0’dan 100’e

JavaScript Motorlarının Çalışma Prensipleri

Pazartesi, 26.02.2018 / Frontend İstanbul

Oğuz Kılıç | oguzz.kilic@gmail.com

Kimdir?

Oğuz KILIÇ

Frontend Developer

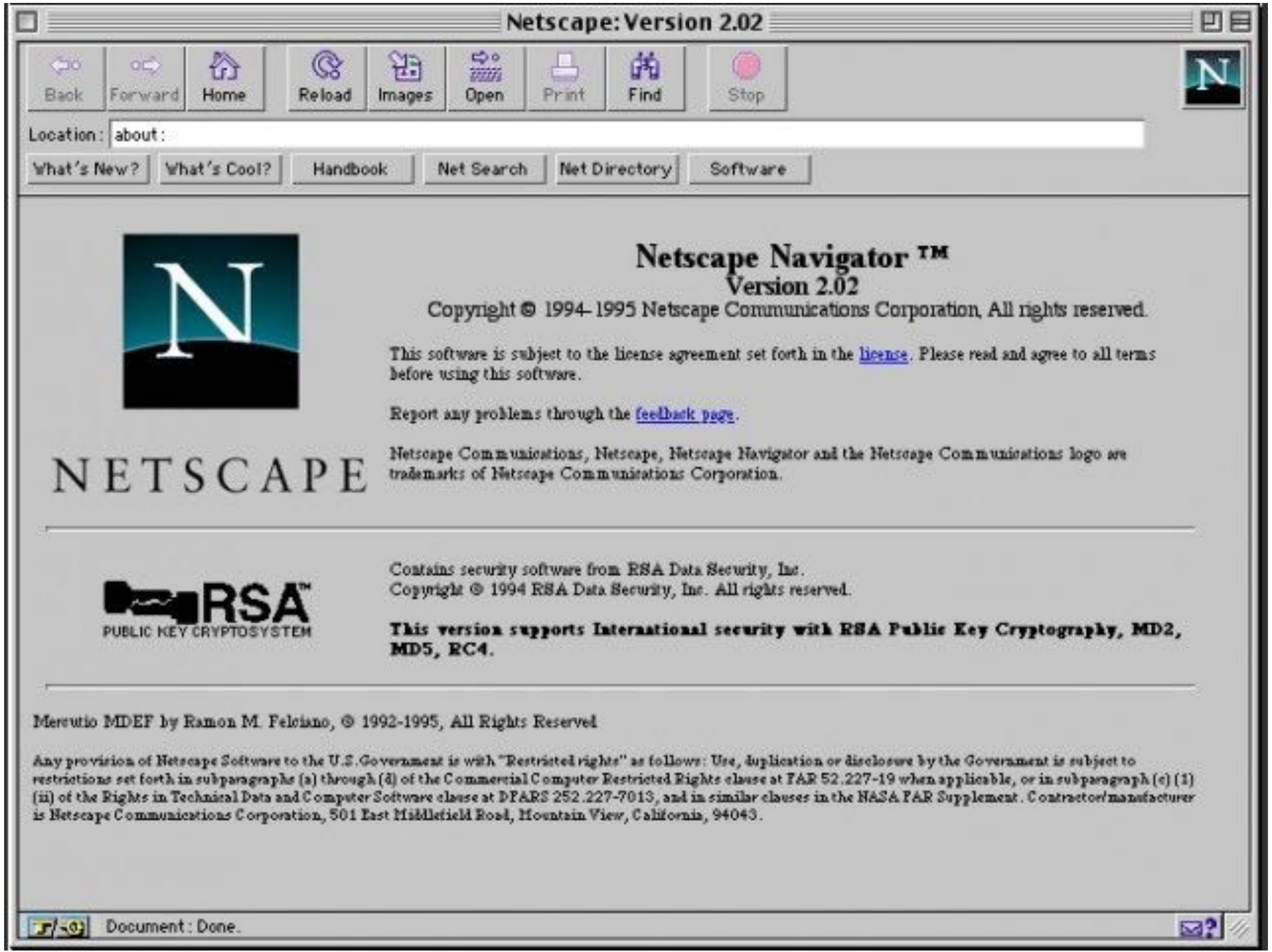
Kısa Bilgi - JavaScript'in Tarihi

Brendan Eich

Creator of JavaScript Language

Mocha

1995



Netscape Navigator ™
Version 2.02
Copyright © 1994-1995 Netscape Communications Corporation, All rights reserved.

This software is subject to the license agreement set forth in the license. Please read and agree to all terms before using this software.

Report any problems through the feedback page.

Netscape Communications, Netscape, Netscape Navigator and the Netscape Communications logo are trademarks of Netscape Communications Corporation.

Contains security software from RSA Data Security, Inc.
Copyright © 1994 RSA Data Security, Inc. All rights reserved.

This version supports International security with RSA Public Key Cryptography, MD2, MD5, RC4.

Mercutio MDEF by Ramon M. Felciano, © 1992-1995, All Rights Reserved

Any provision of Netscape Software to the U.S.Government is with "Restricted rights" as follows: Use, duplication or disclosure by the Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Restricted Rights clause at FAR 52.227-19 when applicable, or in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clauses in the NASA FAR Supplement. Contractor/manufacturer is Netscape Communications Corporation, 501 East Middlefield Road, Mountain View, California, 94043.

