



# Evolução das Versões

Com a padronização, foi formado um comitê chamado de TC39 que passou a trabalhar na especificação **ECMA-262**

ECMAScript Wiki - People Worl x +

Not Secure | tc39wiki.calculist.org/about/people/

	<a href="#">Erik Arvidsson</a>	Google
	Gavin Barraclough	Apple
	Nebojša Ćirić	Google
	<a href="#">Doug Crockford</a>	eBay
	<a href="#">Brendan Eich</a>	Mozilla
	<a href="#">Stephan Herhut</a>	Intel
	<a href="#">Dave Herman</a>	Mozilla
	Luke Hoban	Microsoft
	Rick Hudson	Intel
	Oliver Hunt	Apple
	Waldemar Horwat	Google
	<a href="#">Yehuda Katz</a>	jQuery Foundation

# ECMAScript 1

(110 páginas)



ES1  
1997



Oficialização do que já havia sido feito  
até o momento da padronização pela ECMA

## 4

## Overview

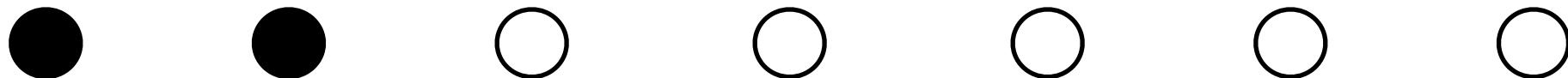
ECMAScript is an object-oriented programming language for performing computations and manipulating computational objects within a host environment. ECMAScript as defined here is not intended to be computationally self-sufficient; indeed, there are no provisions in this specification for input of external data or output of computed results. Instead, it is expected that the computational environment of an ECMAScript program will provide not only the objects and other facilities described in this specification but also certain environment-specific *host* objects, whose description and behavior are beyond the scope of this specification except to indicate that they may provide certain properties that can be accessed and certain functions that can be called from an ECMAScript program.

A *scripting language* is a programming language that is used to manipulate, customize, and automate the facilities of an existing system. In such systems, useful functionality is already available through a user interface, and the scripting language is a mechanism for exposing that functionality to program control. In this way, the existing system is said to provide a host environment of objects and facilities which completes the capabilities of the scripting language. A scripting language is intended for use by both professional and non-professional programmers, and therefore there may be a number of informalities and built into the language.

ECMAScript was originally designed to be a *Web scripting language*, providing a mechanism to enliven Web pages in browsers and to perform server computation as part of a Web-based client-server architecture. ECMAScript can

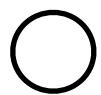
# ECMAScript 2

(117 páginas)



