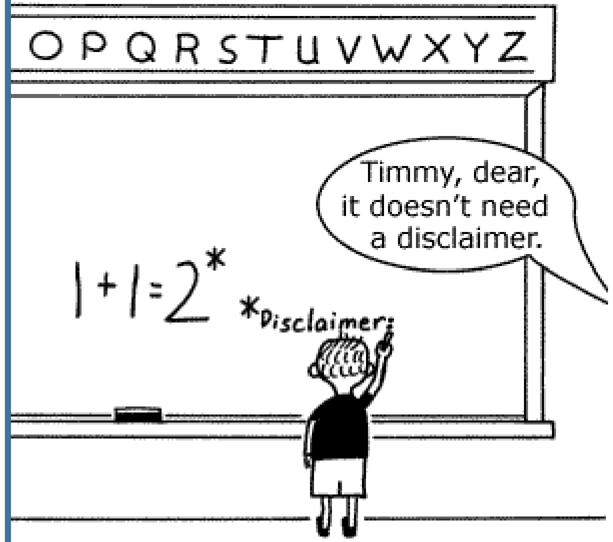


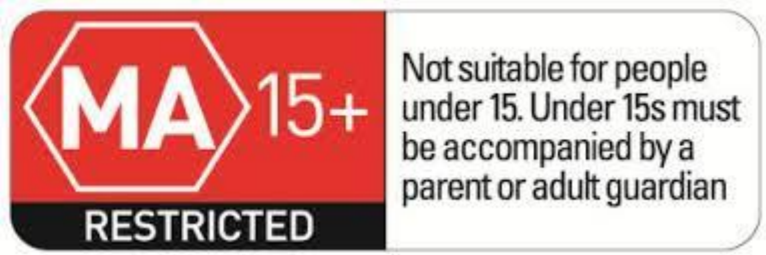
BootCamp

FastTrack

Javascript Intro*

Revision 0

Rating

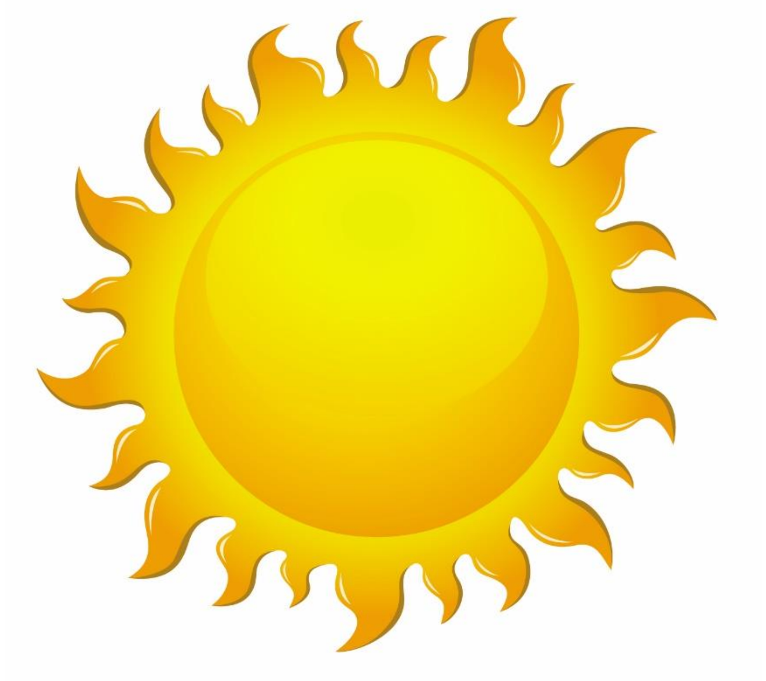


MA 15+

RESTRICTED

Not suitable for people under 15. Under 15s must be accompanied by a parent or adult guardian

