





Matthias Endler

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



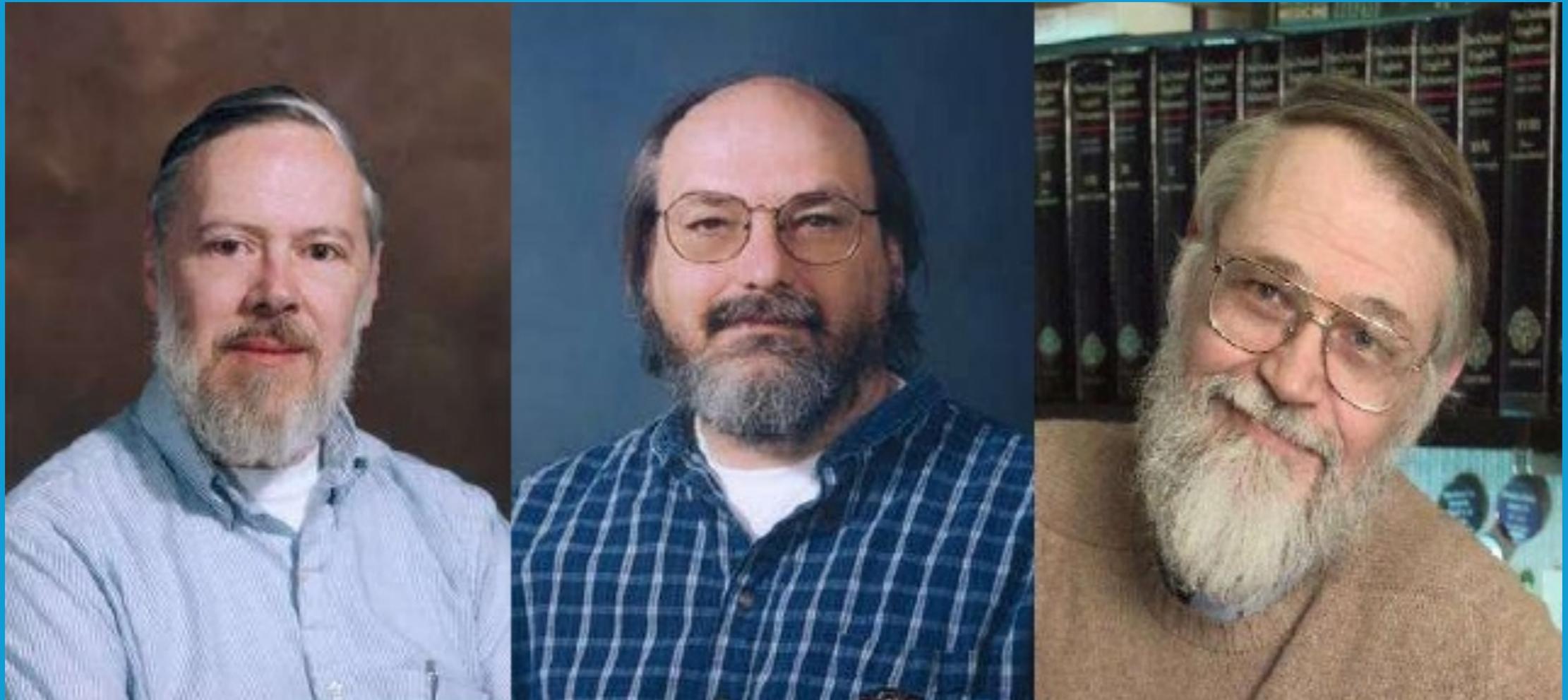
WTF is WASM?











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"