ES1  
1997

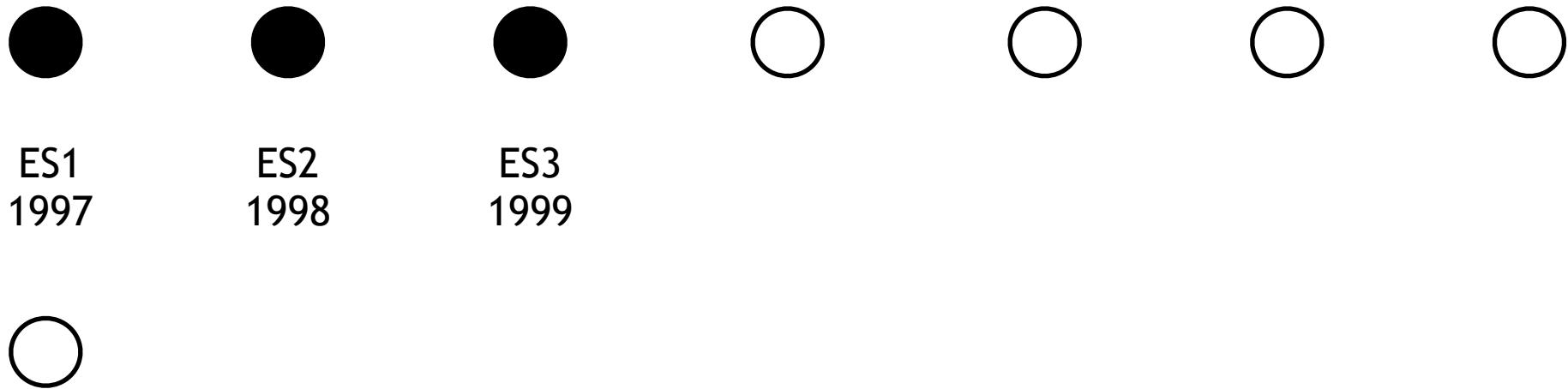
ES2  
1998



Adequação com a normativa ISO/IEC 16262

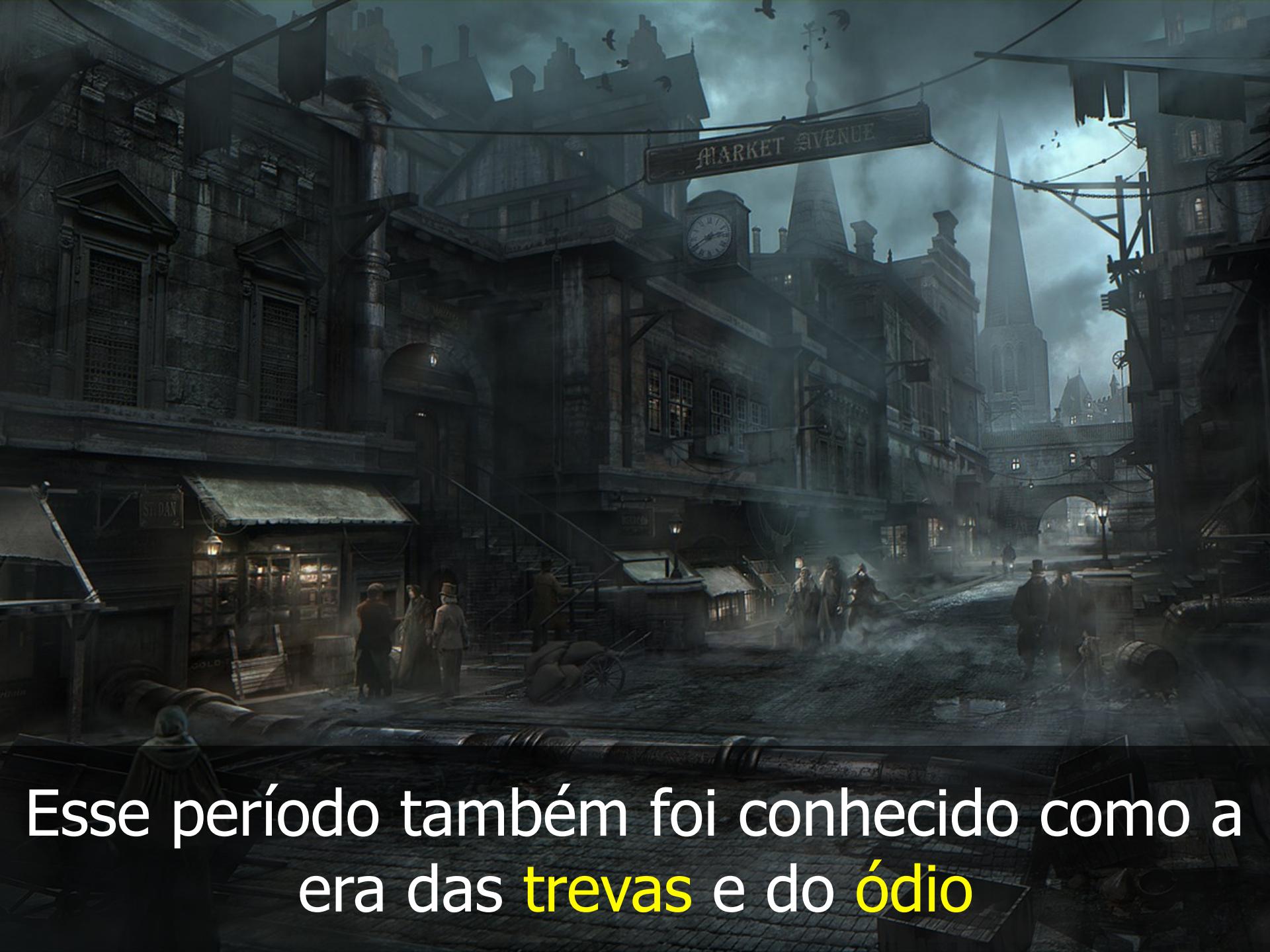
# ECMAScript 3

(188 páginas)



Exception Handling (throw/try/catch), Regular Expression, switch, do-while, ...

