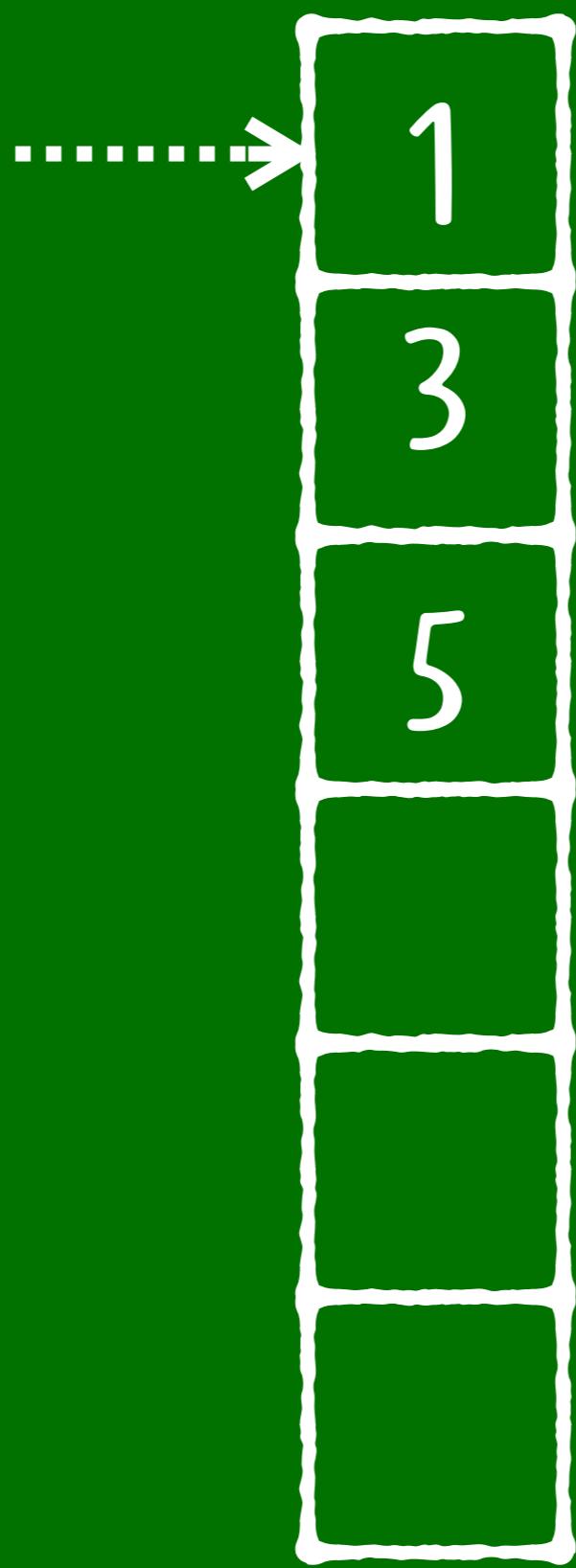


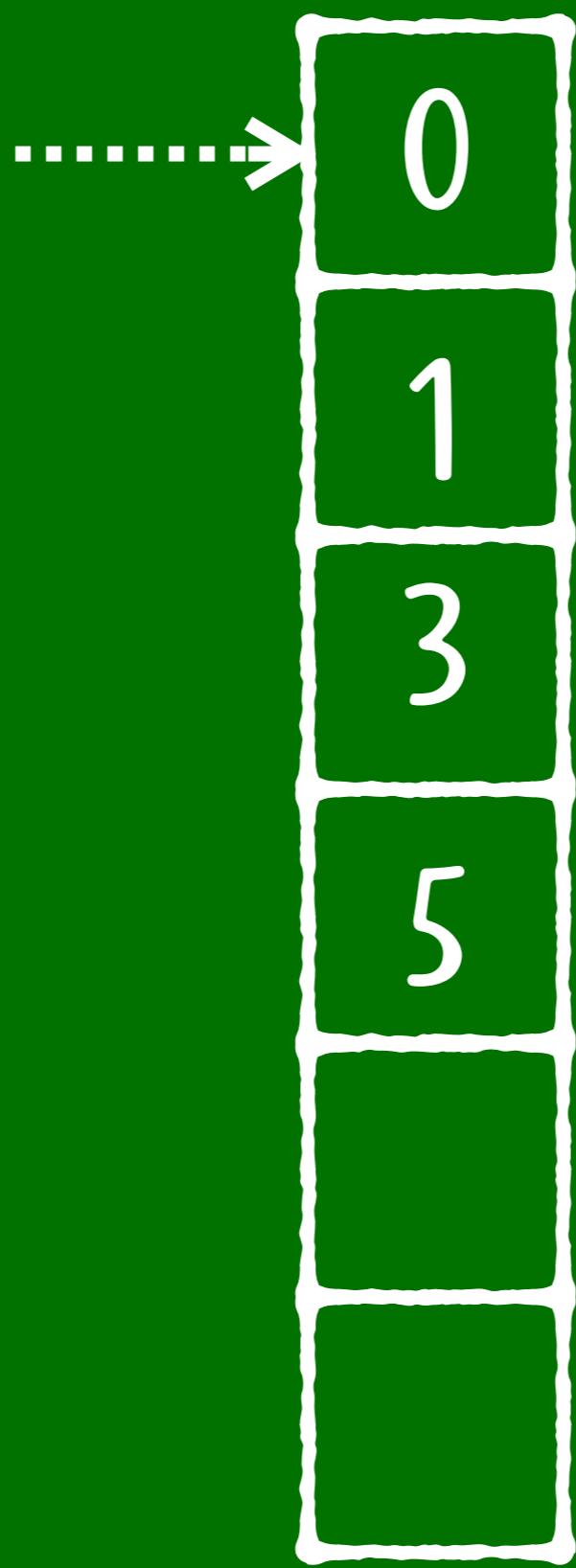
Execution

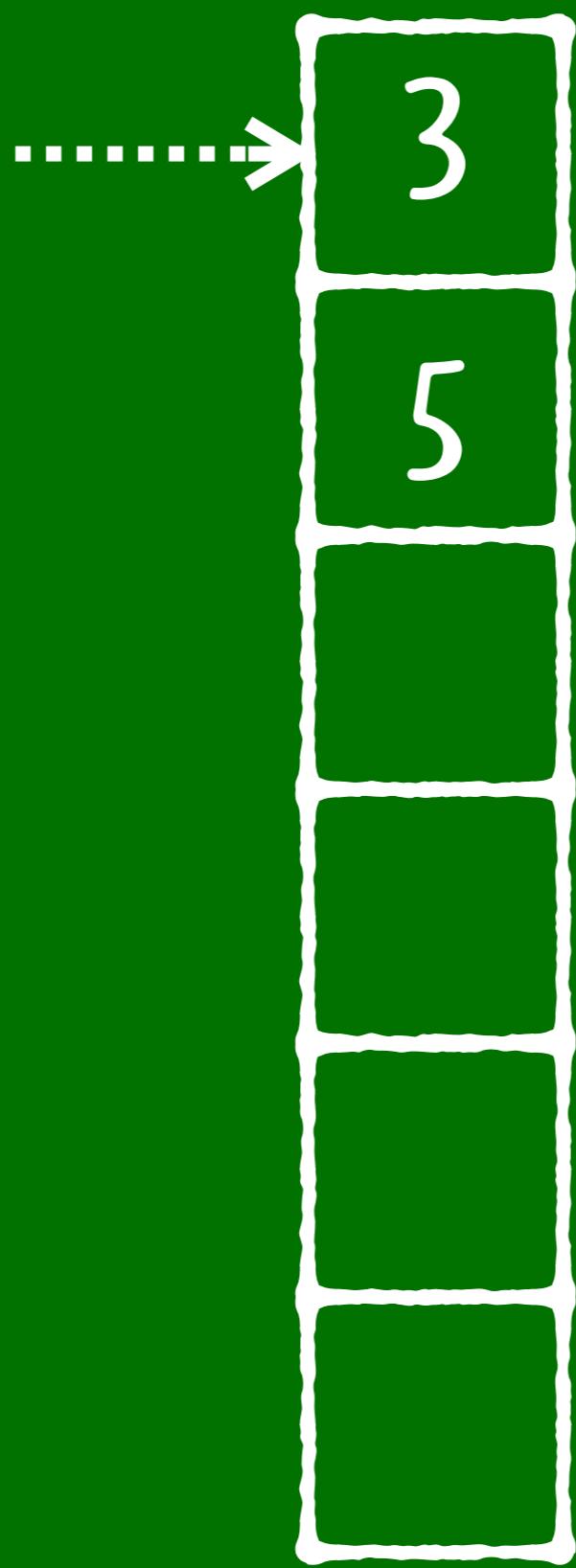


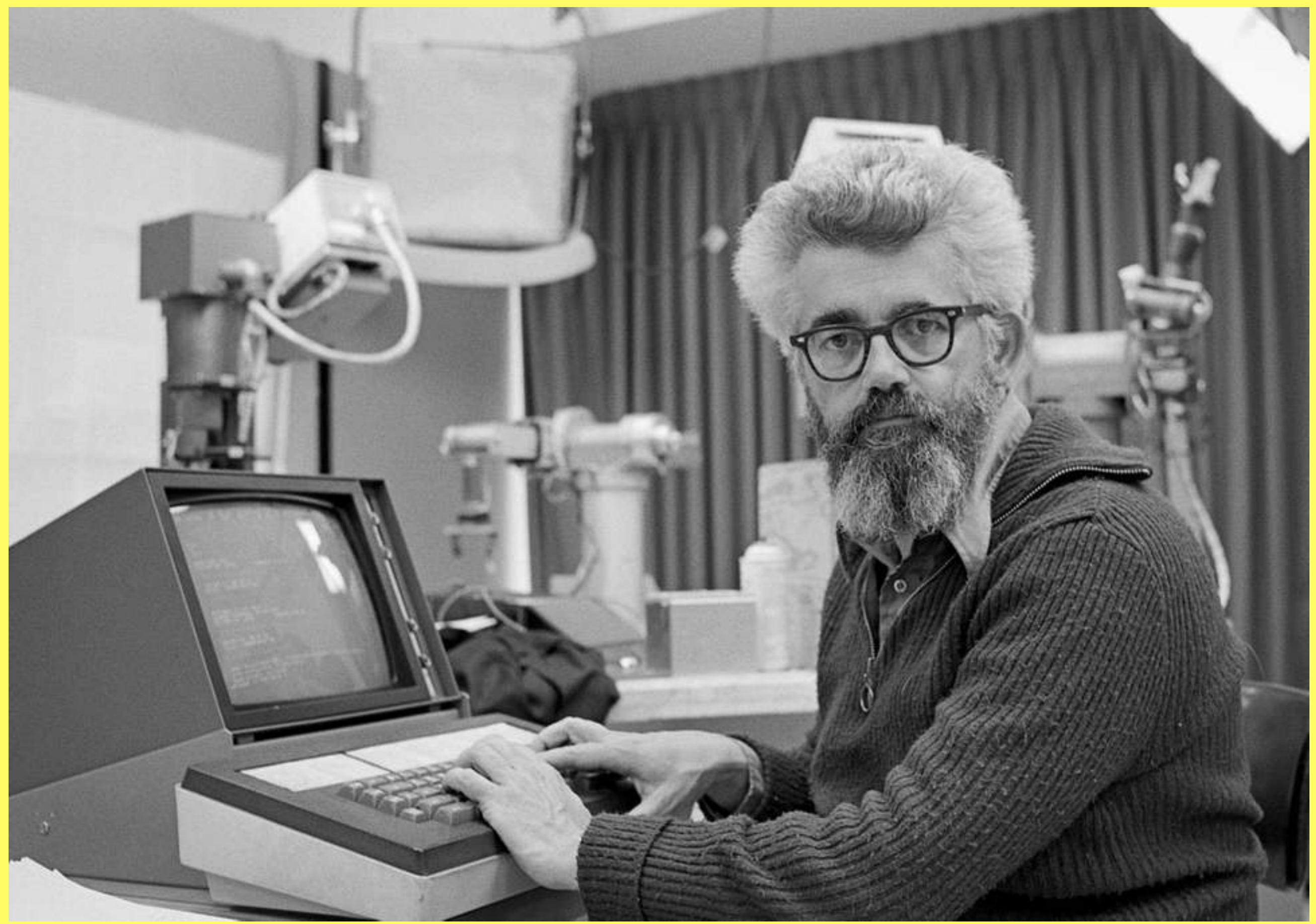
Datatypes!

i32, i64, f32, f64
that's it.









LISP Theory & Practice

```
(LAMBDA
  (X (N . 0))
  (COND
    ((OR
      (ATOM X)
      (CHARCOUNT
        X
        (DIFFERENCE LINEWIDTH N)))
     (PRINO X)))
    (T
      (PRINO LPAR)
      (SPRINT (CAR X) N)
      (SETQ N (PLUS N 3)))
    (LOOP
      (SETQ X (CDR X)))
    (COND
      ((AND X (ATOM X))
       (PRINO PERIOD X)))
    (UNTIL
      (ATOM X)
      (PRINO RPAR)))
    (XTAB N)
    (SPRINT (CAR X) N))))))
```

(module)

0000000: 0061 736d ; WASM_BINARY_MAGIC

0000004: 0100 0000 ; WASM_BINARY_VERSION

(**func** <signature> <locals> <body>)

```
(func (param i32) (param i32) (result f64)
  <body>
)
```

```
(func (param i32) (param f32) (local f64)
  get_local 0
  get_local 1
  get_local 2
)
```

```
(func (param $foo i32) (param $bar f32) (local $baz f64)
  get_local $foo
  get_local $bar
  get_local $baz
)
```

```
(module
(
    func (param $a i32) (param $b i32) (result i32)
        get_local $a
        get_local $b
        i32.add
)
)
```

```
(module
  (func $add (param $a i32) (param $b i32) (result i32)
    get_local $a
    get_local $b
    i32.add)
  (export "add" (func $add)))
)
```