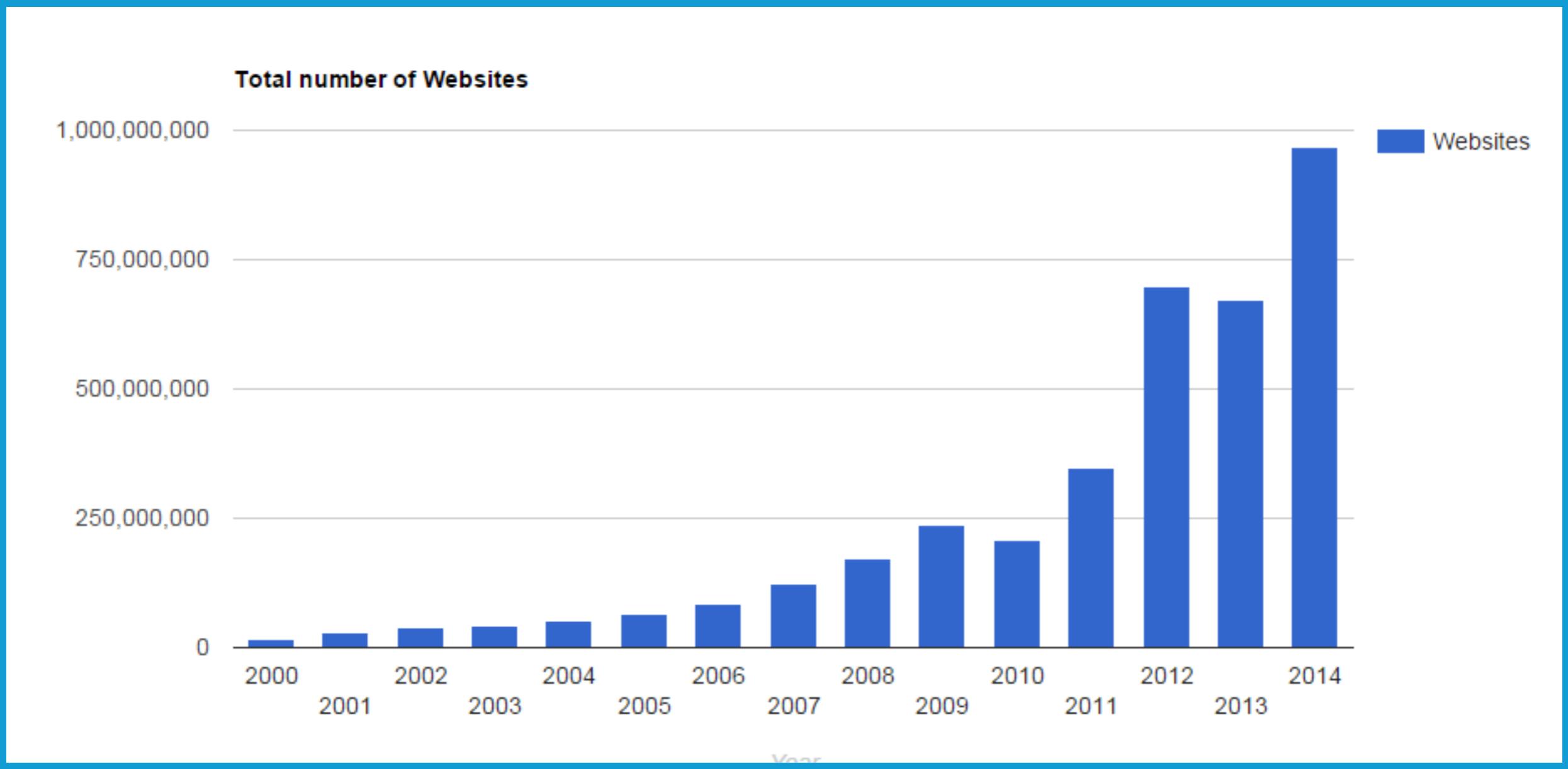
-- Dennis Ritchie

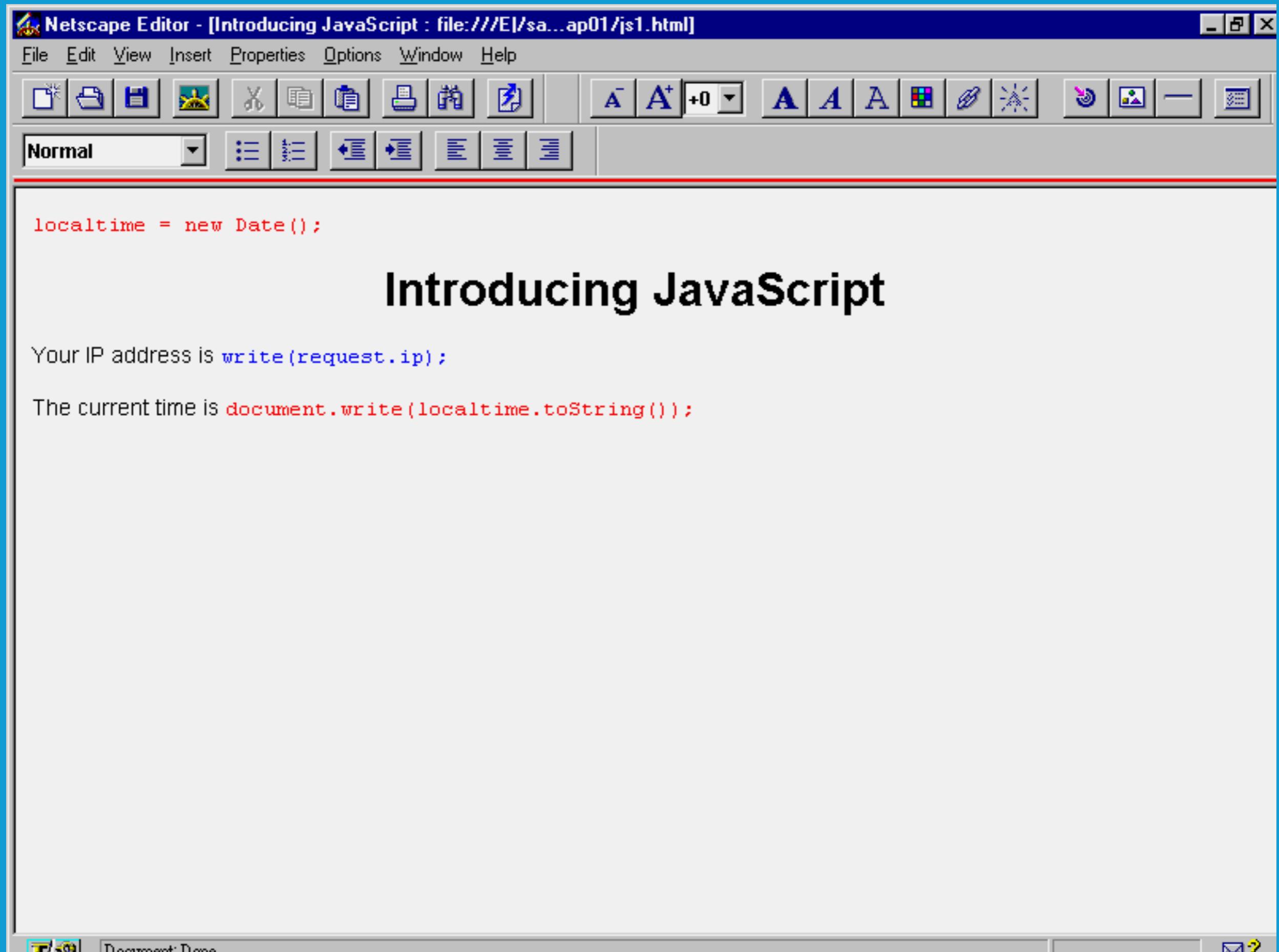




```
<HTML>
  <HEAD>
    <TITLE>WWW</TITLE>
  </HEAD>
  <BODY>
    <BLINK>Hello, World!</BLINK>
  </BODY>
</HTML>
```

