



# Introdução a JavaScript

**J**avaScript é uma das linguagens de programação mais influentes no desenvolvimento web moderno, sendo responsável pela criação de páginas dinâmicas e interativas. Nesta aula introdutória, vamos explorar os principais conceitos que definem o JavaScript, sua aplicabilidade no front-end e sua crescente presença no back-end com o Node.js. Discutiremos como essa linguagem interpretada se integra com HTML e CSS para adicionar funcionalidades dinâmicas, tornando possível a criação de aplicações interativas que respondem em tempo real às ações dos usuários. Prepare-se para entender os fundamentos dessa tecnologia indispensável!

## Introdução à Linguagem JavaScript

JavaScript é uma linguagem de programação essencial no desenvolvimento web, responsável por tornar páginas web dinâmicas e interativas. A principal função dessa linguagem é permitir que os sites se adaptem às ações dos usuários, seja para exibir uma mensagem de boas-vindas, atualizar o conteúdo sem recarregar a página ou realizar cálculos em tempo real.

JavaScript é uma linguagem interpretada, o que significa que seu código é executado diretamente pelo navegador, sem a necessidade de compilação prévia. Ela é amplamente usada no lado do cliente (front-end), mas também pode ser aplicada no lado do servidor (back-end) com o uso do Node.js. Uma característica importante do JavaScript é sua integração com HTML e CSS. Enquanto o HTML define a estrutura de uma página e o CSS cuida da aparência, o JavaScript adiciona a lógica e os comportamentos dinâmicos,

como responder a cliques do usuário, enviar e receber dados de servidores e muito mais.

### **Exemplo de uso básico de JavaScript em uma página web:**

```
<!DOCTYPE html>
```

```
<html lang="pt-BR">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Exemplo JavaScript</title>
```

```
  <script>
```

```
    function saudacao() {alert("Bem-vindo ao curso de JavaScript!");}
```

```
  </script>
```

```
</head>
```

```
<body>
```

```
  <h1>Olá, mundo!</h1>
```

```
  <button onclick="saudacao()">Clique aqui</button>
```

```
</body>
```

```
</html>
```

Neste exemplo, ao clicar no botão, uma janela de alerta aparece com a mensagem “Bem-vindo ao curso de JavaScript!”. Isso demonstra o papel do JavaScript em adicionar interatividade a páginas web.

Além disso, o JavaScript é uma linguagem de alto nível, o que significa que ela é mais próxima da linguagem humana do que do código de máquina, facilitando o aprendizado. Sua sintaxe é clara e acessível, tornando-a uma excelente escolha tanto para iniciantes quanto para desenvolvedores avançados. No mercado atual, o JavaScript é amplamente utilizado, e frameworks como React, Angular e Vue.js têm consolidado seu uso em aplicações web modernas.

## **Fundamentos do JavaScript**

Os fundamentos do JavaScript são essenciais para qualquer desenvolvedor que deseja trabalhar com a web. No centro desses fundamentos estão as variáveis, funções, operadores, estruturas de controle (como condicionais e loops) e manipulação do DOM (Document Object Model). Cada um desses elementos desempenha um papel crucial na construção de uma página web dinâmica e interativa.

## **Variáveis e Tipos de Dados**

Em JavaScript, variáveis são usadas para armazenar dados que podem ser manipulados posteriormente no código. Existem três formas de declarar variáveis: `var`, `let` e `const`, cada uma com suas particularidades em relação ao escopo e à mutabilidade. O `var` foi amplamente utilizado nas primeiras versões do JavaScript, mas a partir do ECMAScript 6 (ES6), `let` e `const` passaram a ser recomendados, proporcionando um controle mais rigoroso sobre o escopo das variáveis.

## **Exemplo de variáveis:**

```
var nome = "João"; // Variável global ou local (escopo de função)
```

```
let idade = 25; // Variável com escopo de bloco
```

```
const pi = 3.14; // Constante, valor imutável
```

Aqui, `var` permite que a variável `nome` seja acessada em qualquer parte da função onde foi definida, enquanto `let` limita o uso da variável `idade` ao bloco em que foi declarada. `const`, por sua vez, define uma constante cujo valor não pode ser alterado após a atribuição.

## Funções

Funções são blocos de código reutilizáveis que realizam uma tarefa específica. Elas são fundamentais no JavaScript, permitindo organizar o código em pequenos blocos modulares. Uma função pode receber parâmetros, realizar operações e retornar um valor.