Kısa Bilgi - JavaScript’in Tarihi

LiveScript

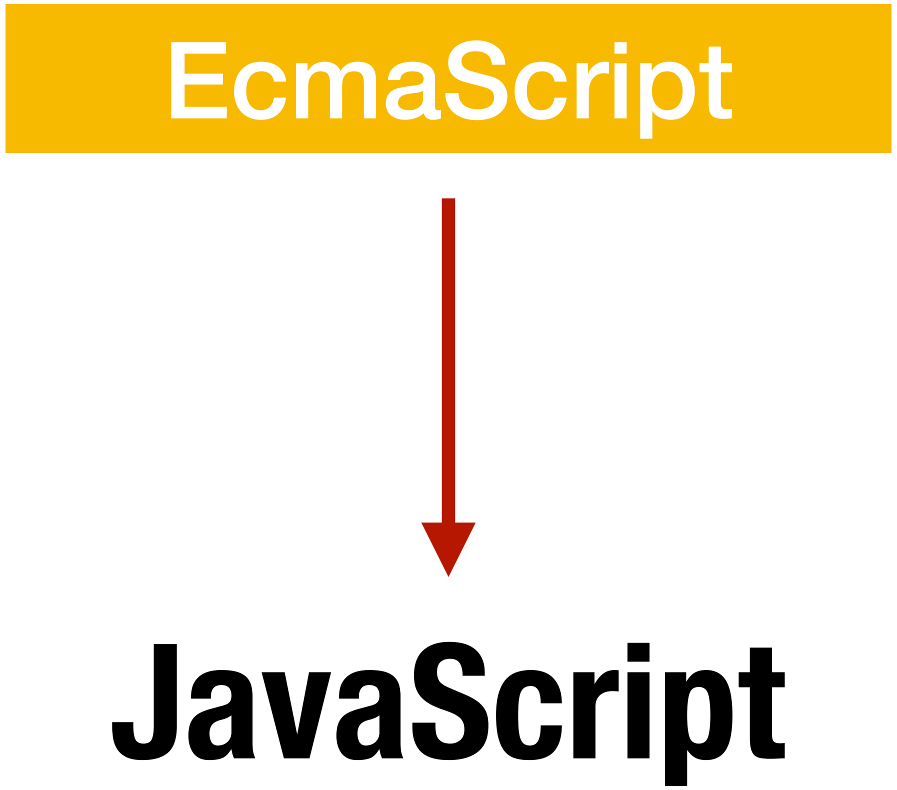
Kısa Bilgi - JavaScript’in Tarihi

JavaScript

Kısa Bilgi - JavaScript’in Tarihi

EcmaScript

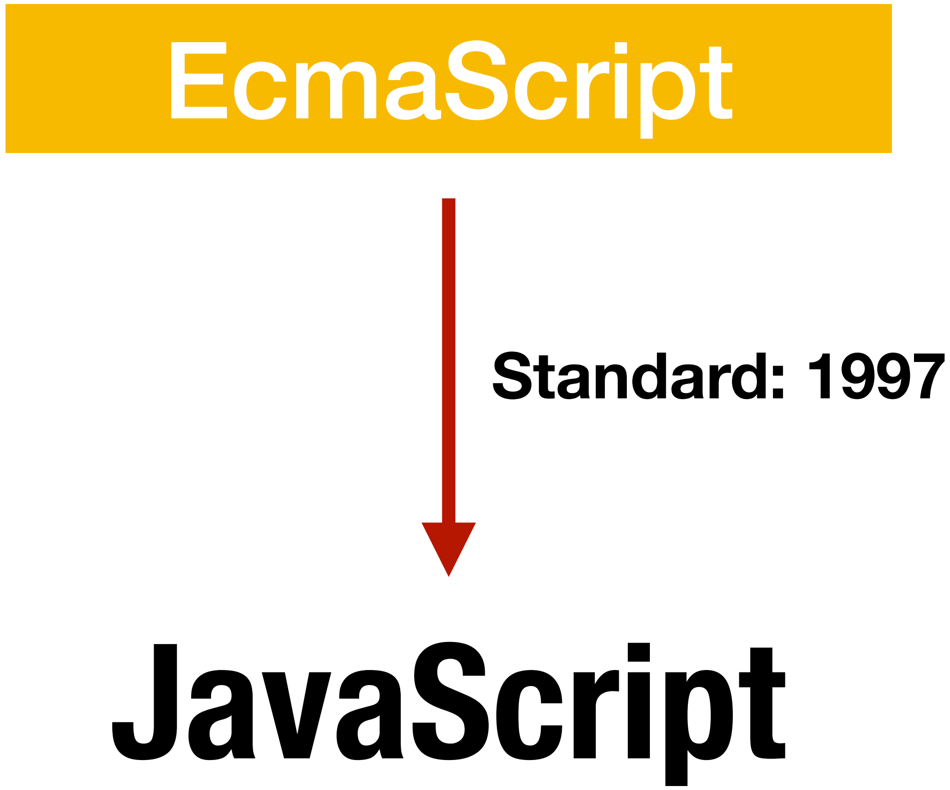
Kısa Bilgi - JavaScript’in Tarihi



EcmaScript

JavaScript

Kısa Bilgi - JavaScript’in Tarihi

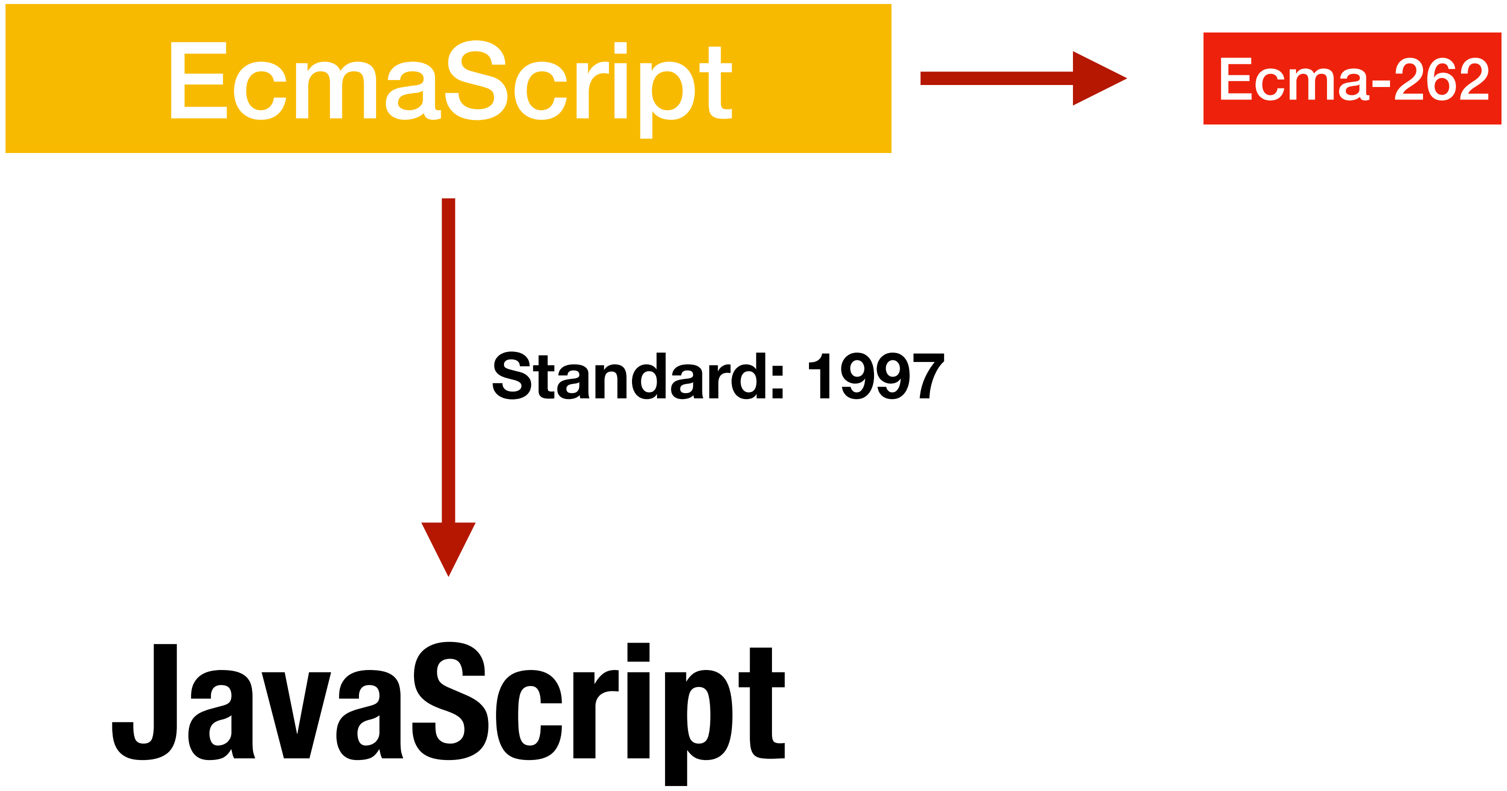


EcmaScript

Standard: 1997

JavaScript

Kısa Bilgi - JavaScript'in Tarihi



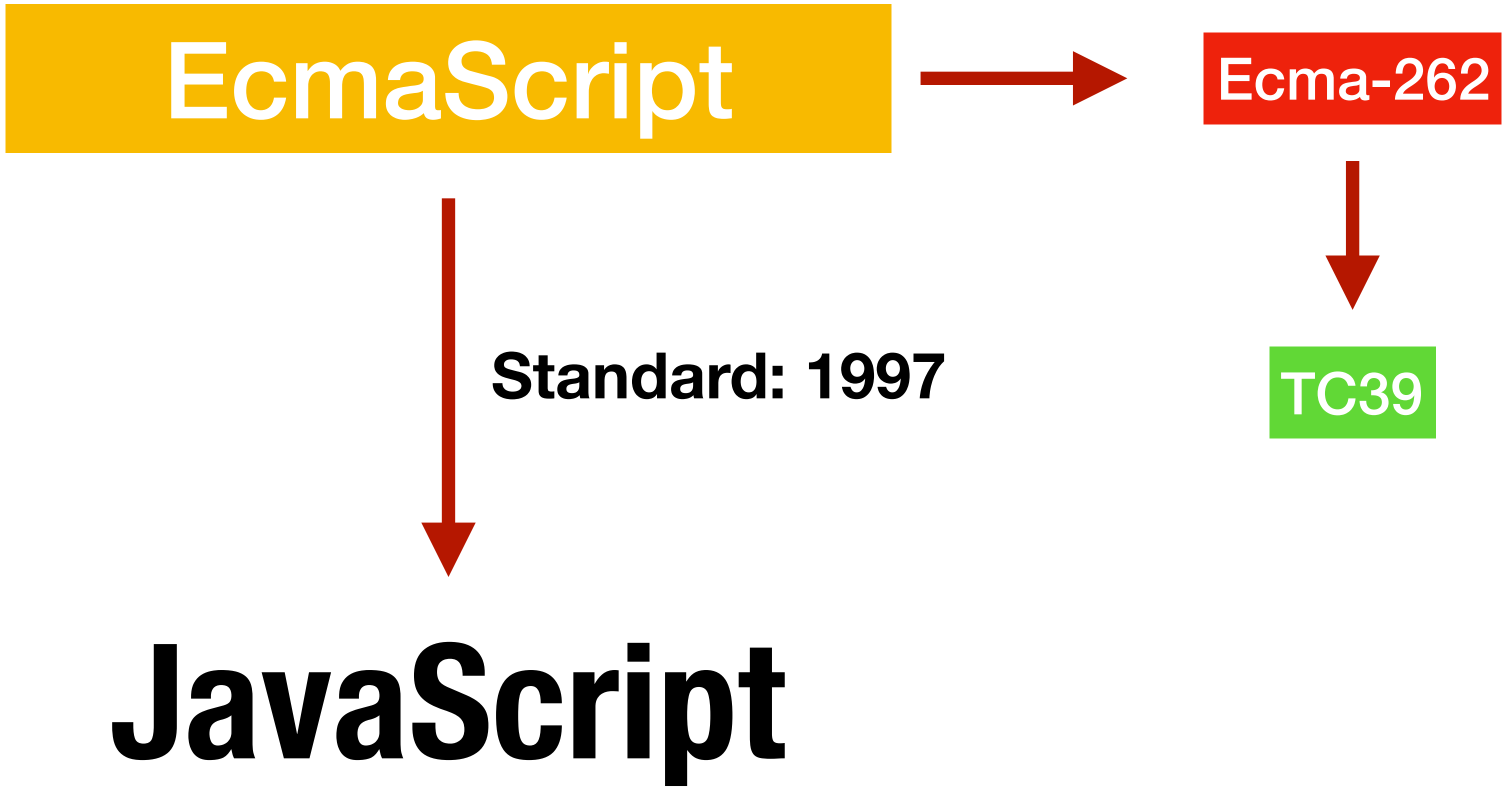
EcmaScript

Ecma-262

Standard: 1997

JavaScript

Kısa Bilgi - JavaScript’in Tarihi



EcmaScript

Ecma-262

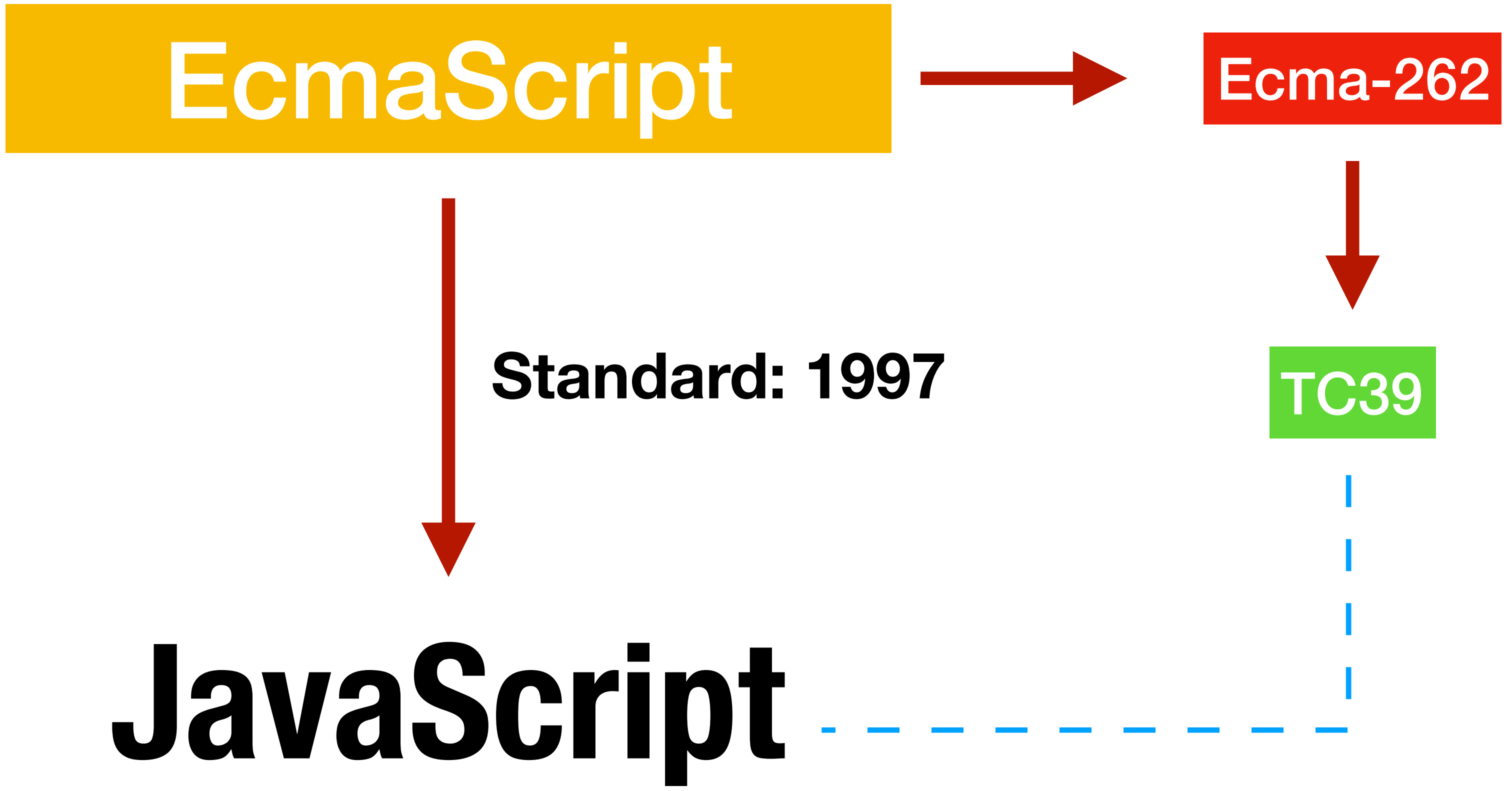
TC39

Standard: 1997

JavaScript

Kısa Bilgi - JavaScript'in Tarihi

http://ecma-international.org/memento/TCs&TGs.htm



EcmaScript

Ecma-262

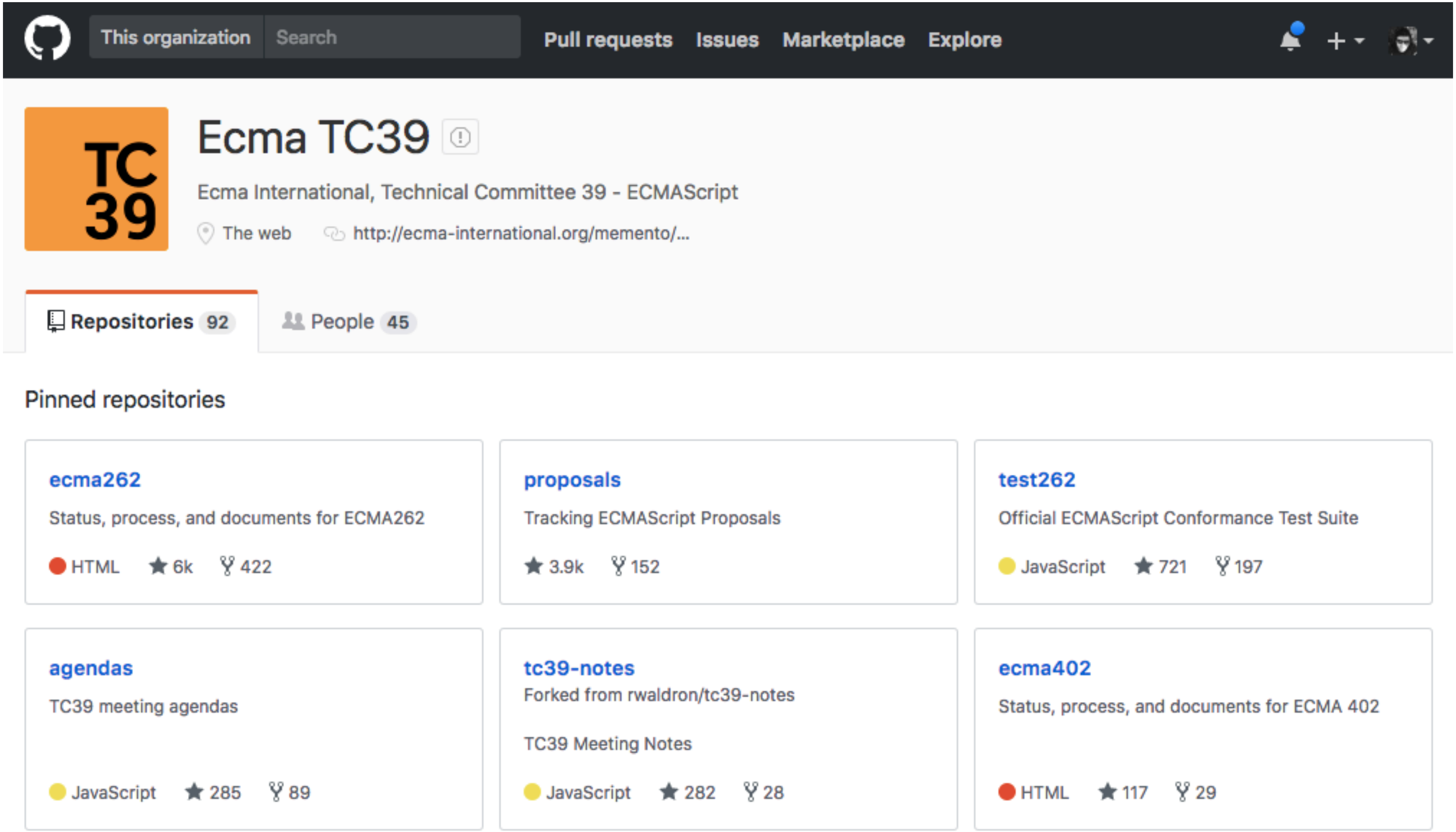
Standard: 1997

TC39

JavaScript

Kısa Bilgi - JavaScript’in Tarihi

https://github.com/tc39



Ecma TC39

Ecma International, Technical Committee 39 - ECMAScript

The web http://ecma-international.org/memento/...

Repositories 92 People 45

Pinned repositories

ecma262
Status, process, and documents for ECMA262
● HTML ★ 6k 422

proposals
Tracking ECMAScript Proposals
★ 3.9k 152

test262
Official ECMAScript Conformance Test Suite
● JavaScript ★ 721 197

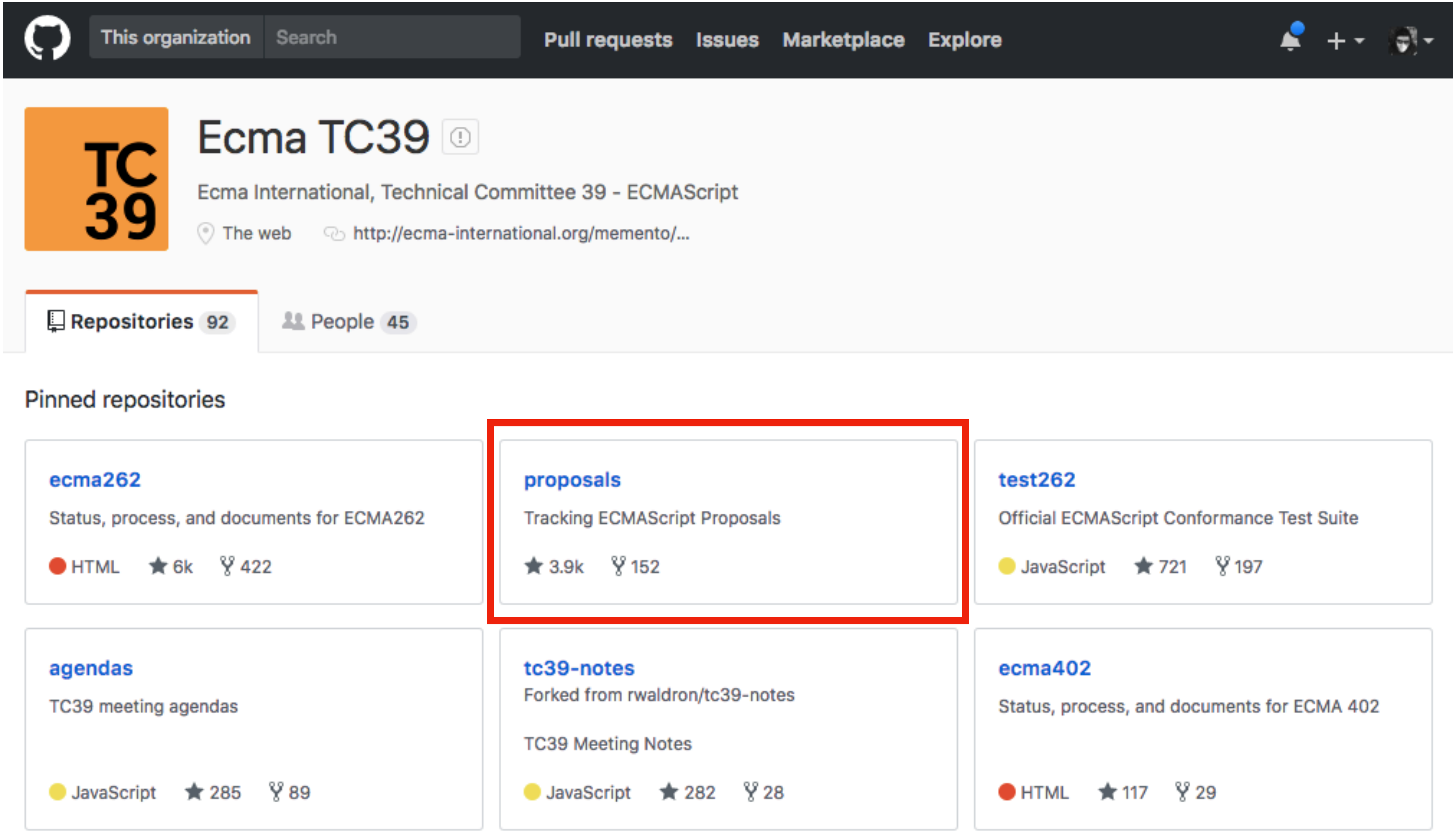
agendas
TC39 meeting agendas
● JavaScript ★ 285 89

tc39-notes
Forked from rwaldron/tc39-notes
TC39 Meeting Notes
● JavaScript ★ 282 28

ecma402
Status, process, and documents for ECMA 402
● HTML ★ 117 29

Kısa Bilgi - JavaScript’in Tarihi

https://github.com/tc39



Ecma TC39

Ecma International, Technical Committee 39 - ECMAScript

The web   http://ecma-international.org/memento/...

Repositories 92   People 45

Pinned repositories

ecma262
Status, process, and documents for ECMA262
● HTML ★ 6k ⑂ 422

proposals
Tracking ECMAScript Proposals
★ 3.9k ⑂ 152

test262
Official ECMAScript Conformance Test Suite
● JavaScript ★ 721 ⑂ 197

agendas
TC39 meeting agendas
● JavaScript ★ 285 ⑂ 89

tc39-notes
Forked from rwaldron/tc39-notes
TC39 Meeting Notes
● JavaScript ★ 282 ⑂ 28

ecma402
Status, process, and documents for ECMA 402
● HTML ★ 117 ⑂ 29

Host Environment

BROWSER

Host Environment

Browser

Goal

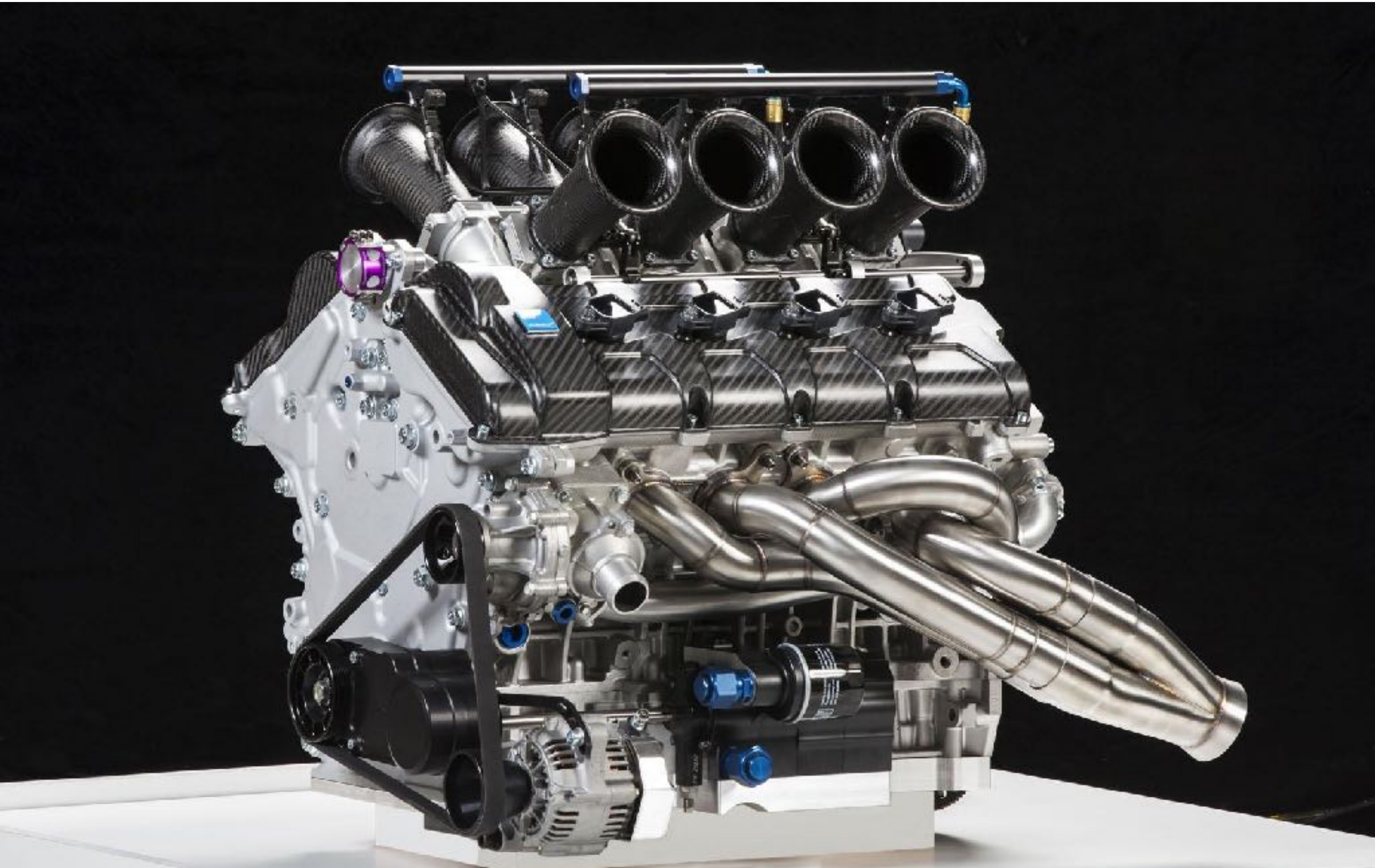
you want to tell the computer what to do