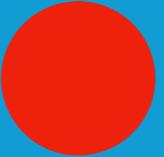




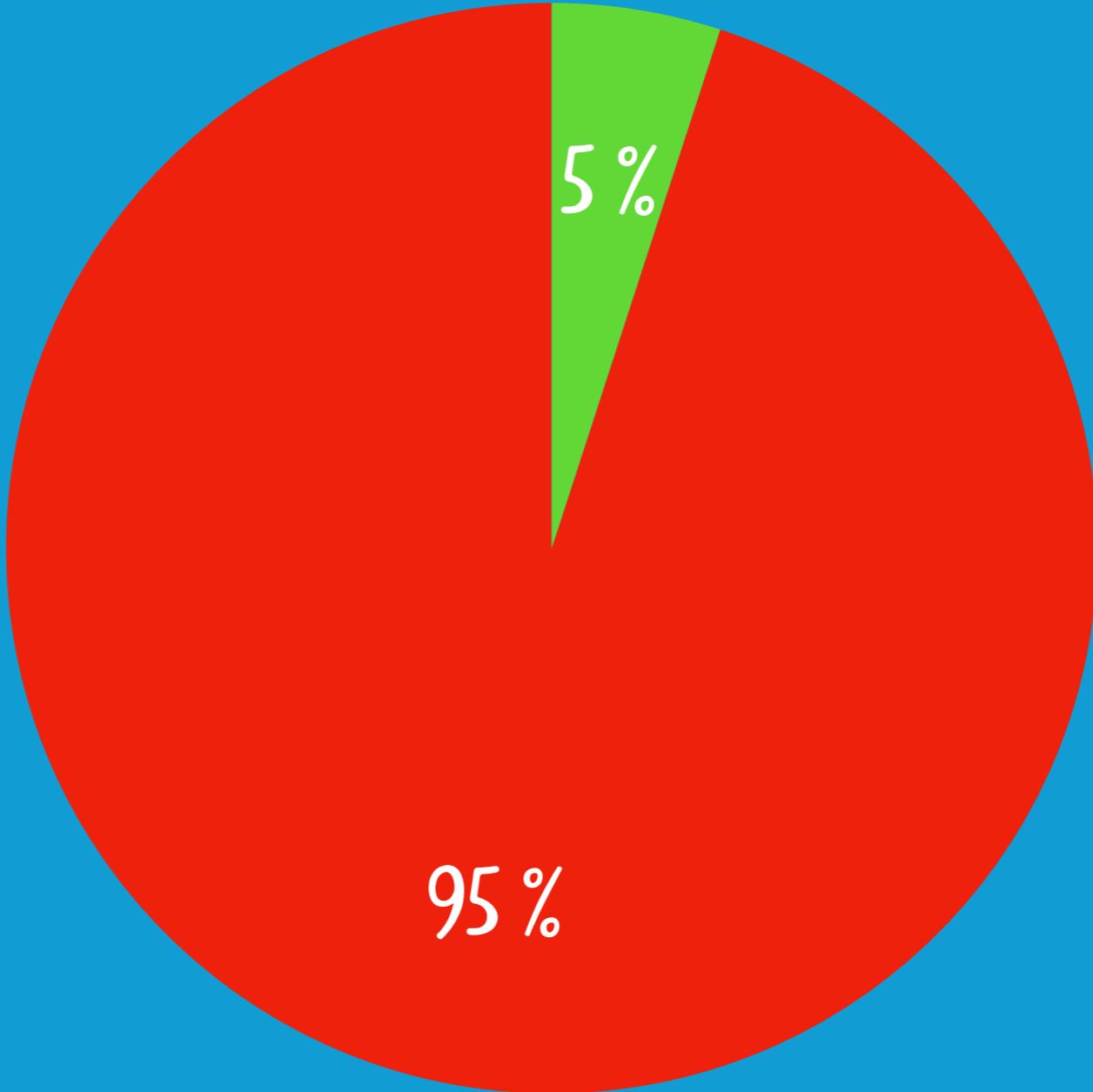




JavaScript



Rest



5 %

95 %



CoffeeScript

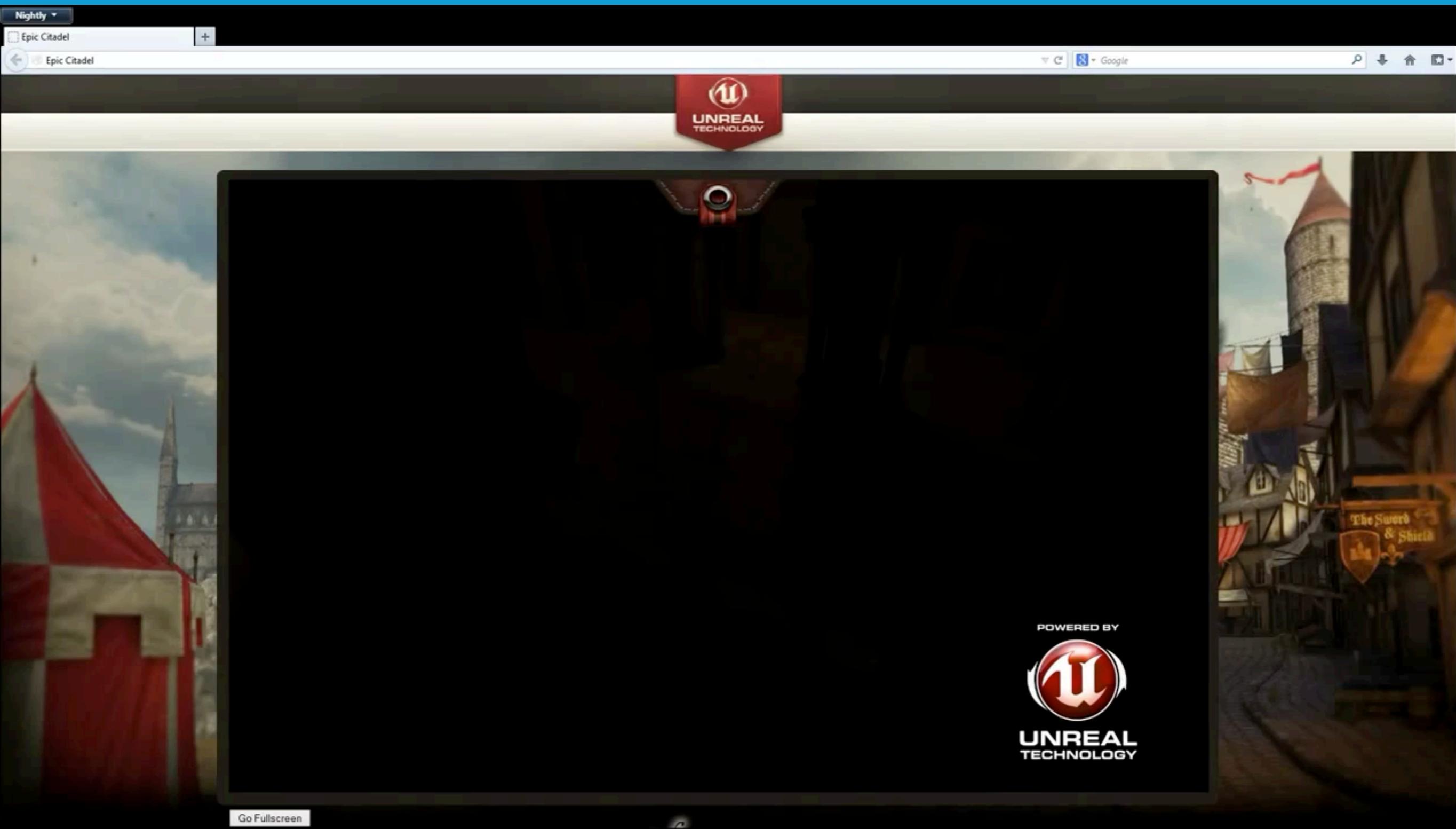
```
# Assignment:  
number: 42  
opposite_day: true  
  
# Conditions:  
number: -42 if opposite_day  
  
# Functions:  
square: x => x * x.  
  
# Arrays:  
list: [1, 2, 3, 4, 5]  
  
# Objects:  
math: {  
    root: Math.sqrt  
    square: square  
    cube: x => x * square(x).  
}  
  
# Array comprehensions: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

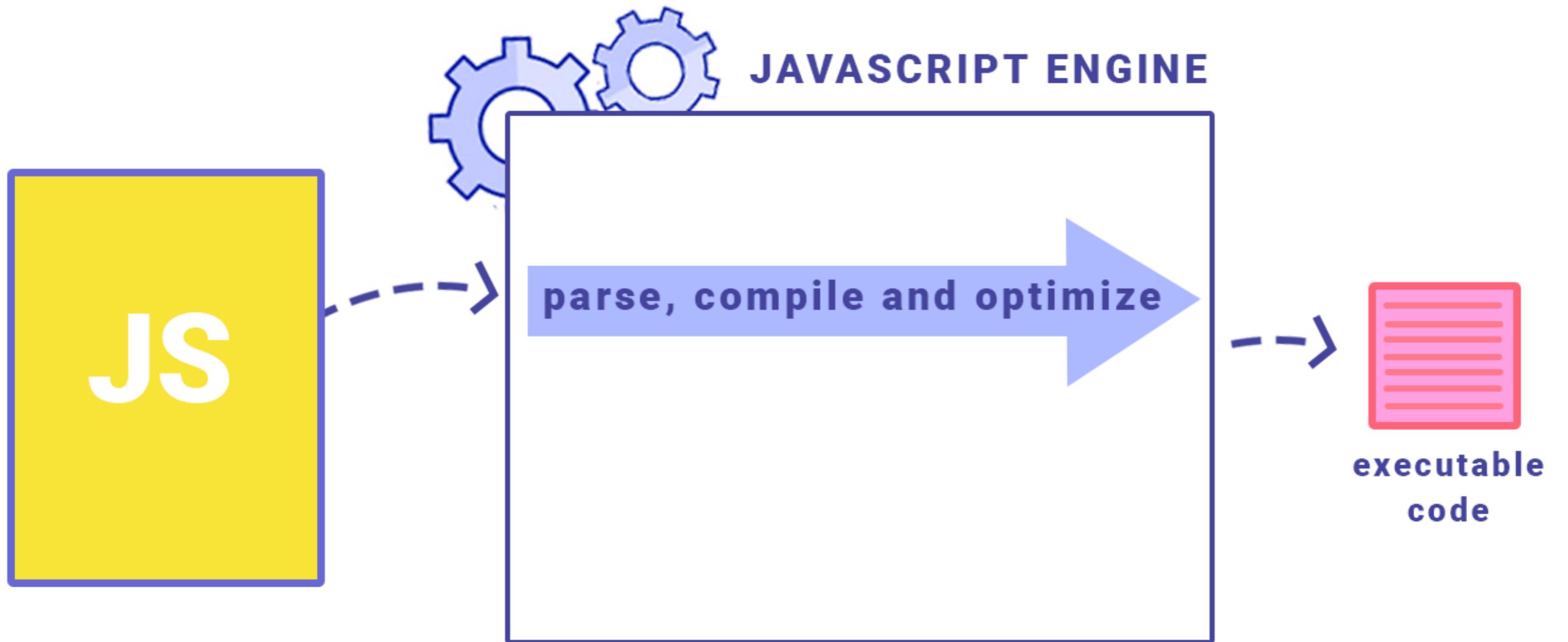


emscripten