Por volta de 2005, com o surgimento do **AJAX**, a linguagem ganhou vida nova



Esse período também foi conhecido como a era das **trevas** e do ódio



Além da linguagem deixar a desejar, pouca gente sabia utilizá-la corretamente

# ECMAScript 5

(252 páginas)



ES1  
1997



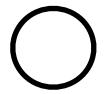
ES2  
1998



ES3  
1999



ES5  
2009



JSON, strict mode, reserved words as property keys,  
multiline string, Object API, Array.prototype.\*



Passaram-se cerca de **10 anos** sem o  
lançamento de novas versões



Por que tanto tempo se passou?



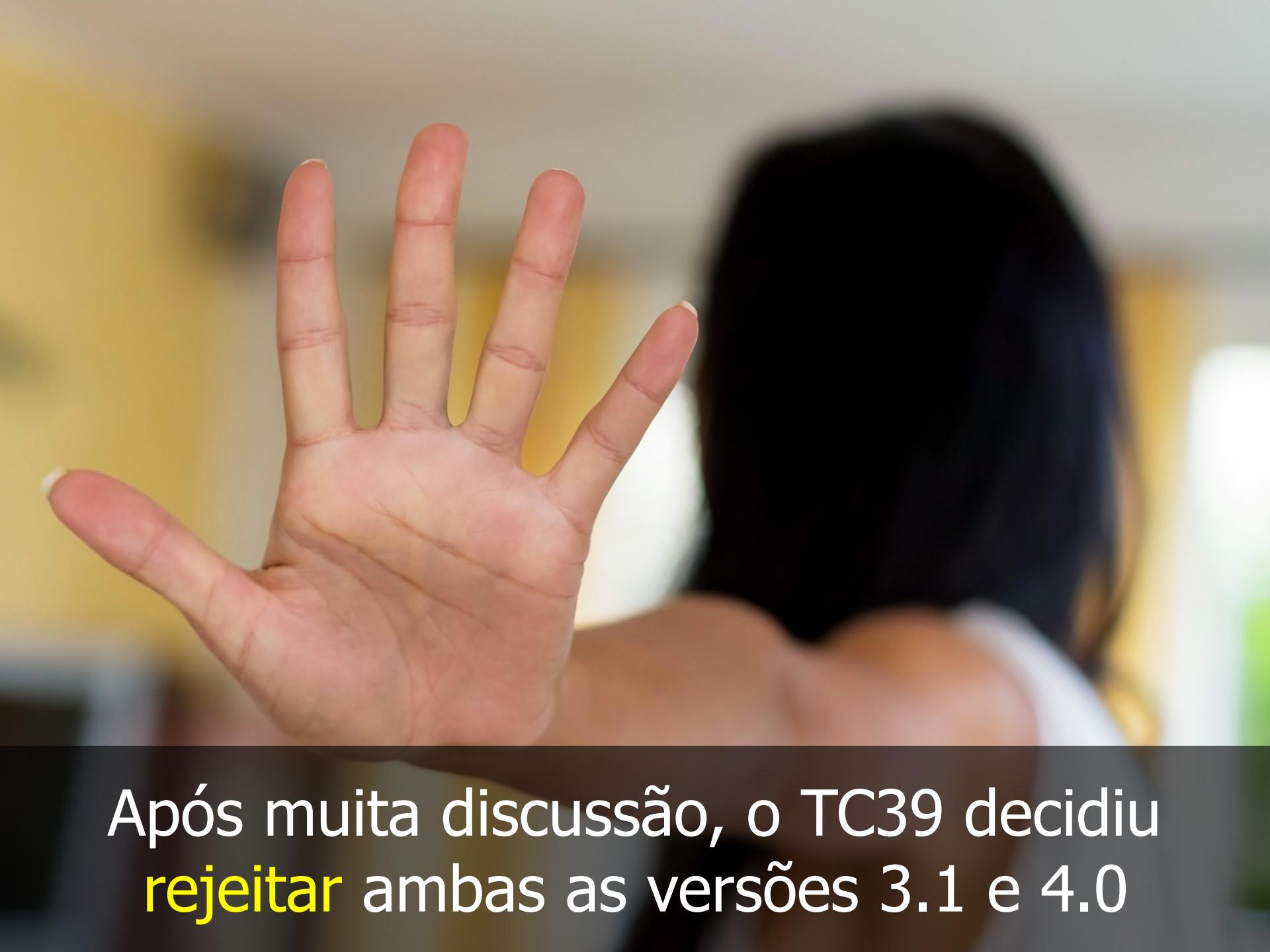
Houve uma **separação** dos grupos que especificavam as versões 3.1 e 4.0