Problem

you and the computer speak different languages

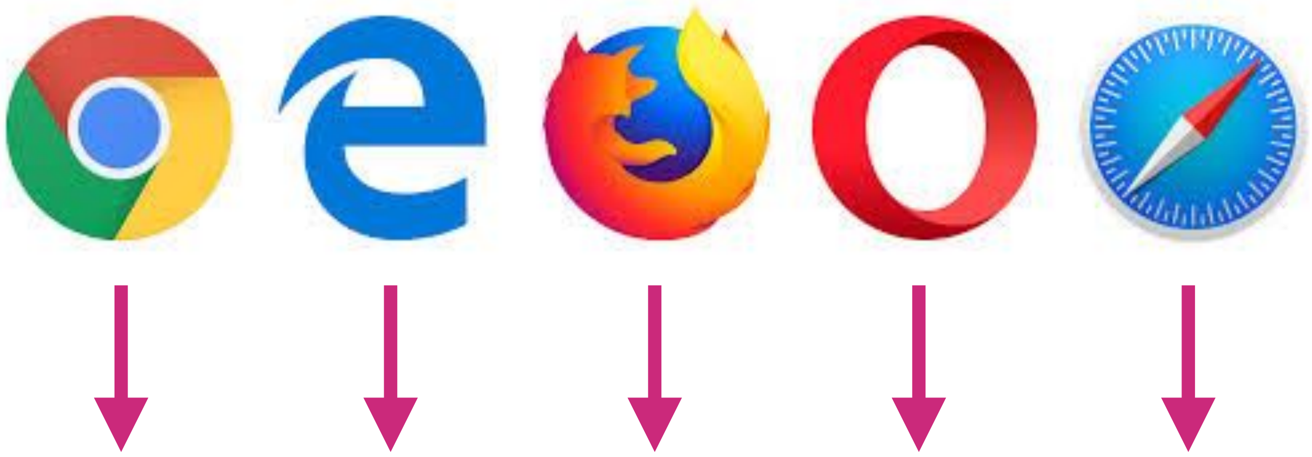
Host Environment
Engine

Host Environment
Engine



Host Environment

Engines



V8 Chakra S.Monkey V8 KJS

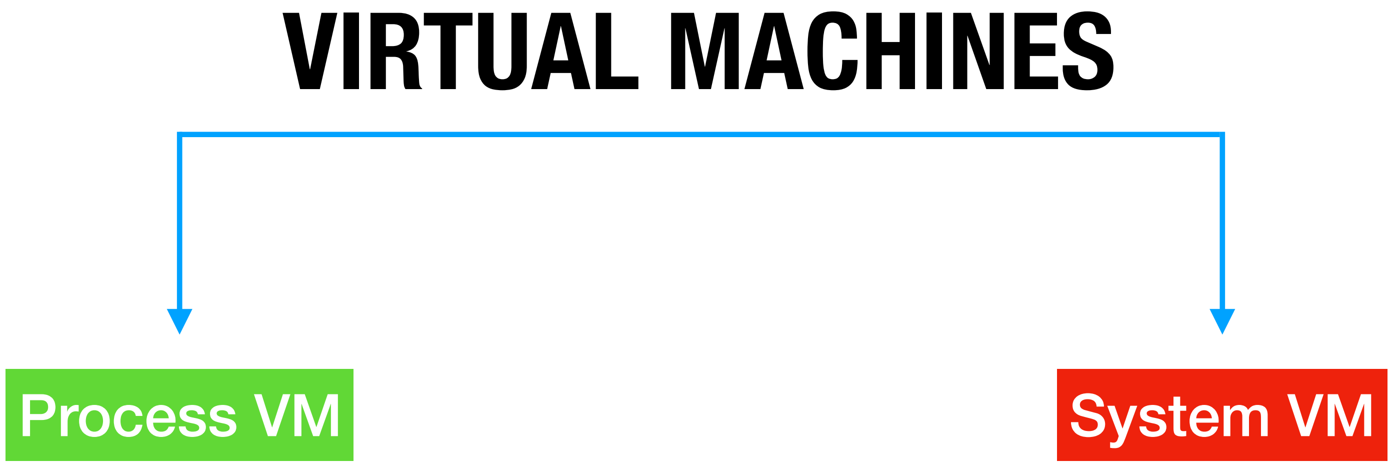
JavaScript Engines

Type

VIRTUAL MACHINES

JavaScript Engines

Type



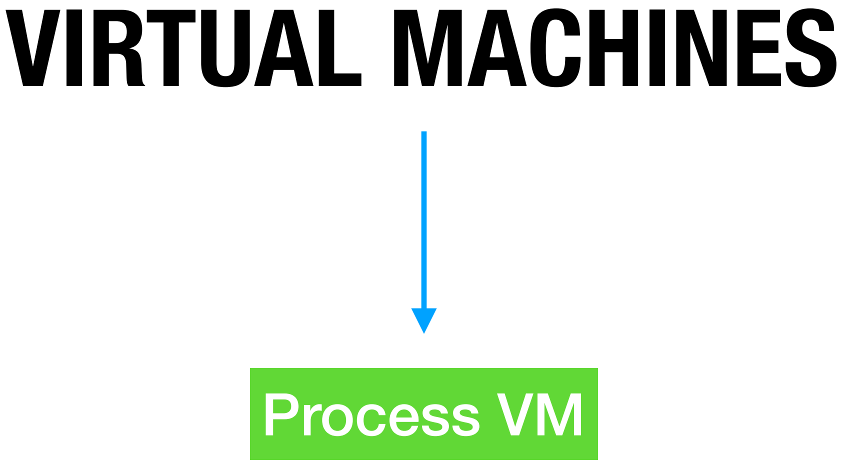
VIRTUAL MACHINES

Process VM

System VM

JavaScript Engines

Type



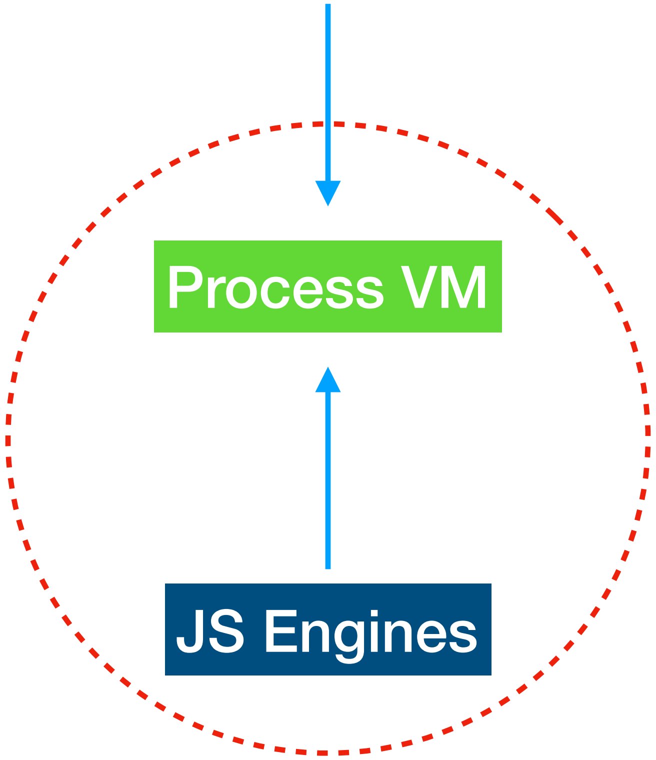
VIRTUAL MACHINES

Process VM

JavaScript Engines

Type

VIRTUAL MACHINES



Process VM

JS Engines

JavaScript Engines

Chrome V8 Engine

Launch with Chrome in 2008

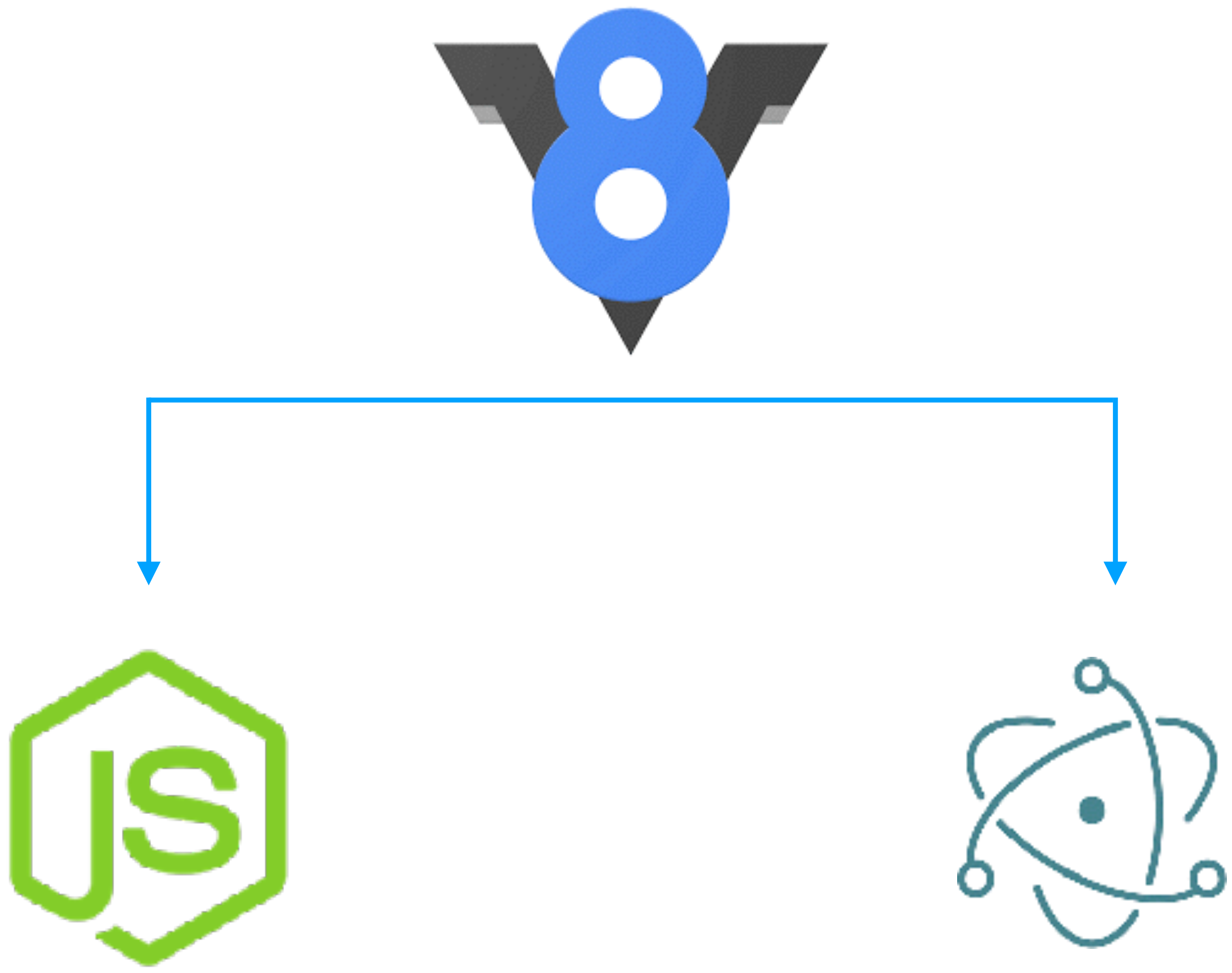
JavaScript Engines

Chrome V8 Engine

C++

JavaScript Engines

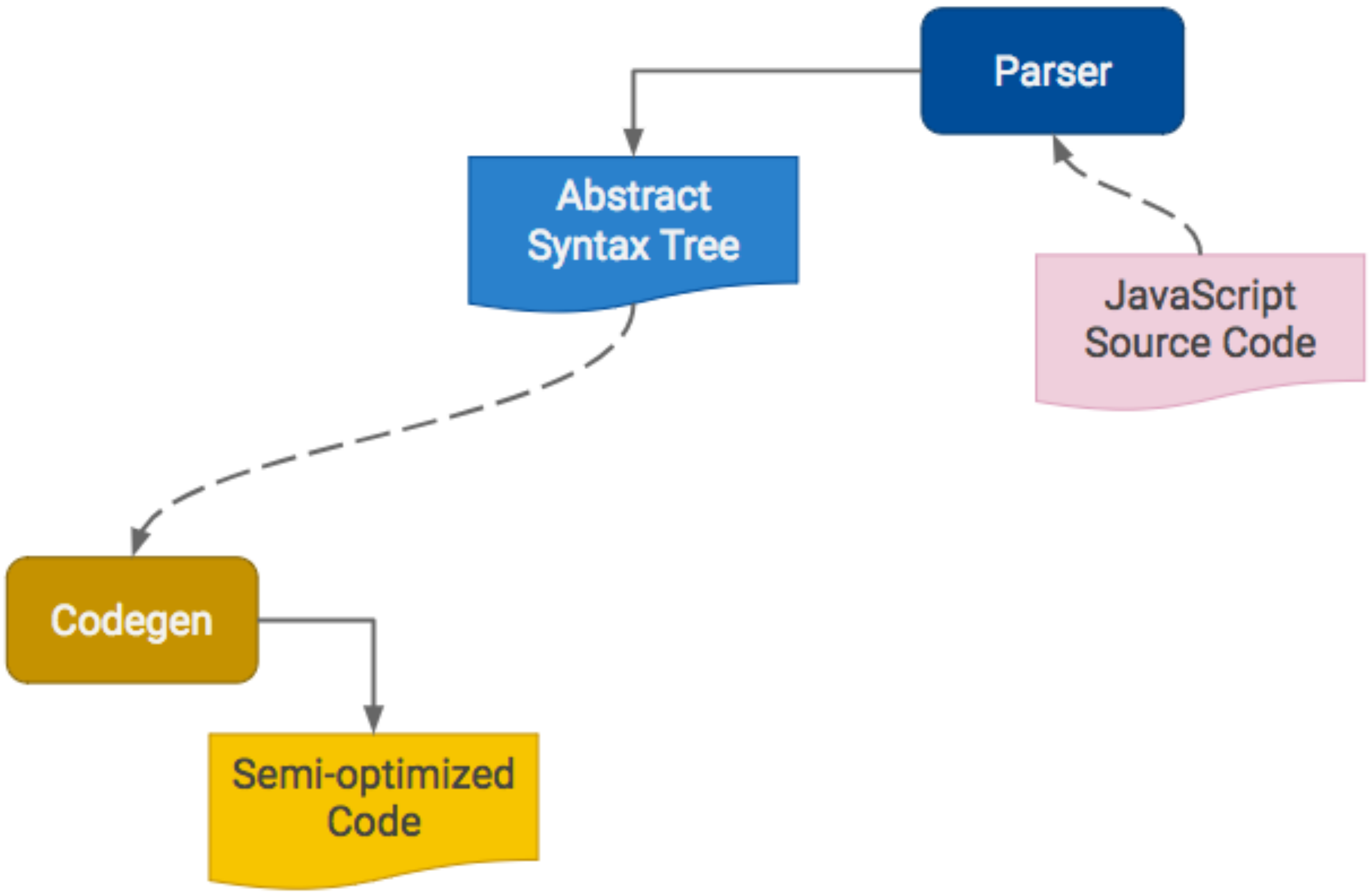
Chrome V8 Engine



