

# Introdução ao JavaScript

## O que é JavaScript?

**JavaScript (JS)** é uma linguagem de programação interpretada, criada para tornar as páginas da web **dinâmicas e interativas**. Enquanto o **HTML** define a estrutura e o **CSS** o estilo, o JavaScript é responsável pelo comportamento — ele faz a página “ganhar vida”.

Hoje em dia, o JavaScript é usado também em aplicativos, jogos, servidores (*Node.js*) e até dispositivos IoT.

## Para que serve o JavaScript?

- Manipular elementos do HTML (adicionar, remover ou alterar conteúdo da página);
- Reagir a eventos (cliques, digitação, rolagem);
- Criar animações e efeitos visuais;
- Enviar e receber dados de servidores (APIs, formulários);
- Desenvolver aplicações web completas, no **frontend** e **backend**.

## Por que o JavaScript foi criado?

Nos primeiros anos da web (década de 1990), as páginas eram estáticas. Em 1995, **Brendan Eich**, programador da **Netscape**, criou o JavaScript para permitir que o navegador executasse scripts e deixasse as páginas mais interativas.

Atualmente, é uma das linguagens mais populares do mundo, presente em todos os navegadores modernos.

# Quem criou o JavaScript?

O criador é **Brendan Eich**. Ele desenvolveu o JavaScript em apenas **10 dias**, enquanto trabalhava na Netscape Communications. Inicialmente chamado de *Mocha*, depois *LiveScript*, foi renomeado para **JavaScript** por razões de marketing.

## Características da linguagem

### Case Sensitive

O JavaScript diferencia maiúsculas e minúsculas:

```
let nome = "Ana";  
console.log(Nome); // Erro! "Nome" "nome"
```

As variáveis `nome`, `Nome` e `NOME` são consideradas diferentes.

### Ponto e vírgula opcional

O ponto e vírgula não é obrigatório, mas é uma **boa prática** sempre utilizá-lo para evitar erros:

```
let a = 10;  
let b = 20;  
console.log(a + b);
```

### Uso de chaves para blocos

As chaves delimitam blocos de código, como estruturas de controle e funções:

```
if (true) {  
    console.log("Este é um bloco de código!");  
}
```

### Escopo

Uma variável declarada dentro de um bloco só pode ser usada dentro dele (escopo léxico):

```
function exemplo() {
  let mensagem = "Olá!";
  console.log(mensagem); // funciona
}
console.log(mensagem); // erro - fora do escopo
```

## 1 Console do Navegador

O console é uma ferramenta usada para testar códigos JavaScript diretamente no navegador.

### Como abrir:

- **Google Chrome / Edge:** F12 ou Ctrl + Shift + I, aba **Console**;
- **Firefox:** Ctrl + Shift + K.

### Exemplos de uso:

```
// Testes rápidos
2 + 2

// Criar variáveis
let nome = "Brunno";

// Criar e chamar funções
function ola() {
  console.log("Olá, mundo!");
}
ola();
```

## Resumo rápido

Conceito	Explicação
Criador	Brendan Eich (Netscape, 1995)
Finalidade	Tornar páginas interativas e dinâmicas
Case Sensitive	nome Nome
Ponto e vírgula	Opcional, mas recomendado
Blocos {}	Delimitam estruturas e escopos
Escopo	Variável interna visível só dentro do bloco
Console	Local para testar comandos e funções

# Variáveis em JavaScript

## O que são variáveis?

Variáveis são **espaços na memória** usados para **armazenar valores** que o programa pode usar e modificar durante a execução. Elas funcionam como “caixinhas” que guardam dados, como números, textos ou resultados de cálculos.

```
let nome = "Ana";  
let idade = 20;
```

## Tipos de variáveis (por tipo de dado)

O JavaScript é uma linguagem **dinamicamente tipada**, ou seja, o tipo é definido automaticamente conforme o valor atribuído.