```
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;
                }
            }
        }
    }
}
```



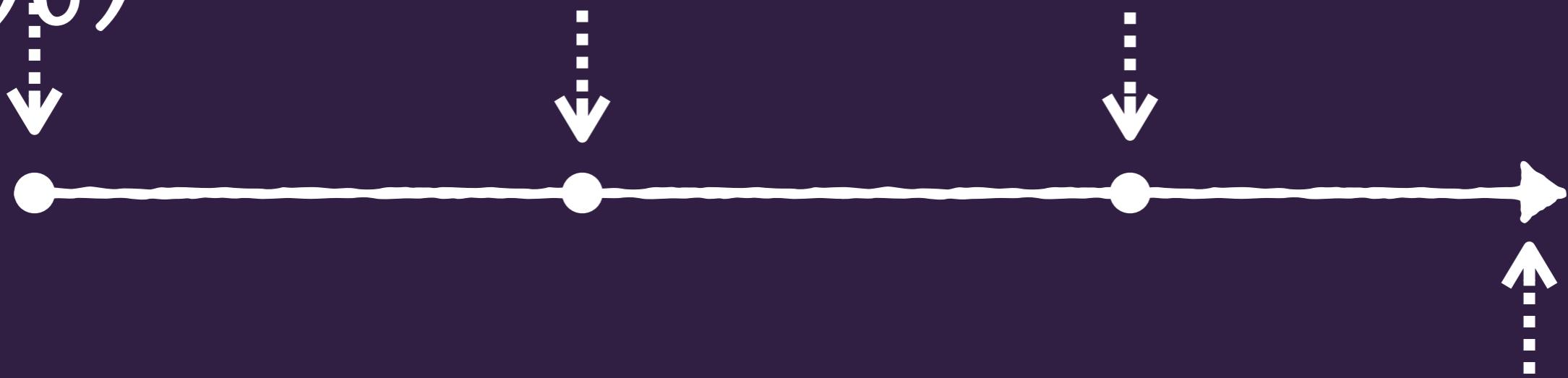
[\[Source\]](#)



Unix
1969

The Web
1989

Wasm
2015