V8 Compiler Pipeline

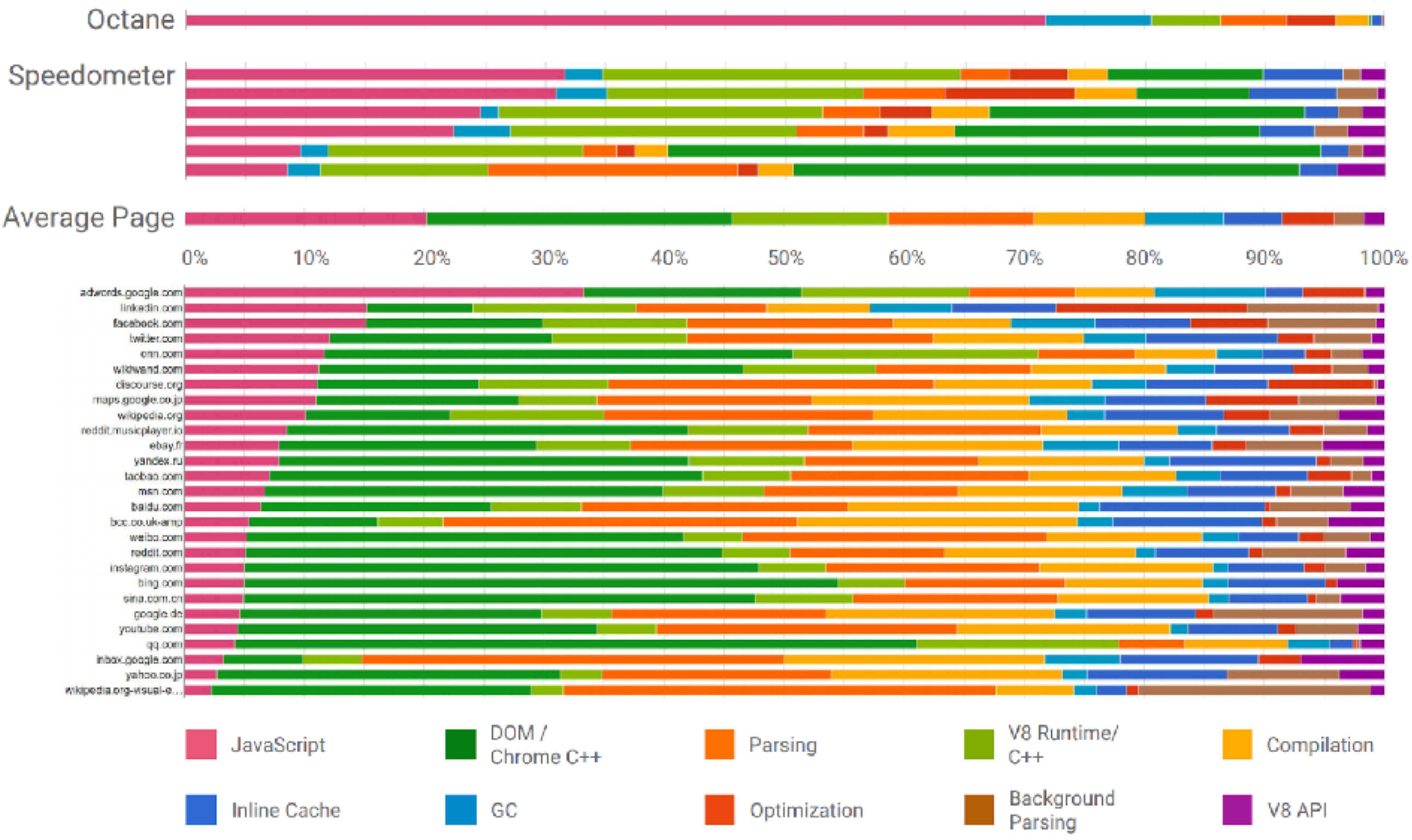
First Pipeline

2008



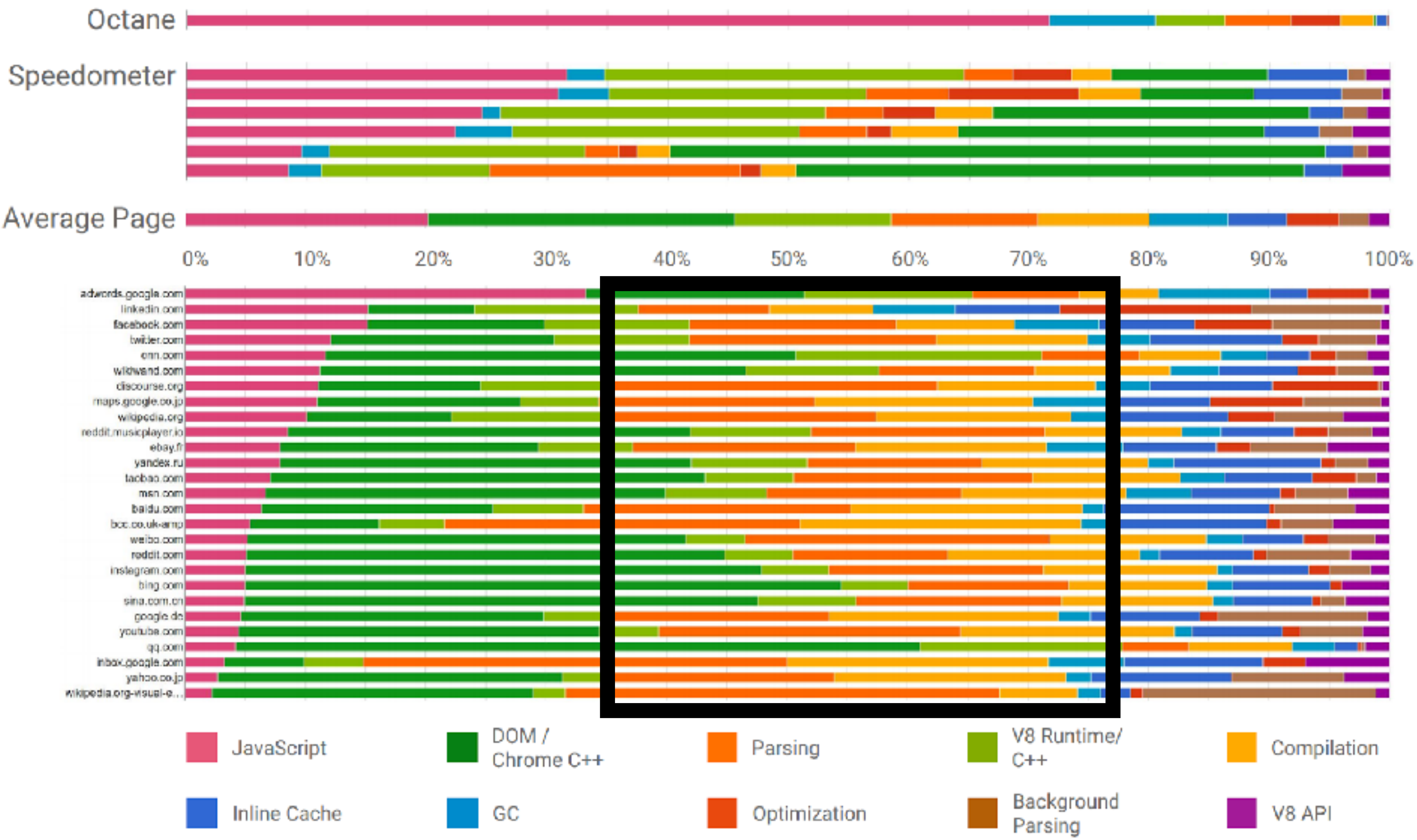
V8 Compiler Pipeline

Parser



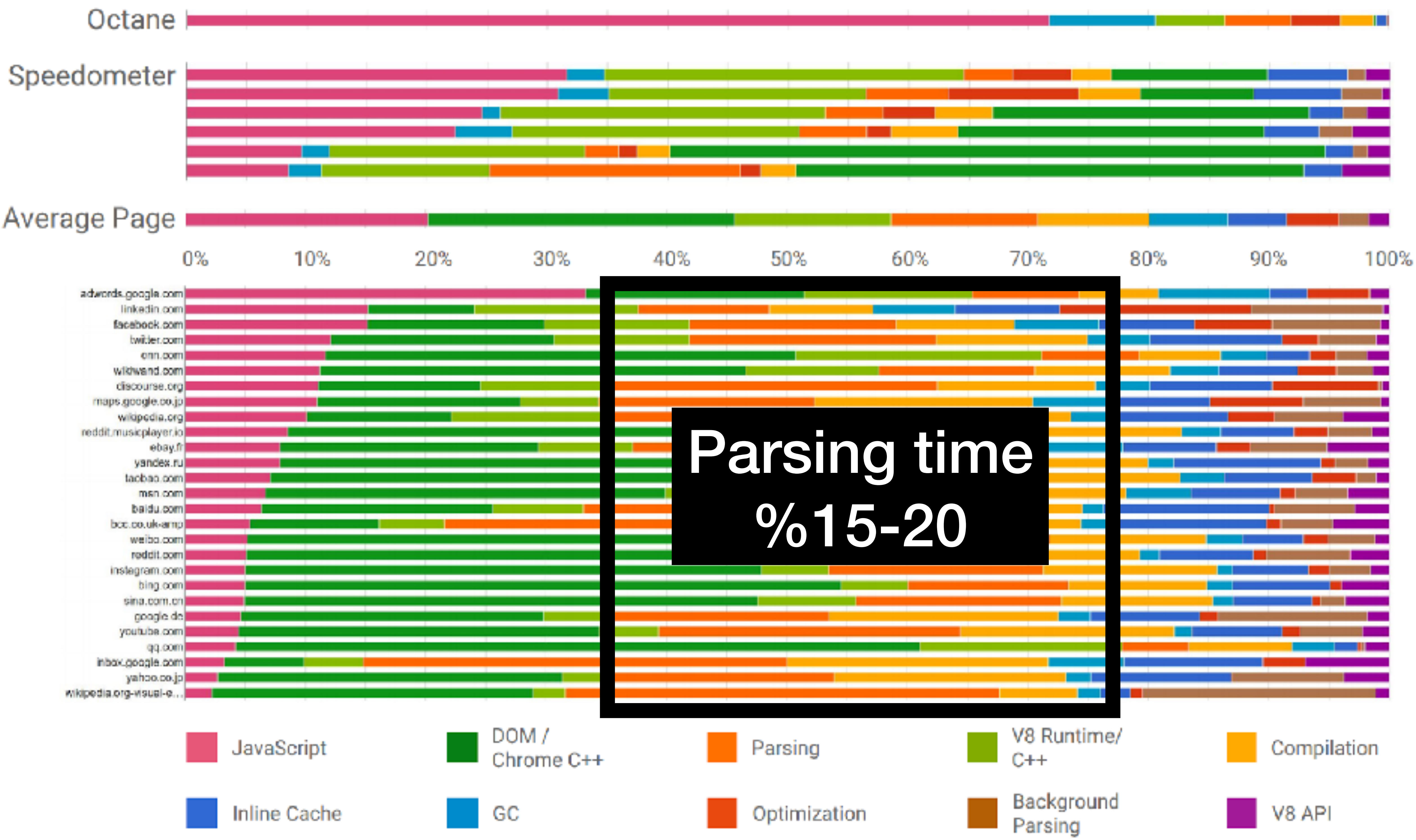
V8 Compiler Pipeline

Parser



V8 Compiler Pipeline

Parser



Parsing time %15-20

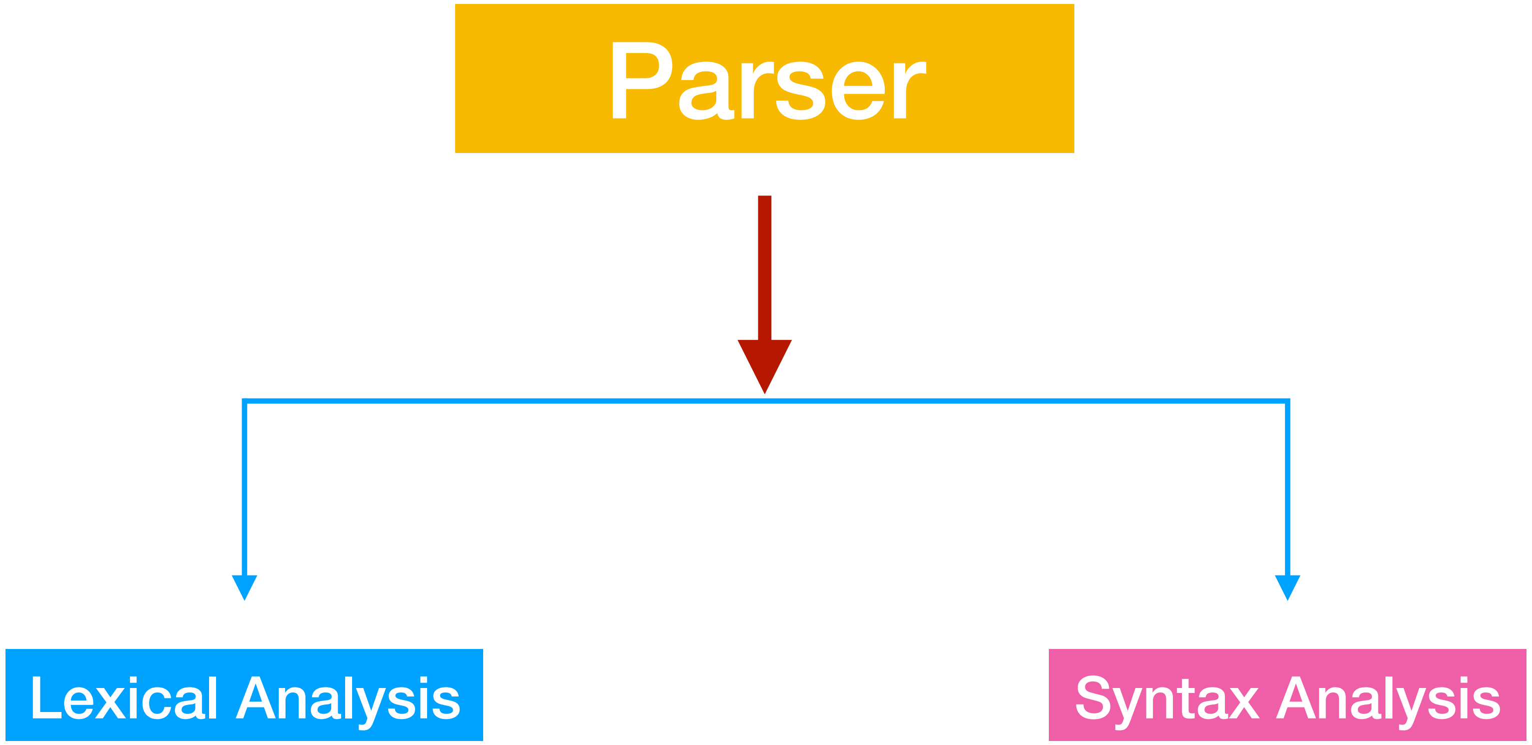
V8 Compiler Pipeline

Parser

Parser

V8 Compiler Pipeline

Parser



Parser

Lexical Analysis

Syntax Analysis

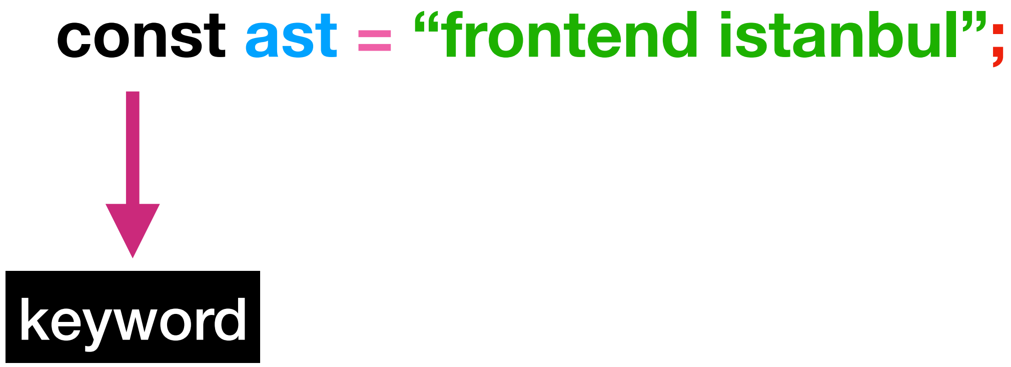
V8 Compiler Pipeline

Parser

```
const ast = “frontend istanbul”;
```

V8 Compiler Pipeline

Parser

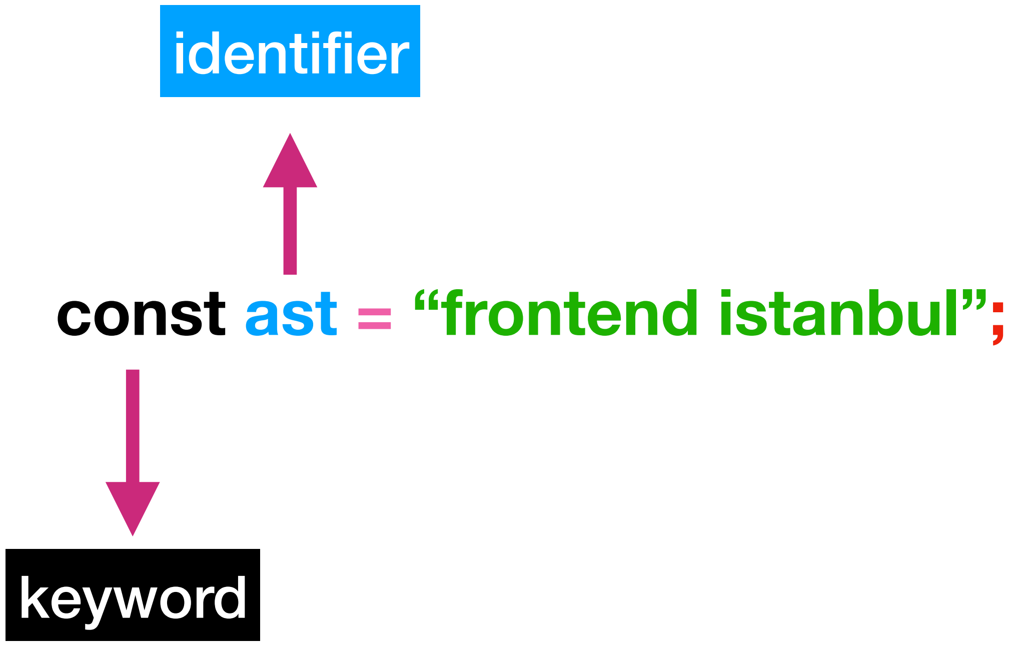


const ast = “frontend istanbul”;

keyword

V8 Compiler Pipeline

Parser



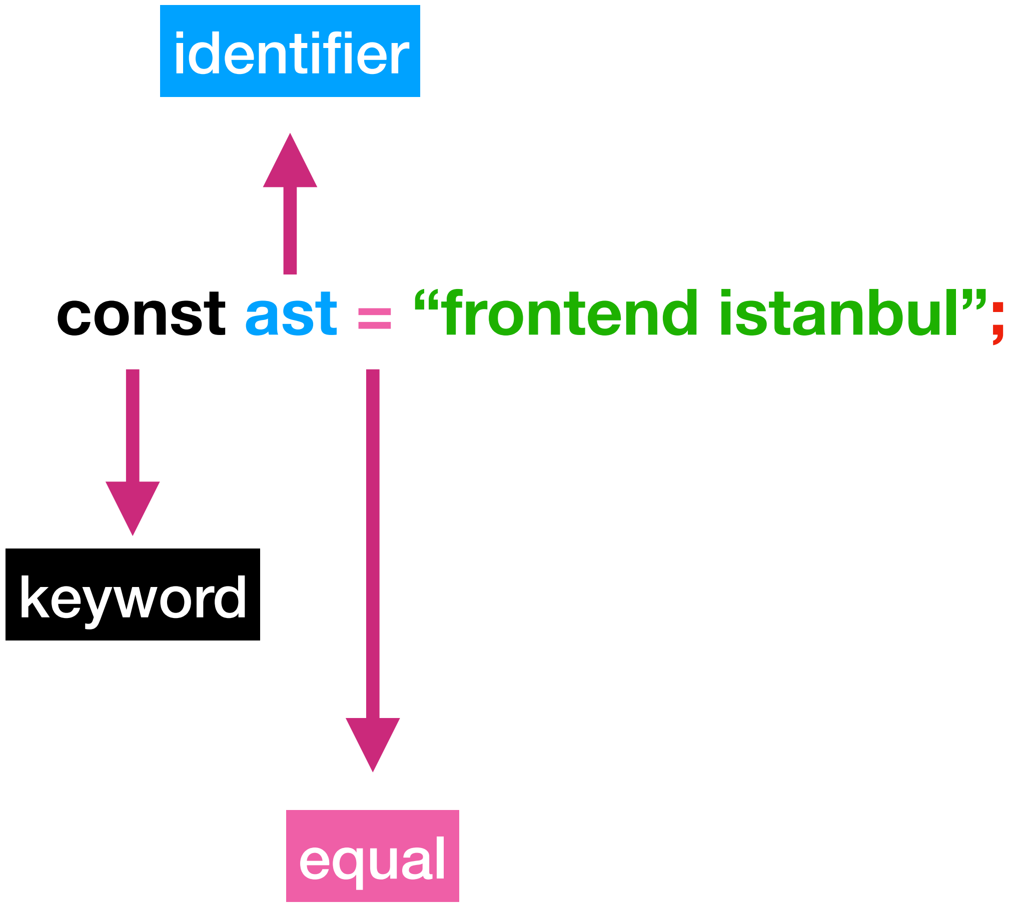
identifier

const ast = “frontend istanbul”;