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

```
(module
  (type $type0 (func (result i32)))
  (table 0 anyfunc)
  (memory 1)
  (export "memory" memory)
  (export "hello" $func0)
  (func $func0 (result i32)
    i32.const 16
  )
  (data (i32.const 16)
    "Hello World\00"
  )
)
```

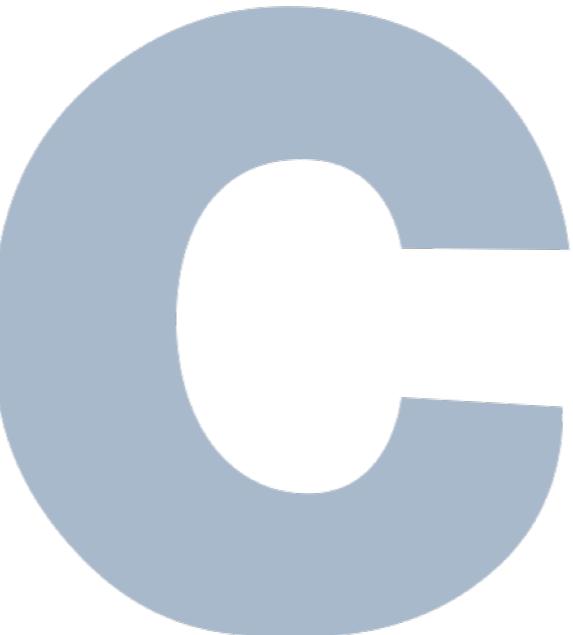
TEXTUAL FORMAT

```
(module
  (func $addTwo (param i32 i32)
    (result i32)
    (i32.add
      (get_local 0)
      (get_local 1)))
  (export "addTwo" $addTwo))
```

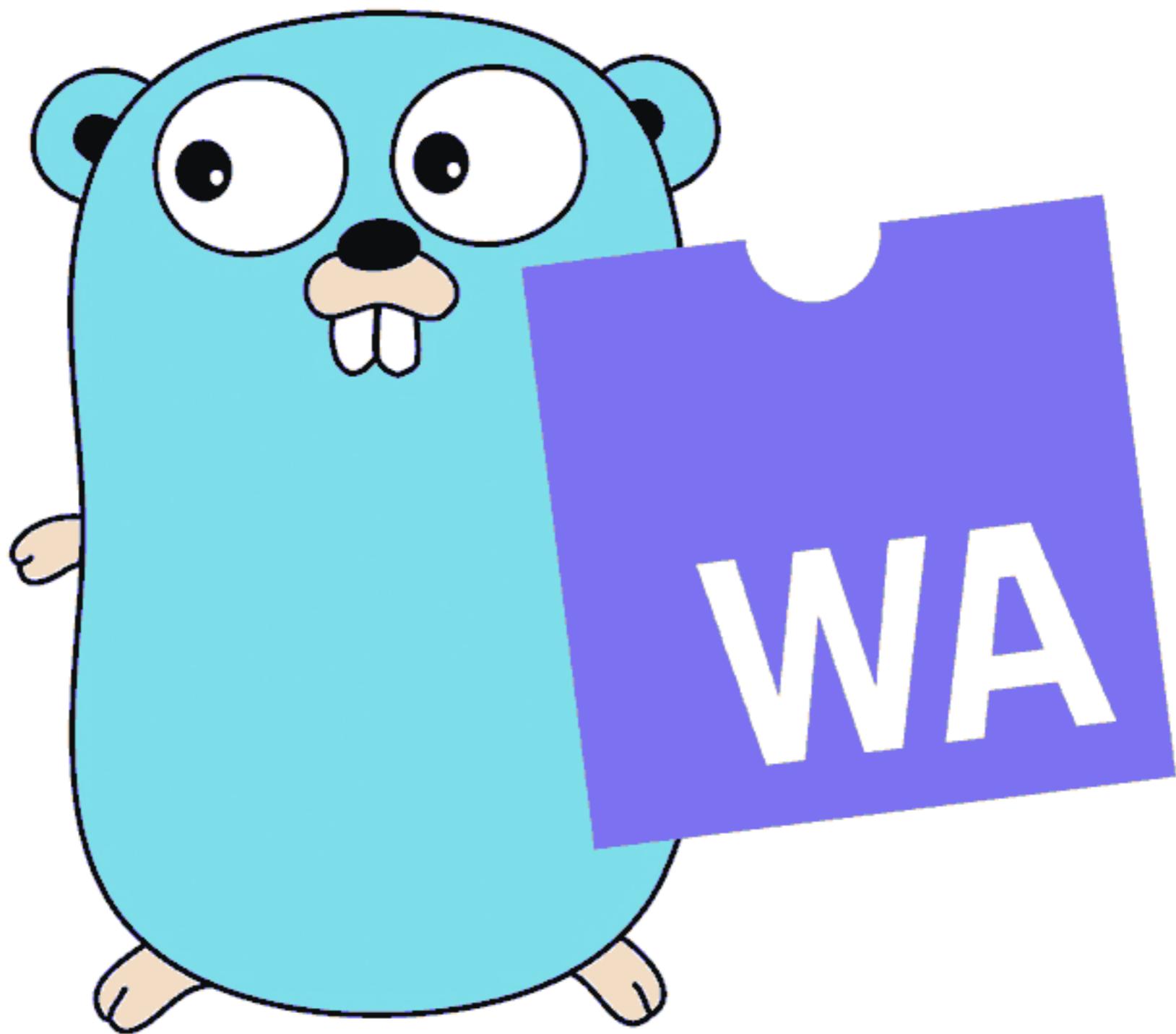
BINARY FORMAT

```
48 83 EC 08
8B CF
8B C1
03 C6
66 90
48 83 C4 08
C3
```

