WEBASSEMBLY AND THE

FUTURE OF THE WEB PLATFORM

WELCOME TO MY TED TALK

I'M AG_DUBS

THE RIGHT LANGUAGE FOR THE JOB

THE RIGHT LANGUAGE FOR THE JOB?

THE RIGHT LANGUAGE FOR THE JOB?

THE RIGHT LANGUAGE FOR THE JOB?

WEBASSEMBLY IS A LANGUAGE THAT WILL REDEFINE THE JOBS WE DO AND THE LANGUAGES WE DO THEM IN

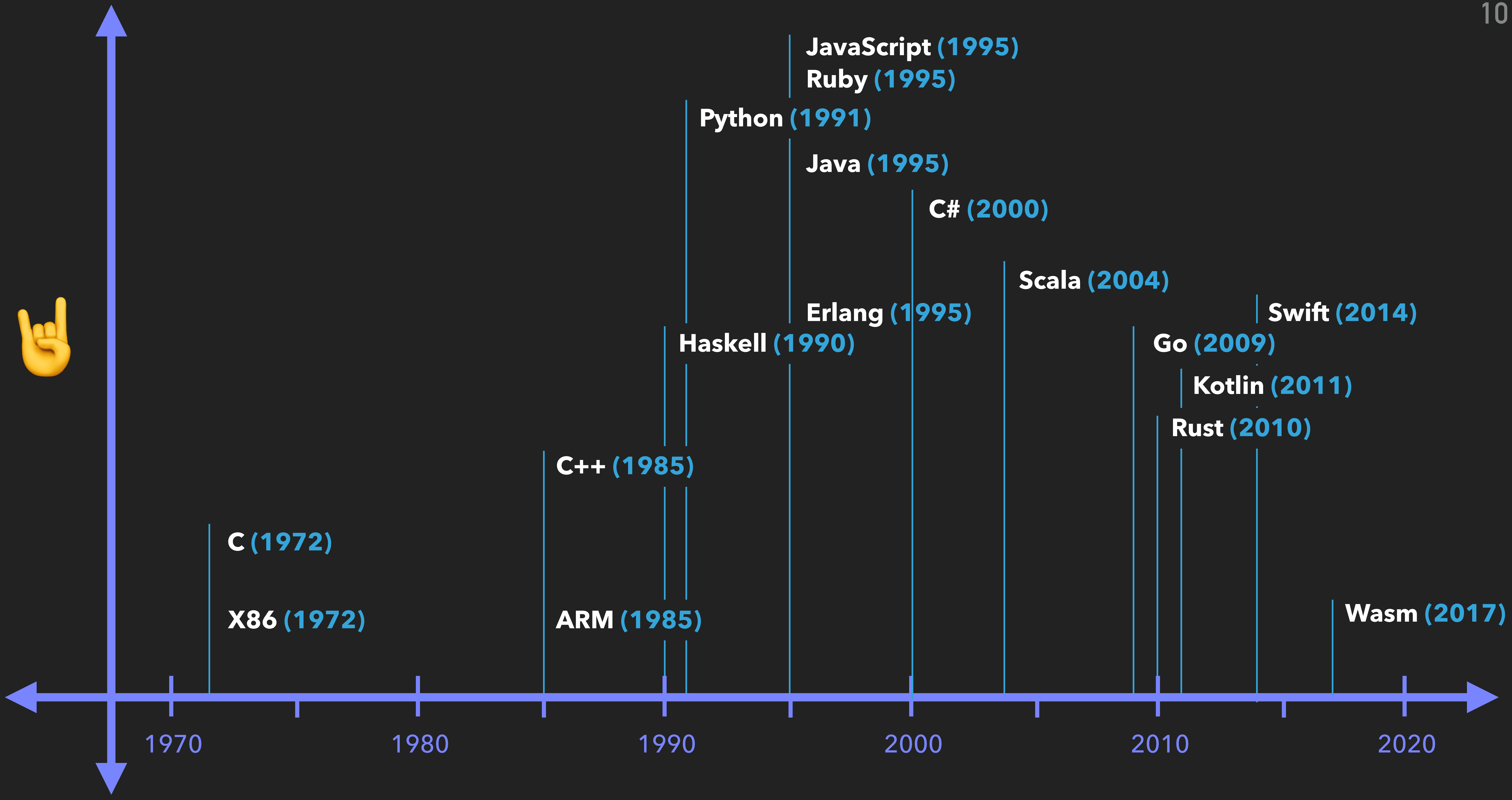
AG_DUBS

LET'S GET READY TO RUMBLE

- Why is WebAssembly different?
- What is WebAssembly?
- What is the Web Platform?
- How is WebAssembly going to change the Web Platform?
- Where we are now, where we going in the future

AN EXTREMELY BRIEF AND INCOMPLETE

HISTORY OF PROGRAMMING LANGUAGE DEVELOPMENT



JavaScript (1995)

Ruby (1995)

Python (1991)

Java (1995)

C# (2000)

Scala (2004)

Erlang (1995)

Swift (2014)

Haskell (1990)

Go (2009)

Kotlin (2011)

Rust (2010)

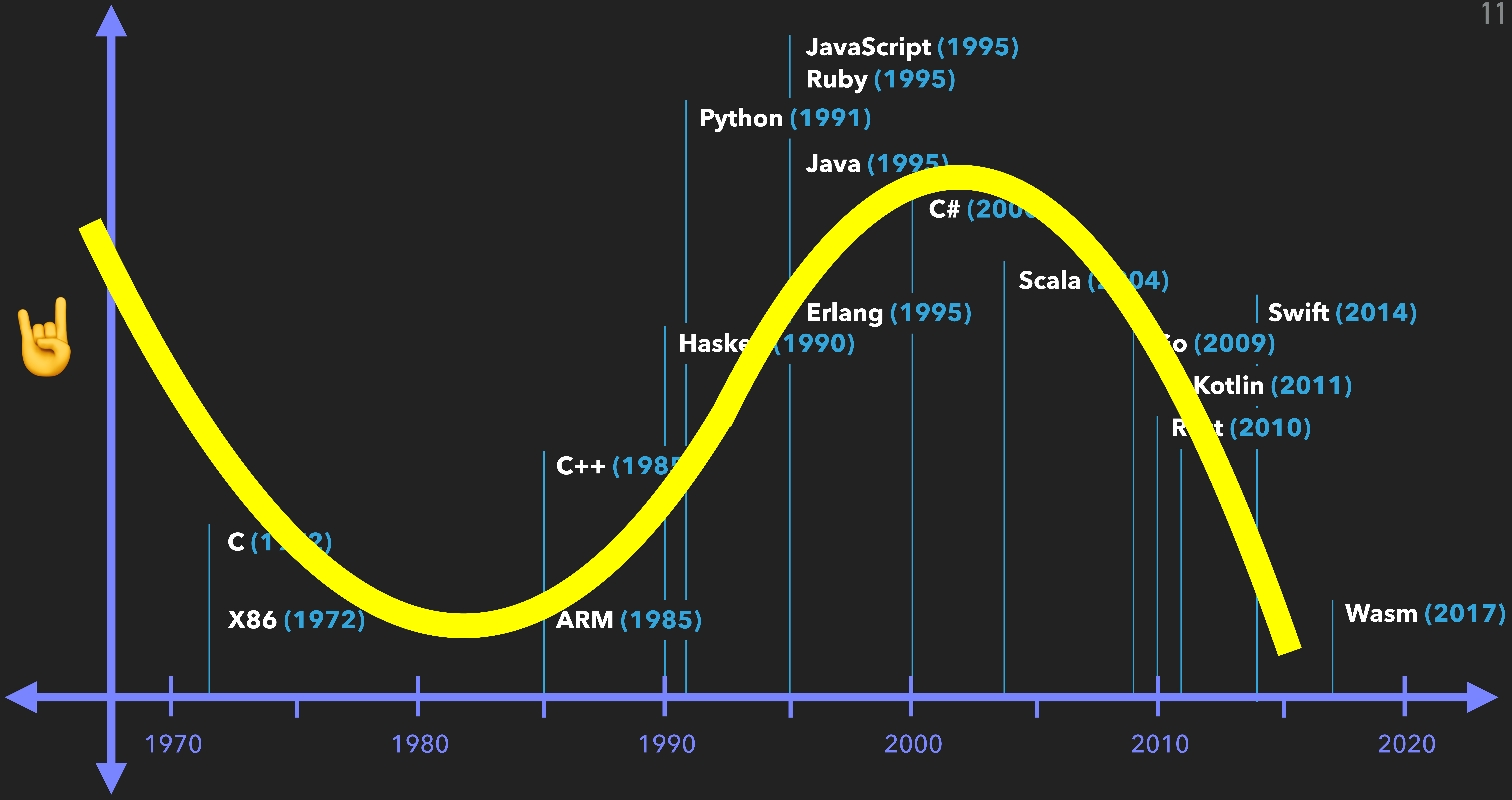
C++ (1985)

