



PUC Minas

# Tecnologias Web

## Linguagem JavaScript

# Javascript – Tópicos

- O que é o JavaScript
- Histórico do JavaScript e ECMA Script (ES)
- Implementações do ECMA Script
- JavaScript vs Java
- Formas de Utilização
- Capacidades do JavaScript
- Console Web do Browser



# O que é JavaScript?

HTML define a **estrutura do conteúdo**

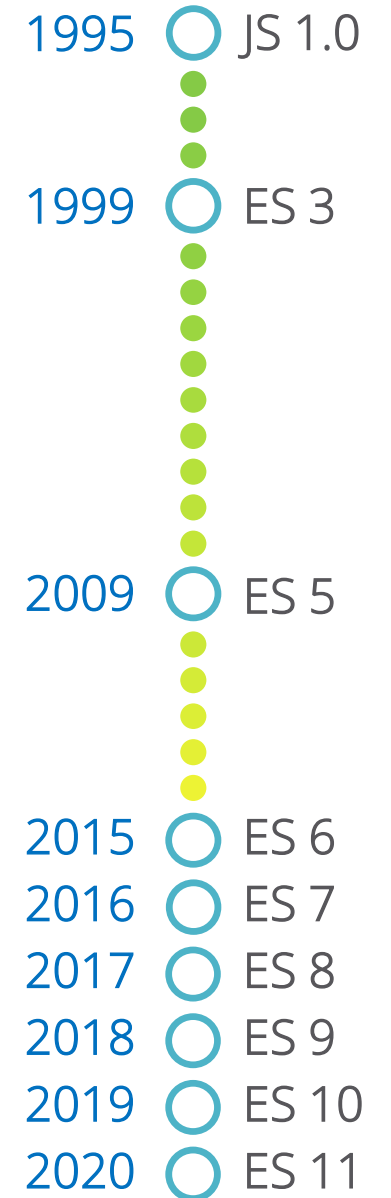
CSS define o **formato de apresentação**

Javascript adiciona a **interatividade** e permite a criação de **aplicações ricas**

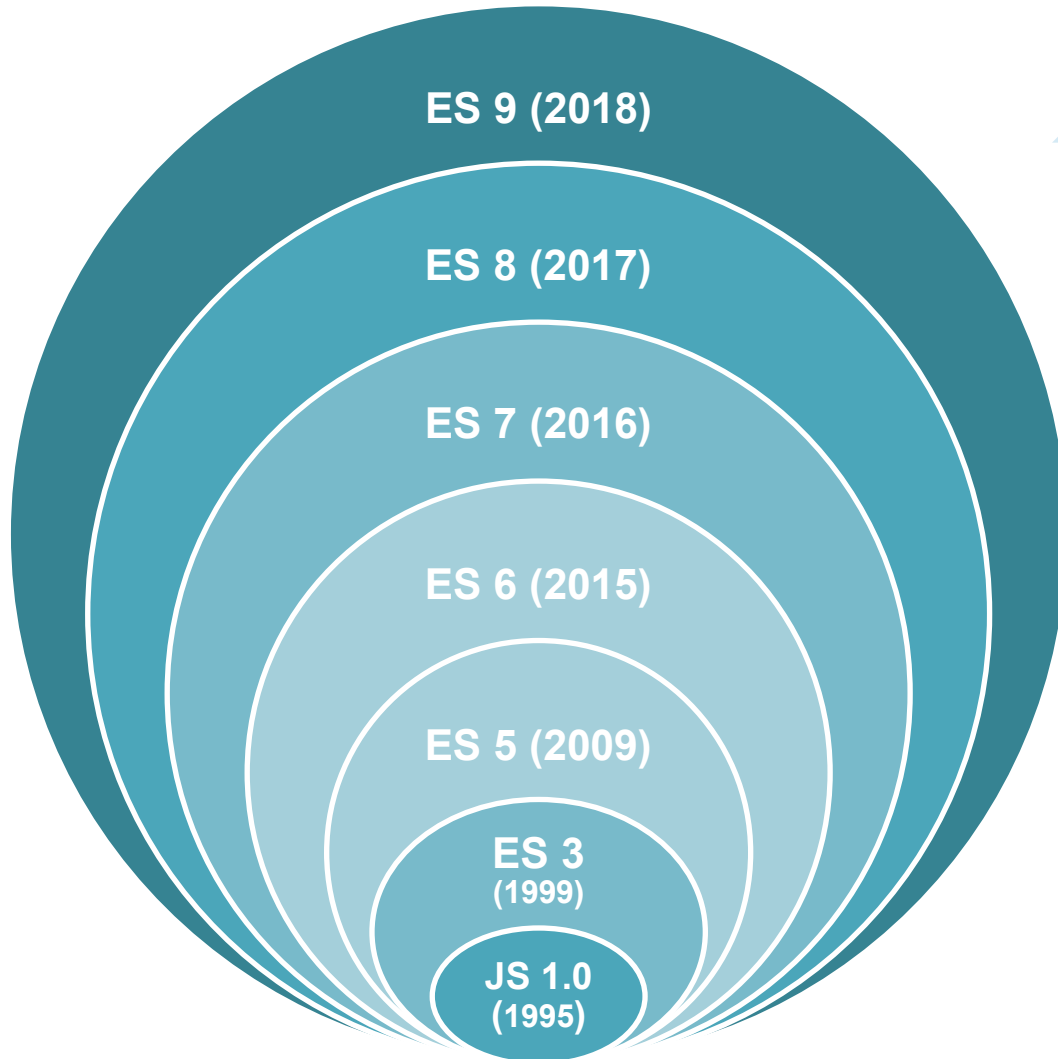
Fonte: Mozilla Developer Network ([https://developer.mozilla.org/en-US/docs/Web/JavaScript/JavaScript\\_technologies\\_overview](https://developer.mozilla.org/en-US/docs/Web/JavaScript/JavaScript_technologies_overview))

