

Curso de

# JavaScript

# Funcionamento do curso

---

**O curso terá 8 aulas de conteúdo e 1 aula de avaliação**

Todas presenciais no Unibave



---

**O curso abordará os seguintes temas**

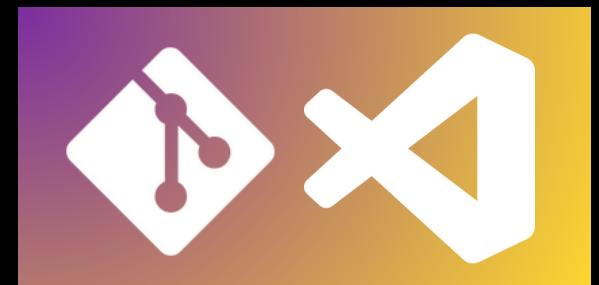
Fundamentos, DOM, AJAX, conceituação de frameworks, dev. front-end e APIs REST



---

**Softwares que usaremos**

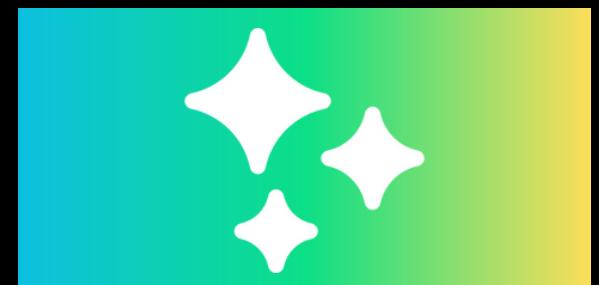
Git, VS code



---

**Material extra**

Apostilas do conteúdo, Guias práticos (cheat sheets) para Git e VS Code



# RoadMap

- **15/08 - Introdução e conceitos básicos**  
Sintaxe, Variáveis, Tipos, Operadores
- **22/08 - DOM (Document Object Model)**  
O que é, Seletores, Criar e modificar elementos na DOM
- **29/08 - Funções e Estruturas de Controle**  
Declaração, Parâmetros e retorno, Funções anônimas e Arrow function
- **05/09 - Objetos**  
Criação e manipulação, Métodos, Arrays
- **12/09 - Estruturas de controle**  
Condicionais, Repetições, Ternario
- **19/09 - AJAX**  
Tipos de eventos, Adição de eventos, requisições, Promessas.
- **26/09 - Frameworks**  
Por que usar?, React, Vue, Angular, Quando usar?
- **03/10 - Front-end X APIs**  
Extrutura de front-end, Consumindo APIs
- **10/10 - Avaliação**