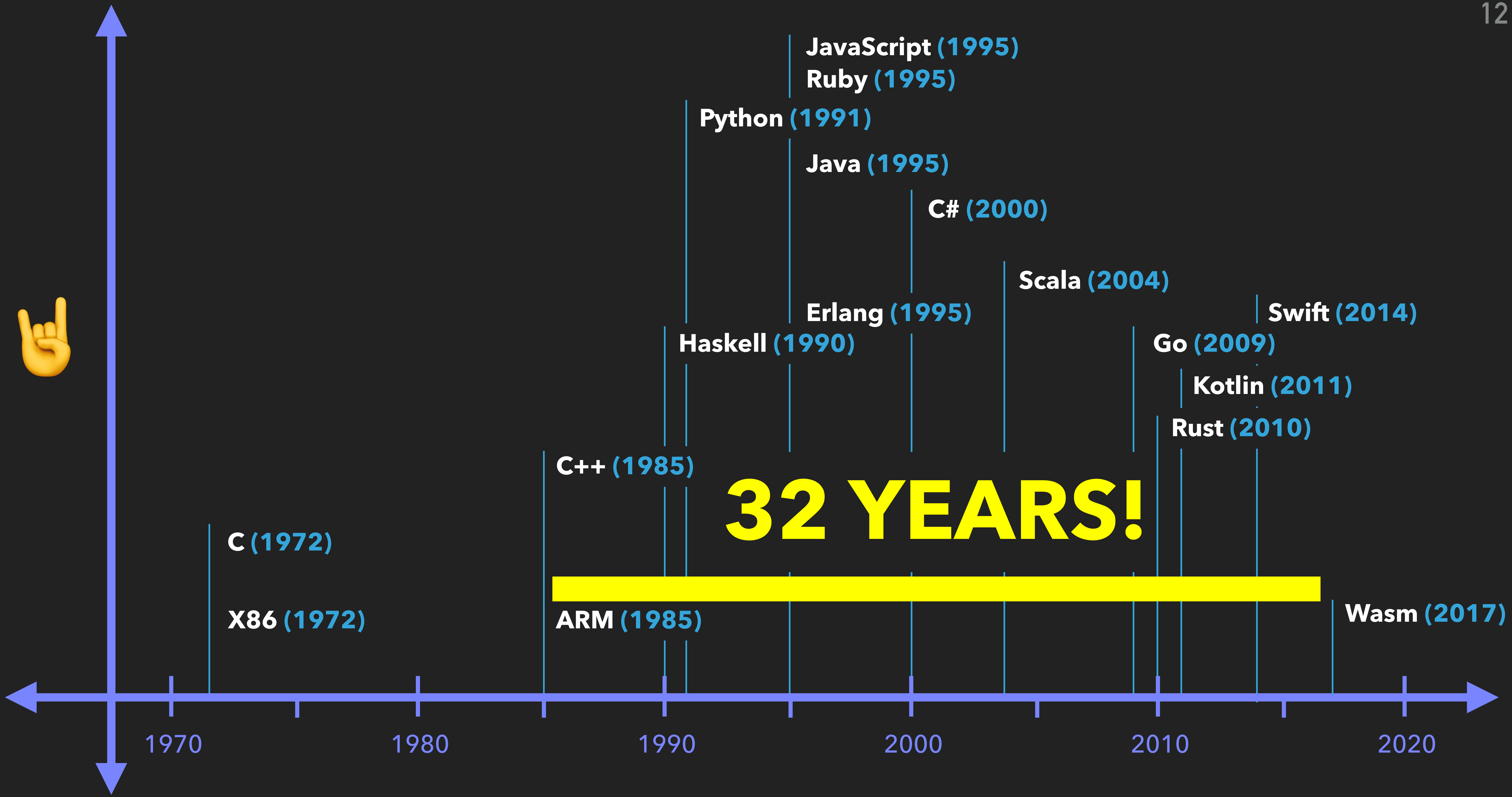
C (1972)

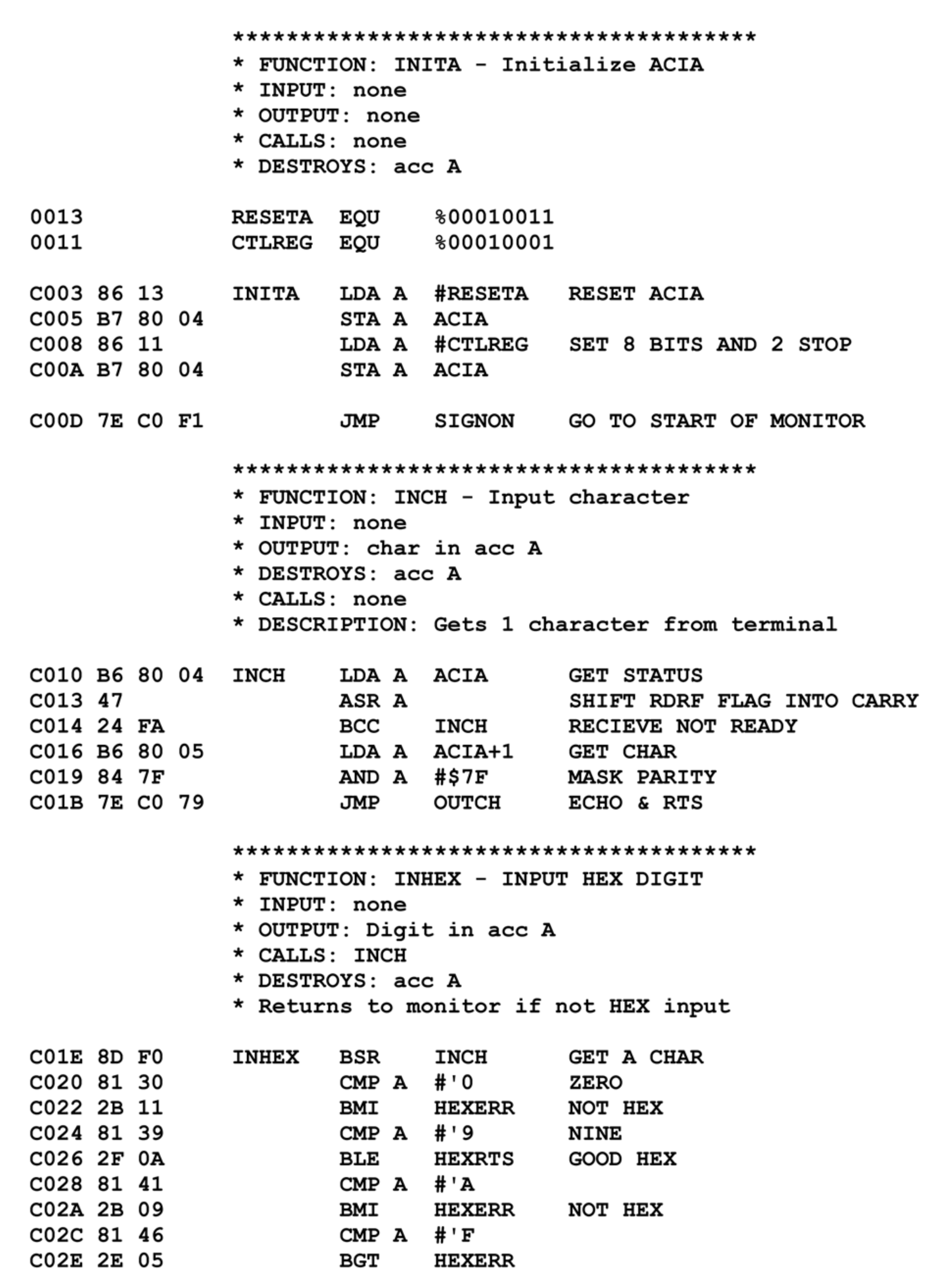
X86 (1972)

ARM (1985)

Wasm (2017)


🤘
JavaScript (1995)
Ruby (1995)
Python (1991)
Java (1995)
C# (2000)
Scala (2004)
Erlang (1995)
Swift (2014)
Haskell (1990)
Go (2009)
Kotlin (2011)
Rust (2010)
C++ (1985)
C (1972)
X86 (1972)
ARM (1985)
Wasm (2017)
1970
1980
1990
2000
2010
2020


🤘
JavaScript (1995)
Ruby (1995)
Python (1991)
Java (1995)
C# (2000)
Scala (2004)
Erlang (1995)
Swift (2014)
Haskell (1990)
Go (2009)
Kotlin (2011)
Rust (2010)
C++ (1985)
32 YEARS!
C (1972)
X86 (1972)
ARM (1985)
Wasm (2017)
1970
1980
1990
2000
2010
2020



```
                ****************************************
                * FUNCTION: INITA - Initialize ACIA
                * INPUT: none
                * OUTPUT: none
                * CALLS: none
                * DESTROYS: acc A

0013            RESETA  EQU    %00010011
0011            CTLREG  EQU    %00010001

C003 86 13      INITA   LDA A  #RESETA   RESET ACIA
C005 B7 80 04           STA A  ACIA
C008 86 11              LDA A  #CTLREG   SET 8 BITS AND 2 STOP
C00A B7 80 04           STA A  ACIA

C00D 7E C0 F1           JMP    SIGNON    GO TO START OF MONITOR

                ****************************************
                * FUNCTION: INCH - Input character
                * INPUT: none
                * OUTPUT: char in acc A
                * DESTROYS: acc A
                * CALLS: none
                * DESCRIPTION: Gets 1 character from terminal

C010 B6 80 04   INCH    LDA A  ACIA      GET STATUS
C013 47                 ASR A            SHIFT RDRF FLAG INTO CARRY
C014 24 FA              BCC    INCH      RECIEVE NOT READY
C016 B6 80 05           LDA A  ACIA+1    GET CHAR
C019 84 7F              AND A  #$7F      MASK PARITY
C01B 7E C0 79           JMP    OUTCH     ECHO & RTS

                ****************************************
                * FUNCTION: INHEX - INPUT HEX DIGIT
                * INPUT: none
                * OUTPUT: Digit in acc A
                * CALLS: INCH
                * DESTROYS: acc A
                * Returns to monitor if not HEX input

C01E 8D F0      INHEX   BSR    INCH      GET A CHAR
C020 81 30              CMP A  #'0       ZERO
C022 2B 11              BMI    HEXERR    NOT HEX
C024 81 39              CMP A  #'9       NINE
C026 2F 0A              BLE    HEXRTS    GOOD HEX
C028 81 41              CMP A  #'A
C02A 2B 09              BMI    HEXERR    NOT HEX
C02C 81 46              CMP A  #'F
C02E 2E 05              BGT    HEXERR
```

WHAT IS AN

ASSEMBLY LANGUAGE?

