

## INFORMAÇÕES

MÓDULO	AULA	INSTRUTOR
04    JavaScript Essencial e Consumo de APIs	01 - Introdução ao JavaScript e Ambiente de Desenvolvimento	Pedro Henrique Amadio

## CONTEÚDO

### 1. Introdução ao JavaScript

O JavaScript tem como objetivo de tornar as páginas da web **interativas e dinâmicas**. Antigamente as páginas eram 100% estáticas, sem interação nenhuma.

Com o JavaScript, passou a ser possível:

- Validar formulários antes de enviá-los;
- Criar animações e efeitos visuais;
- Atualizar partes da página sem recarregar tudo (como faz o Gmail, por exemplo);
- Comunicar-se com **APIs** para buscar e exibir dados dinâmicos.

Em resumo...

HTML é o **esqueleto**, CSS é a **roupa**, e JavaScript é o **cérebro** da página.

Com ele é possível criar variáveis, funções, animações, e muito mais.

### 2. Onde o JavaScript Roda?

O JavaScript nasceu para rodar **no navegador**, interpretado diretamente pelos motores de execução (JavaScript engines) onde cada navegador possui o seu próprio motor.

Mas, com o passar do tempo, o JS saiu do navegador e passou a rodar também no servidor, graças ao Node.js (criado em 2009), que usa o mesmo motor V8 do Chrome, mas fora do ambiente web e hoje está em todo lugar.

O JavaScript é hoje uma das **linguagens mais versáteis do mundo**, usada tanto por desenvolvedores Frontend quanto Backend.

#### CURIOSIDADE:

Pera, JavaScript no backend?

Sim, o JavaScript pode ser usado no BackEnd também em conjunto com o NodeJS + Express que são ambientes e frameworks importantíssimas para garantir seu funcionamento no lado do servidor. Note que para o JavaScript rodar do lado do se

### 3. Como carregar o JavaScript em minha página?

- O que é a tag <script>

A tag <script> é o **ponto de entrada** do JavaScript em uma página web. Quando o navegador lê o HTML e encontra <script>, ele **para o carregamento** do restante da página, **lê o código JavaScript e executa**.

- **Duas formas de carregar o JavaScript**

- **Script interno (inline)**

Você escreve o código **dentro** da própria página HTML.

```
<!doctype html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Script interno</title>
  </head>
  <body>
    <h1>Bem-vindo!</h1>
    <script>
      alert("Olá, mundo!");
      console.log("Script interno rodando!");
    </script>
  </body>
</html>
```

**Vantagem:** rápido para testar.

**Desvantagem:** mistura HTML com código, dificulta manutenção.

**Uso indicado:** exemplos curtos ou testes.

- **Script externo (arquivo separado)**

O código JavaScript fica em um arquivo .js, e a página HTML apenas **referencia** esse arquivo usando o atributo **src**.

```
<!doctype html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Script externo</title>
  </head>
  <body>
    <h1>Usando um arquivo externo</h1>
    <script src="script.js"></script>
  </body>
</html>
```

Arquivo **script.js**:

```
console.log("Arquivo externo carregado com sucesso!");
```

**Vantagens:**

- Facilita a manutenção (HTML limpo e código organizado).
- Pode ser reutilizado em várias páginas.
- Melhora o desempenho (o navegador faz cache do arquivo).

**Uso indicado:** projetos reais, aulas práticas e exercícios.

#### 4. JavaScript: Linguagem “multiparadigma” e “dinâmica”

- **Multiparadigma**

Significa que o JavaScript **aceita diferentes formas de programar** – ele não obriga o desenvolvedor a seguir um único estilo.

Os principais **paradigmas** suportados são:

- **Imperativo:** descreve passo a passo o que o programa deve fazer.

```
let contador = 0;
contador++;
console.log(contador);
```

- **Funcional:** foca em funções puras e imutabilidade.

```
const numeros = [1, 2, 3];
const dobrados = numeros.map(n => n * 2);
console.log(dobrados);
```

- **Orientado a objetos:** organiza o código em classes e instâncias.

```
class Pessoa {
    constructor(nome) {
        this.nome = nome;
    }
    falar() {
        console.log("Olá, eu sou ${this.nome}");
    }
}
const aluno = new Pessoa("Pedro");
aluno.falar();
```

- **Linguagem dinâmica**

O JavaScript é **dinâmico** porque **não exige declarar o tipo de dado de uma variável** - o tipo é determinado **automaticamente** durante a execução.

```
let dado = 10;           // Agora é um number
dado = "dez";           // Agora é uma string
console.log(dado);
```

Isso torna a linguagem mais **rápida para escrever**, mas também pode causar **erros sutis** se o programador não tiver cuidado.

#### 5. Variável

Uma **variável** é um nome que armazena um valor na memória, como uma “caixa com uma etiqueta”.

```
let nome = "Pedro";
let idade = 24;
```

As variáveis permitem **guardar dados e manipulá-los** durante a execução do programa.

- **Declaração moderna (ES6):**
  - let -> variável que pode mudar.
  - const -> variável constante (não muda).
  - var -> forma antiga (evitar).