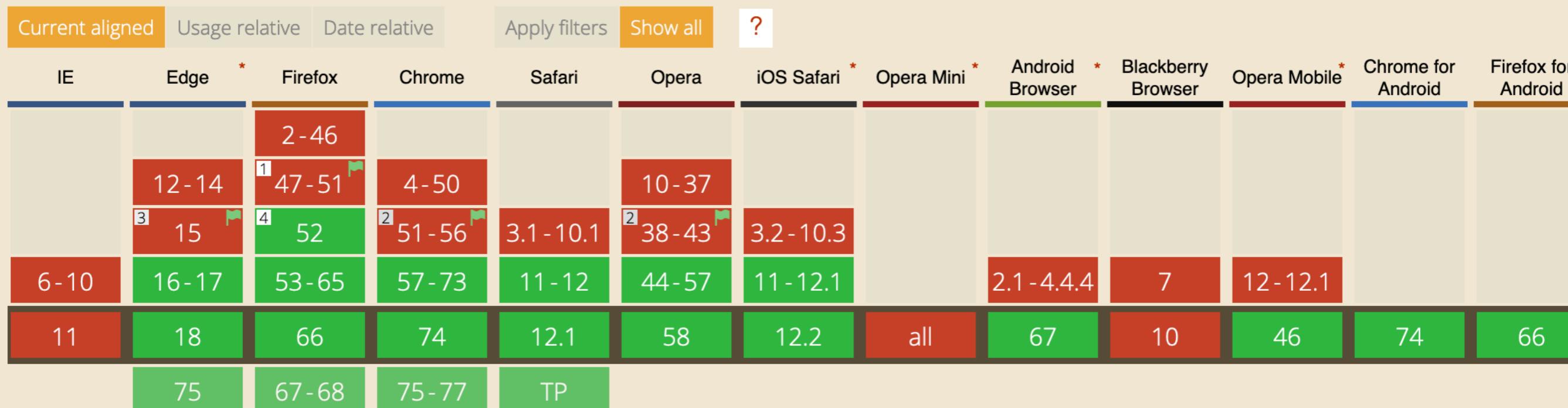


Zombie
apocalypse

WebAssembly

Usage % of all users ▾ ?

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

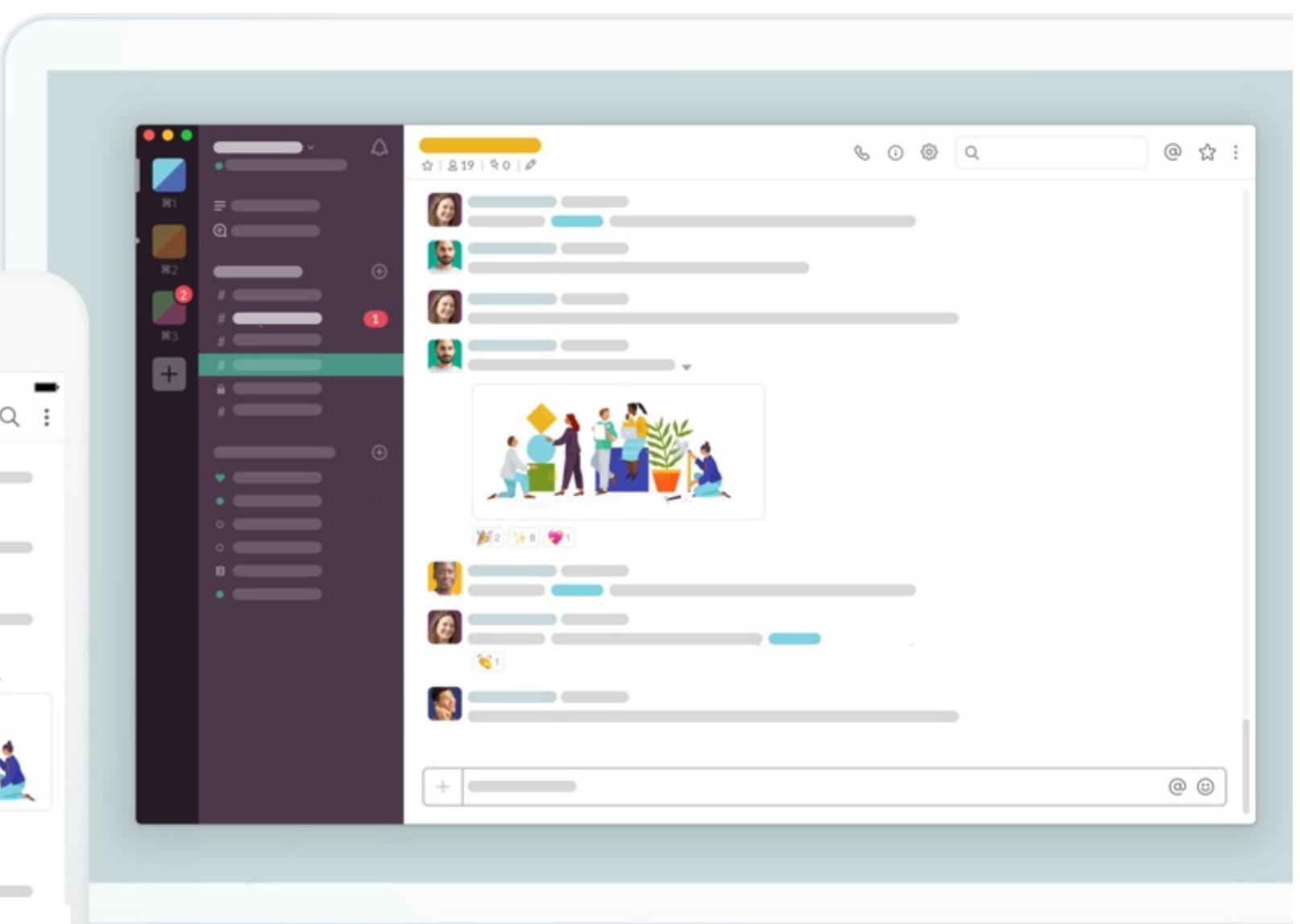
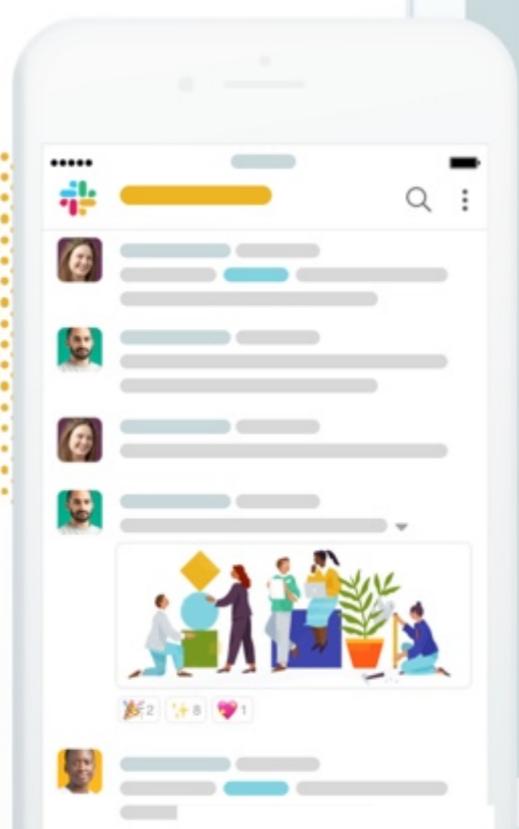