Rust

x86

Rust

x86

```
fn main() {
   let sum = 1 + 1;
}
```

100101010010

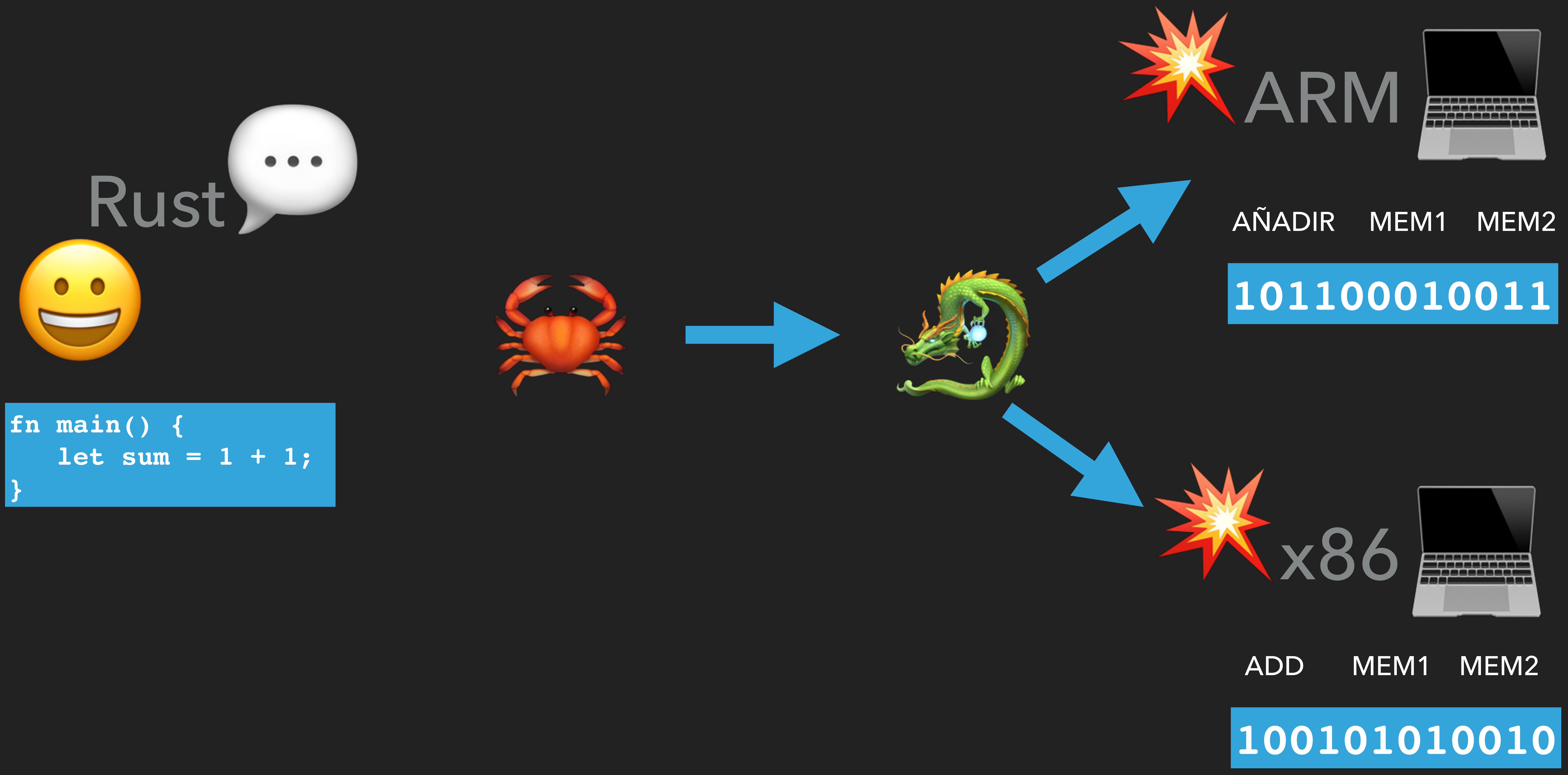
Rust

```
fn main() {
    let sum = 1 + 1;
}
```

x86

ADD MEM1 MEM2

100101010010



Rust

```
fn main() {
   let sum = 1 + 1;
}
```

ARM

AÑADIR MEM1 MEM2

101100010011

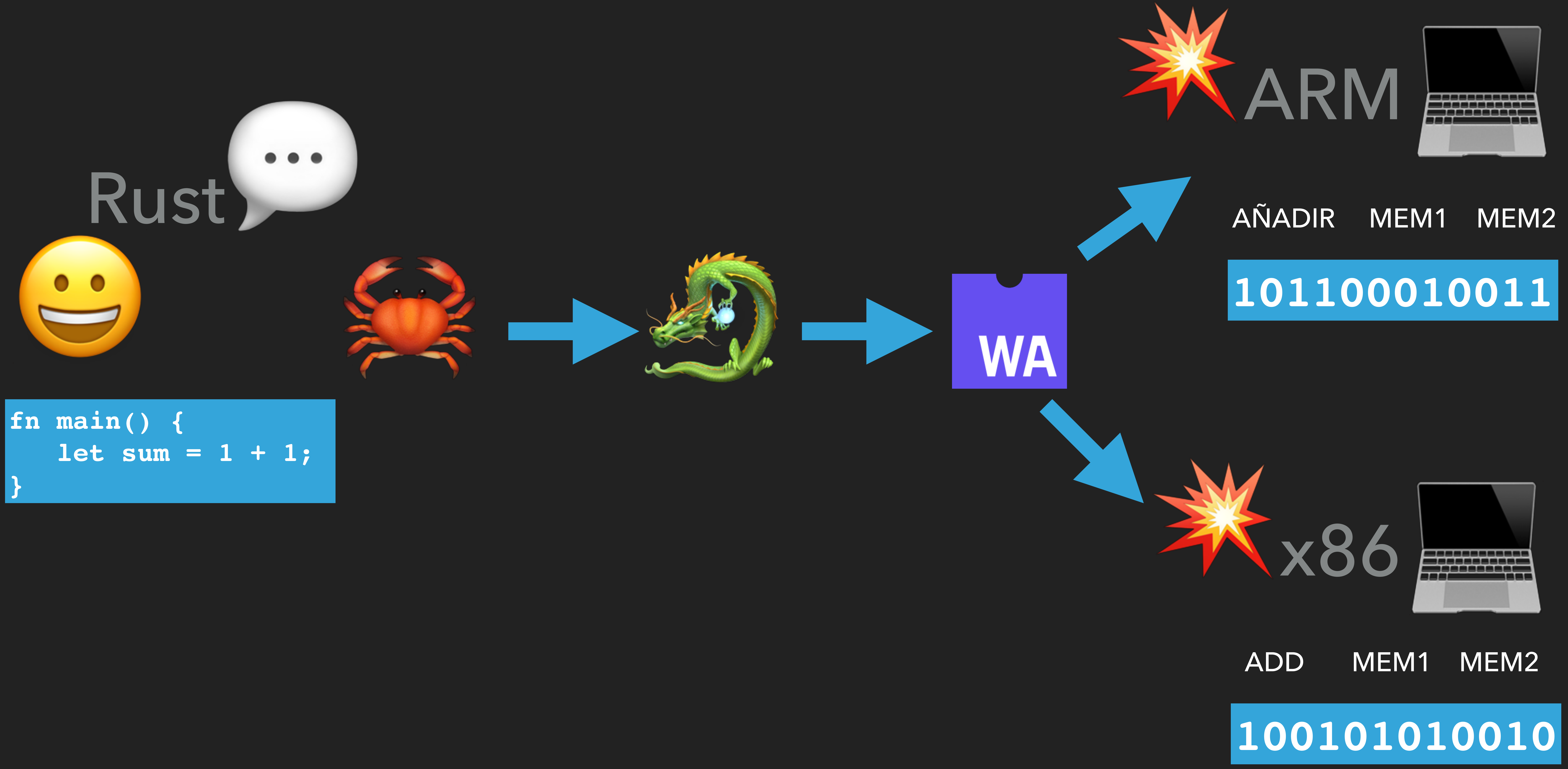
x86

ADD MEM1 MEM2

100101010010

WHAT IS

WEBASSEMBLY?