Exemplo:

```
let linguagem = "JavaScript";
const anoCriacao = 1995;
console.log(linguagem, anoCriacao);
```

- **Onde posicionar o <script> ?**  
Antes do fechamento do <body> (boa prática)

```
<body>
  <h1>Minha página</h1>
  <script src="script.js"></script>
</body>
```

Assim, o HTML é carregado **primeiro**, e o JavaScript só **executa depois**.

## 6. Tipos de dados

O JavaScript possui **tipos primitivos** e **tipos de referência**.

Os principais tipos primitivos são:

Tipo	Exemplo	Descrição
<b>string</b>	"Pedro", 'JS', `texto`	Texto
<b>number</b>	10, 3.14	Números inteiros e decimais
<b>boolean</b>	true, false	Verdadeiro ou falso
<b>undefined</b>	variável declarada, mas sem valor	Indica “sem valor definido”
<b>null</b>	valor intencionalmente vazio	Indica “ausência de valor”
<b>symbol</b>	Symbol('id')	Identificador único (avanhado)
<b>bign</b>	123n	Números muito grandes

## Declarando variáveis e testando tipos de dados

```
let nome = "Lucas";           // string
let idade = 21;              // number
let ativo = true;             // boolean
let telefone;                // undefined
let endereco = null;          // null

console.log(typeof nome);     // "string"
console.log(typeof idade);    // "number"
console.log(typeof ativo);    // "boolean"
console.log(typeof telefone); // "undefined"
console.log(typeof endereco); // "object" (curiosidade
histórica)
```

### Curiosidade:

Existe um erro muito antigo no JavaScript que faz com que o retorno do `typeof null` seja um objeto, e esse erro foi mantido até hoje por questões de compatibilidade

## 7. Conversão de tipos

É possível converter os tipos de dados como no exemplo abaixo

```
let numero = "5";
console.log(Number(numero) + 1); // 6
console.log(String(10)); // "10"
```

- Soma vs. Concatenação
  - Soma:  
`console.log(5 + 1); // 6`
  - Concatenação:  
`console.log("5" + 1); // "51"`

## 8. Conceitos fundamentais

- **alert()**  
Exibe uma mensagem simples.  
`alert("Bem-vindo ao curso!");`  
**OBS: É bloqueante, ou seja pausa a execução até o usuário fechar.**

- **prompt()**  
Permite coletar uma entrada do usuário:  
`let nome = prompt("Qual é o seu nome?");`  
`alert("Olá, " + nome + "!");`  
Retorna sempre **string**, é útil converter se precisar de número:

```
let idade = Number(prompt("Quantos anos você tem?"));
console.log(typeof idade); // number
```

- **console.log()**

Usado para exibir mensagens de depuração no Console (sem interação direta com o usuário):

```
console.log("Mensagem de teste");
```

- Ideal para acompanhar valores de variáveis e verificar se o script está funcionando.
- Também aceita **template strings**:

```
console.log(`Usuário: ${nome} | Idade: ${idade}`);
```

## 9. Atividade prática - “Meu primeiro diálogo JS”

- **Passos:**

1. Criar um arquivo index.html e script.js.
2. No script:

```
alert("Bem-vindo ao sistema!");
let usuario = prompt("Digite seu nome:");
let idade = prompt("Digite sua idade:");
console.log(`Usuário: ${usuario}, ${idade} anos.`);
alert(`Olá ${usuario}, seja bem-vindo(a)!`);
```

## 10. Exercícios

### 1. O que é o JavaScript?

- ( ) Uma linguagem para formatar textos na web.
- ( ) Uma linguagem de marcação para estruturar páginas.
- ( ) Uma linguagem de programação que adiciona interatividade às páginas web.
- ( ) Um editor de código usado para escrever HTML e CSS.

### 2. Onde o JavaScript é executado?

- ( ) No navegador e também em servidores (Node.js).
- ( ) Somente em navegadores antigos.
- ( ) Apenas em aplicativos móveis.
- ( ) No Photoshop e Word.

### 3. Assinale a alternativa verdadeira:

- ( ) O JavaScript é usado para tornar páginas web dinâmicas e interativas.
- ( ) O JavaScript substitui o HTML e o CSS.
- ( ) O JavaScript só funciona se o usuário estiver offline.
- ( ) O JavaScript é usado apenas em servidores.

### 4. Qual das opções abaixo representa um exemplo de variável válida em JavaScript?

- ( ) let nome = "Pedro";
- ( ) var 1idade = 25;
- ( ) const if = "palavra";
- ( ) let sobrenome-usuario = "Oliveira";

### 5. Observe o código abaixo e marque a alternativa correta:

```
let valor = 10;
valor = "dez";
```

```
console.log(typeof valor);  
O que será exibido no console?  
( ) number  
( ) string  
( ) boolean  
( ) undefined
```

6. O que acontece no exemplo a seguir?

```
const cidade = "São Paulo";  
cidade = "Curitiba";  
console.log(cidade);  
( ) Exibe "Curitiba" normalmente.  
( ) Gera erro, pois não é possível alterar uma constante.  
( ) Exibe "São Paulo".  
( ) Retorna undefined.
```

7. O que o código abaixo imprime?

```
let nome = "Maria";  
let idade = 30;  
console.log("Olá " + nome + ", você tem " + idade + " anos!");
```