THE

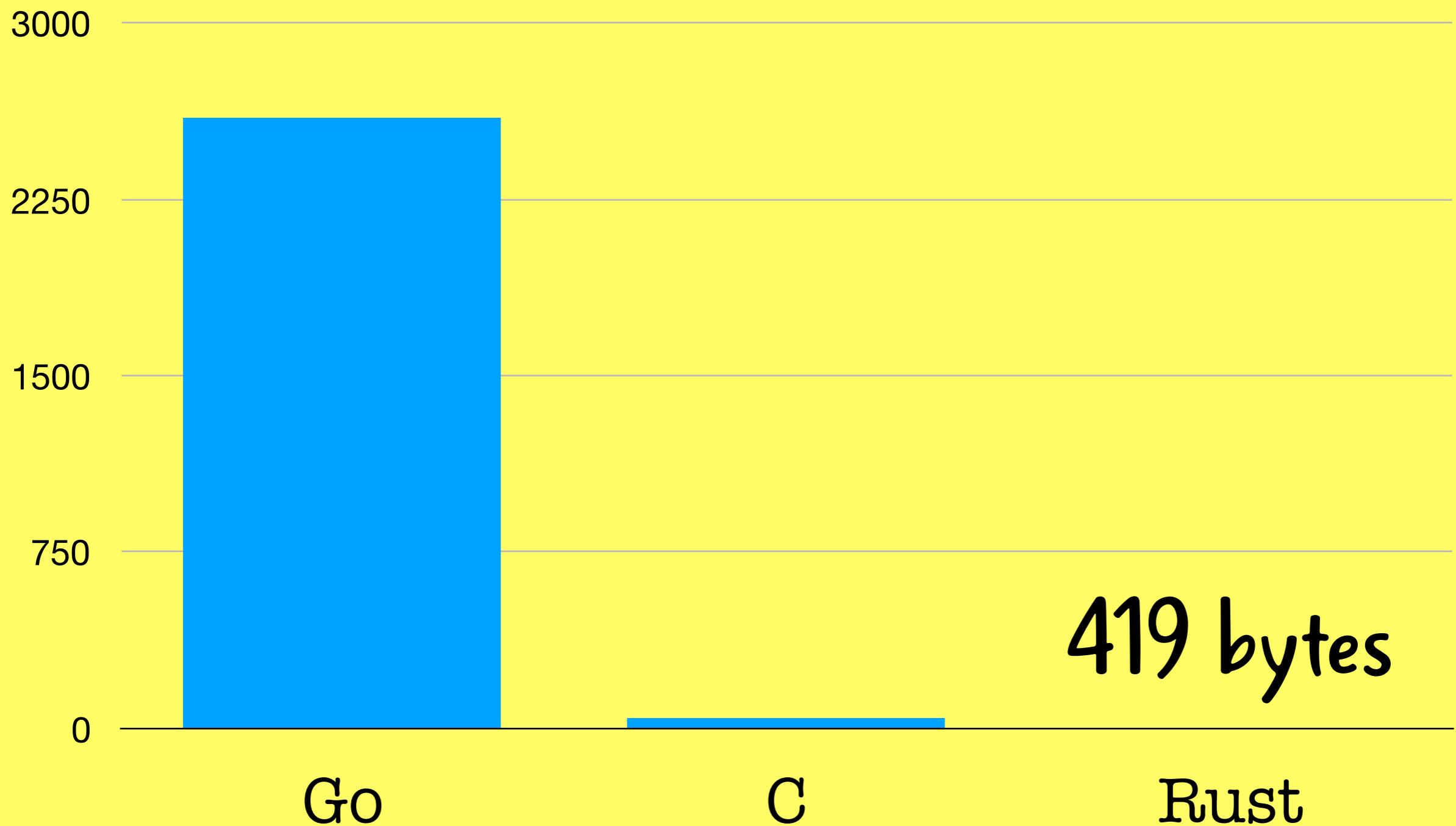


PROGRAMMING
LANGUAGE





How big is "Hello World" (in KB)?