Rust

```
fn main() {
   let sum = 1 + 1;
}
```

ARM

AÑADIR MEM1 MEM2

101100010011

x86

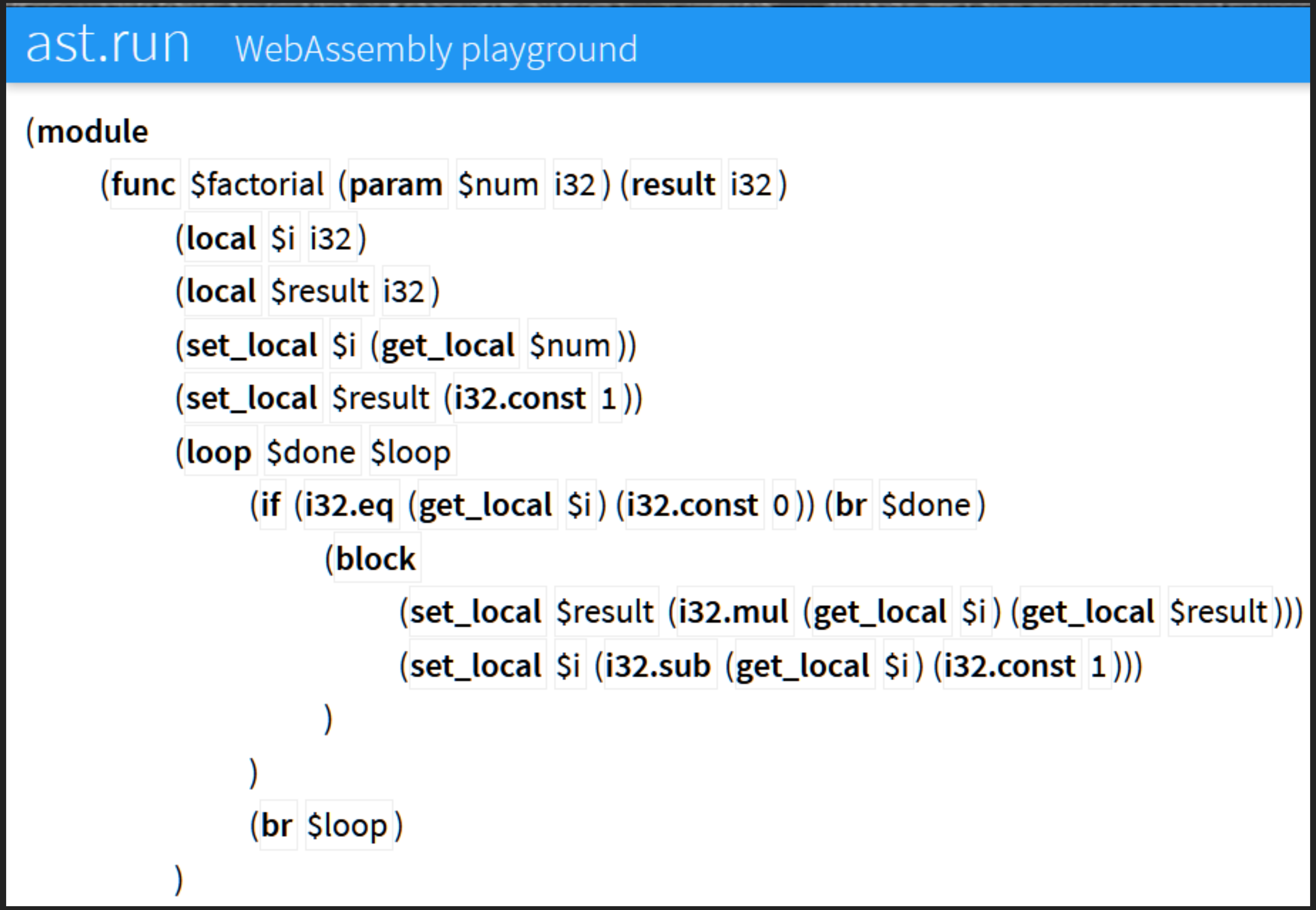
ADD MEM1 MEM2

100101010010

WEBASSEMBLY

- COMPACT BINARY FORMAT
- MULTIPLE “HUMAN-READABLE” TEXT FORMATS
- BUILT IN THE OPEN VIA WEB STANDARDS COMMITTEE
- INTEGRATES WITH EXISTING JAVASCRIPT VIS

HTTPS://WEBASSEMBLY.ORG/



```
(module
    (func $factorial (param $num i32) (result i32)
        (local $i i32)
        (local $result i32)
        (set_local $i (get_local $num))
        (set_local $result (i32.const 1))
        (loop $done $loop
            (if (i32.eq (get_local $i) (i32.const 0)) (br $done)
                (block
                    (set_local $result (i32.mul (get_local $i) (get_local $result)))
                    (set_local $i (i32.sub (get_local $i) (i32.const 1)))
                )
            )
            (br $loop)
        )
```

FACTORIAL

20 00 42 00 51 04 7E 42
01 05 20 00 20 00 42 01

RESOURCES

- HTTP://AST.RUN/
- HTTPS://WEBASSEMBLY.STUDIO/
- HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/WEBASSEMBLY

NO GC (YET)

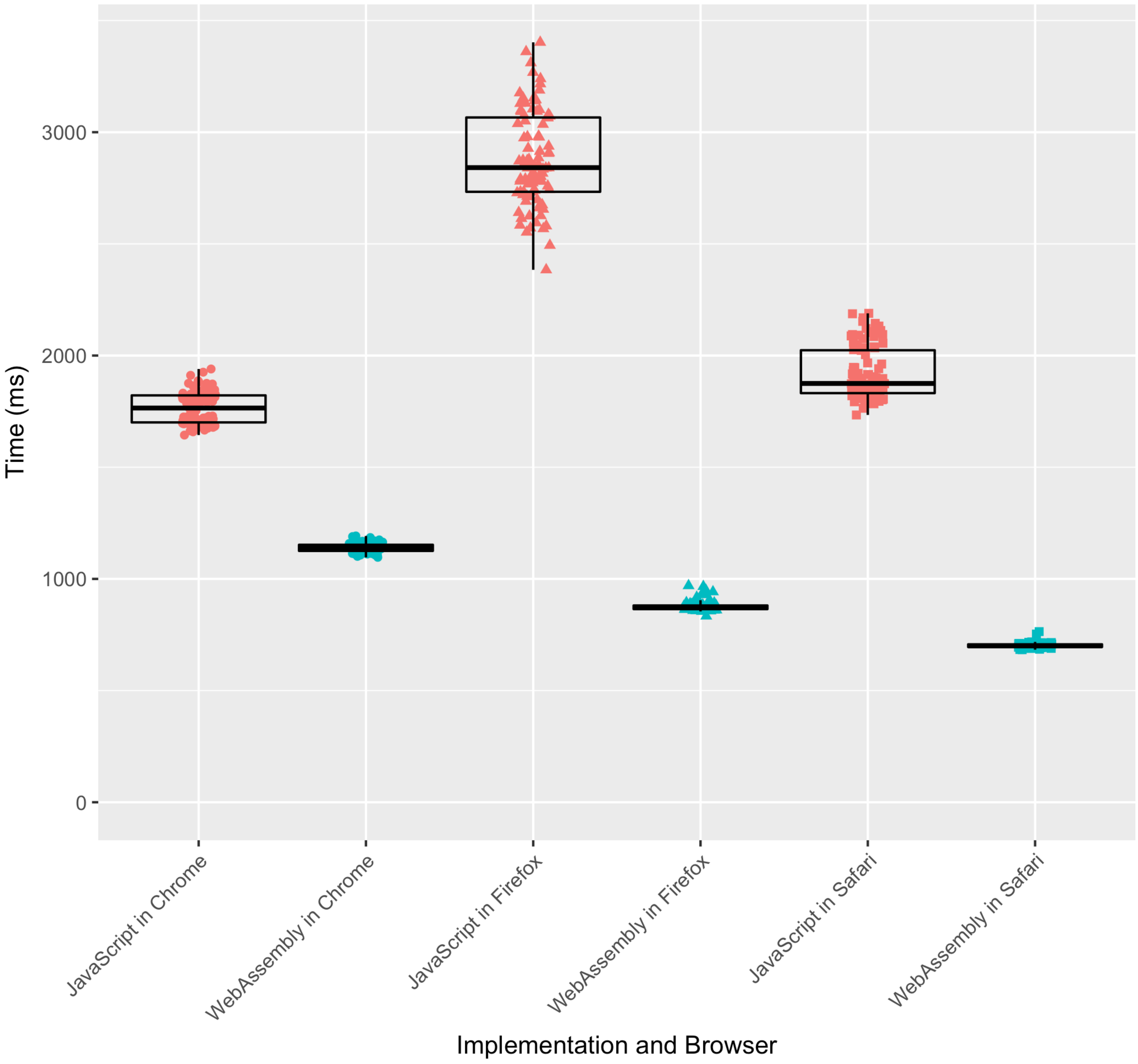
SMALL AND FAST

> IT'S COOL THAT WEBASSEMBLY IS FAST, BUT IT'S KILLER FEATURE IS ITS PREDICTABILITY.

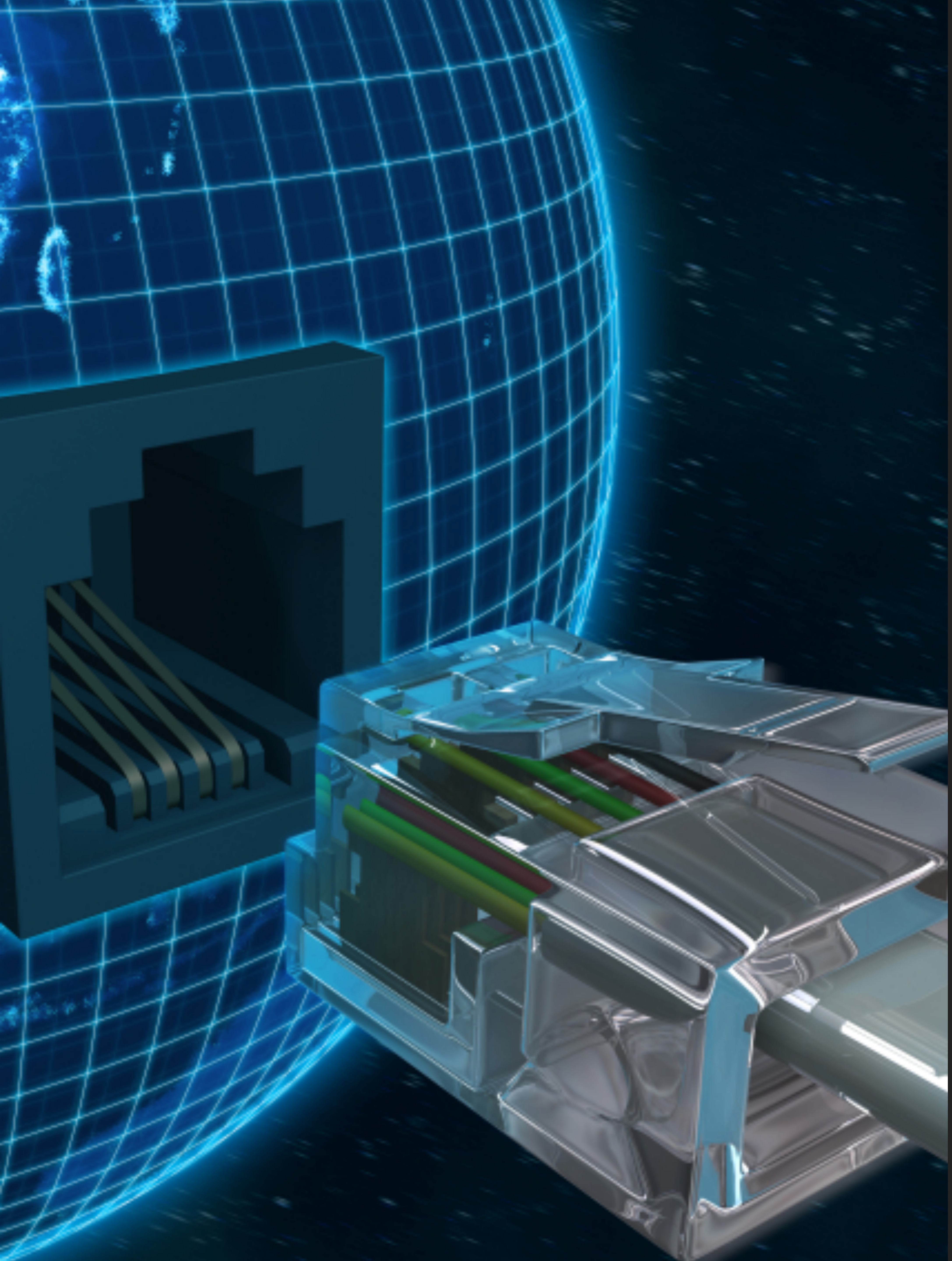
AG_DUBS

Set First Breakpoint

Scala.JS Source Map



BUT WHY???