keyword

V8 Compiler Pipeline

Parser



identifier

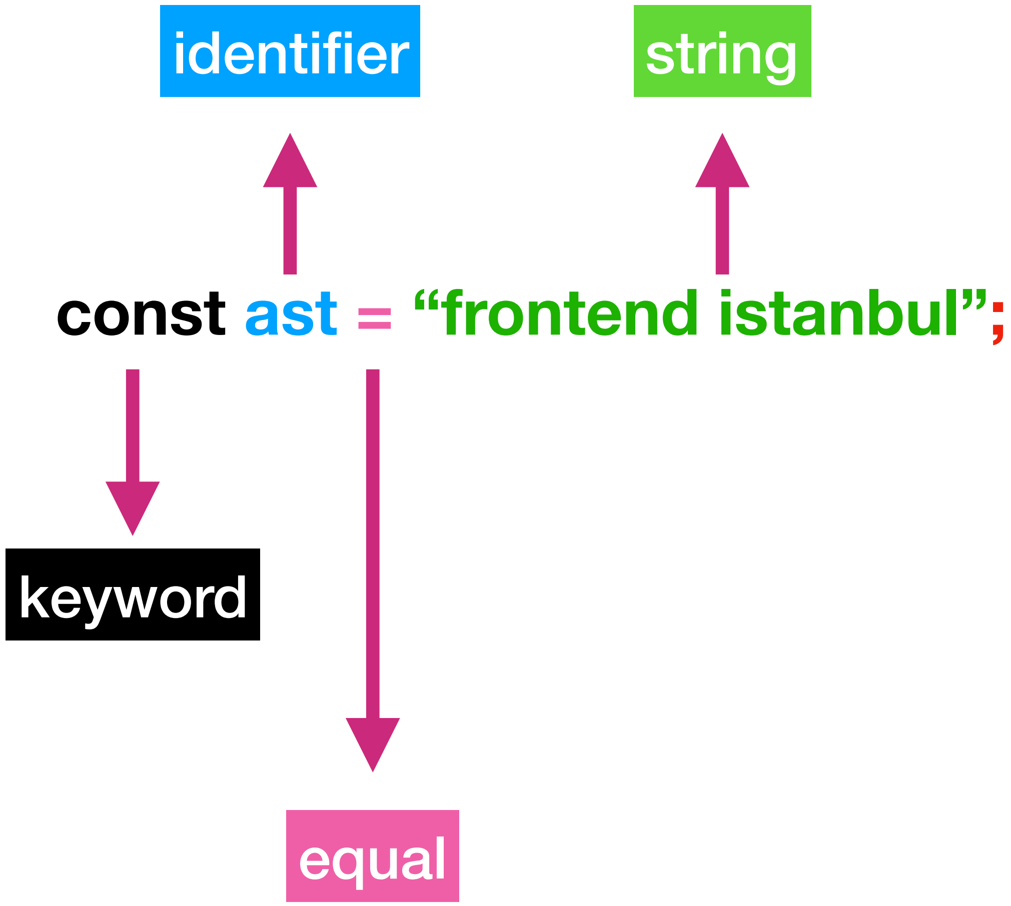
const ast = “frontend istanbul”;

keyword

equal

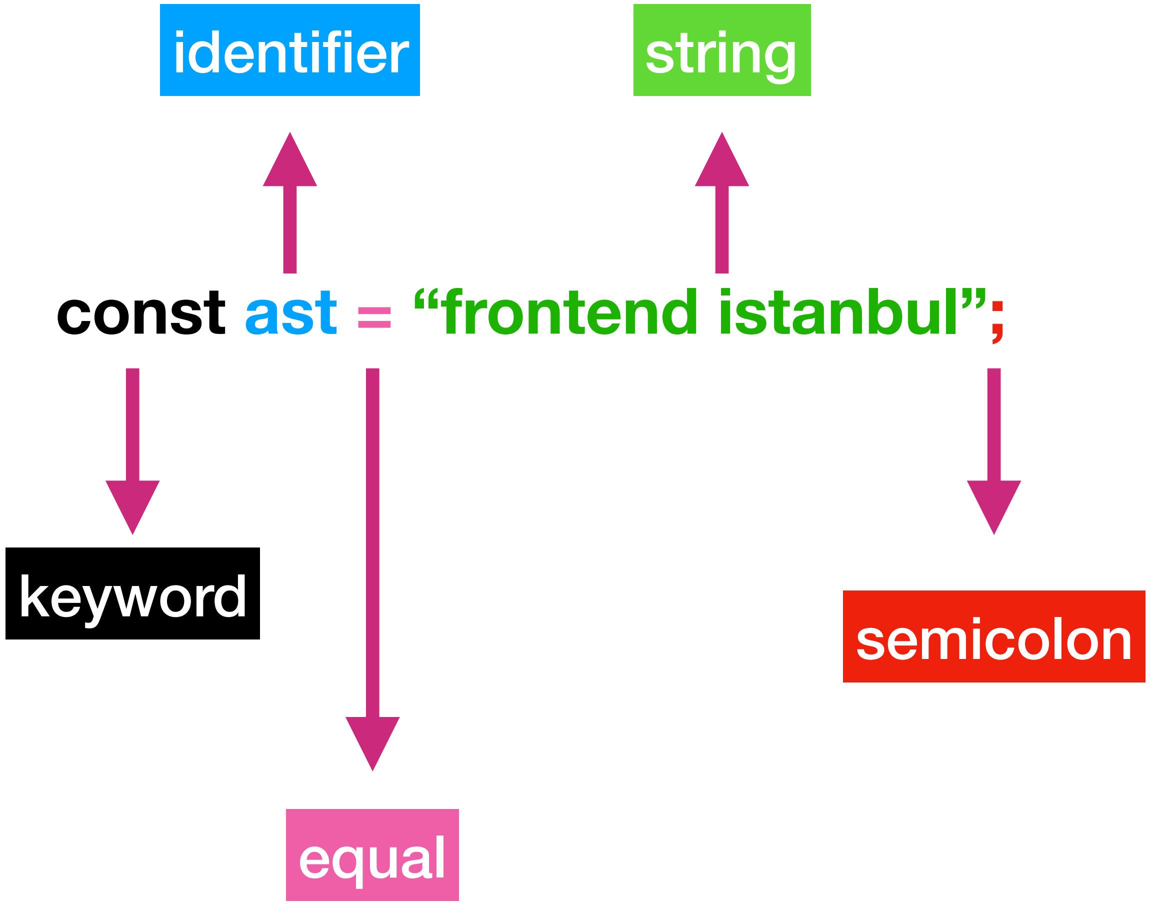
V8 Compiler Pipeline

Parser



V8 Compiler Pipeline

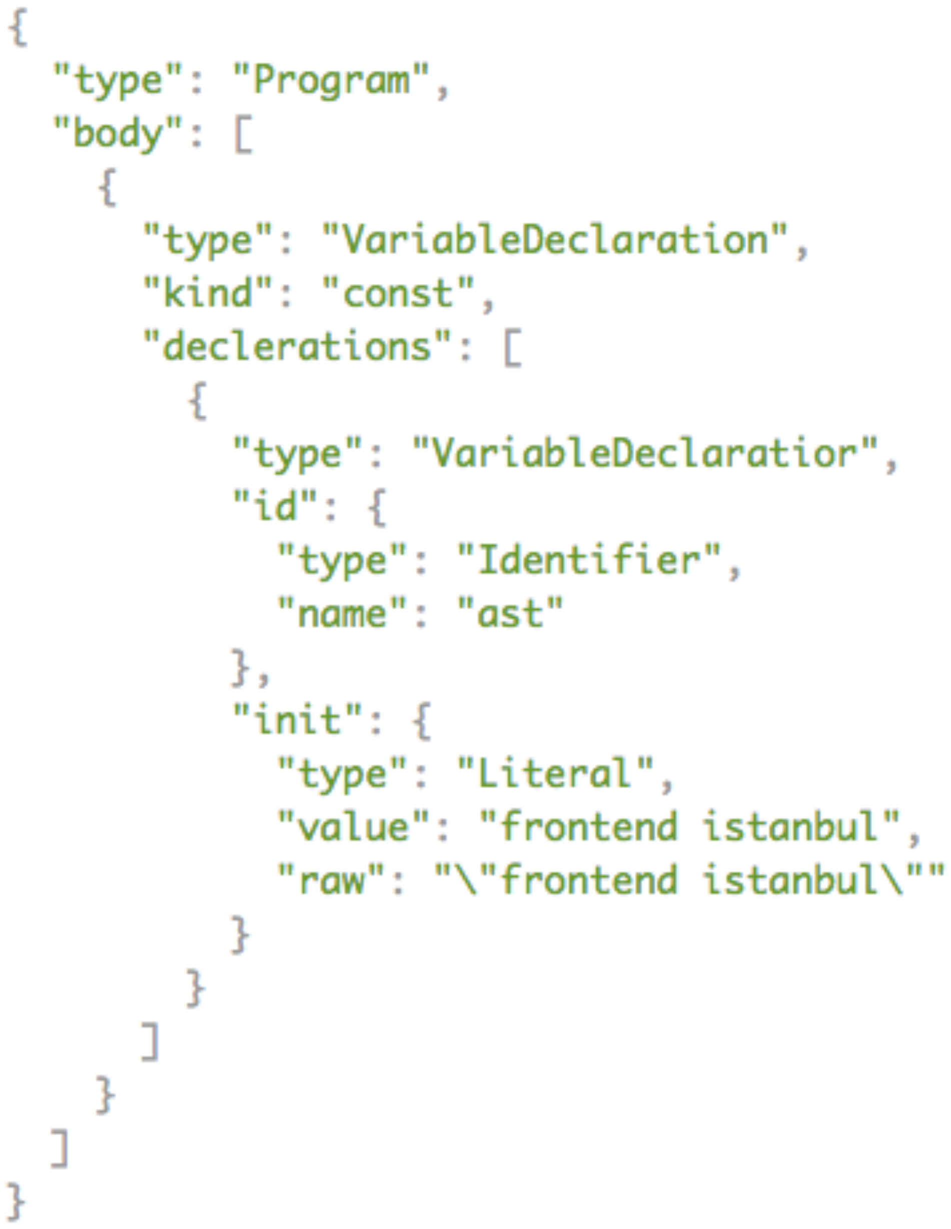
Parser



V8 Compiler Pipeline

AST - Abstract Syntax Tree

```
const ast = “frontend istanbul”;
```

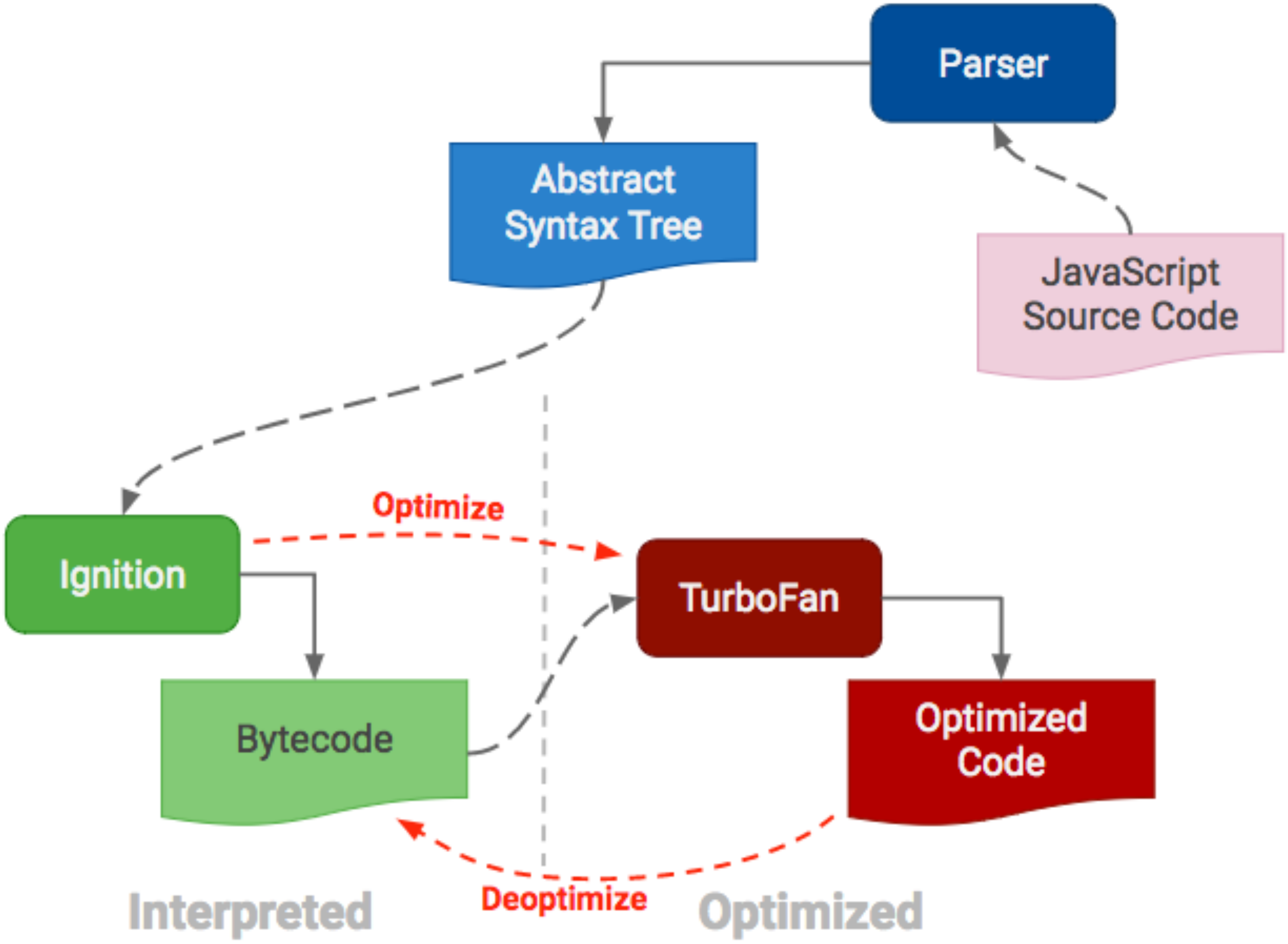


```
{
  "type": "Program",
  "body": [
    {
      "type": "VariableDeclaration",
      "kind": "const",
      "declerations": [
        {
          "type": "VariableDeclaratior",
          "id": {
            "type": "Identifier",
            "name": "ast"
          },
          "init": {
            "type": "Literal",
            "value": "frontend istanbul",
            "raw": "\"frontend istanbul\""
          }
        }
      ]
    }
  ]
}
```

V8 Compiler Pipeline

Last Pipeline

2017



V8 Compiler Pipeline

Interpreter

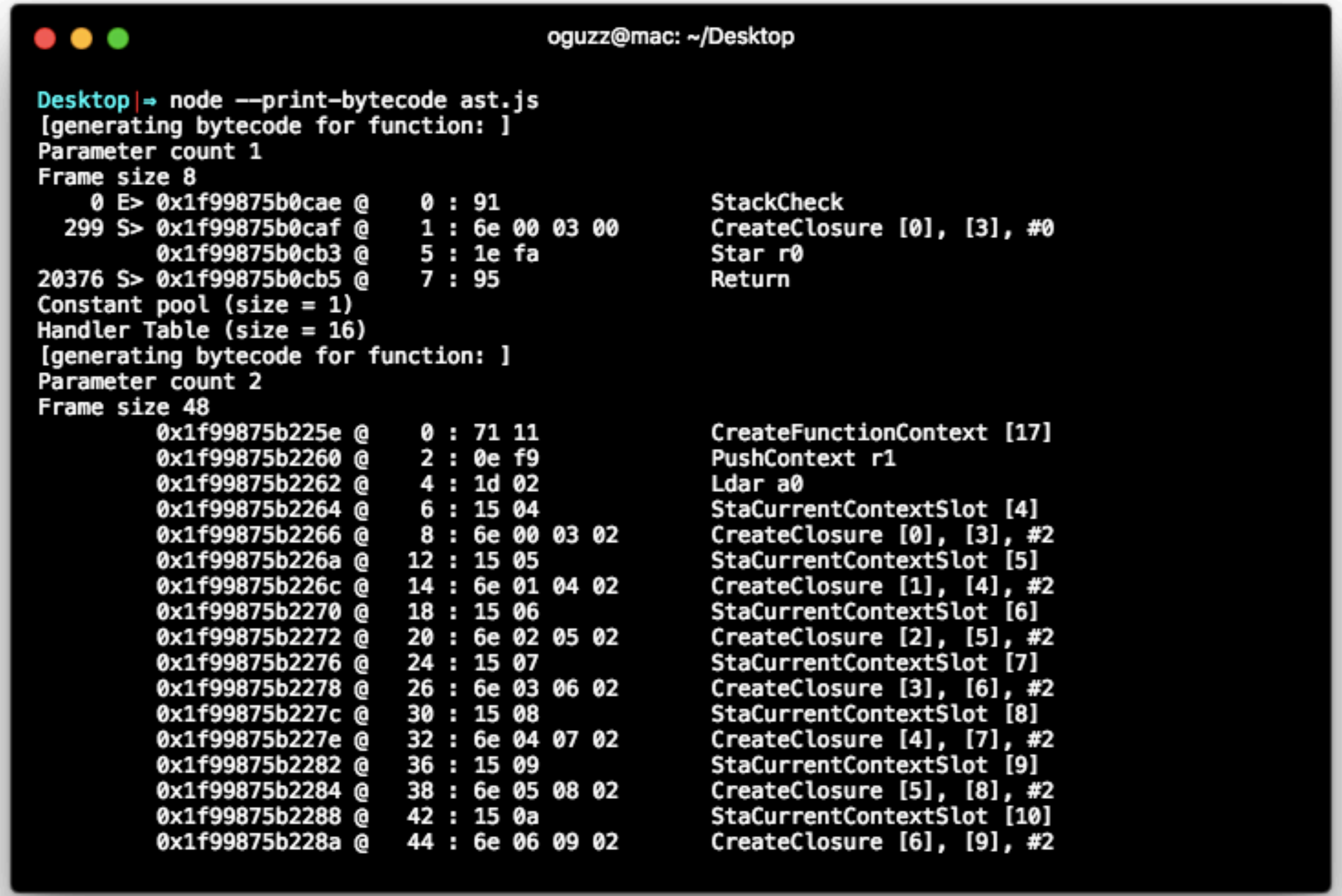
Ignition

written C++

V8 Compiler Pipeline

ByteCode

const ast = “frontend istanbul”;



```
Desktop|⇒ node --print-bytecode ast.js
[generating bytecode for function: ]
Parameter count 1
Frame size 8
    0 E> 0x1f99875b0cae @    0 : 91             StackCheck
  299 S> 0x1f99875b0caf @    1 : 6e 00 03 00    CreateClosure [0], [3], #0
         0x1f99875b0cb3 @    5 : 1e fa          Star r0
20376 S> 0x1f99875b0cb5 @    7 : 95             Return
Constant pool (size = 1)
Handler Table (size = 16)
[generating bytecode for function: ]
Parameter count 2
Frame size 48
         0x1f99875b225e @    0 : 71 11          CreateFunctionContext [17]
         0x1f99875b2260 @    2 : 0e f9          PushContext r1
         0x1f99875b2262 @    4 : 1d 02          Ldar a0
         0x1f99875b2264 @    6 : 15 04          StaCurrentContextSlot [4]
         0x1f99875b2266 @    8 : 6e 00 03 02    CreateClosure [0], [3], #2
         0x1f99875b226a @   12 : 15 05          StaCurrentContextSlot [5]
         0x1f99875b226c @   14 : 6e 01 04 02    CreateClosure [1], [4], #2
         0x1f99875b2270 @   18 : 15 06          StaCurrentContextSlot [6]
         0x1f99875b2272 @   20 : 6e 02 05 02    CreateClosure [2], [5], #2
         0x1f99875b2276 @   24 : 15 07          StaCurrentContextSlot [7]
         0x1f99875b2278 @   26 : 6e 03 06 02    CreateClosure [3], [6], #2
         0x1f99875b227c @   30 : 15 08          StaCurrentContextSlot [8]
         0x1f99875b227e @   32 : 6e 04 07 02    CreateClosure [4], [7], #2
         0x1f99875b2282 @   36 : 15 09          StaCurrentContextSlot [9]
         0x1f99875b2284 @   38 : 6e 05 08 02    CreateClosure [5], [8], #2
         0x1f99875b2288 @   42 : 15 0a          StaCurrentContextSlot [10]
         0x1f99875b228a @   44 : 6e 06 09 02    CreateClosure [6], [9], #2
```