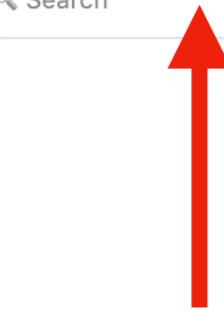


Examples!



MATTHIAS ENDLER

Backend Engineer at trivago.
Likes just-in-time compilers
and hot chocolate. [About me.](#)

 Search

Wasm!

21st of March, 2019

Maybe You Don't Need Kubernetes

Kubernetes is the 800-pound gorilla of container orchestration. It powers some of the biggest deployments worldwide, but it comes with a price tag... [More »](#)

2nd of December, 2018

What Is Rust Doing Behind the Curtains?

Rust allows for a lot of syntactic sugar, that makes it a pleasure to write. It is sometimes hard, however, to look behind the curtain and see what the compiler is really doing with our code. [More »](#)

5th of November, 2018

The Unreasonable Effectiveness of Excel Macros

I never was a big fan of internships, partially because all the exciting companies were far away from my little village in Bavaria and partially because I was too shy to apply. Only once I applied for an internship in Ireland as part of a school program. Our teacher assigned the jobs and so my friend got one at Apple and I ended up at a medium-sized IT distributor — let's call them PcGo. [More »](#)

2nd of September, 2018

Switching from a German to a US Keyboard Layout



Drag & drop or **select an image**

Or try one of these:



Large photo
(2.8mb)



Artwork
(2.9mb)



Device screen
(1.6mb)



SVG icon
(13k)



Chromium Blog

News and developments from the open source browser project

WebAssembly brings Google Earth to more browsers

Thursday, June 20, 2019

About 14 years ago, Google Earth gave users a rush of excitement by allowing them to zoom right in on their childhood homes. But that could happen only once they downloaded and installed the application. Earth was released as a native application because rendering the whole world in real time required advanced technologies that weren't available in the browser.

As the Web progressed, we wanted Earth to be available on the platform so it could reach as many people as possible and let them experience the entire world at their fingertips. Web apps offer a better user experience because they're [linkable](#), meaning you can share access to the whole experience with a single click; they're secure, since users aren't at risk of viruses that can come with software downloads; and they're composable, meaning we can embed them in other parts of the web.

In [WebAssembly](#) (Wasm), the [W3C](#) web standard for bringing native code to the web, the Earth team found a solution to allow Google Earth to move across multiple browsers—something we've been working on for a while now. Earth first came to the Web about two years ago using Native Client (NaCl), a Chrome-only solution—at the time. It was the only way to run native code in the browser and offer the performance users expect in modern web applications. But cross-browser compatibility is not as easy as we would like, since not every browser supports new technologies the

Labels ▾

Archive ▾

Feed

Follow @ChromiumDev

Give us feedback in our [Product Forums](#).

"Serverless"





Search or jump to...

/

Pull requests Issues Marketplace Explore

Bell +

fastly / terrarium-templates

Watch ▾ 15

Star 7

Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Branch: master ▾

terrarium-templates / 003_empty_ts / assembly / index.ts

Find file Copy path



tyler Explicitly order the example via the dir names

2f56712 on Dec 5, 2018

1 contributor

13 lines (10 sloc) | 289 Bytes

Raw

Blame

History



```
1 import { run_user, Request, Response } from "./http_guest";
2
3 function user_entrypoint(req: Request): Response {
4     let resp = new Response();
5     resp.status = 200;
6     resp.body_string = "Hello, world!";
7     return resp;
8 }
9
10 export function run(): void {
11     run_user(user_entrypoint);
12 }
```





Search or jump to...