WHAT IS THE

WEB PLATFORM?

WHAT IS A PLATFORM?

ISA (INSTRUCTION SET ARCHITECTURE)

RUNTIME

TOOLING

JAVA

ISA (INSTRUCTION SET ARCHITECTURE)

JAVA BYTE CODE

RUNTIME

JVM

TOOLING

JAVAC

X86 ON UNIX

ISA (INSTRUCTION SET ARCHITECTURE)

X86

RUNTIME

UNIX

TOOLING

SHELL, C, GCC

WEB PLATFORM

ISA (INSTRUCTION SET ARCHITECTURE)

????

RUNTIME

JAVASCRIPT, PLATFORM APIS

TOOLING

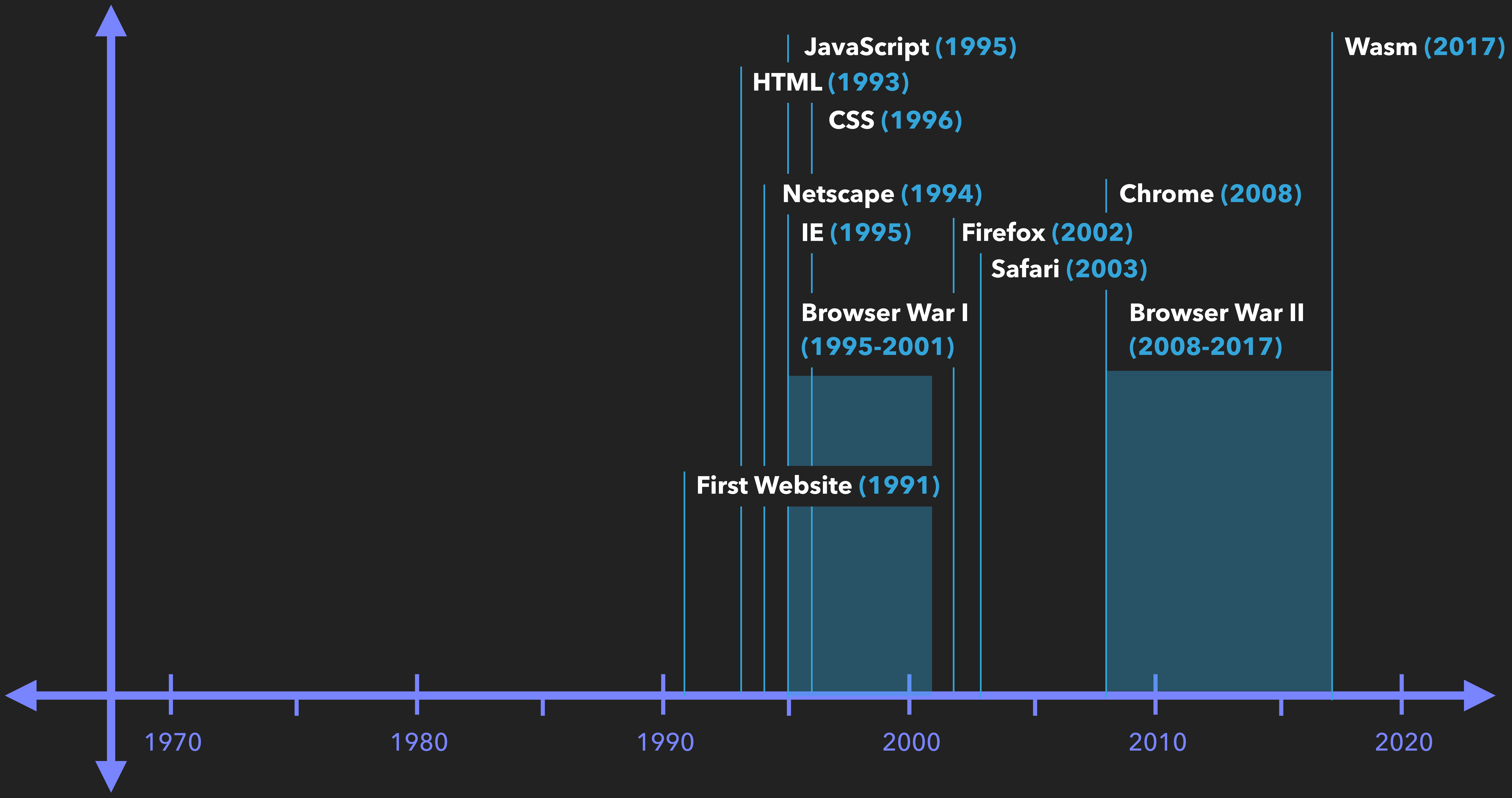
SO MUCH.