V8 Compiler Pipeline

JIT Compiler

TurboFan

written C++

V8 Compiler Pipeline

Turbo Fan

Modern Code Generation Architecture

V8 Compiler Pipeline

Turbo Fan

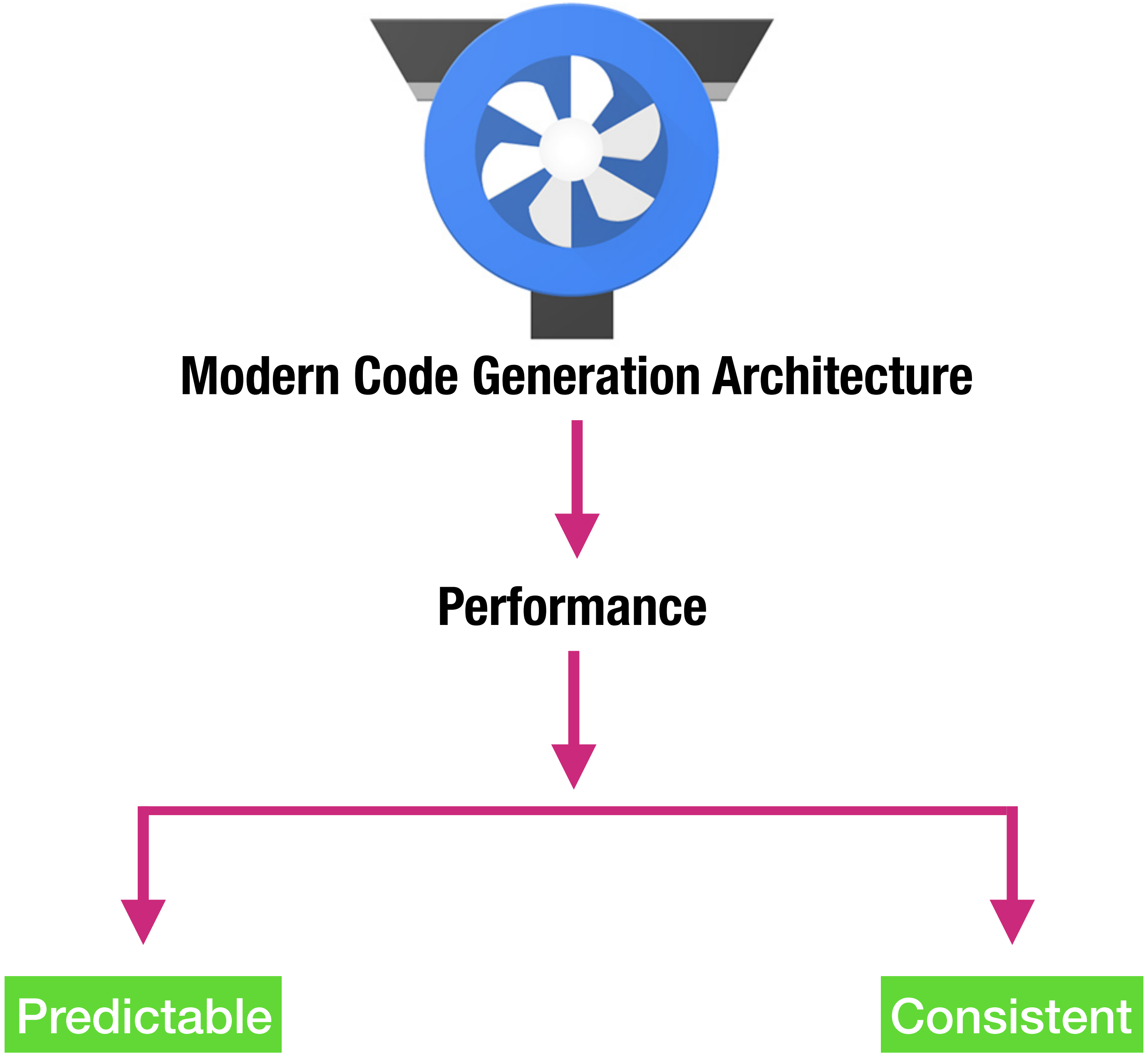
Modern Code Generation Architecture

↓

Performance

V8 Compiler Pipeline

Turbo Fan



Modern Code Generation Architecture

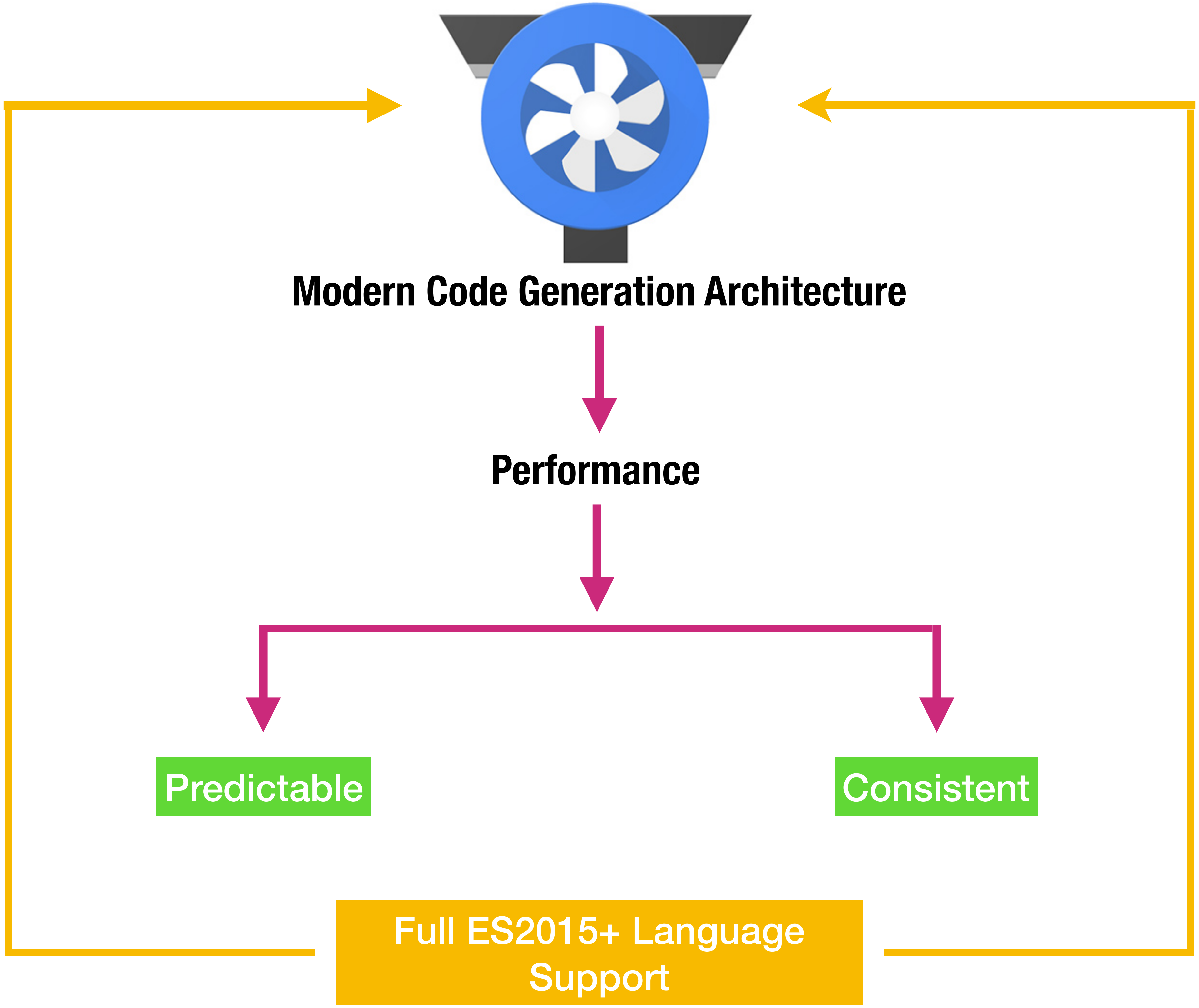
Performance

Predictable

Consistent

V8 Compiler Pipeline

Turbo Fan



Modern Code Generation Architecture

Performance

Predictable

Consistent

Full ES2015+ Language Support

V8 Compiler Pipeline

LifeCycle Example

```
const f = (a, b) => a + b * 3;
```

V8 Compiler Pipeline

LifeCycle

```
const f = (a, b) => a + b * 3;
```

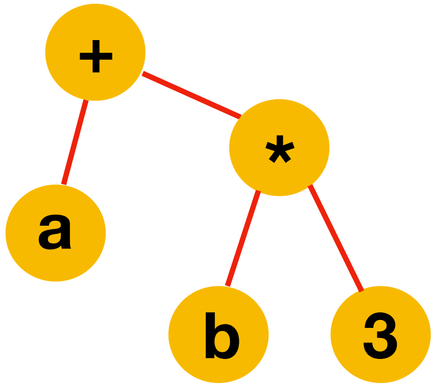
Abstract Syntax Tree

V8 Compiler Pipeline

LifeCycle

```
const f = (a, b) => a + b * 3;
```

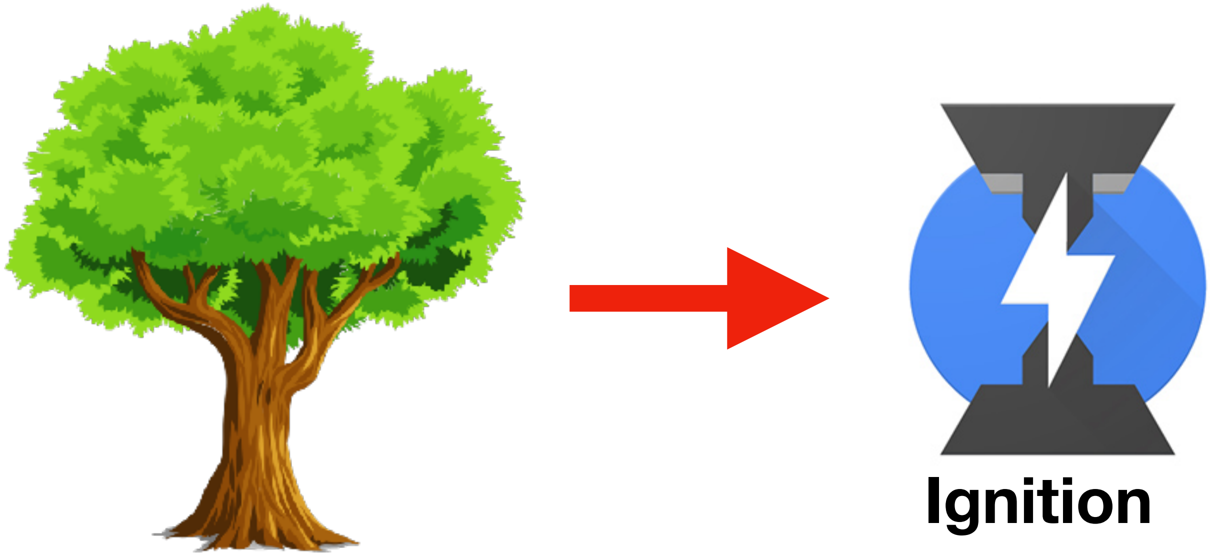
Abstract Syntax Tree



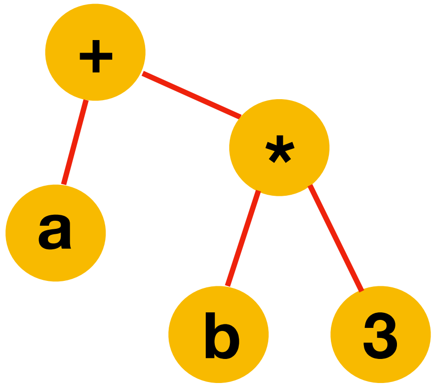
V8 Compiler Pipeline

LifeCycle

```
const f = (a, b) => a + b * 3;
```



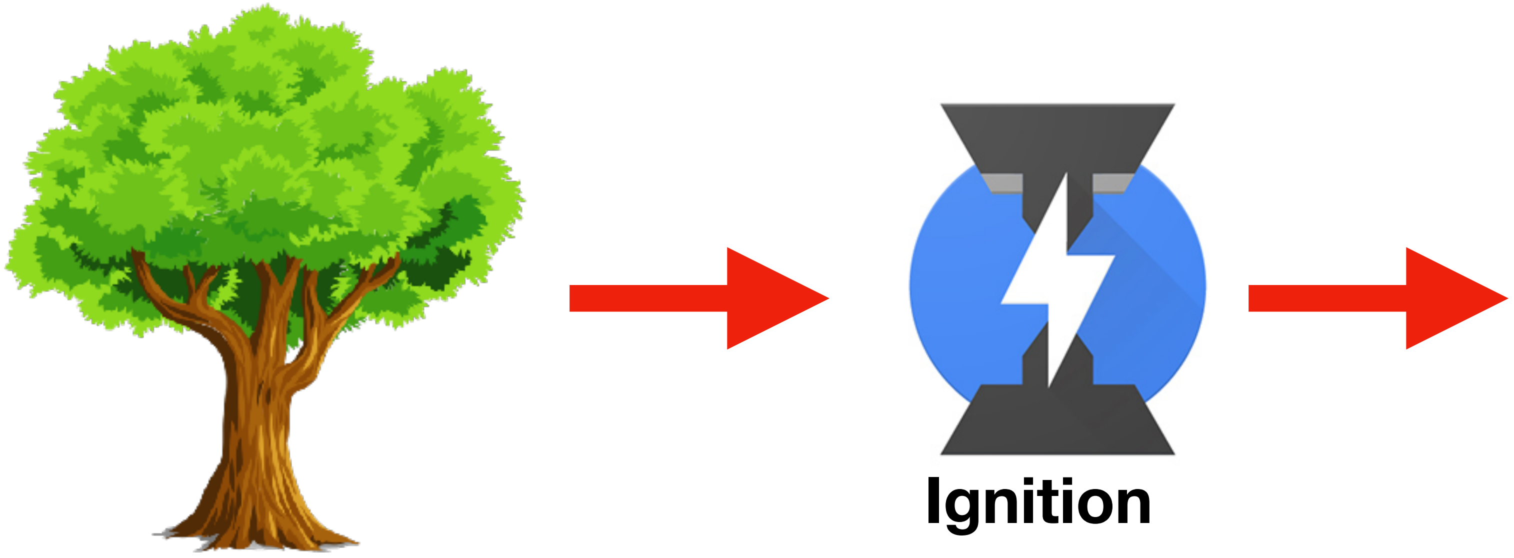
Abstract Syntax Tree



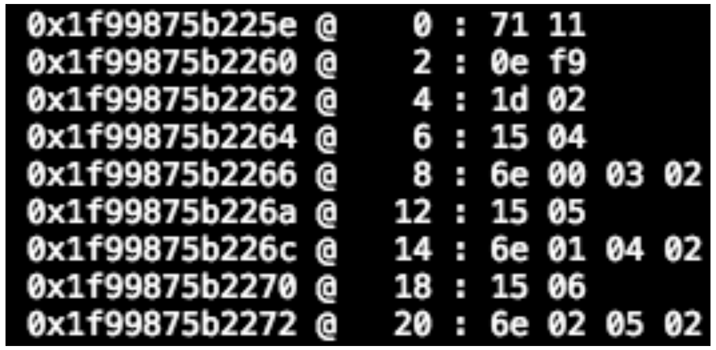
V8 Compiler Pipeline

LifeCycle

```
const f = (a, b) => a + b * 3;
```

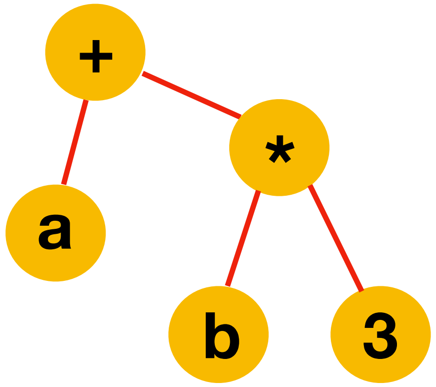


ByteCode



```
0x1f99875b225e @    0 : 71 11
0x1f99875b2260 @    2 : 0e f9
0x1f99875b2262 @    4 : 1d 02
0x1f99875b2264 @    6 : 15 04
0x1f99875b2266 @    8 : 6e 00 03 02
0x1f99875b226a @   12 : 15 05
0x1f99875b226c @   14 : 6e 01 04 02
0x1f99875b2270 @   18 : 15 06
0x1f99875b2272 @   20 : 6e 02 05 02
```