Introduction



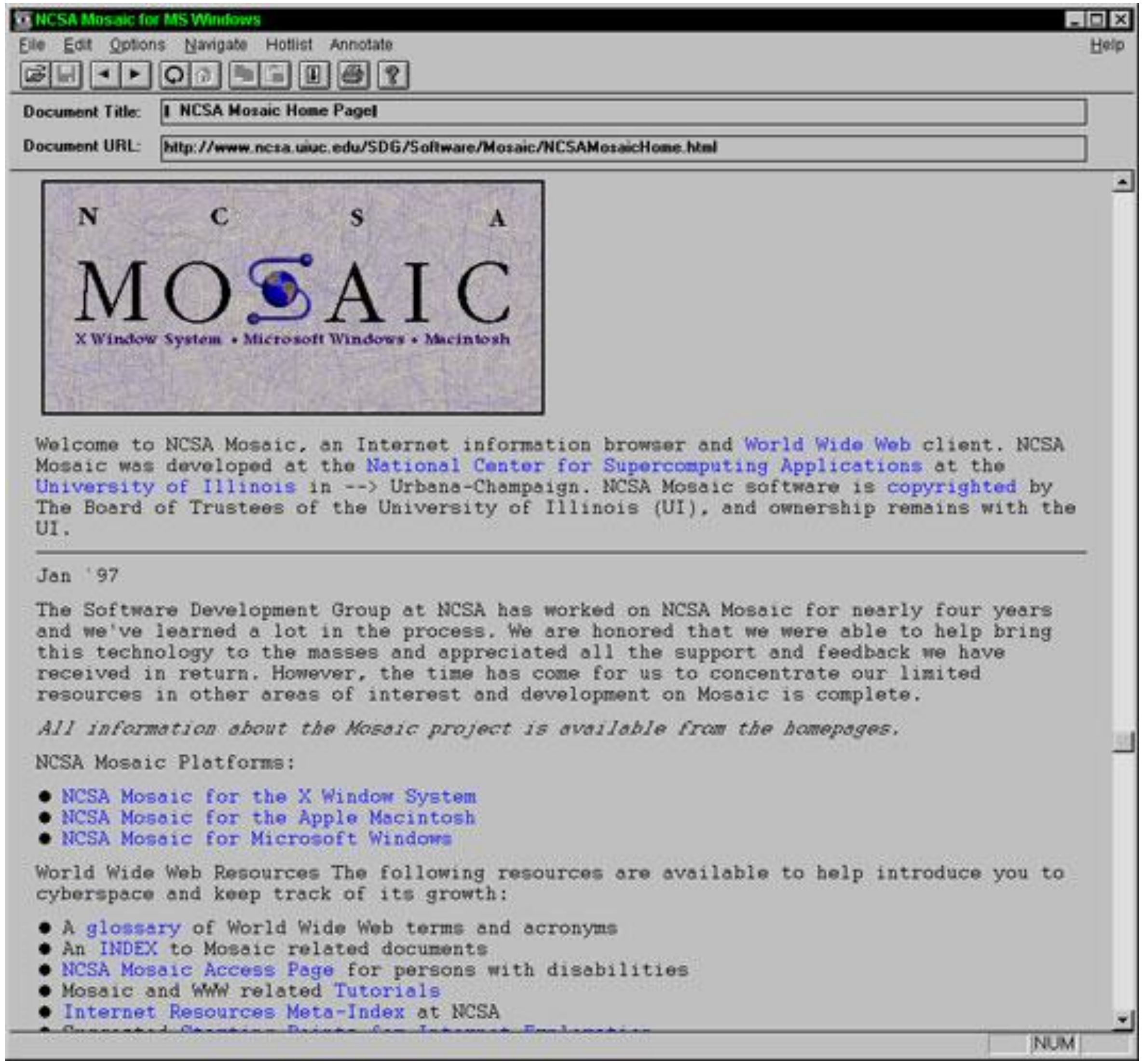
Introduction

Introduction

Introduction

History

NCSA: Mosaic


NCSA Mosaic for MS Windows
File Edit Options Navigate Hotlist Annotate
Help
N C S A
MOSAIC
X Window System • Microsoft Windows • Macintosh
NUM

Document Title: NCSA Mosaic Home Page

Document URL: http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html

Welcome to NCSA Mosaic, an Internet information browser and World Wide Web client. NCSA Mosaic was developed at the National Center for Supercomputing Applications at the University of Illinois in --> Urbana-Champaign. NCSA Mosaic software is copyrighted by The Board of Trustees of the University of Illinois (UI), and ownership remains with the UI.

Jan '97

The Software Development Group at NCSA has worked on NCSA Mosaic for nearly four years and we've learned a lot in the process. We are honored that we were able to help bring this technology to the masses and appreciated all the support and feedback we have received in return. However, the time has come for us to concentrate our limited resources in other areas of interest and development on Mosaic is complete.

All information about the Mosaic project is available from the homepages.

NCSA Mosaic Platforms:

- NCSA Mosaic for the X Window System
- NCSA Mosaic for the Apple Macintosh
- NCSA Mosaic for Microsoft Windows

World Wide Web Resources The following resources are available to help introduce you to cyberspace and keep track of its growth:

- A glossary of World Wide Web terms and acronyms
- An INDEX to Mosaic related documents
- NCSA Mosaic Access Page for persons with disabilities
- Mosaic and WWW related Tutorials
- Internet Resources Meta-Index at NCSA
- Suggested Starting Points for Internet Exploration

Standardization

- ECMA – European computer manufacturer association
- TC39 committee (1997, 21 y)
    - ECMAScript (ECMA 262)
- 6. June 2015 → ES2015 / ES6
- 7. June 2016 → ES2016 / ES7
- 8. June 2017 → ES2017 / ES8

ecma INTERNATIONAL

Standard ECMA-262

8th Edition / June 2017

ECMAScript® 2017
Language Specification

Standard

Rue du Rhône 114 CH-1204 Geneva T: +41 22 849 6000 F: +41 22 849 6001

How is your code run?