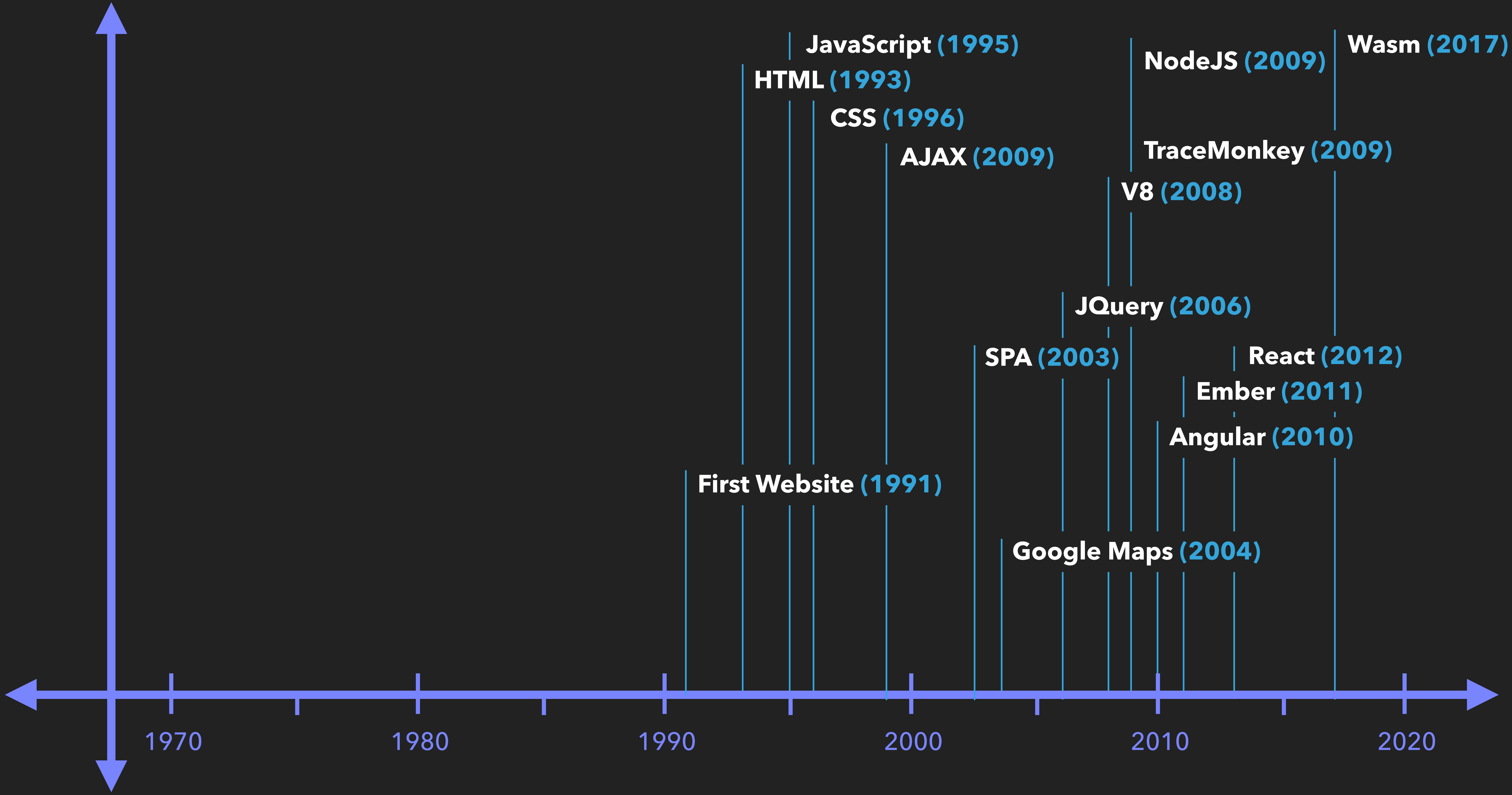
AN EXTREMELY BRIEF AND INCOMPLETE

HISTORY OF THE WEB PLATFORM

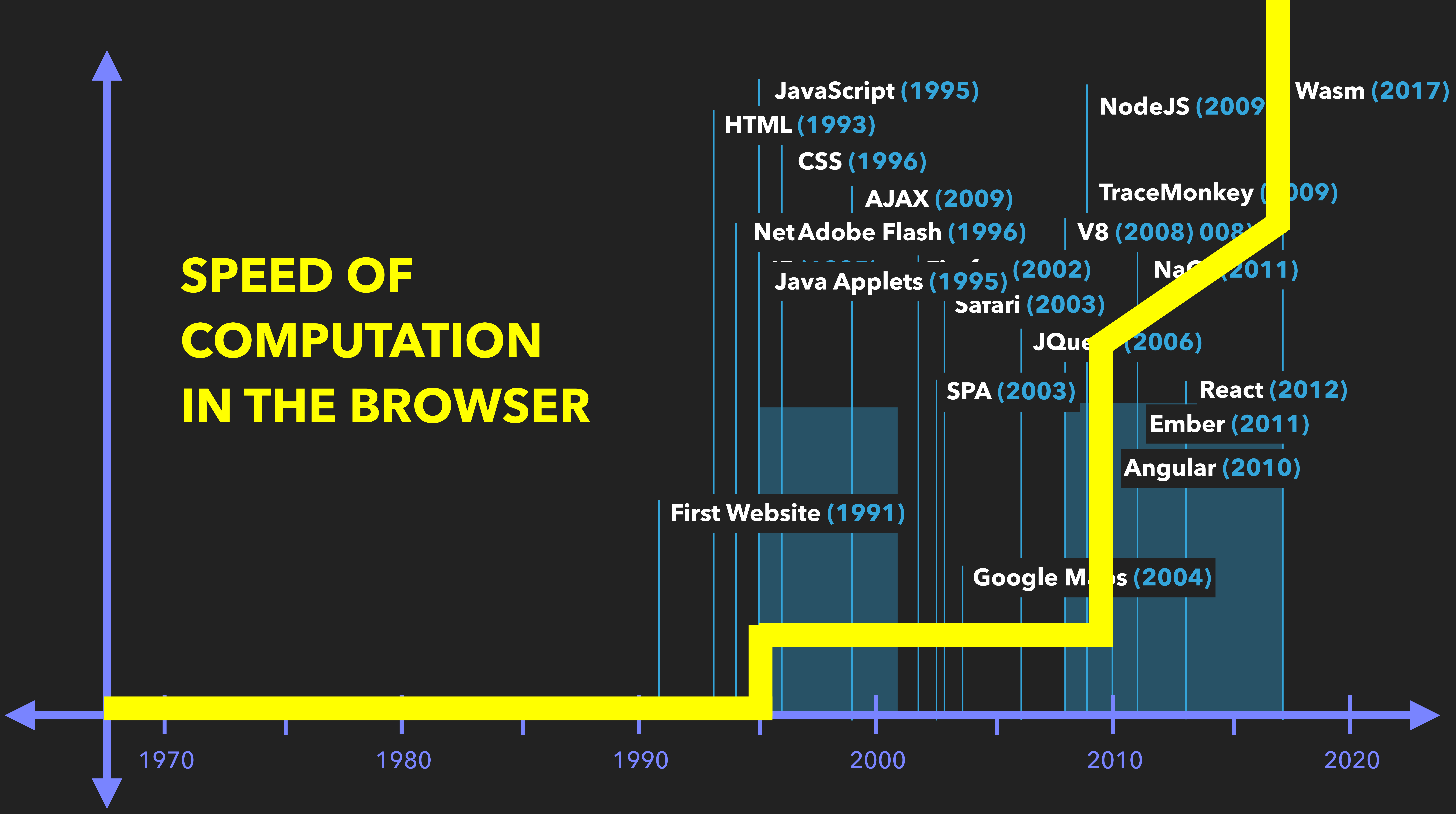
THE WEB PLATFORM IS UNUSUAL IN THAT IT WAS DEVELOPED OUT OF ORDER AND IT HAS, HISTORICALLY, HAD MISSING PIECES.

@steveklabnik

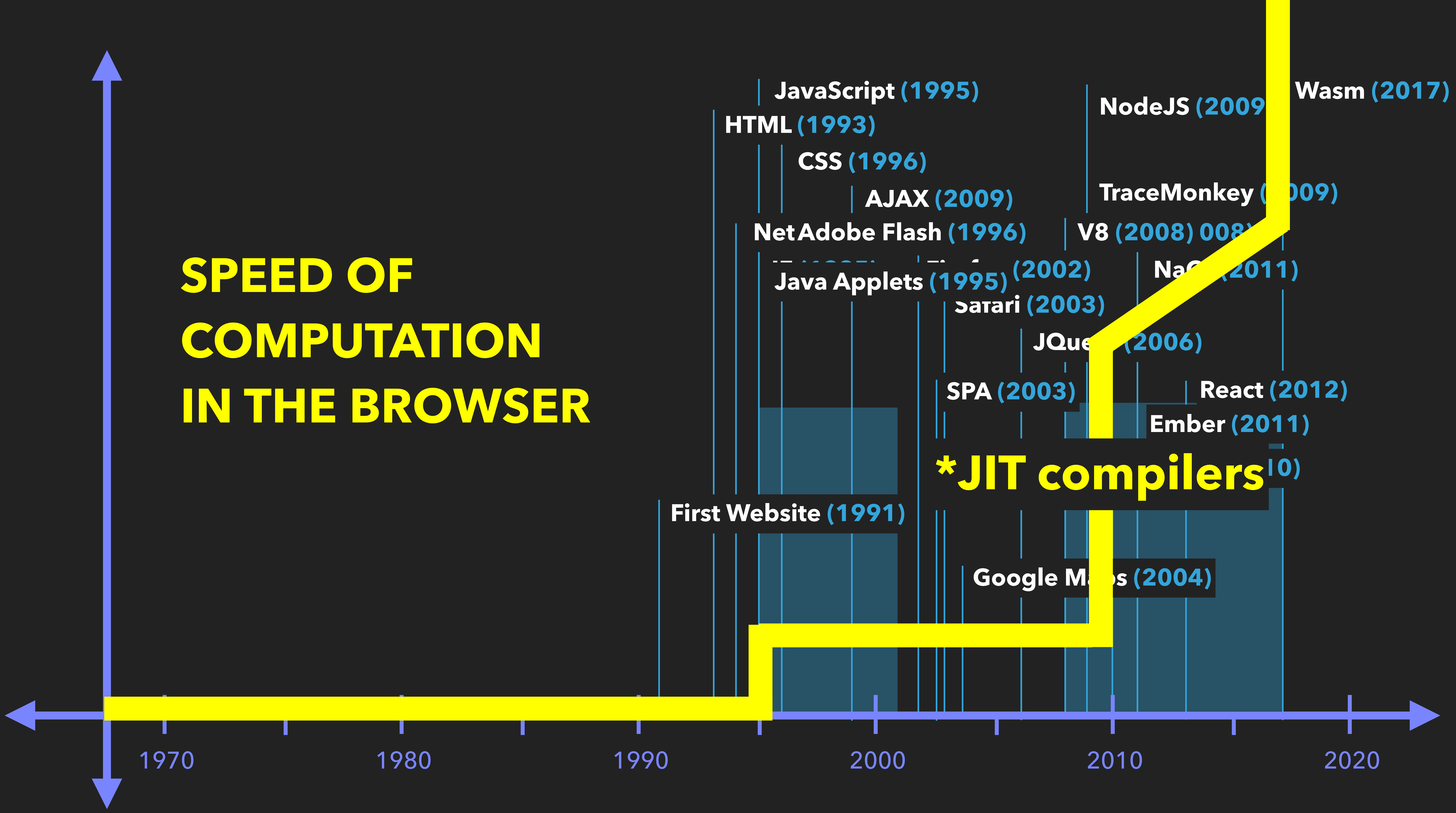

JavaScript (1995)
Wasm (2017)
HTML (1993)
CSS (1996)
Netscape (1994)
Chrome (2008)
IE (1995)
Firefox (2002)
Safari (2003)
Browser War I (1995-2001)
Browser War II (2008-2017)
First Website (1991)
1970
1980
1990
2000
2010
2020


JavaScript (1995)
HTML (1993)
CSS (1996)
AJAX (2009)
NodeJS (2009)
Wasm (2017)
TraceMonkey (2009)
V8 (2008)
JQuery (2006)
SPA (2003)
React (2012)
Ember (2011)
Angular (2010)
First Website (1991)
Google Maps (2004)
1970
1980
1990
2000
2010
2020

SPEED OF COMPUTATION IN THE BROWSER



SPEED OF COMPUTATION IN THE BROWSER



JAVASCRIPT AS AN ISA?

I WAS NOT DESIGNED FOR THIS TASK.

JavaScript

WEB PLATFORM

ISA (INSTRUCTION SET ARCHITECTURE)

WEB ASSEMBLY

RUNTIME

JAVASCRIPT, PLATFORM APIS

TOOLING

SO MUCH.

INTERPRETERS AND COMPILERS

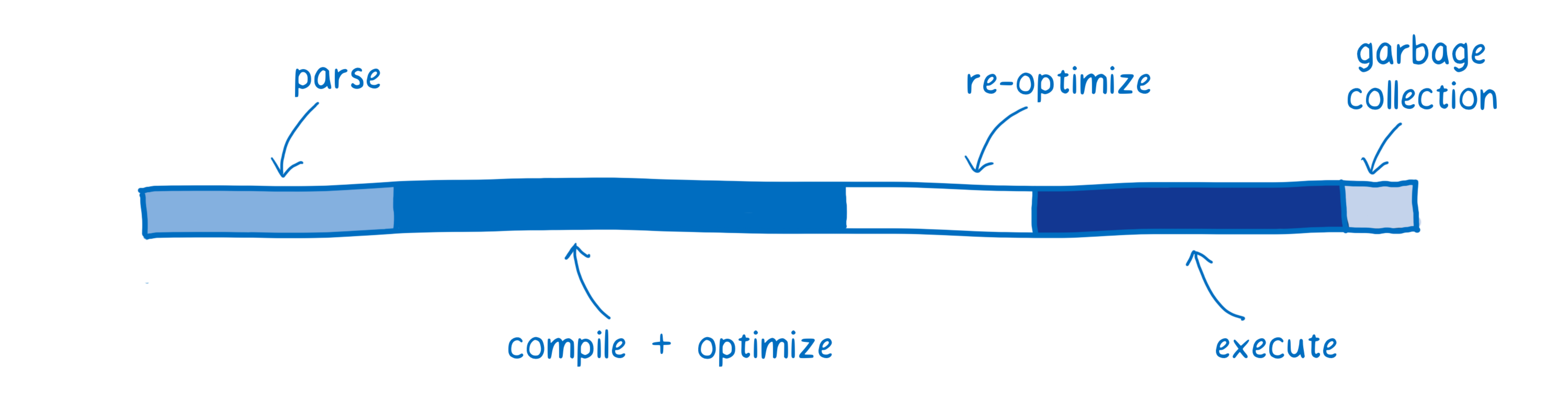
BROWSER ENGINES ARE FRIGGIN MAGICAL OK?