xenitaph





Wasm is . . .

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

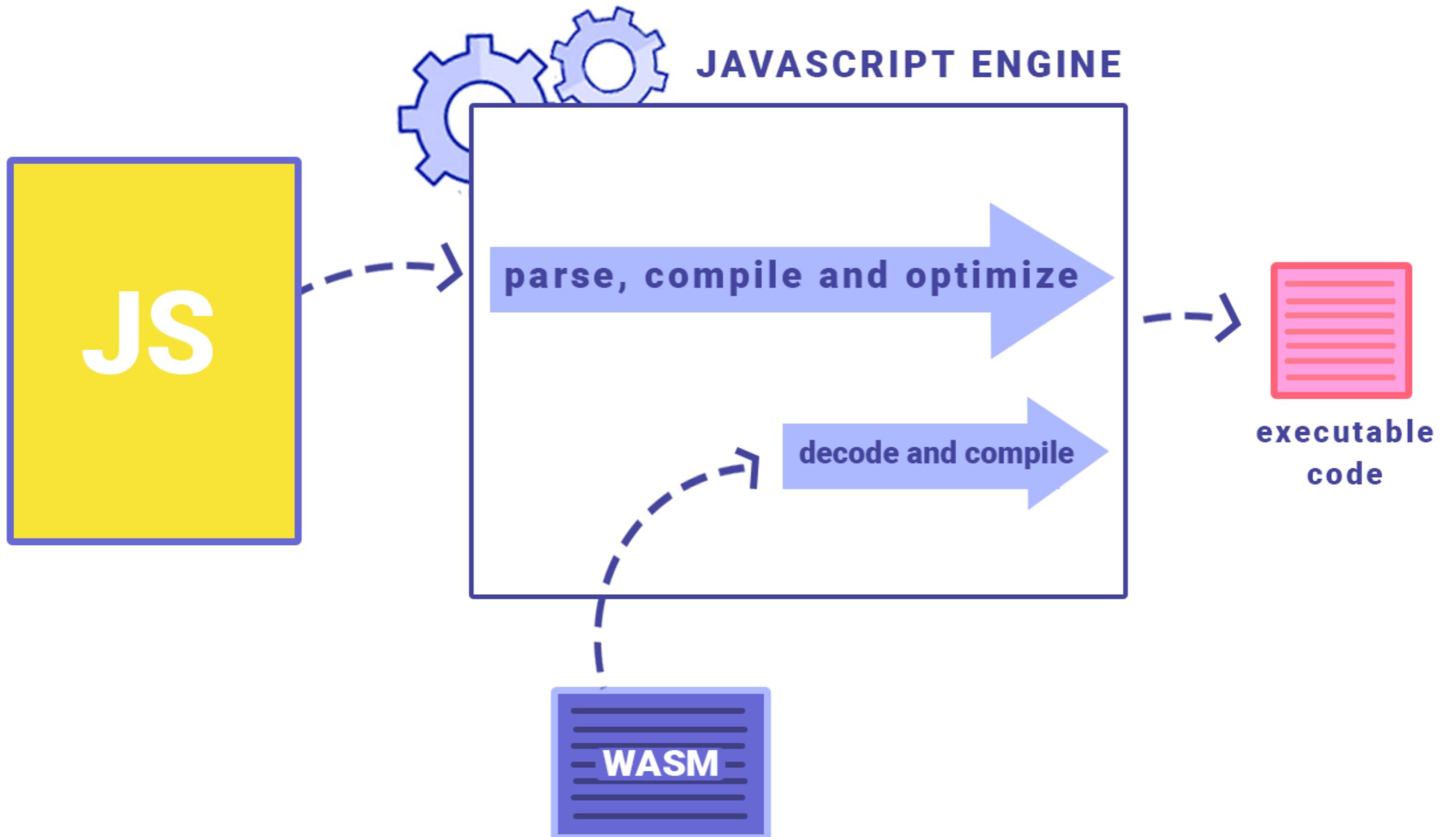
Decoding



Validation

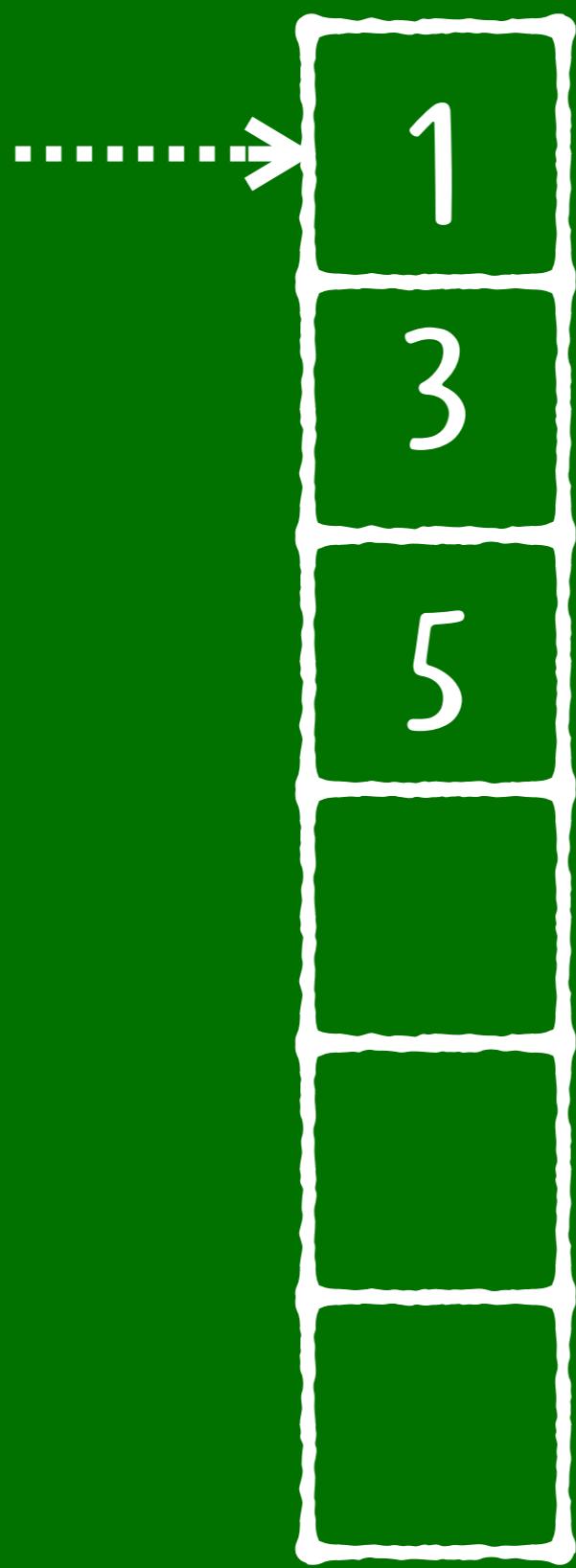