Single Threaded

Synchronous

Engines

V8

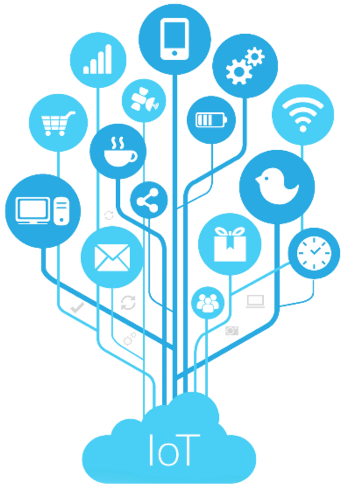
SpiderMonkey

Chakra

JavaScriptCore

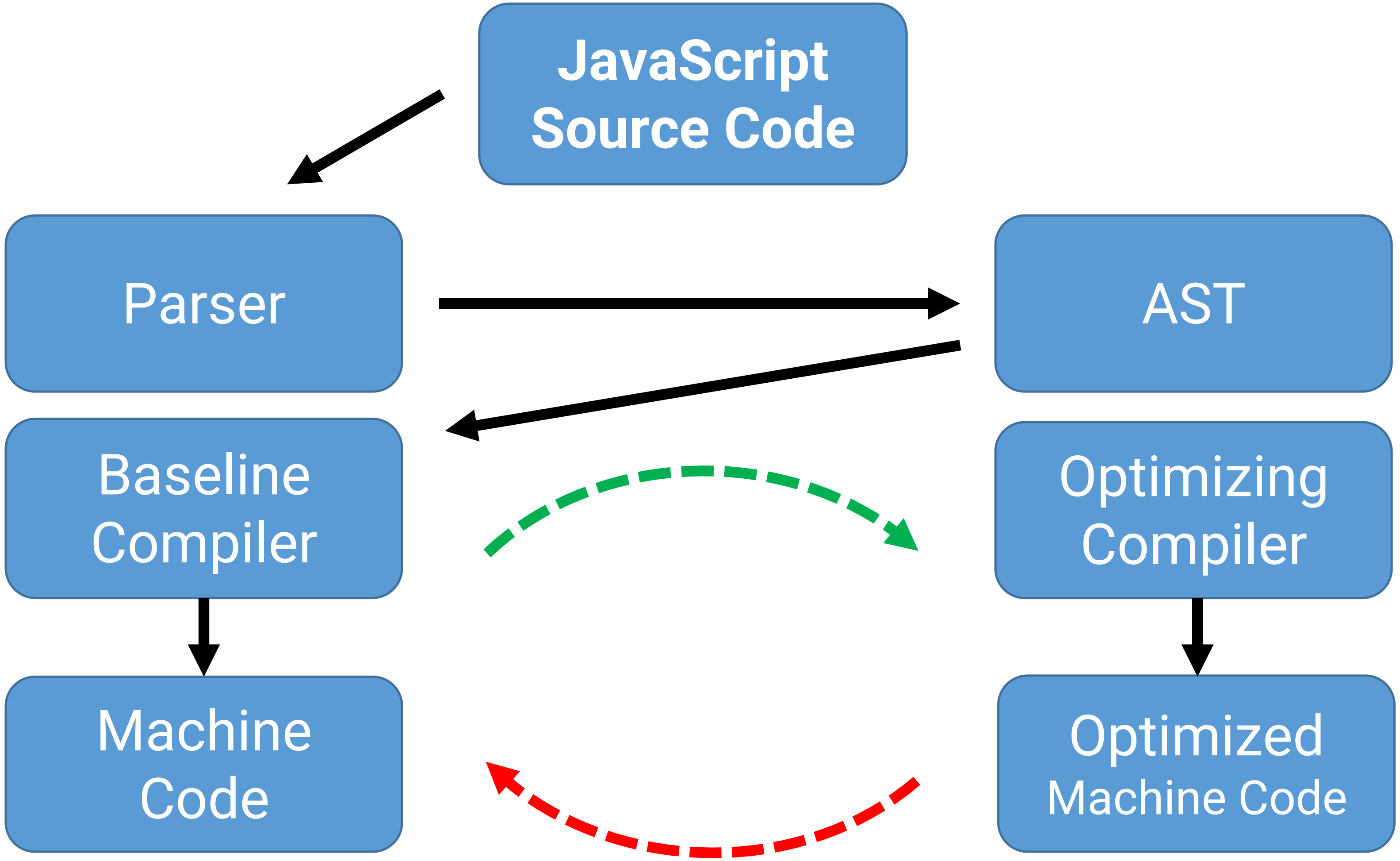
Crakan

JerryScript



Duktape

Engines - Internals



JavaScript Source Code

Parser

AST