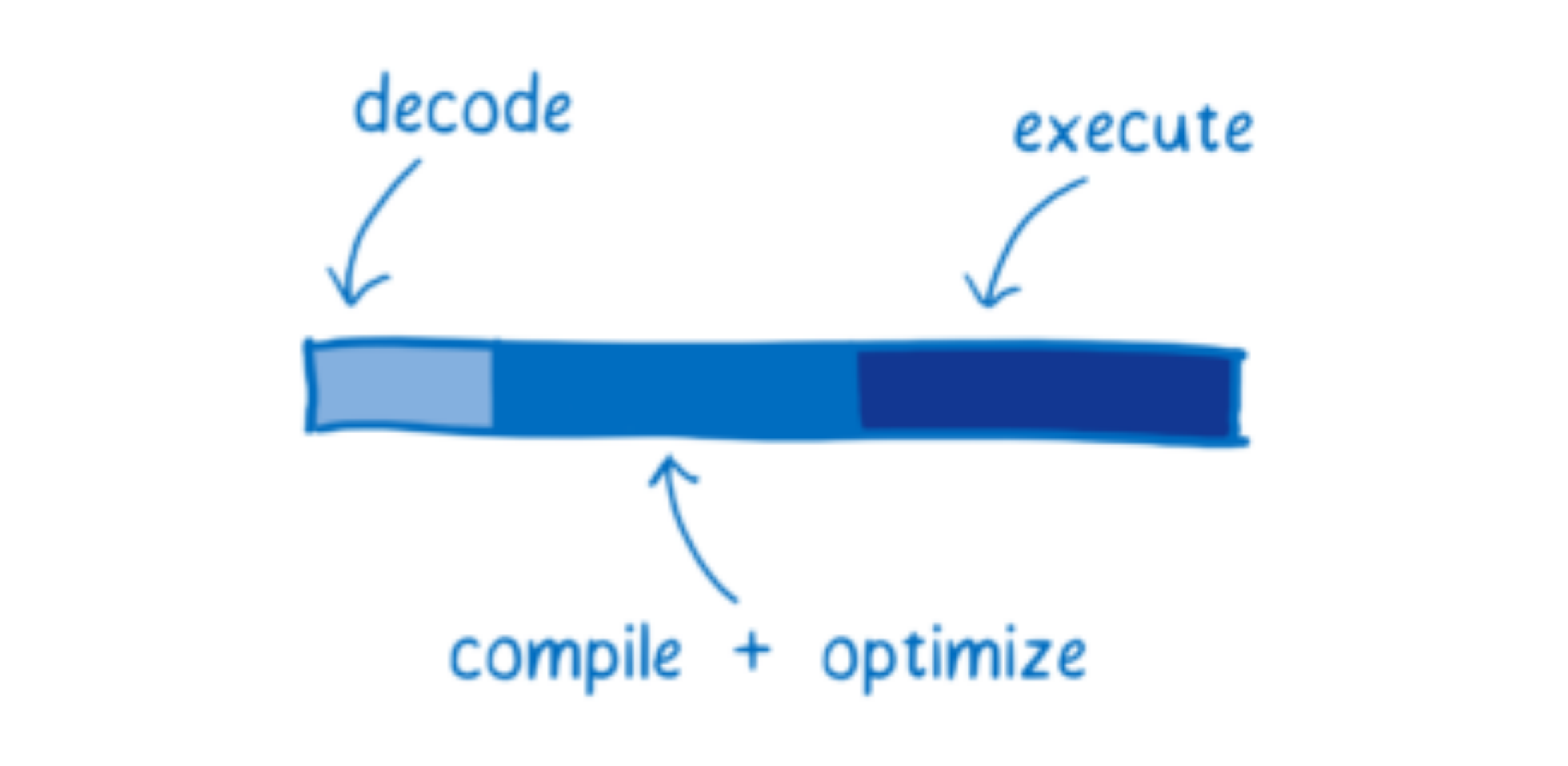
AG_DUBS

HOW DO BROWSERS EVEN?

WHAT IS A

JIT?





WHY IS WEBASSEMBLY FAST?

Lin Clark. https://hacks.mozilla.org/2017/02/what-makes-webassembly-fast/

HOW DOES COMPUTATION ON THE WEB WORK NOW

SERVER ORIENTED

SERVER ORIENTED

SERVER ORIENTED

SERVER ORIENTED

SERVER ORIENTED

SERVER ORIENTED

WITH CACHE

SERVER ORIENTED

WITH CACHE

SERVER ORIENTED

WITH CACHE

SERVER ORIENTED

WITH CACHE

SERVER ORIENTED

WITH CACHE

SERVER ORIENTED

WITH CACHE

SERVER ORIENTED

WITH CACHE

SERVER ORIENTED

WITH CACHE

SERVER ORIENTED

WITH CACHE

CLIENT ORIENTED

CLIENT ORIENTED

CLIENT ORIENTED

CLIENT ORIENTED

CLIENT ORIENTED

CLIENT ORIENTED

CLIENT ORIENTED

WITH CACHE

CLIENT ORIENTED

WITH CACHE

CLIENT ORIENTED

WITH CACHE

CLIENT ORIENTED

WITH CACHE

CLIENT ORIENTED

WITH CACHE

CLIENT ORIENTED

WITH CACHE

CLIENT ORIENTED

WITH CACHE

CLIENT ORIENTED

WITH CACHE

HOW COULD COMPUTATION ON THE WEB WORK

https://gimme.pizza.now

EDGE COMPUTE

https://gimme.pizza.now

EDGE COMPUTE

https://gimme.pizza.now

EDGE COMPUTE

https://gimme.pizza.now

EDGE COMPUTE

EDGE COMPUTE

https://gimme.pizza.now

EDGE COMPUTE

BUT WHY???

WEBASSEMBLY AND JS AS A UNIFIED WEB PLATFORM MAKES OFFERING EDGE COMPUTE MORE ECONOMICALLY FEASIBLE

BUT WHY???

LESS LATENCY MEANS MORE ACCESS FOR MORE PEOPLE

HOW CAN I USE WEB ASSEMBLY TODAY?

WHAT IS

EMSCRIPTEN?

EMSCRIPTEN

- LLVM BACKEND THAT MAKES ASM.JS OR WEBASSEMBLY
- USED BY PNACL
- USED BY UNITY, UNREAL, DOSBOX
- C, C++, RUST

HTTPS://EMSCRIPTEN.ORG/

SHOULD I

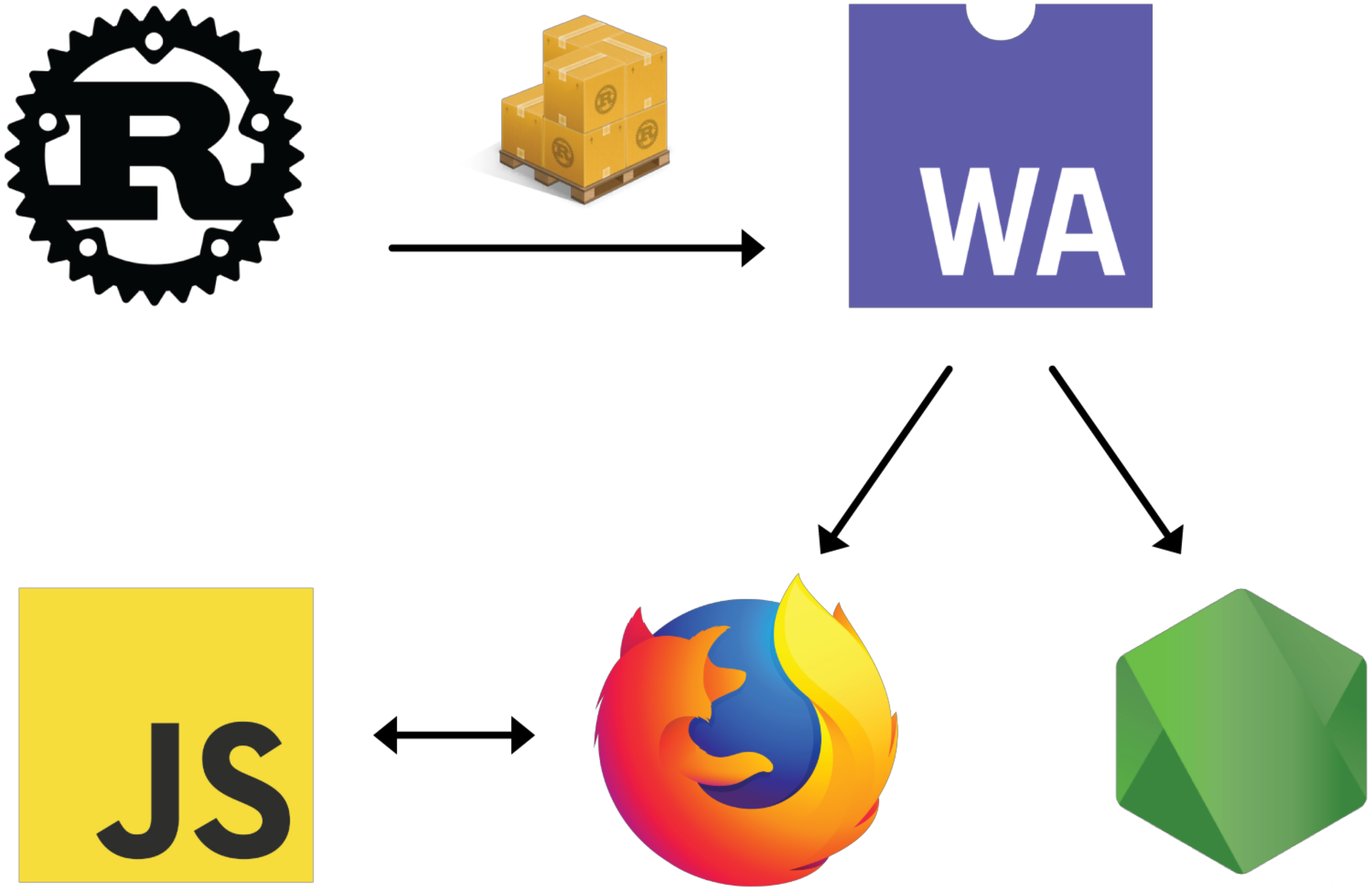
REWRITE IT IN RUST?