# Histórico do JavaScript

- O **JavaScript** foi criado em 1995 por **Brendan Eich** na Netscape
- Em 1996, foi transferido para o **ECMA International** para padronização, sendo a primeira versão do ES lançada em 1997.
- Em 1998, a linguagem é aprovada como a **ISO/IEC 16262** e batizada então como **ECMA-262**
- Em 2005, o advento do **AJAX** alavancou o uso do Javascript
- Em 2009 foi lançado o **ECMA Script 5 (ES 5)**, versão com maior compatibilidade nos browsers atualmente
- Em 2009 foi criado o projeto **CommonJS** com o objetivo de levar o JavaScript para além do Browser.
- Em 2015 foi lançado o **ECMA Script 6 (ES6)**, atualmente em estágio avançado de adoção pelos browsers
- O TC39 (<https://github.com/tc39>) é o Comitê Técnico responsável pela evolução do ECMAScript.



# Histórico do JavaScript

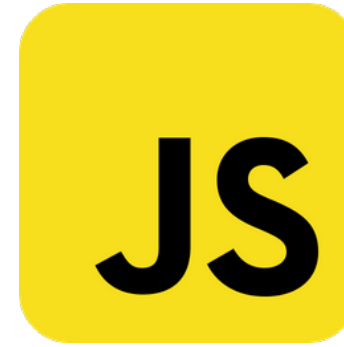


- **ECMA Script 9 (2018)**  
Regex Changes, Promise Finally, Async Iteration
- **ECMA Script 8 (2017)**  
Async functions, Shared Memory, Atomics
- **ECMA Script 7 (2016)**  
Array Includes, Exponential operator (\*\*)
- **ECMA Script 6 (2015)**  
Classes, Modules, Arrow Functions, Const e Let, Template Literals, Promises
- **ECMA Script 5**  
Strict mode, Suporte ao JSON
- **ECMA Script 3**  
Expressões regulares (RegExp), Métodos da classe String, Tratamento de exceção (try/catch)
- **JavaScript 1.0**

# Browsers Engines e ECMAScript engines

Browser / Ambiente	Web Engine	ECMAScript Engine
Mozilla Firefox	Gecko	<u><a href="https://developer.mozilla.org/pt-BR/docs/Mozilla/Projects/SpiderMonkey">Spider Monkey</a></u> <a href="https://developer.mozilla.org/pt-BR/docs/Mozilla/Projects/SpiderMonkey">https://developer.mozilla.org/pt-BR/docs/Mozilla/Projects/SpiderMonkey</a>
Google Chrome	Blink	<u><a href="https://developers.google.com/v8/?hl=pt">Google V8</a></u> <a href="https://developers.google.com/v8/?hl=pt">https://developers.google.com/v8/?hl=pt</a>
Apple Safari	WebKit	<u><a href="https://developer.apple.com/documentation/javascriptcore">JavaScriptCore</a></u> <a href="https://developer.apple.com/documentation/javascriptcore">https://developer.apple.com/documentation/javascriptcore</a>
Internet Explorer	Trident	Chakra Core <a href="https://github.com/Microsoft/ChakraCore">https://github.com/Microsoft/ChakraCore</a>
Edge	EDGE	Chakra Core <a href="https://github.com/Microsoft/ChakraCore">https://github.com/Microsoft/ChakraCore</a>
		<u><a href="#">Rhino</a></u>
Opera	Presto → Blink	