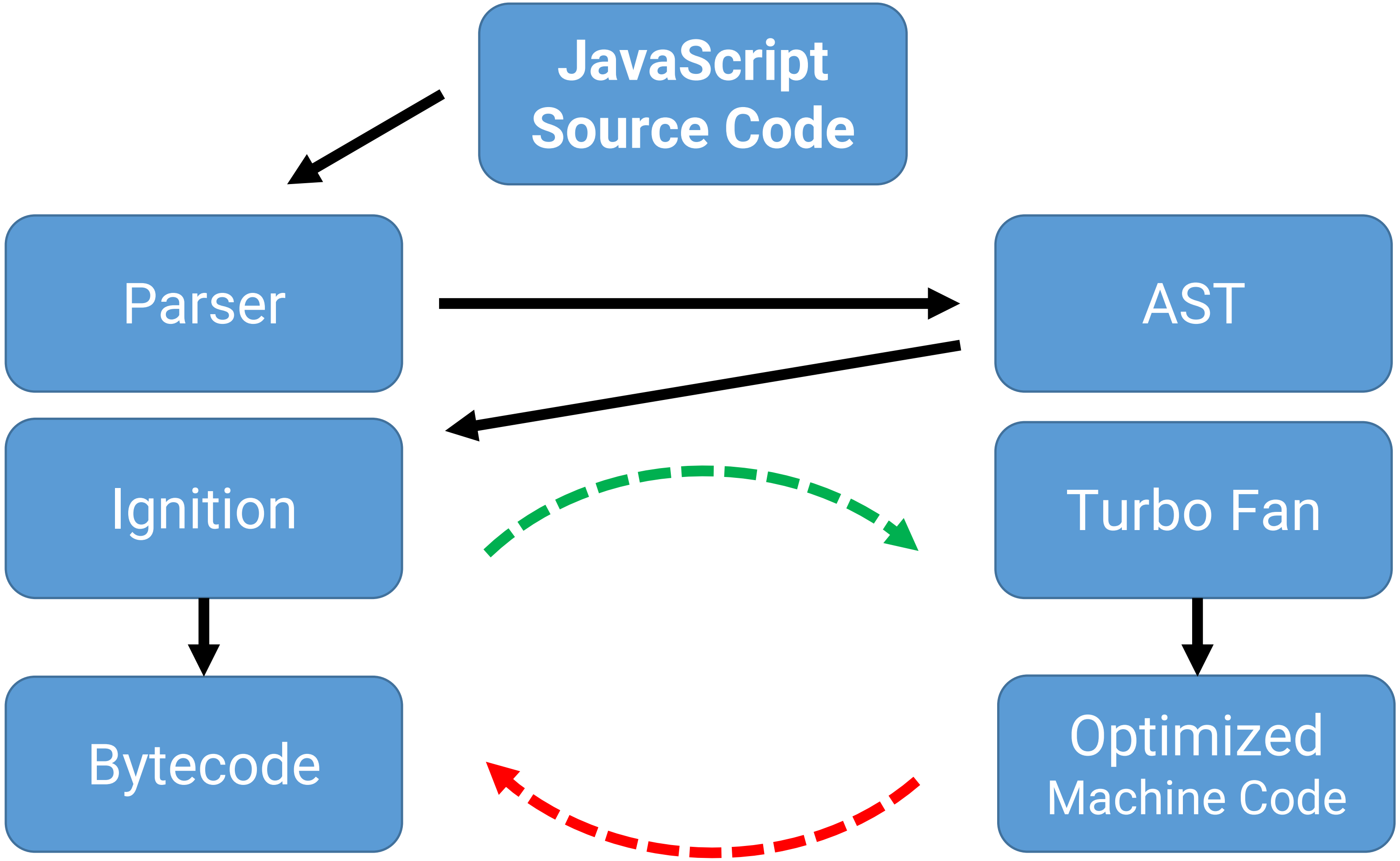
Baseline Compiler

Optimizing Compiler

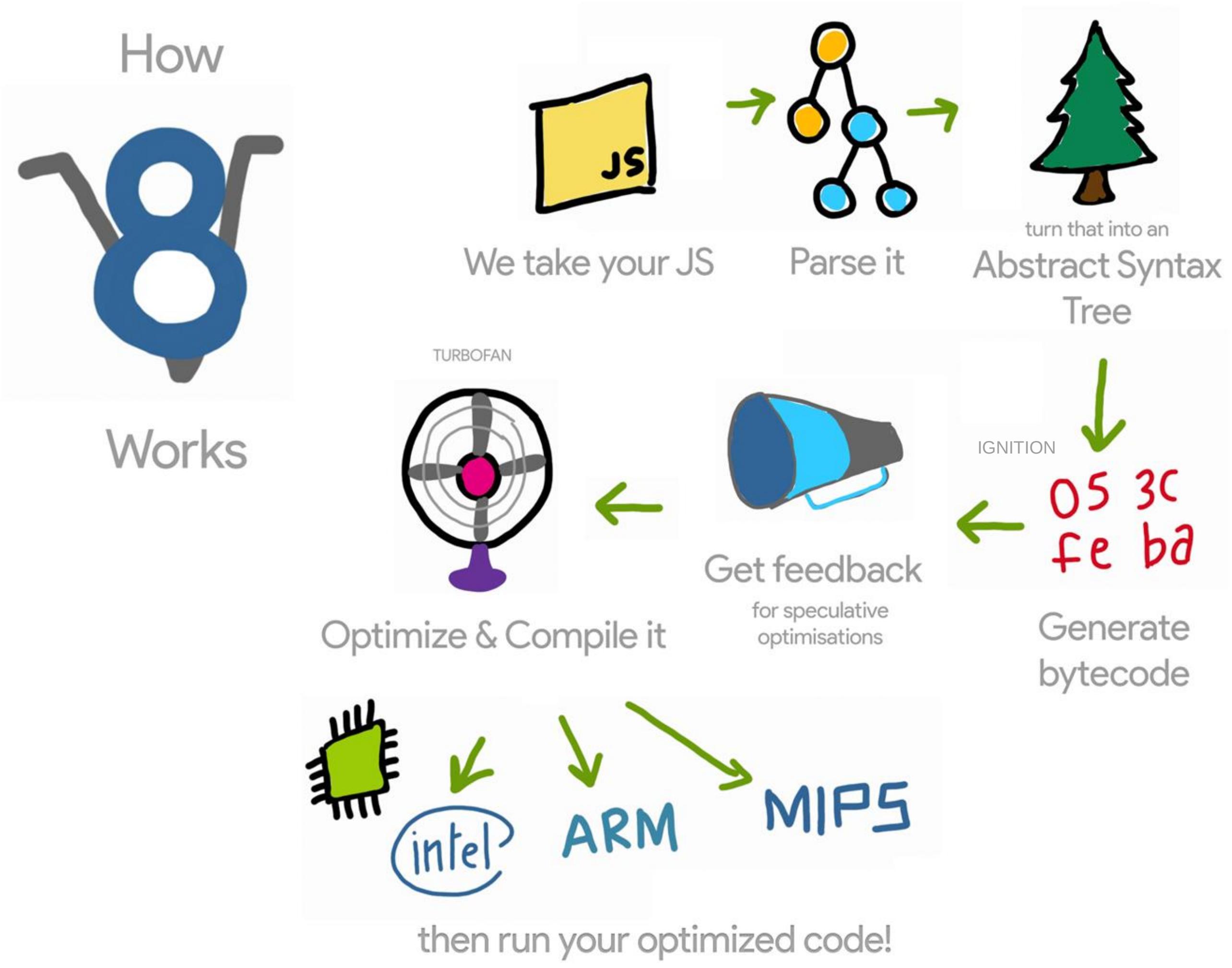
Machine Code

Optimized Machine Code

V8


JavaScript Source Code
Parser
AST
Ignition
Turbo Fan
Bytecode
Optimized Machine Code