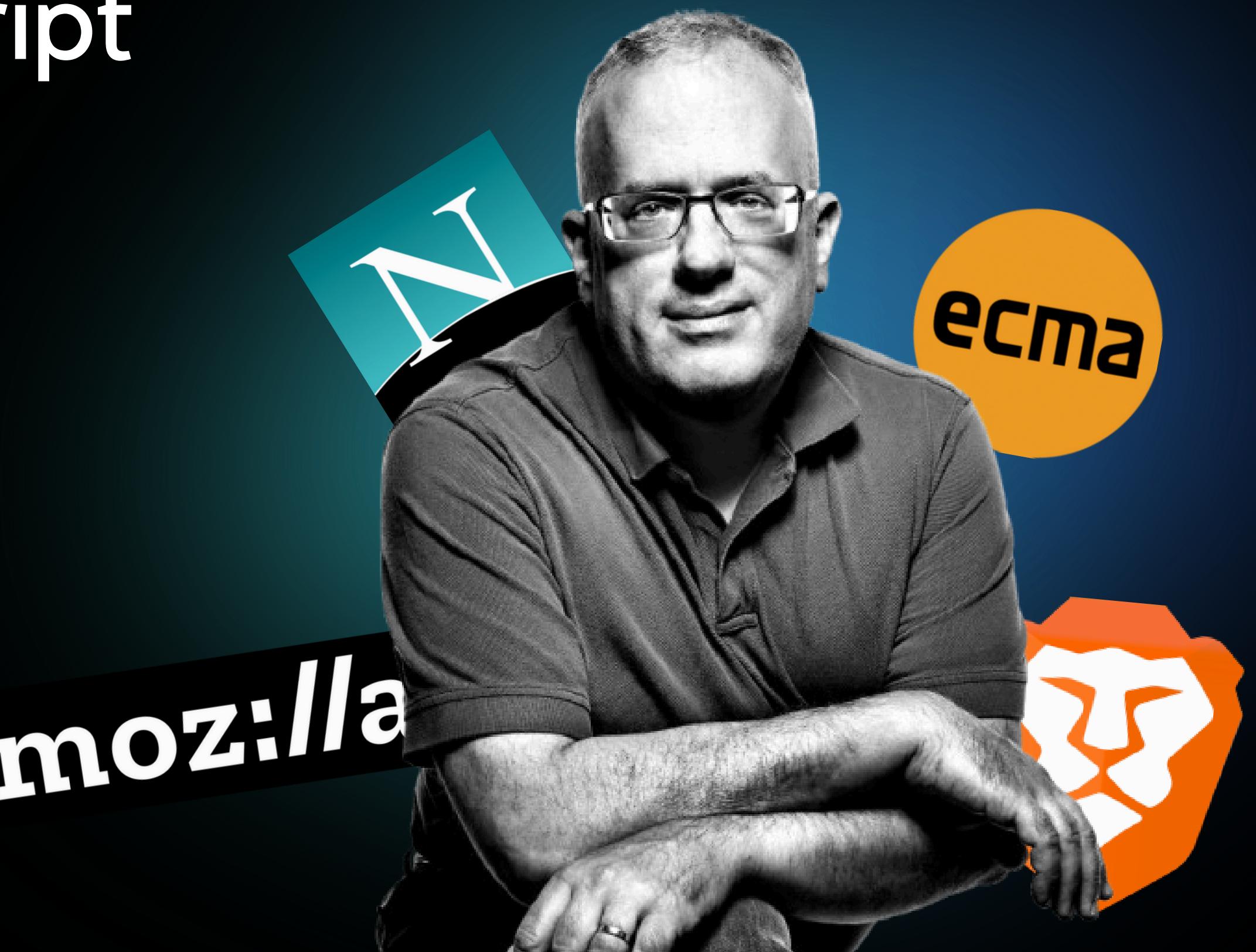
# Introdução e conceitos básicos

# O que é JavaScript

Criado por Brendan Eich em 1995, JavaScript (na época com o codinome Mocha) foi desenvolvido em **10 dias**, sendo disponibilizada para uso com o lançamento da beta do netscape navigator 2.0 sob o nome LiveScript, subsequentemente recebendo o nome JavaScript por motivos de marketing.

JavaScript foi desenvolvido com o objetivo de deixar a web mais dinâmica e amigável para programadores e usuários, mas sua adoção foi conturbada.

Apenas quando a Netscape submeteu o JavaScript para padronização pela ECMA que, 1997, surgiu a primeira padronização ECMAScript, servindo de base para a implementação de JavaScript na web.



# Dados Interessantes

## Linguagem Interpretada

O JavaScript precisa de um motor de execução para funcionar, os mais conhecidos são o V8 (Chrome e Node.JS), SpiderMonkey (Firefox), JavaScriptCore (Apple Safari) e Chakra (Internet Explorer)

## Presente fora da web

Inicialmente projetado para front-end, o JavaScript se popularizou em outras frentes, no back-end com o surgimento do Node.JS, nos desktops com Electron e mobile com React Native

## Extremamente Popular

JavaScript é a linguagem mais popular entre programadores segundo a pesquisa do Stack OverFlow de 2024.

# Sintaxe

# Comentários



```
// Eu sou um comentário em JavaScript  
  
/*  
Eu sou um bloco de comentários em JavaScript  
*/
```

## Case Sensitive



Exemplo != exemplo

## Tipagem



```
const numero = 1  
const texto = "texto"  
const char = "A"  
const double = 3.14
```

# Variáveis

Existem 3 formas de declarar uma variável no JavaScript, elas são o **const**, **let** e **o var**.

Se você é acostumado com outras linguagens de programação, como Java, pode achar estranho não declararmos a variável com o tipo de dado dela, ex: int para Inteiros, mas isso acontece pois as variáveis em JS “deduzem” o seu tipo conforme é preenchido o valor dela, isso é chamado de **tipagem fraca**.