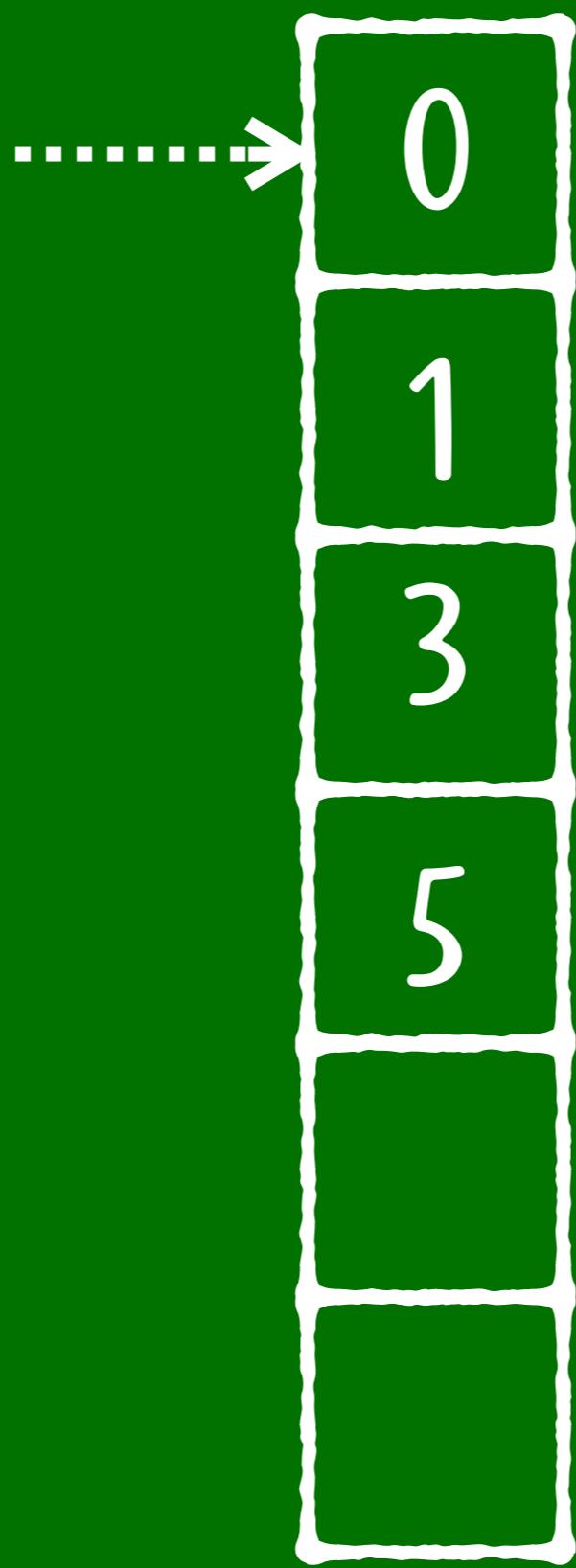
Execution

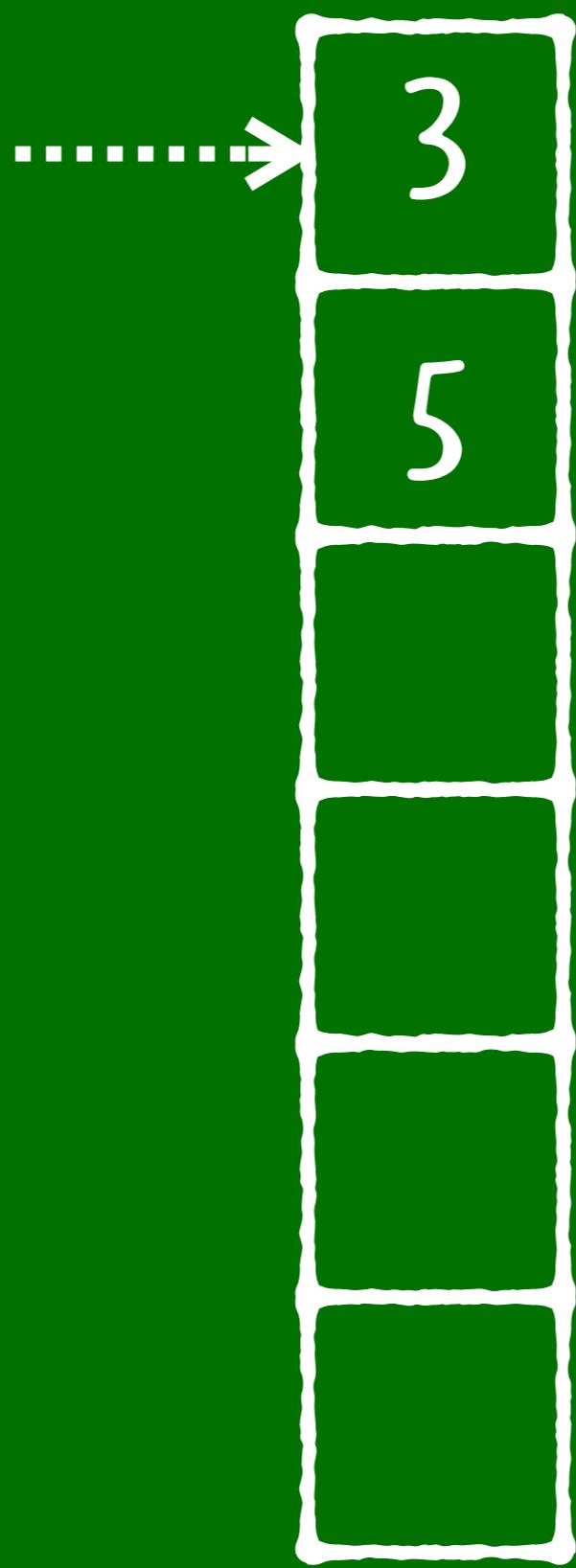


Datatypes!

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

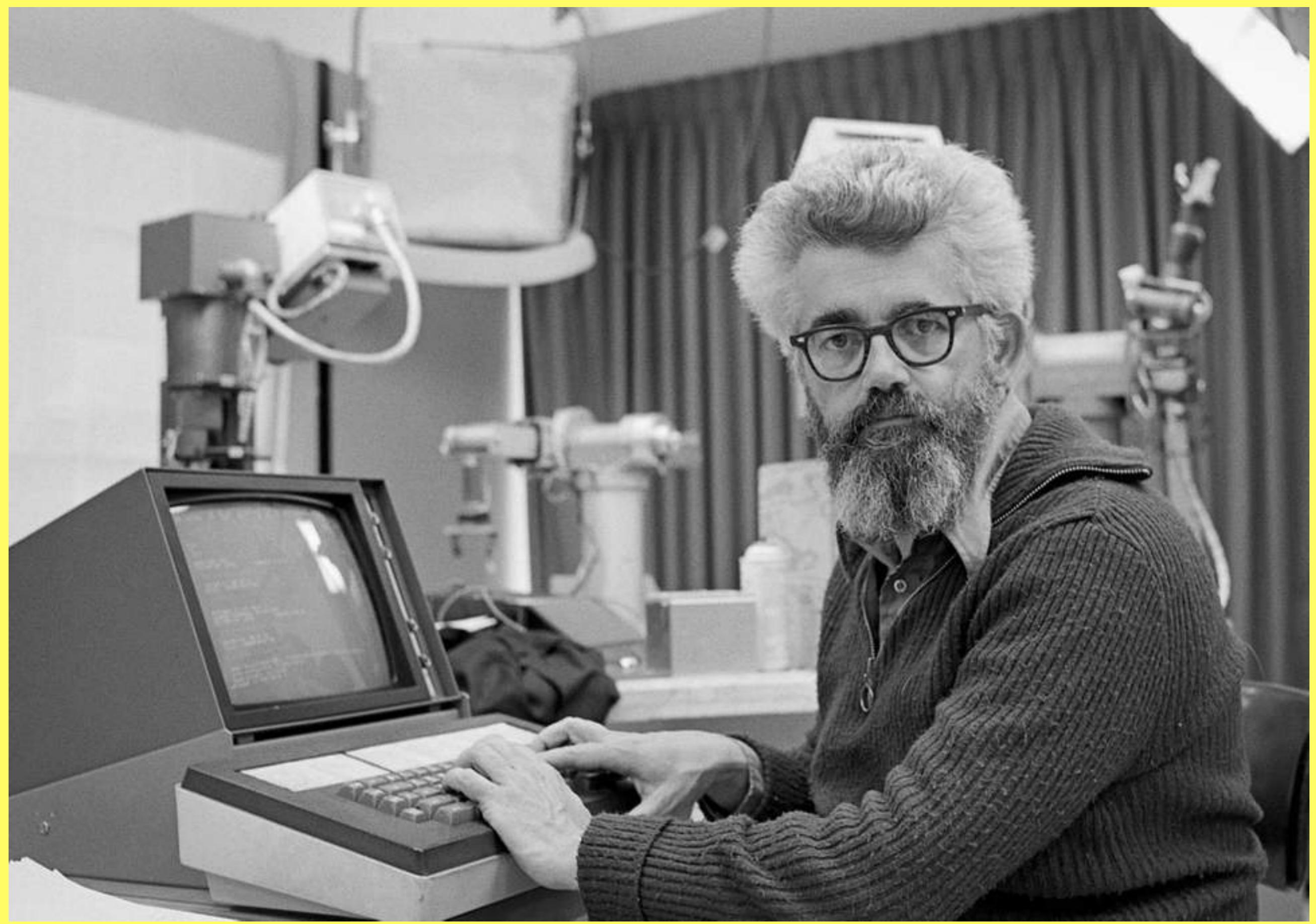






LISP Theory & Practice