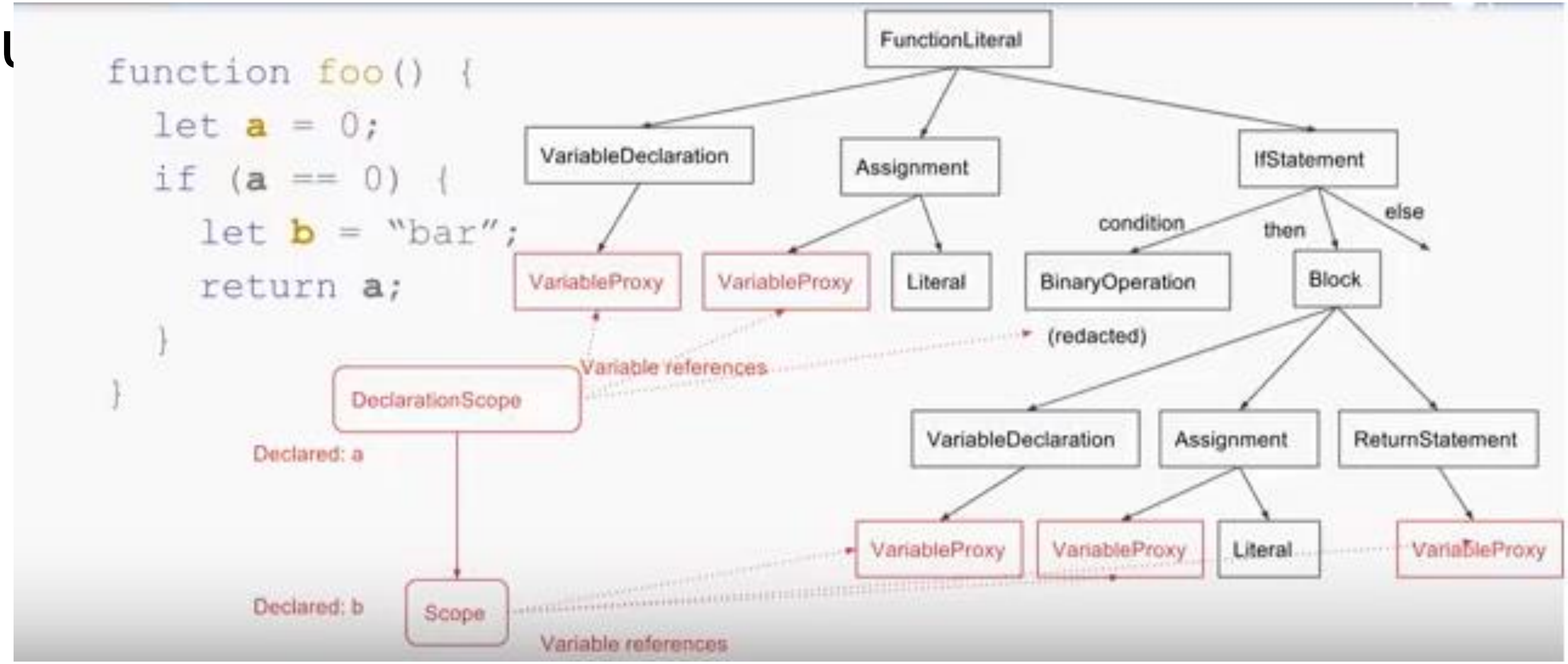
V8



By @addyosmani

Parser / AST

fu



```
function foo() {
  let a = 0;
  if (a == 0) {
    let b = "bar";
    return a;
  }
}
```