## Const

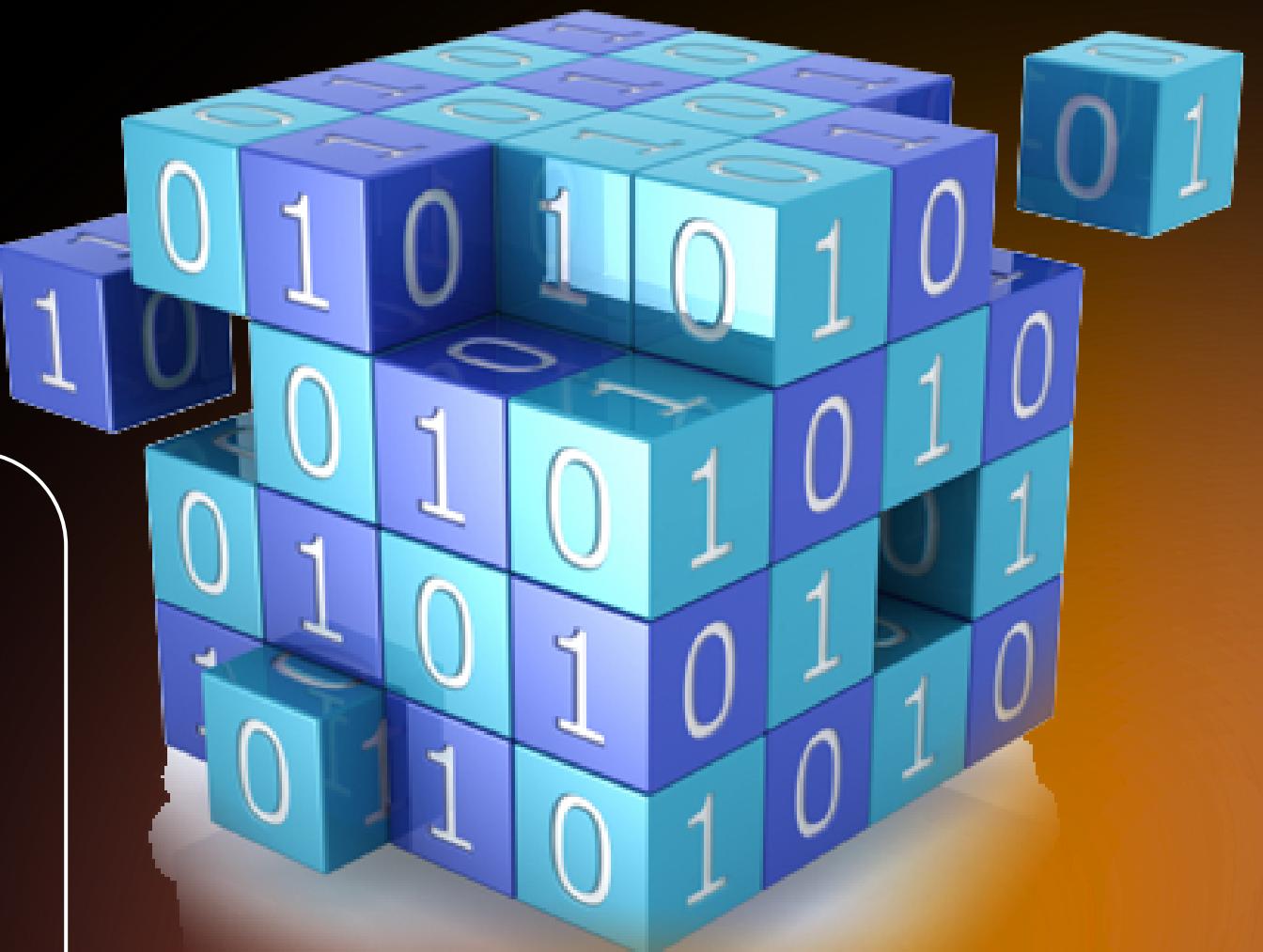
É um tipo de variável que após receber um valor, não é possível mudar seu valor subsequentemente no código, sendo assim uma **constante**, tem escopo de bloco, “{ }”

## Let

É uma forma moderna de declarar uma variável “comum” no JS, seu escopo é de bloco, “{ }”,

## Var

A forma original de declarar uma variável em JS, seu escopo é de função.



# Diferença entre Let e Var

Pode parecer redundante ter duas formas de declarar variáveis na mesma linguagem, mas a introdução da variável let no ECMAScript 6 veio para tornar a linguagem mais robusta, veja as diferenças entre as duas:

## Var

Presente desde a introdução do JavaScript

Seu escopo é de função, ou global caso seja declarado na função principal do código.

É possível redeclarar a variável var dentro do seu escopo.

Esses fatores fazem com que variáveis var possam gerar erros difíceis de rastrear.

Ela é mantida para manter compatibilidade com código legado.

## Let

Introduzida no padrão ECMAScript 6

Seu escopo é de bloco, o que dá mais controle sobre seu uso no código.

Essa variável não permite redeclaração dentro do seu escopo.

Ela foi introduzida para ser uma forma mais moderna e segura de declarar variáveis, eliminando problemas presentes nas variáveis var, é amplamente recomendado o seu uso ao invés do var.

# Operadores

## Aritméticos

Utilizados em operações matemáticas.

- + : adição ( $2 + 2 = 4$ ).
- : subtração ( $2 - 2 = 0$ ).
- \* : multiplicação ( $2 * 2 = 4$ ).
- / : divisão ( $2 / 2 = 1$ ).
- % : módulo ( $10 \% 2 = 0$ ).

## Comparação

Compara duas preposições e retorna verdadeiro ou falso.

- == : igual "fraco", converte o valor entre tipos diferentes.
- === : igual "forte", não faz conversão entre tipos.
- != : diferente.
- < : menor que.
- > : maior que.

## Lógicos

Utilizados em operações de matemática booleana.

- && : e lógico, ambas preposições precisam ser verdadeiras.
- || : ou lógico, qualquer preposição pode ser verdadeira.
- ! : negação lógica, inverte o valor da preposição