Microsoft e Yahoo estavam na 3.1 e Adobe,  
Mozilla, Opera e Google na 4.0

A close-up photograph of a light-colored wooden board with a grid of circular holes. A dark, rectangular piece of wood is partially inserted into one of the holes, creating a visual metaphor for integration or compatibility.

Após mais de um ano de trabalho em  
paralelo, as versões não eram compatíveis



Após muita discussão, o TC39 decidiu  
**rejeitar** ambas as versões 3.1 e 4.0

# ECMAScript Harmony

Brendan Eich [brendan at mozilla.org](mailto:brendan@mozilla.org)

Wed Aug 13 14:26:56 PDT 2008

- Previous message: [Old documents marked "New"](#)
  - Next message: [ECMAScript Harmony](#)
  - **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#)
- 

It's no secret that the JavaScript standards body, Ecma's Technical Committee 39, has been split for over a year, with some members favoring ES4, a major fourth edition to ECMA-262, and others advocating ES3.1 based on the existing ECMA-262 Edition 3 (ES3) specification. Now, I'm happy to report, the split is over.

The Ecma TC39 meeting in Oslo at the end of July was very productive, and if we keep working together, it will be seen as seminal when we look back in a couple of years. Before this meeting, I worked with John Neumann, TC39 chair, and ES3.1 and ES4 principals, especially Lars Hansen (Adobe), Mark Miller (Google), and Allen Wirfs-Brock (Microsoft), to unify the committee around shared values and a common roadmap. This message is my attempt to announce the main result of the meeting, which I've labeled "Harmony".

## Executive Summary

The committee has resolved in favor of these tasks and conclusions:

1. Focus work on ES3.1 with full collaboration of all parties, and target two interoperable implementations by early next year.
2. Collaborate on the next step beyond ES3.1, which will include syntactic extensions but which will be more modest than ES4 in both semantic and syntactic innovation.

# ECMAScript 5.1

(258 páginas)