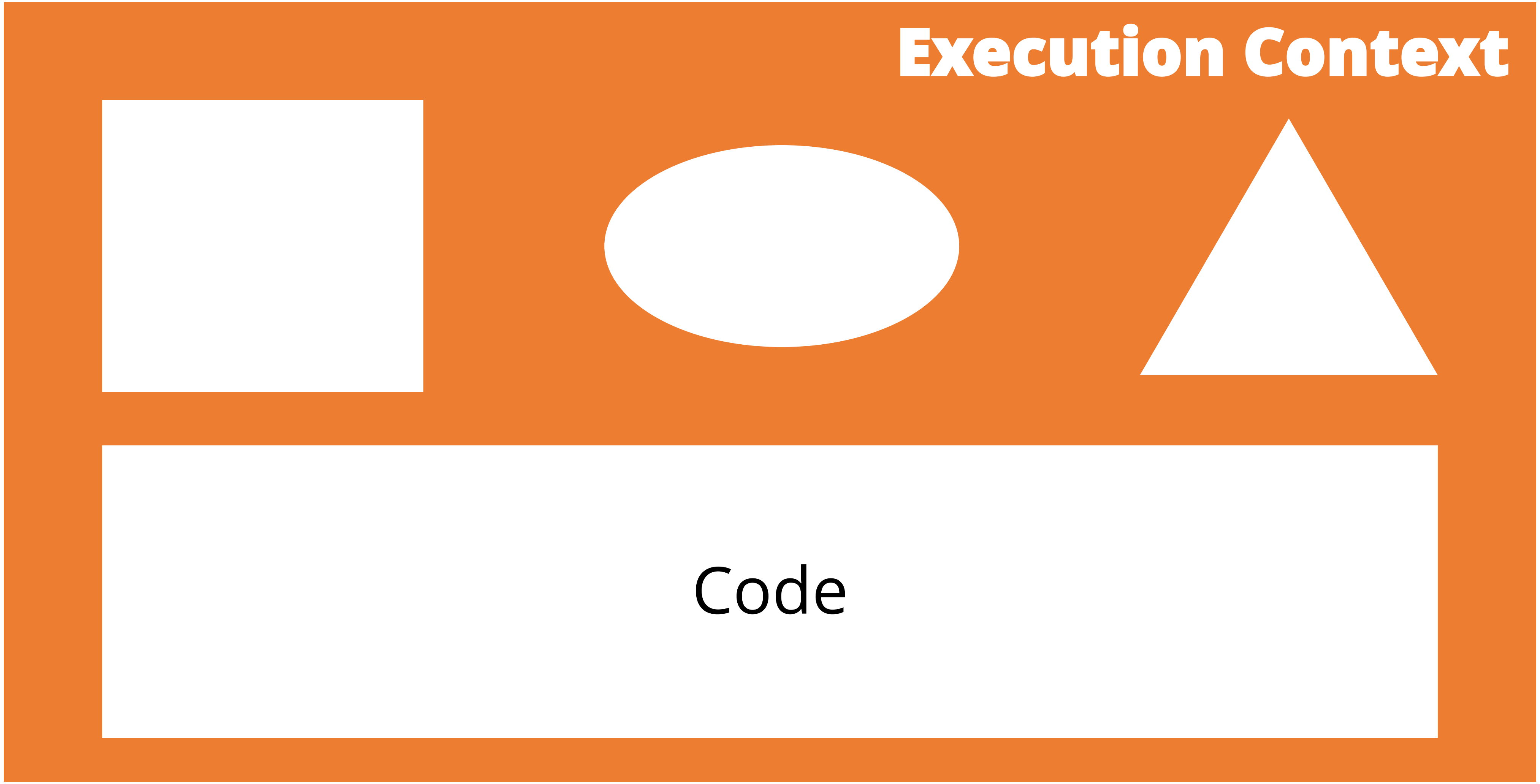
}

Definitions

- **Syntax Parser**
  - A program that reads your code and determines what it does and if its grammar is valid
- **Lexical Environment**
  - Where something sits physically in the code that you write
- **Execution Context**
  - A wrapper to help manage the code that is running

Executing Code

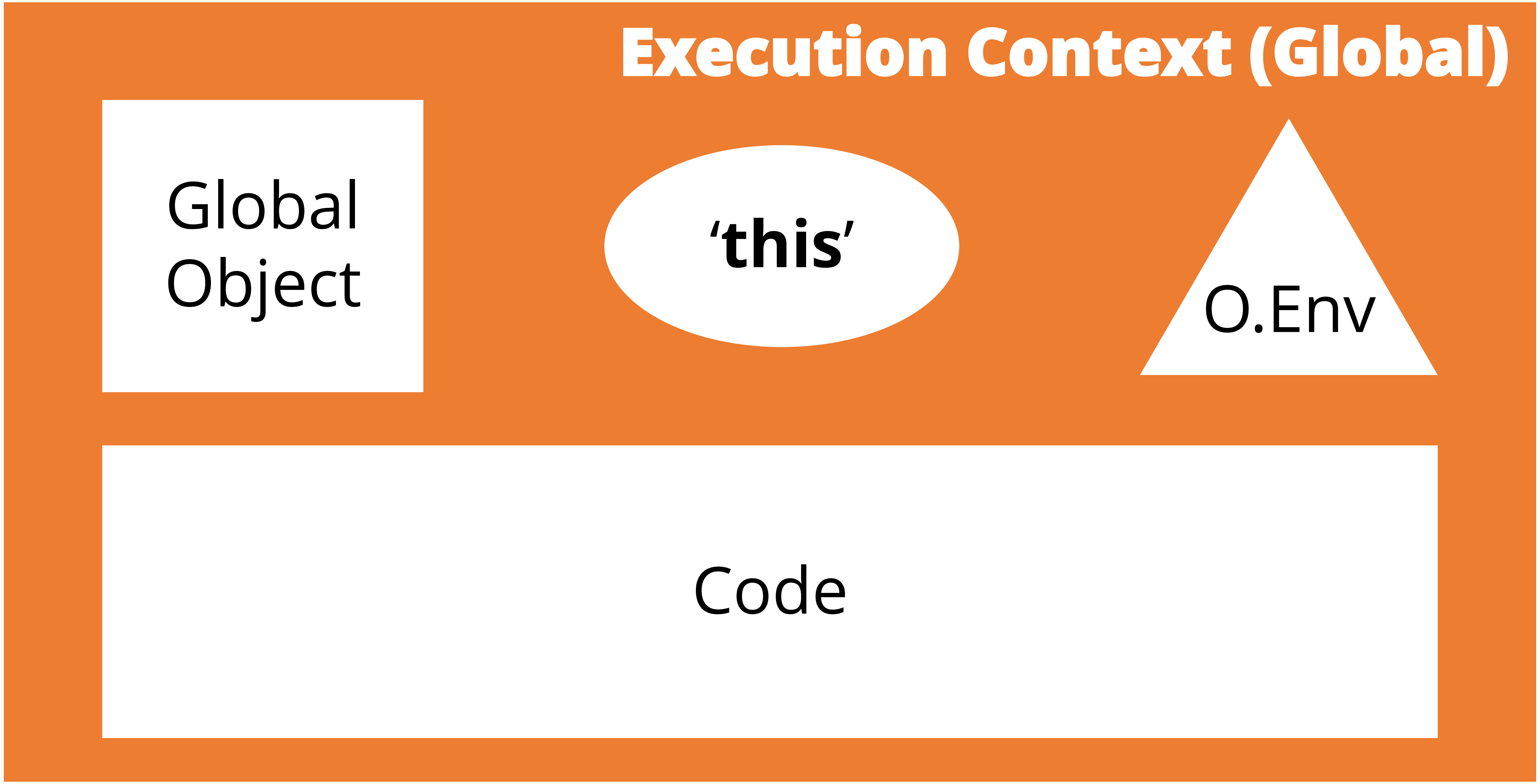
Executing Code



Execution Context

Code

Executing Code



Execution Context (Global)