/ Pull requests Issues Marketplace Explore

Bell +

fastly / terrarium-templates

Watch

15

Star

7

Fork

0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Branch: master ▾

terrarium-templates / 001_empty_rust / src / lib.rs

Find file Copy path



tyler Explicitly order the example via the dir names

2f56712 on Dec 5, 2018

1 contributor

14 lines (10 sloc) | 296 Bytes

Raw

Blame

History



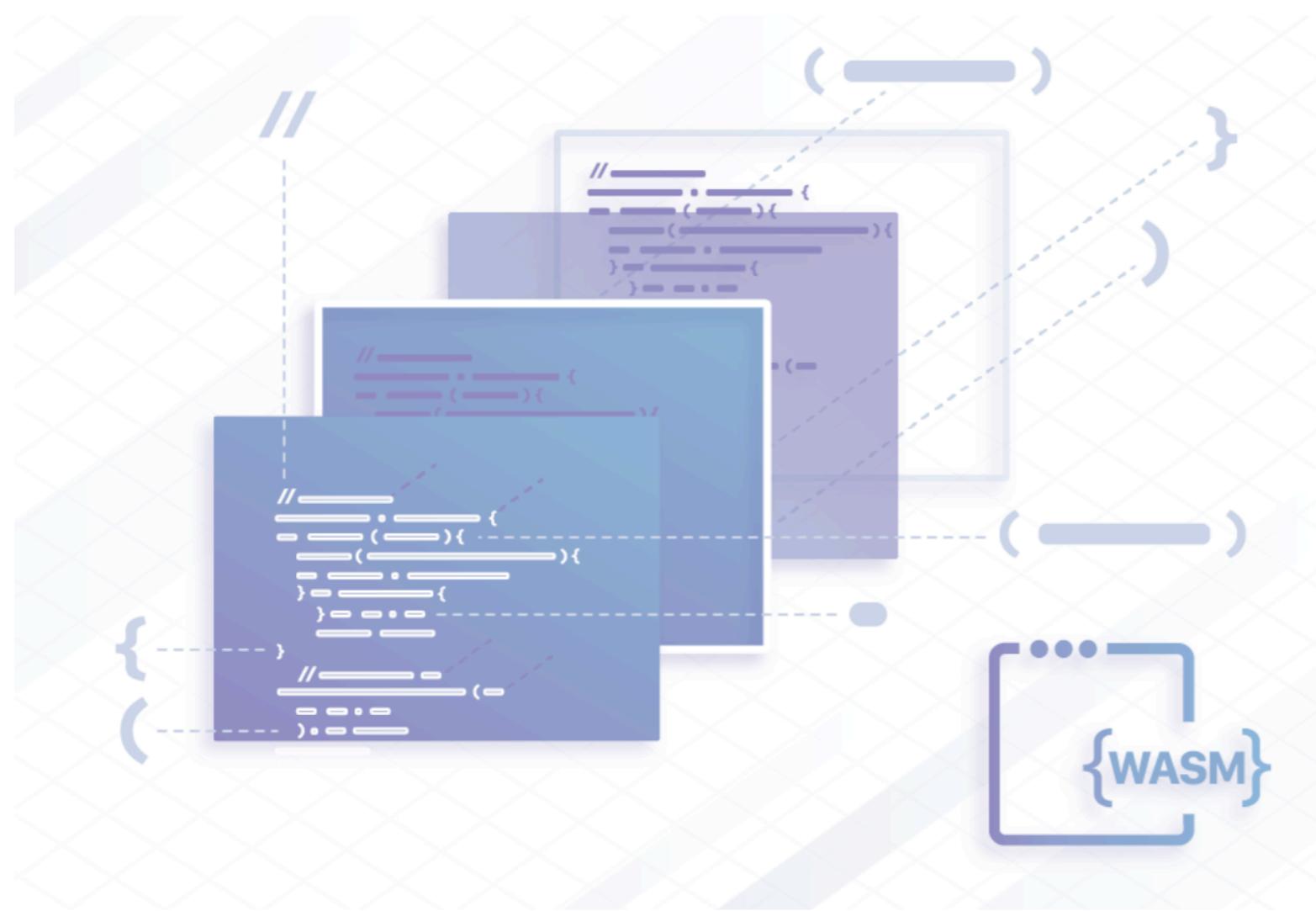
```
1 #[macro_use]
2 extern crate http_guest;
3
4 use http_guest::{Request, Response};
5
6 pub fn user_entrypoint(_req: &Request<Vec<u8>>) -> Response<Vec<u8>> {
7     Response::builder()
8         .status(200)
9         .body("Hello, world!".as_bytes().to_owned())
10        .unwrap()
11    }
12
13 guest_app!(user_entrypoint);
```



WebAssembly on Cloudflare Workers

01 Oct 2018 by [Kenton Varda](#).

[Tweet](#)



We just announced ten major new products and initiatives over [Crypto Week](#) and [Birthday Week](#), but our work is never finished. We're continuously upgrading our existing products with new functionality.

Search

Categories

- [Product News](#) ▶
- [Security](#) ▶
- [Performance](#) ▶
- [Reliability](#) ▶
- [Network](#) ▶
- [Serverless](#) ▶
- [International](#) ▶
- [Cloudflare Apps](#) ▶

Enter your email address

[Subscribe to this blog](#)

US callers

1 (888) 99-FLARE

UK callers

+44 (0)20 3514 6970

Singapore callers

+65 3158 3954

Find a Local Number

Fast Google Fonts with Cloudflare Workers

22 Nov 2018 by [Patrick Meenan](#).

[Tweet](#)

Google Fonts is one of the most common third-party resources on the web, but carries with it significant user-facing performance issues. Cloudflare Workers running at the edge is a great solution for fixing these performance issues, without having to modify the publishing system for every site using Google Fonts.

This post walks through the implementation details for how to fix the performance of Google Fonts with Cloudflare Workers. More importantly, it also provides code for doing high-performance content modification at the edge using Cloudflare Workers.