Tipo	Exemplo	Descrição
String	"Olá"	Textos entre aspas
Number	10, 3.14	Números inteiros ou decimais
Boolean	true, false	Verdadeiro ou falso
Null	null	Valor nulo (vazio intencionalmente)
Undefined	variável sem valor	Indica que não há valor definido
Symbol	Symbol("id")	Identificador único
BigInt	123n	Números muito grandes

## Como declarar variáveis

Existem três formas principais:

**var** Forma antiga, tem escopo de função e permite redeclaração (não recomendada).

```
var nome = "Brunno";
```

**let** Introduzido no ES6, tem escopo de bloco e não pode ser redeclarado.

```
let idade = 22;
```

**const** Usado para valores constantes (não podem ser alterados).

```
const PI = 3.14159;
```

## Inicialização de variáveis

Sempre inicie as variáveis para evitar erros.

**Correto:**

```
let contador = 0;
```

**Incorreto:**

```
let contador;  
console.log(contador); // undefined
```

## Boas práticas ao nomear variáveis

1. Use nomes descritivos: `idade`, `precoProduto`.
2. Evite espaços ou acentos: `nomeDoAluno`.
3. Use padrão **camelCase**: `numeroDeAlunos`.
4. Não comece com número: `correto valor1`, `incorreto 1valor`.
5. Evite palavras reservadas (`for`, `while`, `return`, etc).

## Exemplo completo

```
// Declaração e inicialização  
let nome = "Ana";  
let idade = 20;  
const cidade = "São Paulo";  
  
// Uso  
console.log("Nome:", nome);  
console.log("Idade:", idade);  
console.log("Cidade:", cidade);
```

## Resumo

Conceito	Explicação
Variáveis	Espaços na memória para armazenar dados
Tipos principais	String, Number, Boolean, Null, Undefined, Symbol, BigInt
Declaração	<code>var</code> , <code>let</code> , <code>const</code>
Inicialização	Sempre atribuir um valor inicial
Nomeação	<code>camelCase</code> , sem acentos, nomes descritivos

# Entrada de valores e concatenação em JavaScript

## Entrada de valores — `prompt()`

O comando `prompt()` é usado para solicitar que o usuário digite um valor. O valor retornado é sempre uma **string** (texto), mesmo que o usuário digite um número.

```
let nome = prompt("Qual é o seu nome?");
alert("Olá, " + nome + "! Seja bem-vindo!");
```

**Dica:** o comando `alert()` exibe uma mensagem na tela.

## Concatenação

Para unir textos ou valores, usa-se o operador de concatenação `+`.

```
let nome = "Ana";
let idade = 20;
console.log("Meu nome é " + nome + " e tenho " + idade + " anos.");
```

Saída:

Meu nome é Ana e tenho 20 anos.

## Operadores matemáticos

Operador	Significado	Exemplo	Resultado
<code>+</code>	Adição	<code>5 + 2</code>	7
<code>-</code>	Subtração	<code>5 - 2</code>	3
<code>*</code>	Multiplicação	<code>5 * 2</code>	10
<code>/</code>	Divisão	<code>10 / 2</code>	5
<code>%</code>	Módulo (resto)	<code>10 % 3</code>	1

## Cuidado: tudo que vem do `prompt()` é texto!

Mesmo que o usuário digite números, o resultado será tratado como **string**. Assim, a soma se tornará uma concatenação:

```
let n1 = prompt("Digite um número:");
let n2 = prompt("Digite outro número:");
console.log(n1 + n2); // "53" se o usuário digitar 5 e 3
```

## Convertendo valores

`parseInt()` Converte o texto em número inteiro.

```
let n1 = parseInt(prompt("Digite um número:"));
let n2 = parseInt(prompt("Digite outro número:"));
let soma = n1 + n2;
console.log("A soma é: " + soma);
```

`toString()` Converte número em texto.

```
let numero = 123;
let texto = numero.toString();
console.log("Número convertido em texto: " + texto);
```

## Exemplo completo

```
let nome = prompt("Digite seu nome:");
let idade = parseInt(prompt("Digite sua idade:"));
let anoNascimento = 2025 - idade;

console.log("Olá, " + nome + "!");
console.log("Você nasceu em " + anoNascimento + ".");
```

## Resumo

Comando	Função
<code>prompt()</code>	Captura valor digitado pelo usuário (string)
<code>+</code>	Concatena textos ou soma números
<code>-</code> , <code>*</code> , <code>/</code> , <code>%</code>	Operadores matemáticos
<code>parseInt()</code>	Converte texto em número inteiro
<code>toString()</code>	Converte número em texto