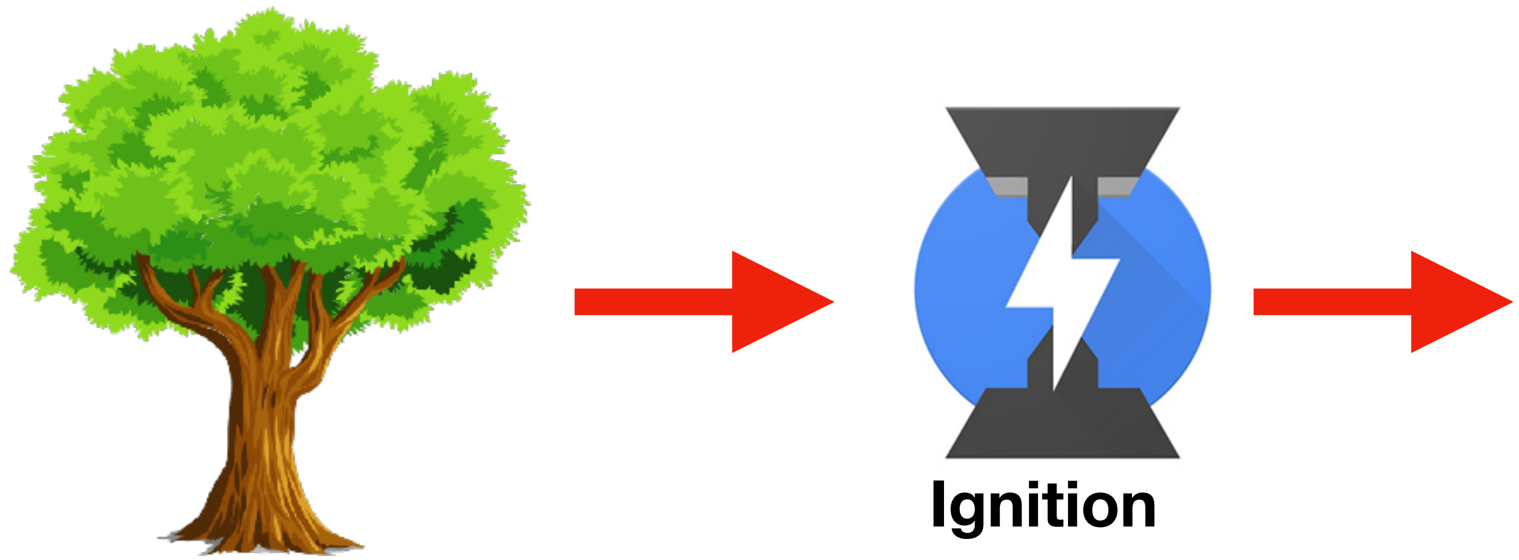
Abstract Syntax Tree



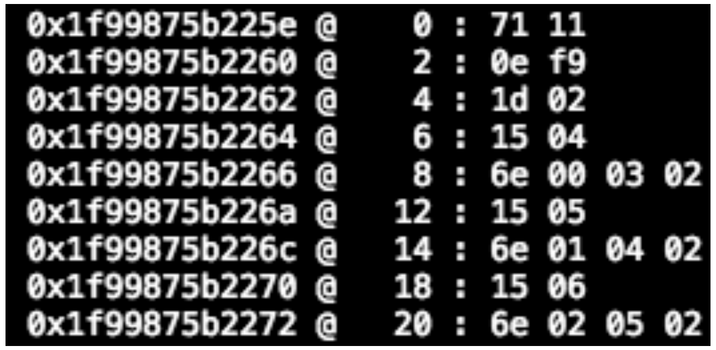
V8 Compiler Pipeline

LifeCycle

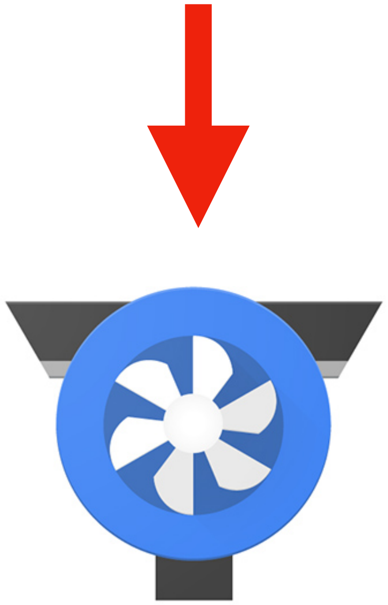
```
const f = (a, b) => a + b * 3;
```



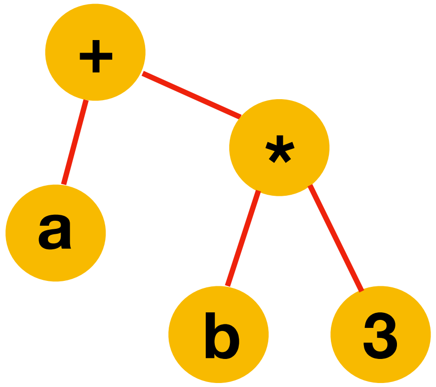
ByteCode



```
0x1f99875b225e @    0 : 71 11
0x1f99875b2260 @    2 : 0e f9
0x1f99875b2262 @    4 : 1d 02
0x1f99875b2264 @    6 : 15 04
0x1f99875b2266 @    8 : 6e 00 03 02
0x1f99875b226a @   12 : 15 05
0x1f99875b226c @   14 : 6e 01 04 02
0x1f99875b2270 @   18 : 15 06
0x1f99875b2272 @   20 : 6e 02 05 02
```



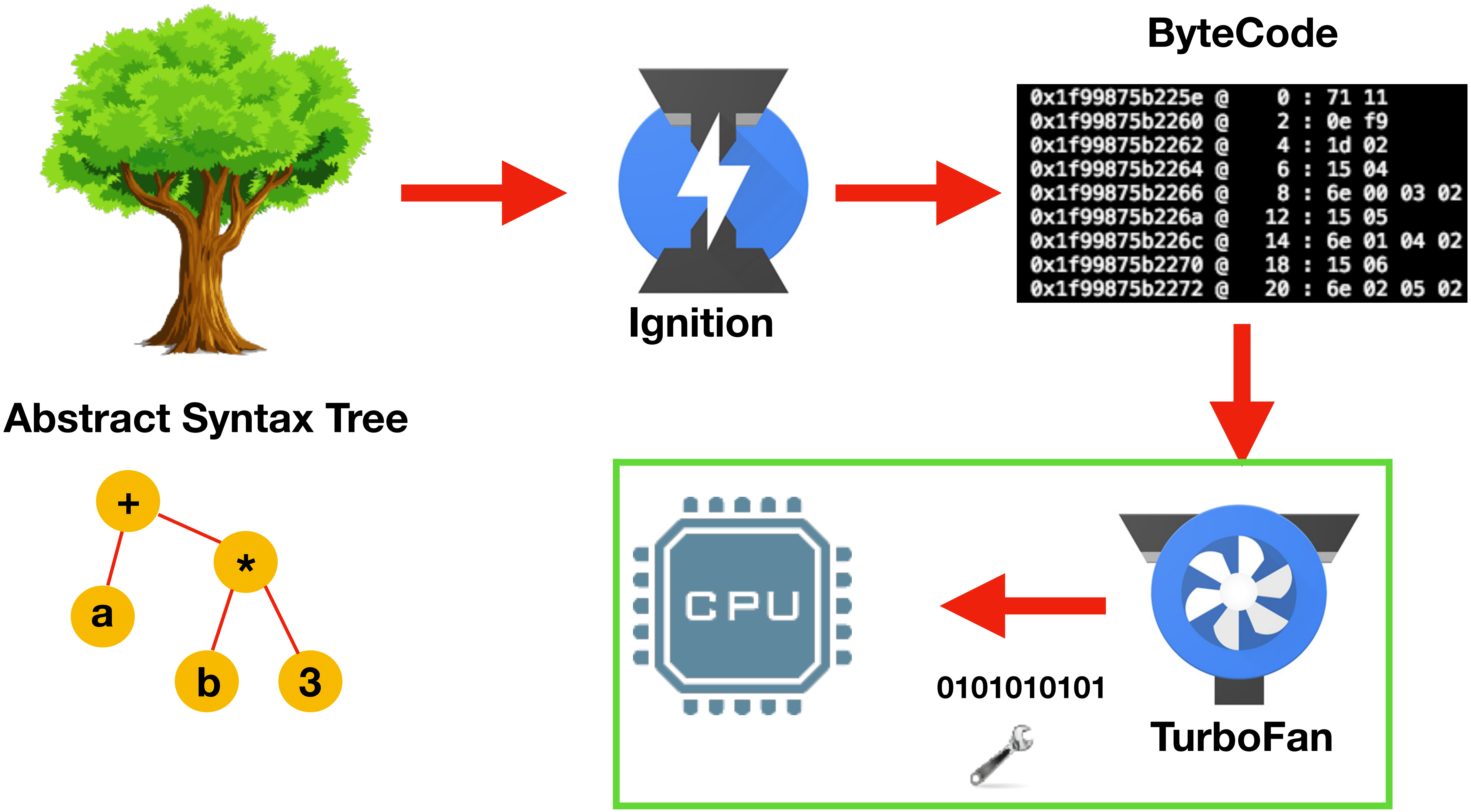
Abstract Syntax Tree



V8 Compiler Pipeline

LifeCycle

```
const f = (a, b) => a + b * 3;
```



Abstract Syntax Tree

Ignition

ByteCode

```
0x1f99875b225e @    0 : 71 11
0x1f99875b2260 @    2 : 0e f9
0x1f99875b2262 @    4 : 1d 02
0x1f99875b2264 @    6 : 15 04
0x1f99875b2266 @    8 : 6e 00 03 02
0x1f99875b226a @   12 : 15 05
0x1f99875b226c @   14 : 6e 01 04 02
0x1f99875b2270 @   18 : 15 06
0x1f99875b2272 @   20 : 6e 02 05 02
```