Google fonts are SLOW

First, some background. [Google Fonts](#) provides a rich selection of royalty-free fonts for sites to use. You select the fonts you want to use, and end up with a simple stylesheet URL to include on your pages, as well as styles to use for applying the fonts to parts of the page:

```
<link href="https://fonts.googleapis.com/css?family=Open+Sans|Roboto+Slab"
      rel="stylesheet">
<style>
body {
```

Search

Categories

- [Product News](#) ▶
- [Security](#) ▶
- [Performance](#) ▶
- [Reliability](#) ▶
- [Network](#) ▶
- [Serverless](#) ▶
- [International](#) ▶
- [Cloudflare Apps](#) ▶

Enter your email address

[Subscribe to this blog](#)

US callers

1 (888) 99-FLARE

UK callers

+44 (0)20 3514 6970

Singapore callers

+65 3158 3954

[Find a Local Number](#)



is the WebAssembly Package Manager

Recently updated packages

navigaid/wapm

golang wasm example

 navigaid published 0.1.4 • 3 days ago

drbh/flate

 Compress files on the client side super fast using WASM. Supports GZIP, ZLIB and DEFLATE compression and decompression

 drbh published 0.1.0 • 7 days ago

fortune

fortune is a program that displays a pseudorandom message from a database of quotations

 syrusakbary published 0.1.0 • 8 days ago

jsc

The JavaScript engine that powers Safari

 syrusakbary published 0.1.0 • 9 days ago

<https://wapm.io/>



moz://a

HACKS

[Download Firefox](#)

Search Mozilla Hacks

Standardizing WASI: A system interface to run WebAssembly outside the web



By [Lin Clark](#)

Posted on March 27, 2019 in [Code Cartoons](#), [Featured Article](#), and [WebAssembly](#)

Share This



Today, we announce the start of a new standardization effort—WASI, the WebAssembly system interface.

Why: Developers are starting to push WebAssembly beyond the browser, because it provides a fast, scalable, secure way to run the same code across all machines.

But we don't yet have a solid foundation to build upon. Code outside of a browser needs a way to talk to the system—a system interface. And the WebAssembly platform doesn't have that yet.

What: WebAssembly is an assembly language for a conceptual machine, not a physical one. This is why it can be run across a variety of different machine architectures.

Just as WebAssembly is an assembly language for a conceptual machine, WebAssembly needs a system interface for a conceptual operating system, not

<https://hacks.mozilla.org/2019/03/standardizing-wasi-a-webassembly-system-interface/>

If WASM+WASI existed in 2008, we wouldn't have needed to created Docker. That's how important it is. Webassembly on the server is the future of computing. A standardized system interface was the missing link. Let's hope WASI is up to the task! twitter.com/linclark/status/1103901100000000000



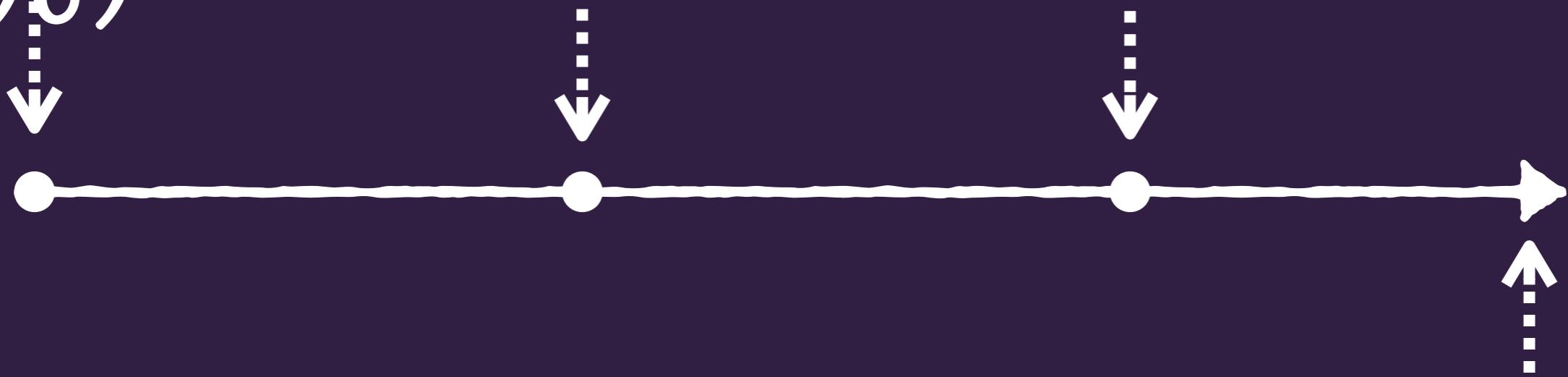
Solomon Hykes @solomonstre

9:39pm - 27 Mar 2019

Unix
1969

The Web
1989

Wasm
2015



Zombie
apocalypse