ES1  
1997



ES2  
1998



ES3  
1999



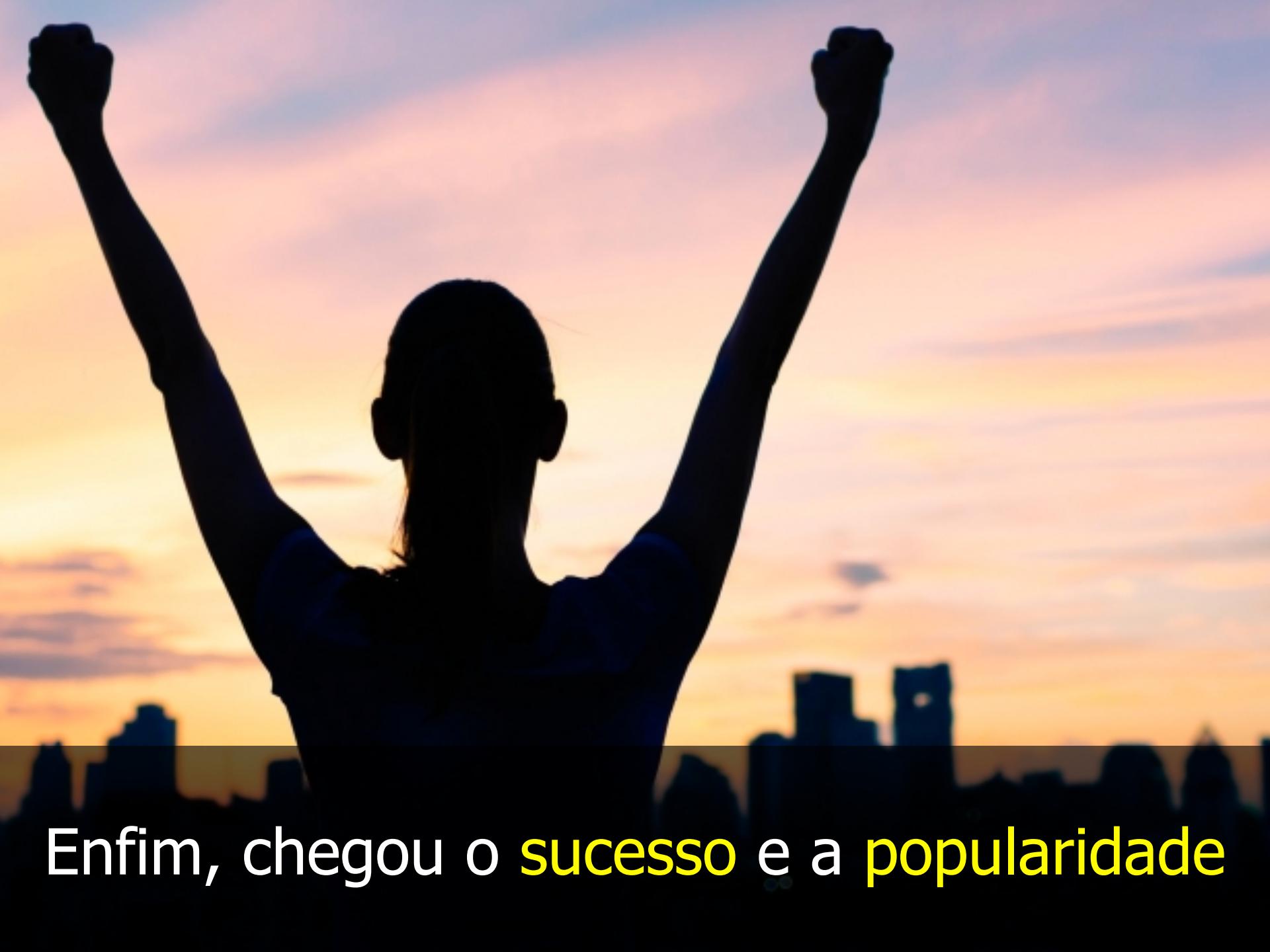
ES5  
2009



ES5.1  
2011



Adequação com a normativa ISO/IEC 16262

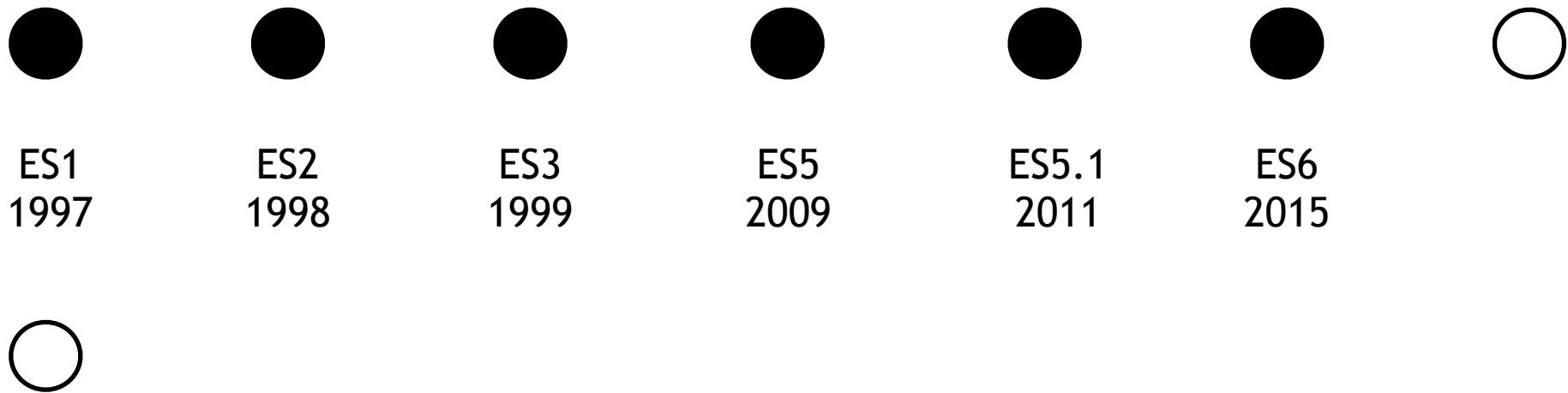


Enfim, chegou o sucesso e a popularidade



# ECMAScript 6 / ECMA2015

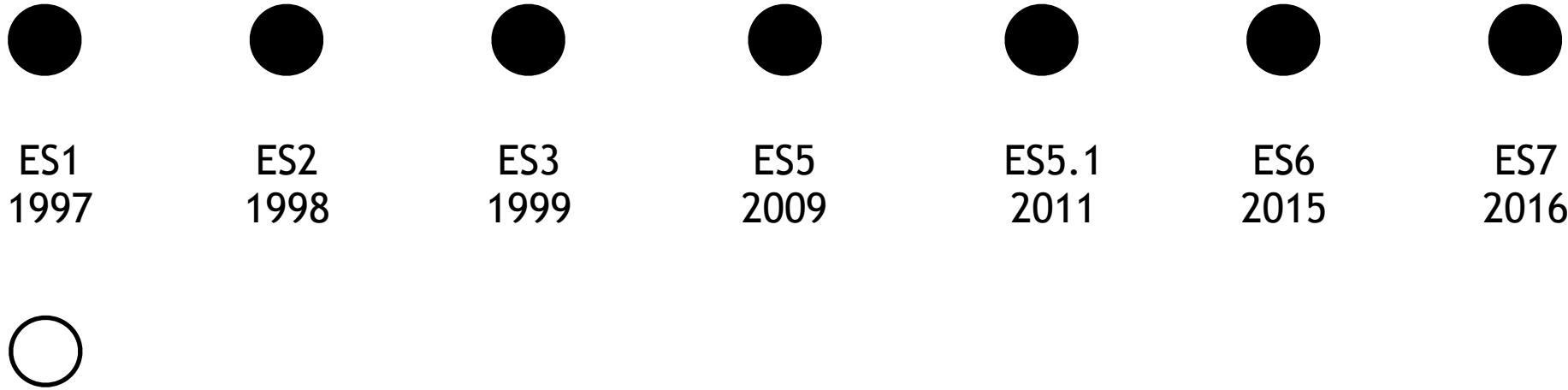
(566 páginas)



Class, Arrow Function, Proxy, Reflect, Map, Set,  
Destructuring, Rest Parameter, Default Value, Template  
Literal, Spread Operator, Generators, Promises, Modules

# ECMAScript 7 / ECMA2016

(586 páginas)



Array.prototype.includes, Exponentiation operator...

# ECMAScript 8 / ECMA2017

(885 páginas)

●	●	●	●	●	●	●
ES1 1997	ES2 1998	ES3 1999	ES5 2009	ES5.1 2011	ES6 2015	ES7 2016



ES8  
2017

Async/Await, Object.values, Object.entries,

String.prototype.padStart, String.prototype.padEnd,

Trailling commas in parameters list, objects and arrays...



Como o ES6 ficou grande demais, a partir do ES7, as versões passaram a ser anuais



CAUTION

Cuidado com problemas de compatibilidade  
ao utilizar novas funcionalidades