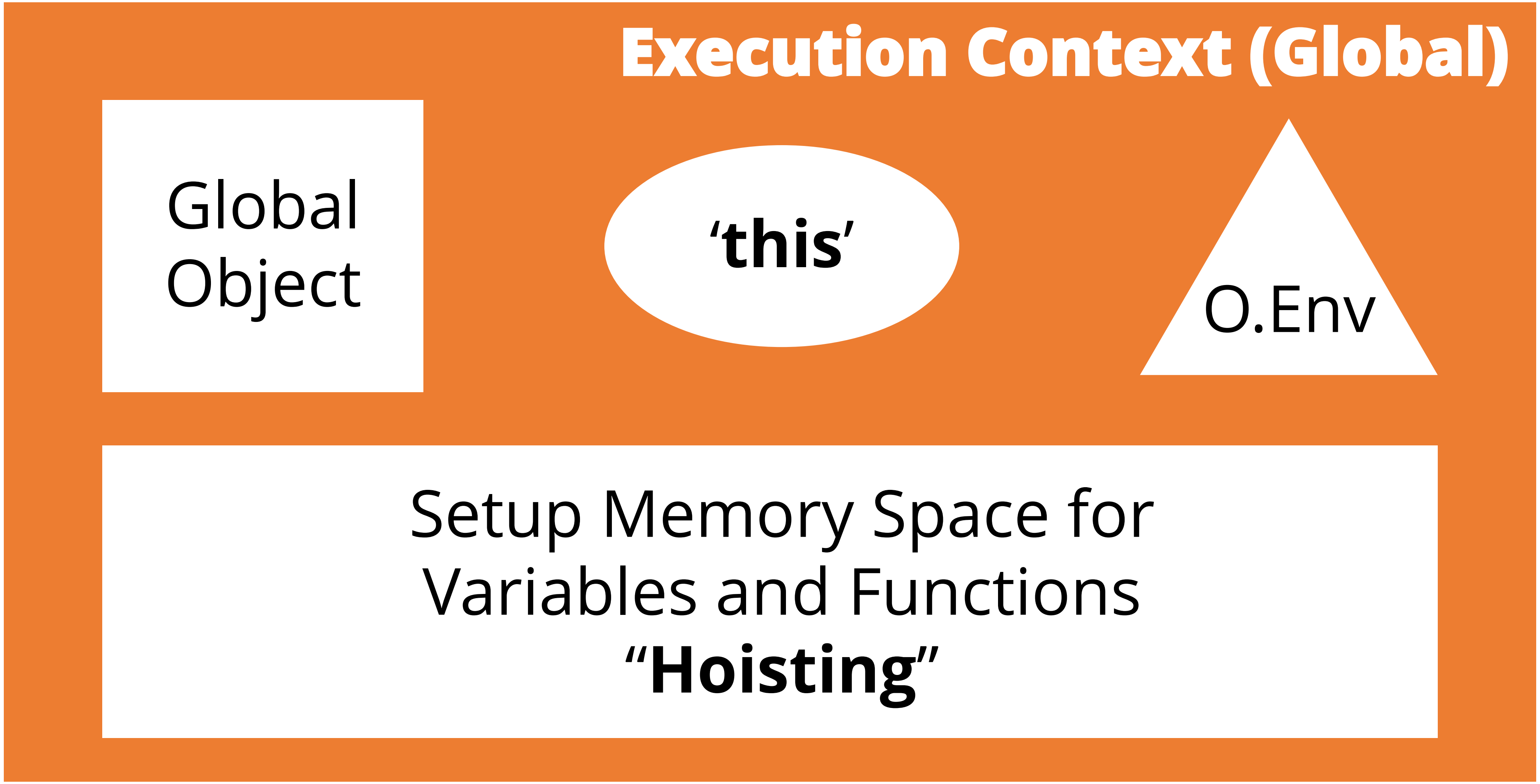
Global Object

'this'

O.Env

Code

Execution Context: Creation



Execution Context (Global)