```
(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) ... )
```

```
(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 $lhs i32) (param $rhs i32) (result i32)
    get_local $lhs
    get_local $rhs
    i32.add
)
)
```

```
(module
  (func $add (param $lhs i32) (param $rhs i32) (result i32)
    get_local $lhs
    get_local $rhs
    i32.add)
  (export "add" (func $add)))
)
```

```
// JavaScript
function add(lhs, rhs) {
  return lhs + rhs;
}
```

```
(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"
  )
)
```

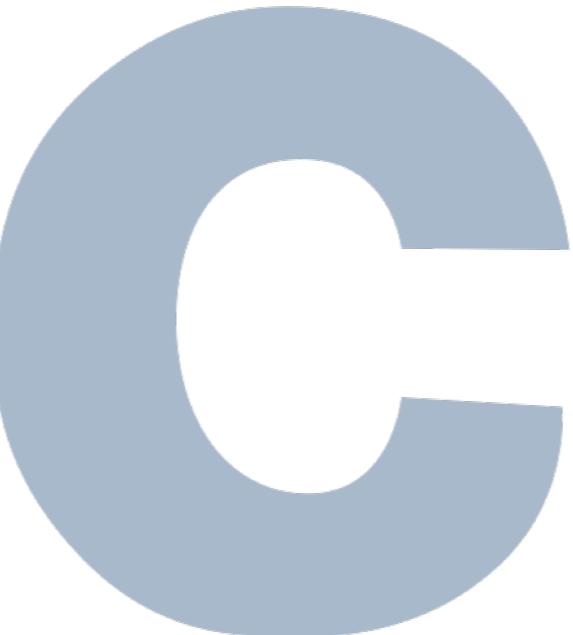
TEXTUAL FORMAT

```
(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

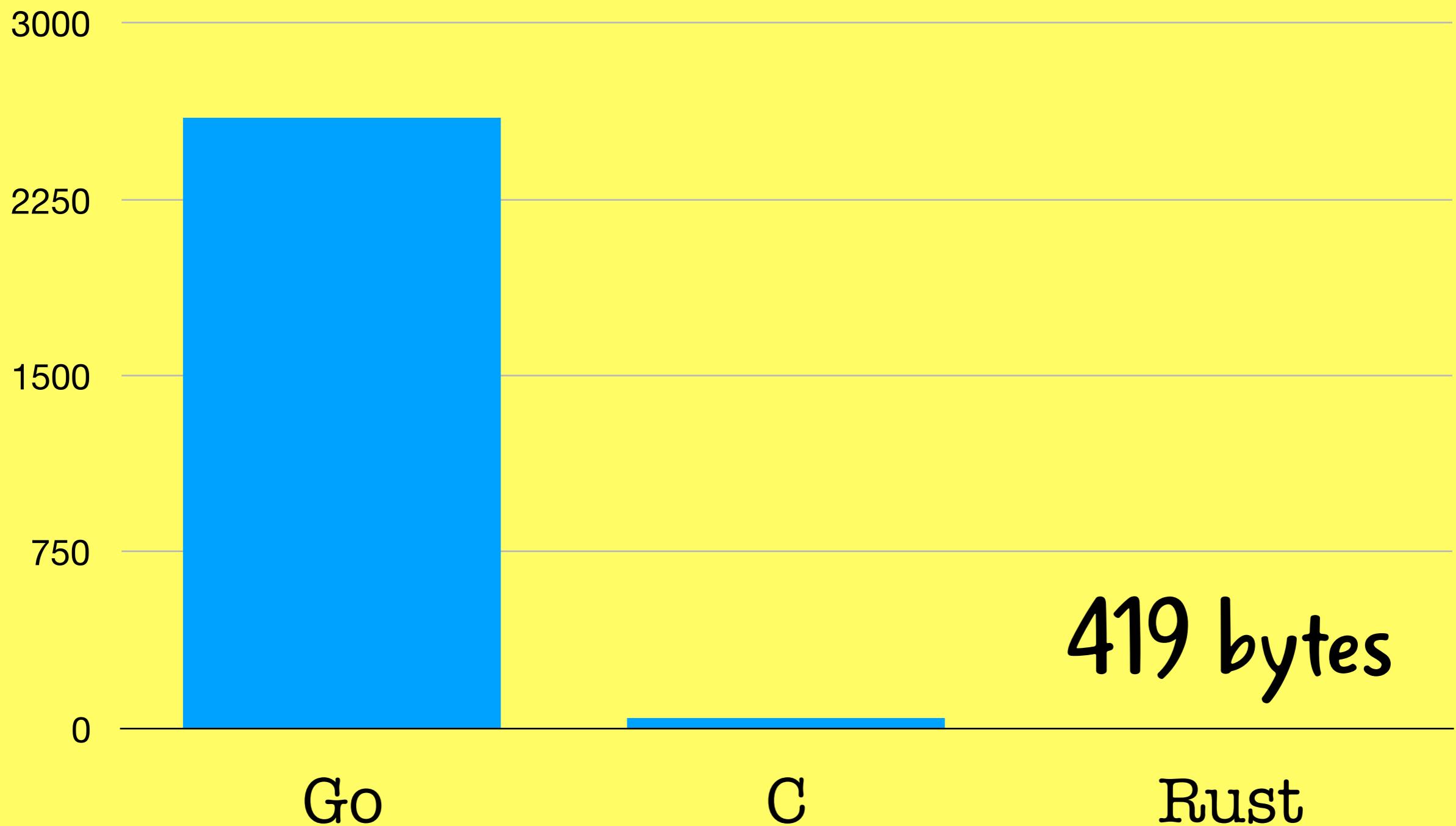


PROGRAMMING
LANGUAGE





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





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



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/>



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/>