# JavaScript vs Java



- Apesar de ter uma sintaxe semelhante à Java, é totalmente diferente no conceito e uso
- Javascript é baseada em objetos enquanto Java é baseada em classes
- Oferece tipagem dinâmica, tipos de variáveis não são definidos previamente
- Javascript é uma linguagem interpretada, Java é compilada
- O código Javascript é executado no Navegador, enquanto o código Java é executado na JVM (Java Virtual Machine)
- O aplicativo Javascript depende da página Web enquanto o aplicativo Java é independente

# Formas de Utilização do JavaScript

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exemplo Javascript</title>
    <script type="text/javascript" src="script.js"></script>

    <script type="text/javascript">
      /* código Javascript */
      alert ('Passei por aqui!');
    </script>
  </head>
  <body>
    <p onClick="alert('Voce clicou no parágrafo');">...</p>
  </body>
</html>
```

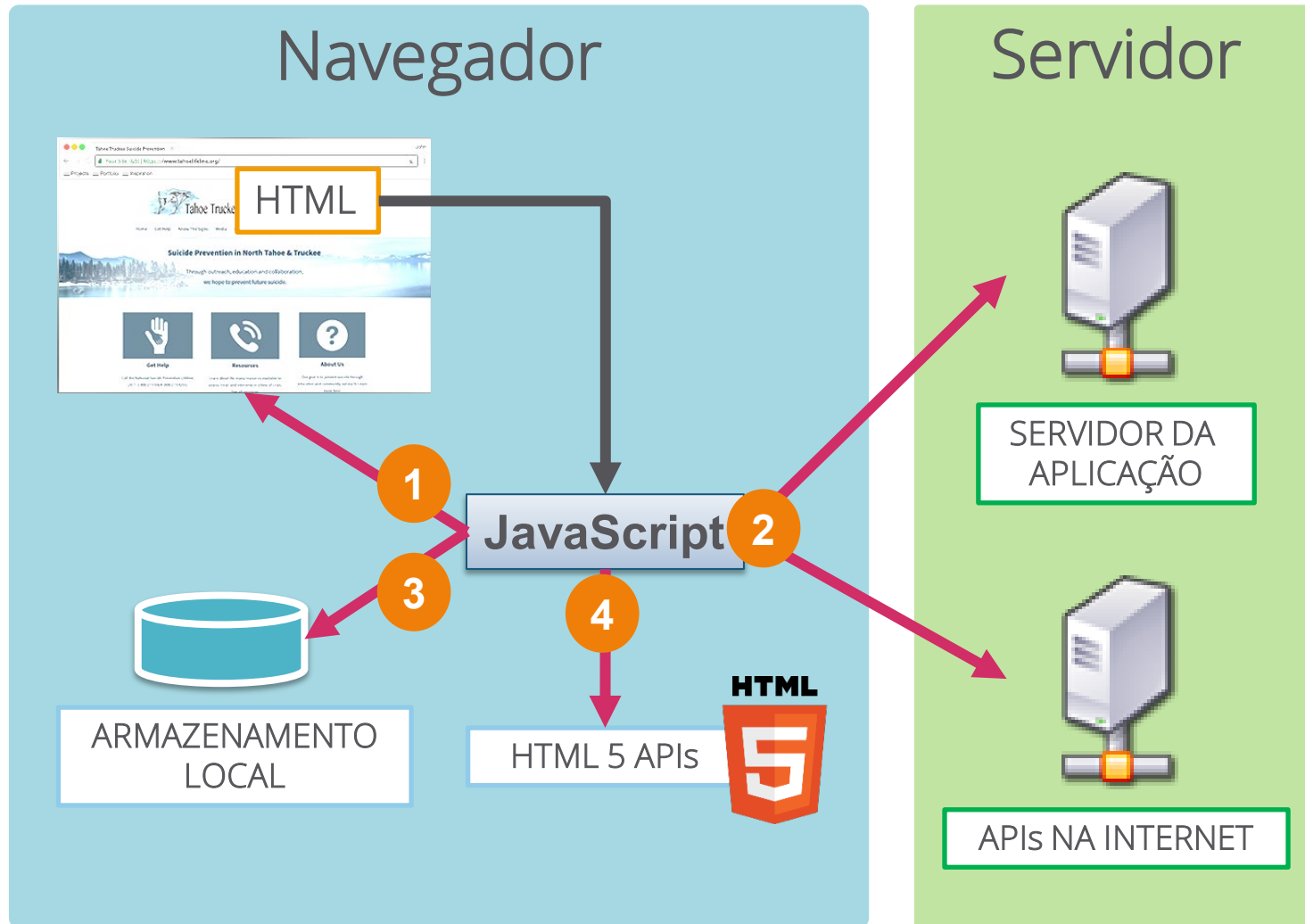
Arquivo JS externo

Bloco interno

Inline

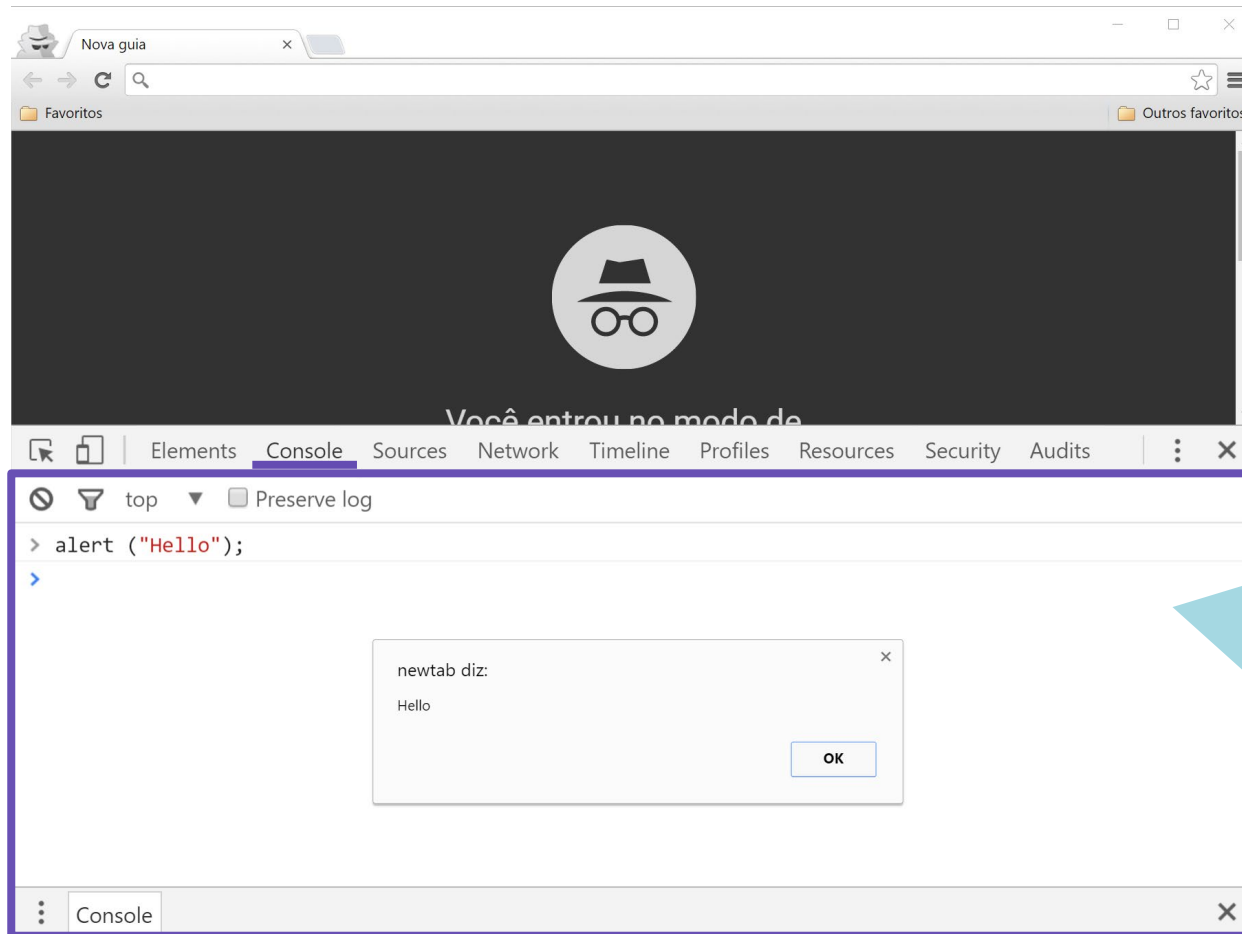


# Capacidades do JavaScript



- 1 Manipulação de objetos da página HTML e tratamento de eventos (**DOM Document Object Model**)
- 2 Comunicação com o servidor e uso de Application Programming Interface (API) via AJAX (**XMLHttpRequest**)
- 3 Persistência de dados em estruturas providas pelo Browser (**Indexed DB e LocalStorage**)
- 4 Interação com recursos das novas APIs do HTML 5 (**Canvas, Media, File, Drag and Drop, Geolocation, Web Workers, History**)

# Console Web do Browser



## Console

Para ativar, selecione a opção Ferramentas do Desenvolvedor (Google Chrome) ou Web Console (Mozilla Firefox)

Apple: ⌘ + I

Windows: Ctrl + Shift + I

## Funcionalidades

Mostra informações sobre a página corrente Inclui uma linha de comando para testes de expressões Javascript

# Obrigado!