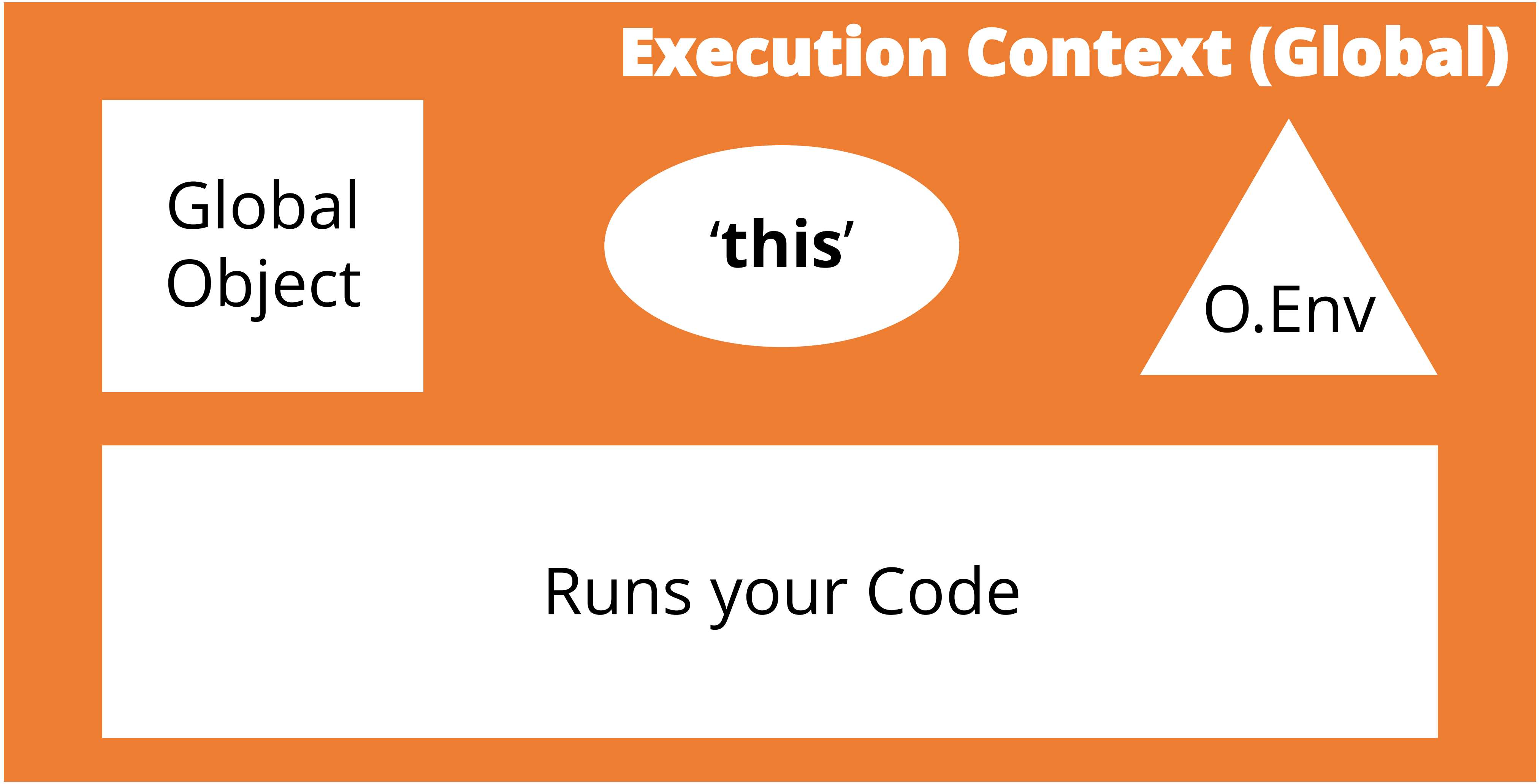
Global Object

‘**this**’

O.Env

Setup Memory Space for Variables and Functions “**Hoisting**”

Execution Context: Execution


Execution Context (Global)

Global Object

‘this’

O.Env

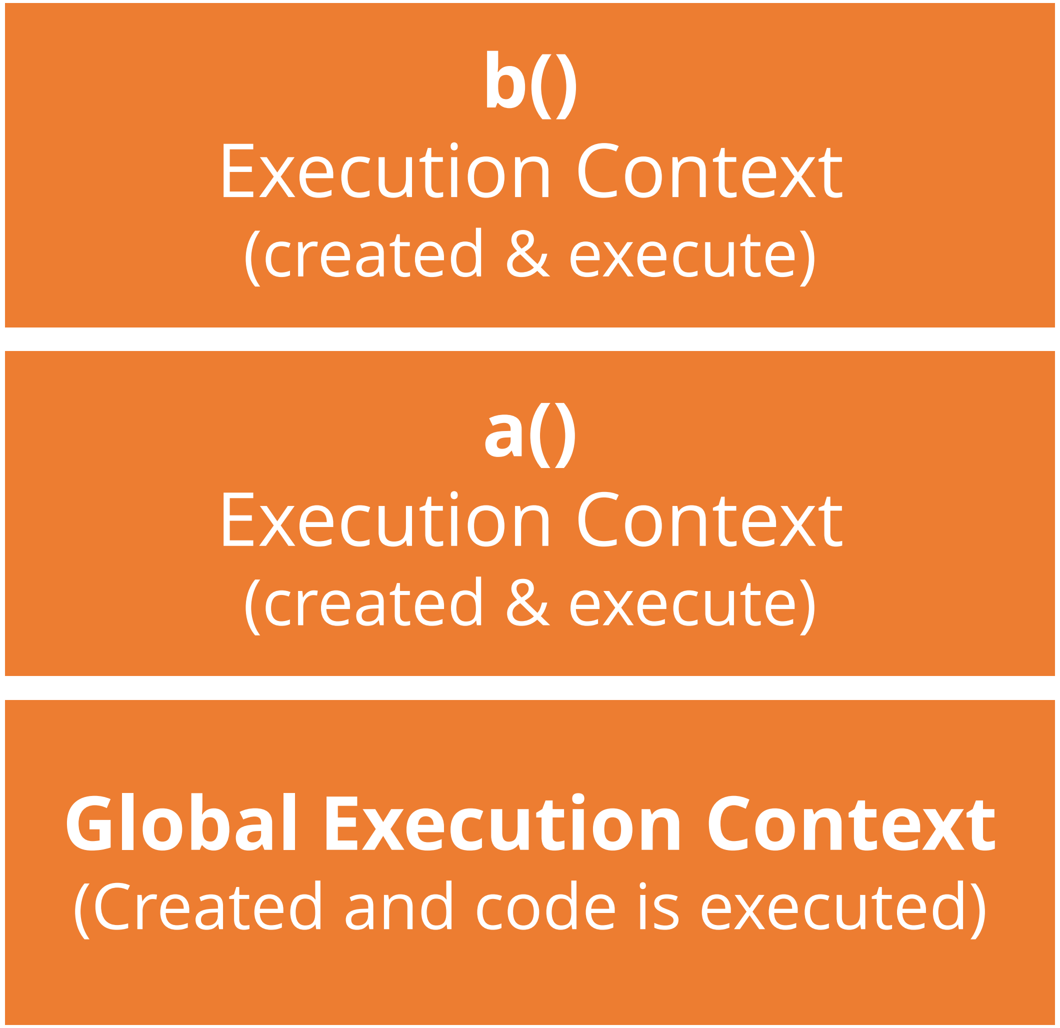
Runs your Code

Functions

```
function b() {
}

function a() {
    b();
}

a();
```



b()
Execution Context
(created & execute)

a()
Execution Context
(created & execute)

Global Execution Context
(Created and code is executed)

Interview Question 1

```
var a = 3;

function b() {
  a = 40;
  return;

  function a() {}
}

b();

console.log(a);
```

Interview Question 2

```
function foo(){

  function bar() {
    return 4;
  }

  return bar();

  function bar() {
    return 9;
  }
}

alert(foo());
```

Interview Question 3

```
function parent() {
    var hoisted = "I'm a variable";

    function hoisted() {
        return "I'm a function";
    }

    return hoisted();
}

console.log(parent());
```

Interview Question 4

```
alert(foo());

function foo() {

  var bar = function() {
    return 4;
  };

  return bar();

  var bar = function() {
    return 9;
  };

}
```

Interview Question 5

```
var myVar = 'foo';

(function() {
  console.log('Original value was: ' + myVar);
  var myVar = 'bar';
  console.log('New value is: ' + myVar);
})();
```