

# JavaScript history

[tsevdos.me](http://tsevdos.me) / [@tsevdos](https://twitter.com/tsevdos)

# Rules

Feel free to interrupt me for:

- questions
- relevant comments

# Agenda

- Brendan Eich and Netscape
- JavaScript's timeline
- Advantages and Disadvantages

# Introduction

**JavaScript** is a versatile, high-level programming language that is integral to modern web development, enabling **interactive and dynamic content in websites and applications.**

# Brendan Eich and Netscape

**Brendan Eich**, was hired in April 1995 by **Netscape**, and was initially commissioned to insert **Scheme**, a functional programming language, in the **Netscape browser**.





# Brendan Eich and Netscape

In just **10 days** Brendan created one with the Scheme functionality, Self's typical object orientation and Java-like syntax. The first version was called **Mocha**, then renamed **LiveScript** and finally **JavaScript**.

# Timeline

- **May 1995:** Project Mocha starts
- **September 1995:** Project release as LiveScript
- **December 1995:** Renamed to JavaScript
- **1996:** it was standardized as ECMA Script to ECMA (European Computer Manufacturers Association)
- **ECMA Script** is a standardized specification ensuring compatibility across implementations and settings, while **JavaScript** implements with environment-specific features

# Timeline

- **ECMA Technical Committee 39** – also known as **TC39** – is the committee which defines **ECMAScript (JavaScript)** and related standards
- **ES1**: June 1997 (ECMA-262)
- **ES2**: June 1998
- **ES3**: December 1999
- **ES4**: Abandoned

# Timeline

- **ES5:** December 2009
  - Many improvements in OOP
  - Strict mode
  - JSON
- The same year (2009), **node.js** a JavaScript runtime is running JS code on the server!

# Timeline

- **ES2015 (or ES6): June 2015**
  - classes
  - arrow functions
  - promises
  - template strings
  - transpilers (babel and tracer)
- **ES2016 - ES2024**

# JavaScript Usage

- **Web Development:** Client-side (React, Angular) and Server-side (Node.js, Deno)
- **Mobile Apps:** PWAs, React Native, Ionic
- **Desktop Apps:** Electron, NW.js
- **IoT:** Johnny-Five, Espruino

# JavaScript Advantages

- Native language for browsers: Universally supported by all major browsers
- Open standard: Governed by ECMA International, promoting broad compatibility
- Huge community: Extensive resources, support, and contributions from developers worldwide
- Rich ecosystem: Numerous tools, frameworks, libraries, and plugins available for diverse needs

# JavaScript Disadvantages

- Quirks: Issues with inheritance, type coercion, and browser inconsistencies
- Fragmentation: Multiple environments (browser engines, Node.js, Deno, Bun) can lead to compatibility challenges
- Overabundance of tools: The vast number of tools and frameworks can be overwhelming and lead to decision fatigue

**Happy coding!**