TurboFan

0101010101

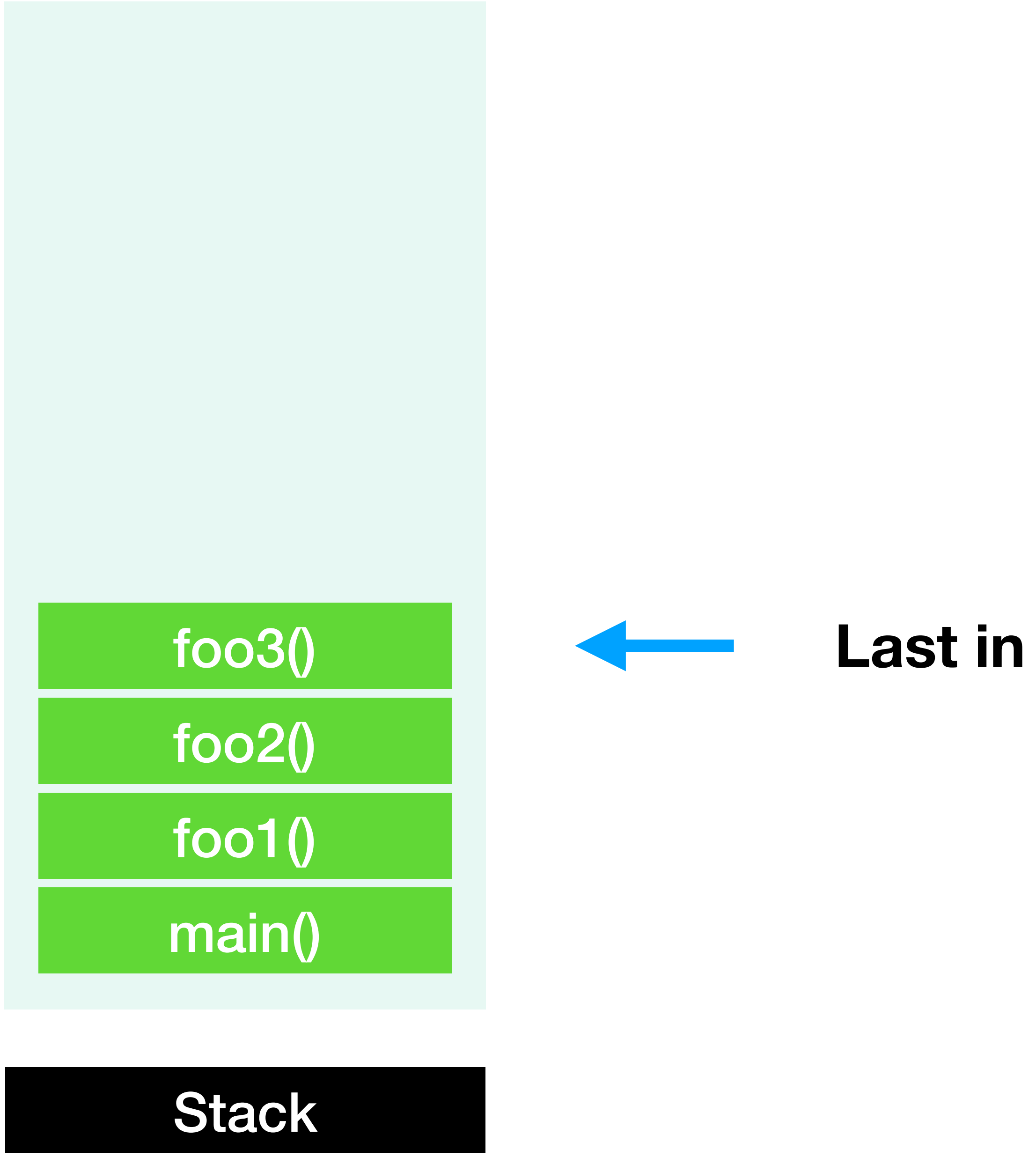
Optimization = CPU

Concurrency Model

Call Stack

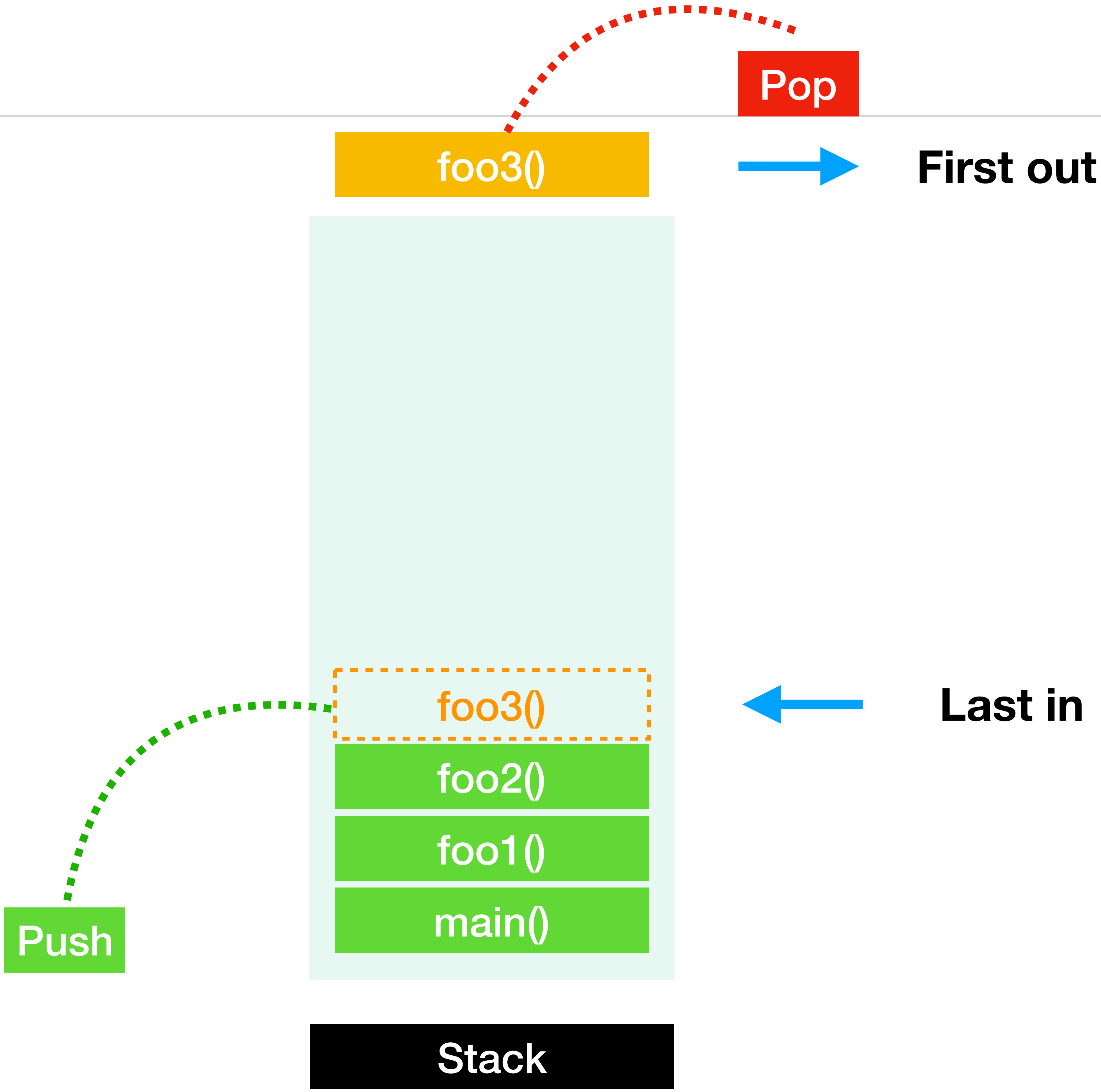
JavaScript Single-Thread Language

Concurrency Model
Call Stack



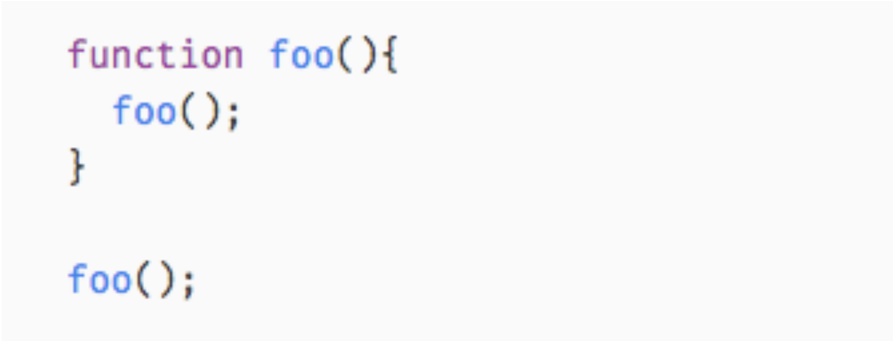
Concurrency Model

Call Stack

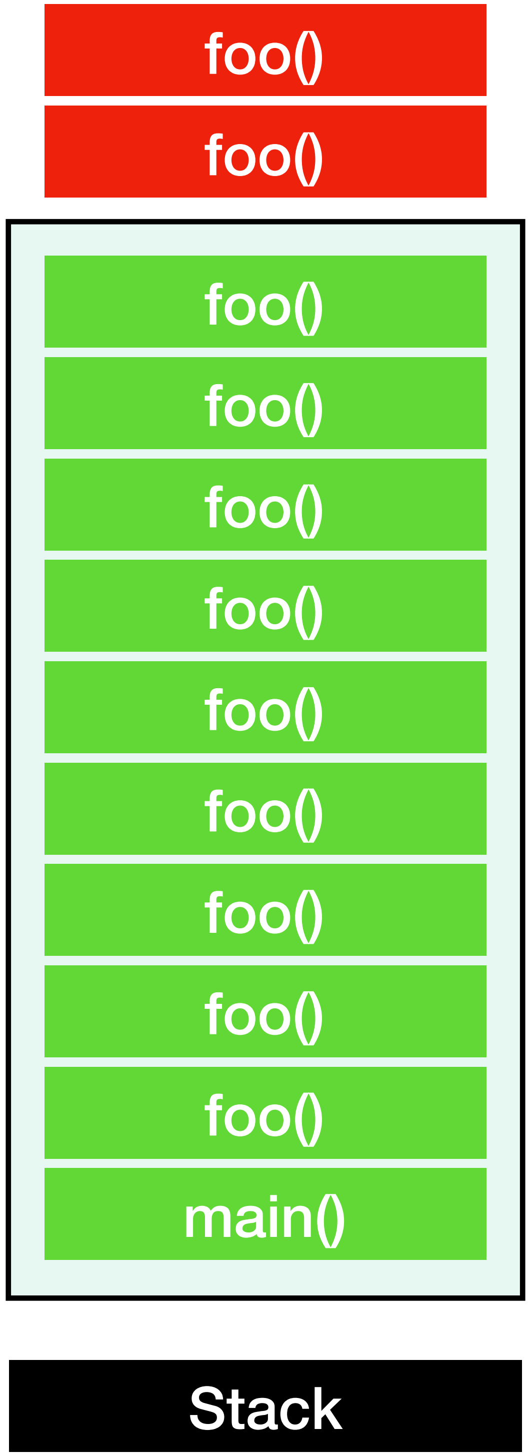


Concurrency Model

Stack Overflow

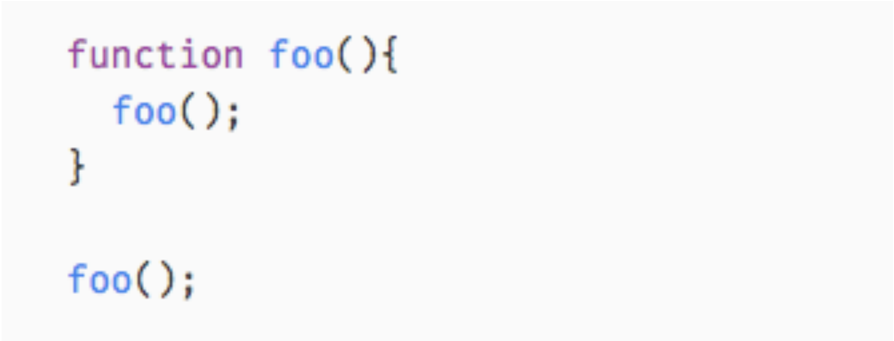


```
function foo(){
  foo();
}

foo();
```

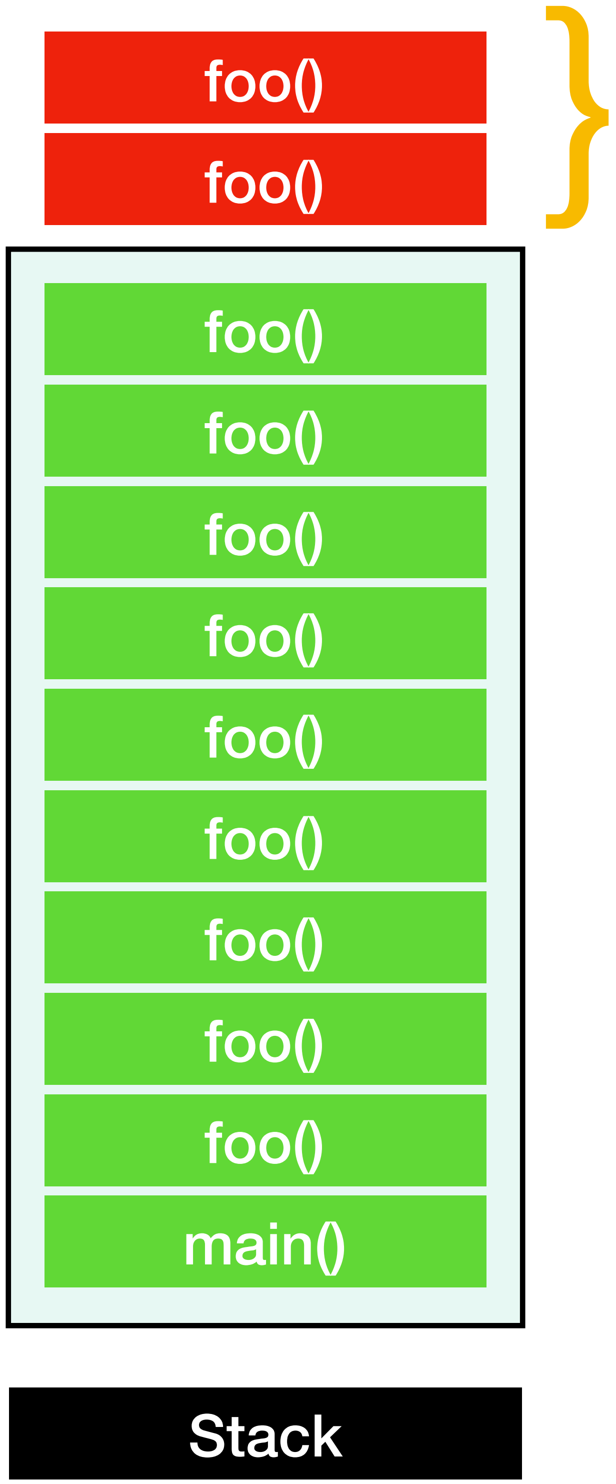


Concurrency Model

Stack Overflow



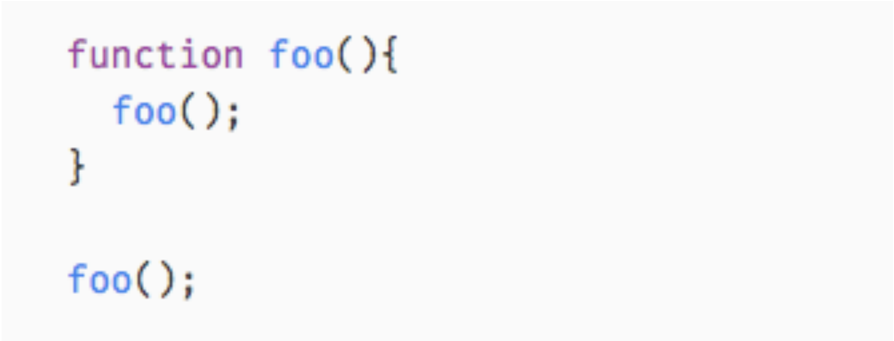
```
function foo(){
  foo();
}

foo();
```



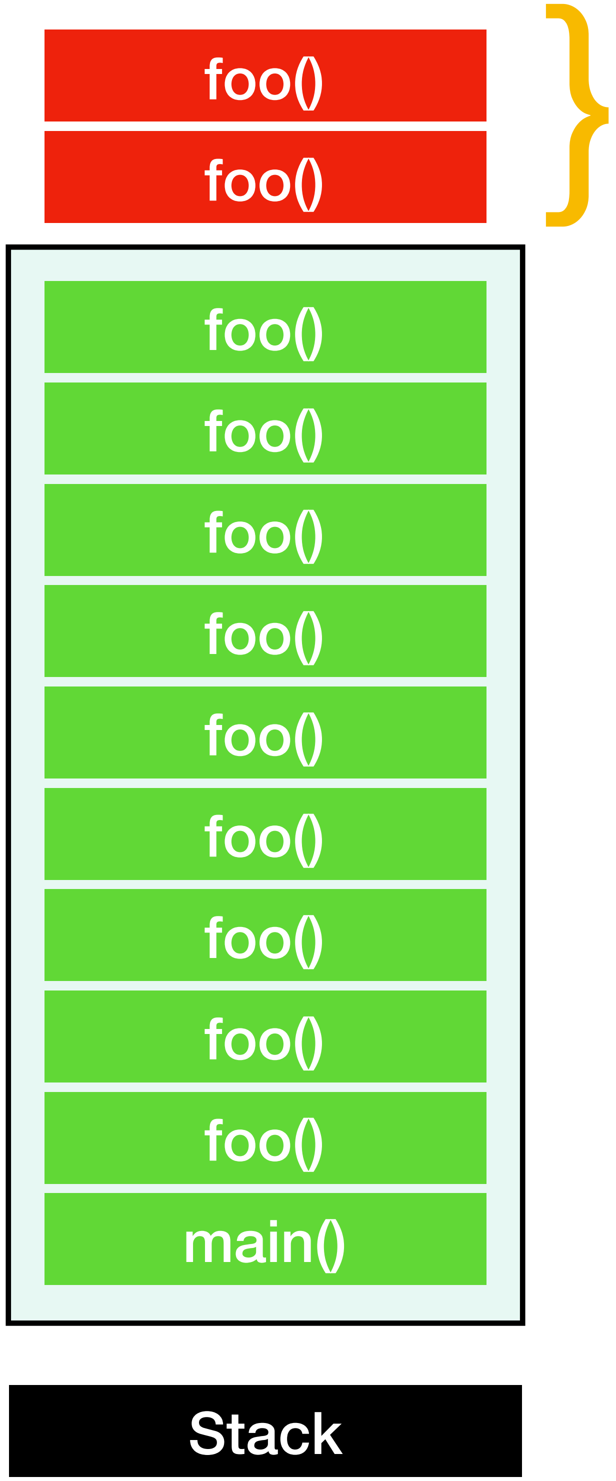
foo()
foo()
foo()
foo()
foo()
foo()
foo()
foo()
foo()
foo()
foo()
main()
Stack

Concurrency Model

Stack Overflow



```
function foo(){
  foo();
}

foo();
```



foo()

foo()

foo()

foo()

foo()

foo()

foo()

foo()

foo()

foo()

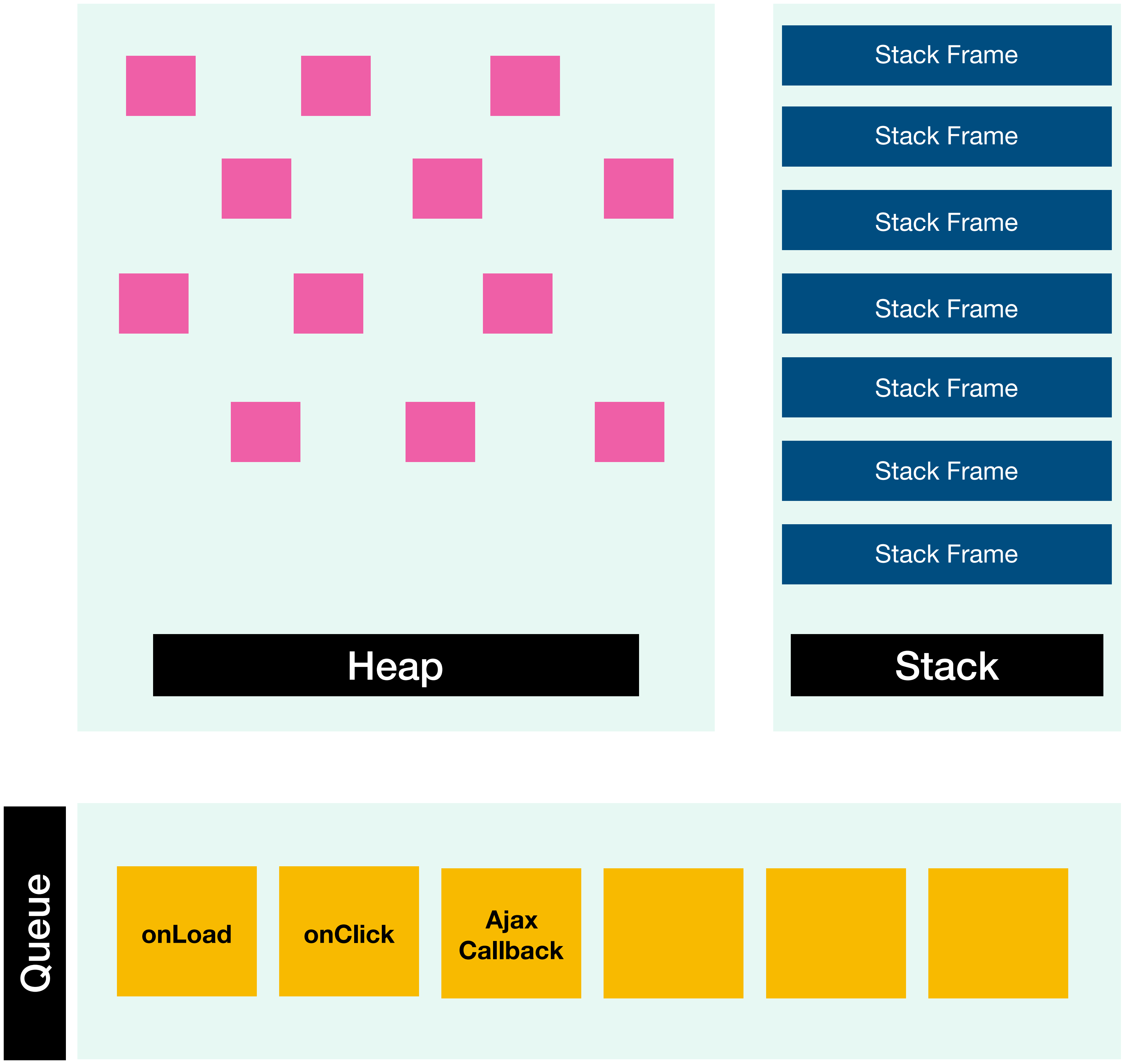
foo()

main()

Stack

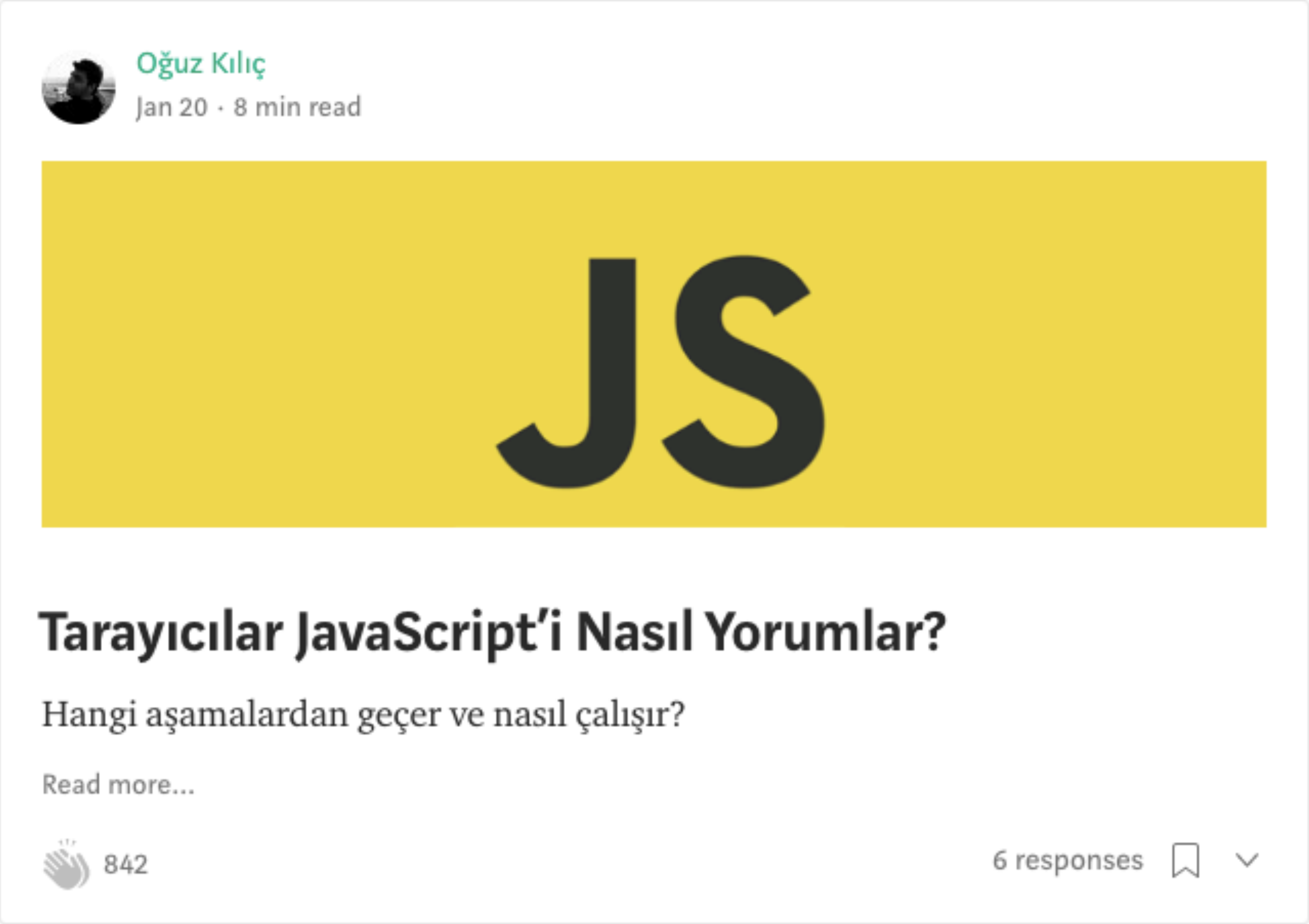
Range Error: maximum call stack size exceeded

Concurrency Model



Memory Management

Garbage Collection



Oğuz Kılıç

Jan 20 · 8 min read

Tarayıcılar JavaScript'i Nasıl Yorumlar?

Hangi aşamalardan geçer ve nasıl çalışır?

Read more…

842

6 responses

https://medium.com/@oguzkilic

The End