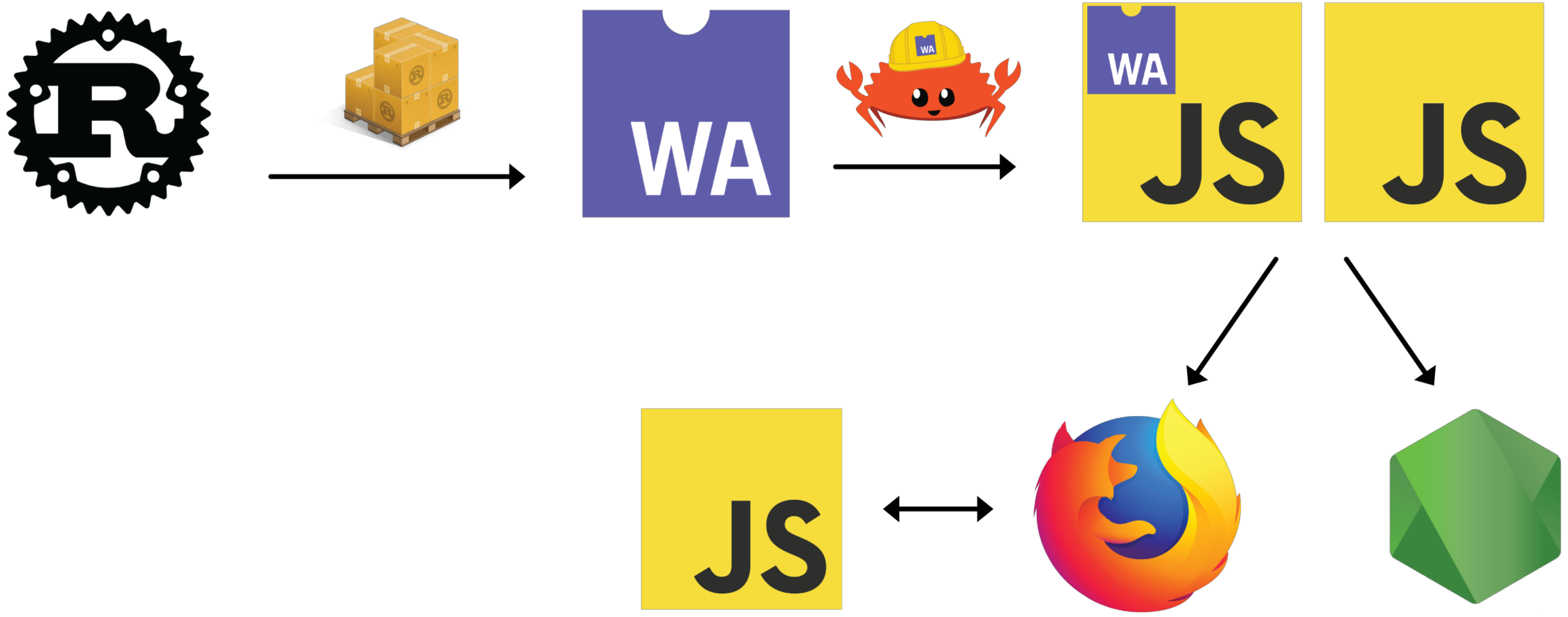
YES!

RUSTWASM

- RUST COMPILER (W/LLVM) TARGETING WEBASSEMBLY
- GENERATES JAVASCRIPT BINDINGS TO WASM
- INTEROPERATES WITH JS TOOLS: NPM, WEBPACK, GREENKEEPER
- HEAVY FOCUS ON DEVELOPER TOOLS AND EXPERIENCE

HTTPS://RUSTWASM.GITHUB.IO/


WA
JS


WA
WA
WA
JS
JS
JS

HTTPS://RUSTWASM.GITHUB.IO/WASM-PACK

DEMO TIME

WHAT ABOUT

JAVASCRIPT?

CAN JS COMPILE TO WEBASSEMBLY?

- ANYTHING CAN COMPILE TO WEBASSEMBLY
- ONE MUST COMPILE THE RUNTIME IN ADDITION TO THEIR PROGRAM WHEN USING AN INTERPRETED LANGUAGE

WHAT IS

ASSEMBLYSCRIPT?

ASSEMBLYSCRIPT

- SUBSET OF TYPESCRIPT COMPILES WEBASSEMBLY VIA BINARYEN
- LOWEST BARRIER TO ENTRY FOR JS DEVELOPERS

HTTPS://ASSEMBLYSCRIPT.ORG/

EVERY PLATFORM WILL BECOME THE WEB PLATFORM.

AG_DUBS

THE DREAM OF

UNIVERSAL BINARIES

I WANT A WEBASSEMBLY OPERATING SYSTEM AND I WANT IT YESTERDAY.

AG_DUBS

SOON, "WEB APPLICATION" DOES NOT NEED TO BE SYNONYMOUS WITH "NETWORKED APPLICATION" OR "WEB PAGE" OR ANYTHING ELSE WE ASSOCIATE WITH THE WEB PLATFORM TODAY.

@steveklabnik

“NO INSTALL NEEDED”

WE SHOULD BE DESIGNING LANGUAGES TODAY FOR THE WEBASSEMBLY TOMORROW

DEVELOPER EXPERIENCE MATTERS

RUNTIMES ARE EXPENSIVE

RUNTIMES ARE CONFUSING

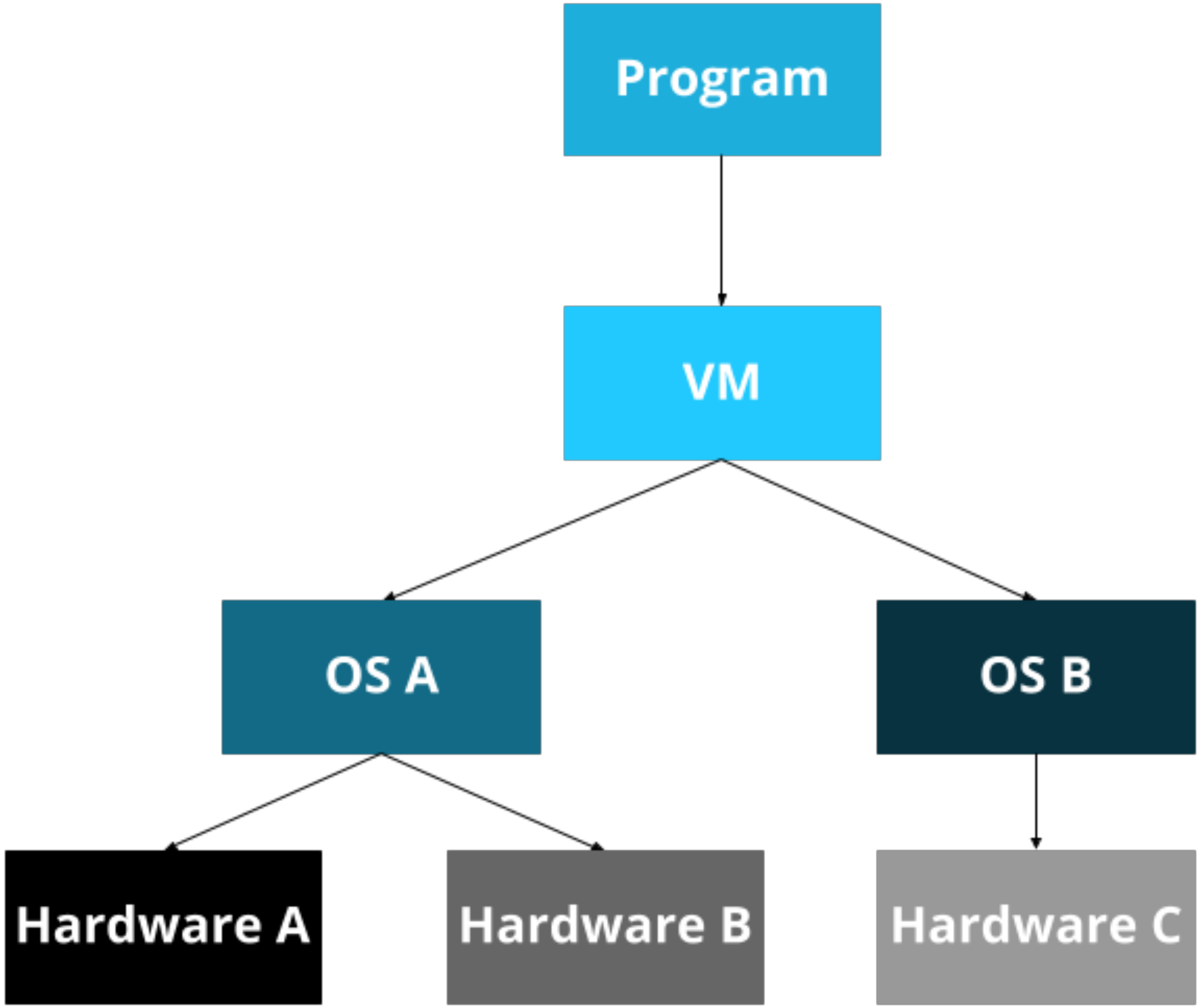
CODE FOR HUMANS IS DIFFERENT THAN CODE FOR COMPUTERS

WE CAN MAKE PROGRAMMING AND PROGRAMS MORE ACCESSIBLE WITH THE POWER OF COMPILERS

I WELCOME OUR COMPILED FUTURE

WEBASSEMBLY IS A LANGUAGE THAT WILL REDEFINE THE JOBS WE DO AND THE LANGUAGES WE DO THEM IN

AG_DUBS



Program

VM

OS A

OS B

Hardware A

Hardware B

Hardware C

SUPPORT MORE HARDWARE WITH LESS

LESS LATENCY MEANS MORE ACCESS FOR MORE PEOPLE

WEBASSEMBLY MEANS A MORE DEMOCRATIC INTERNET

THANKS!

@AG_DUBS