Exemplo de função:

```
function somar(a, b) {return a + b;}
```

```
let resultado = somar(5, 3); // Chamada da função
```

```
console.log(resultado); // Exibe 8
```

Nesse exemplo, a função `somar` recebe dois parâmetros (`a` e `b`) e retorna a soma deles. A função pode ser chamada várias vezes com diferentes valores, tornando o código mais eficiente e fácil de manter.

## Estruturas de Controle

As estruturas de controle, como condicionais (if, else) e loops (for, while), permitem tomar decisões e repetir blocos de código com base em condições específicas.

Exemplo de estrutura condicional:

```
let nota = 85;
```

```
if (nota >= 60) {console.log("Aprovado");} else {console.log("Reprovado");}
```

Neste caso, o código verifica se a nota é maior ou igual a 60 e exibe "Aprovado" ou "Reprovado" dependendo do valor da variável nota.

Esses são apenas alguns dos fundamentos do JavaScript, mas eles formam a base para a criação de qualquer aplicação web.

## **Constantes e Variáveis**

Em JavaScript, variáveis e constantes são utilizadas para armazenar e manipular informações. Entender como e quando utilizá-las é essencial para escrever um código eficiente e seguro.

var, let e const

Há três maneiras principais de declarar variáveis em JavaScript: var, let e const.

var: Declara uma variável com escopo de função ou global. Antes do ES6, essa era a principal maneira de declarar variáveis. Contudo, seu comportamento pode ser confuso, pois permite que variáveis sejam redeclaradas e acessadas fora do bloco onde foram definidas.

let: Introduzida no ES6, let tem escopo de bloco, ou seja, a variável só está acessível dentro do bloco em que foi declarada. É mais segura que var porque impede certos erros de lógica.

const: Também introduzida no ES6, const declara uma constante, ou seja, um valor que não pode ser alterado após sua atribuição. Ideal para valores que não devem mudar durante a execução do programa.

Exemplo prático:

```
let idade = 30;
```

```
const pi = 3.14159;
```

```
idade = 31; // Válido
```

```
pi = 3.2; // Erro! Não é possível alterar uma constante
```

## Diferença de Escopo

O escopo de uma variável refere-se à parte do código onde ela pode ser acessada. Em JavaScript, o escopo pode ser global (válido em todo o código), de função (válido dentro de uma função) ou de bloco (válido apenas dentro de um bloco, como em um if ou for).

```
function testeEscopo() {if (true) {var varGlobal = "Escopo global";
```

```
    let letBloco = "Escopo de bloco";}}
```

```
console.log(varGlobal); // Funciona
```

```
console.log(letBloco); // Erro: letBloco não está acessível aqui}
```

```
testeEscopo();
```

No exemplo acima, `varGlobal` pode ser acessada fora do bloco `if`, mas `letBloco` não, pois `let` tem escopo restrito ao bloco.

## **Anotações no Código**

Documentar o código com comentários é uma prática essencial para garantir que ele seja compreensível por outros desenvolvedores e pelo próprio autor, caso precise revisitar o projeto no futuro. Comentários servem para explicar o que uma determinada parte do código faz, quais são suas responsabilidades e por que decisões foram tomadas.

### **Existem dois tipos de comentários em JavaScript:**

- Comentário de linha única: utilizado para explicar trechos curtos de código. Inicia-se com `//`.

Exemplo:

```
let total = 10; // Armazena o valor total
```

- Comentário de bloco: Utilizado para descrever trechos maiores de código ou fornecer explicações mais detalhadas. Envolve o texto entre `/*` e `*/`.

Exemplo:

```
/*Função que calcula a média de dois números.
```

A função recebe dois parâmetros numéricos e retorna o resultado da divisão entre a soma dos dois e o número de parâmetros.\*/

```
function calcularMedia(a, b) {return (a + b) / 2;
```

Comentários ajudam a manter o código legível e a evitar confusões futuras. Além disso, quando se trabalha em equipe, comentários claros e objetivos são essenciais para garantir que todos compreendam o código.

## **Conteúdo Bônus**

A Khan Academy oferece um curso básico de JavaScript focado no desenvolvimento de animações e jogos. É uma forma divertida e interativa de aprender a linguagem.

Título: Unidade 1: Introdução a JavaScript: Desenho & Animação

Plataforma: Khan Academy

## **Referência Bibliográfica**

DEITEL, P. J.; DEITEL, H. M. Ajax, rich internet applications e desenvolvimento web para programadores. Pearson: 2008

FELIX, R. (Org.). Programação orientada a objetos. Pearson: 2016



FERREIRA, R. D. Linguagem de programação. Contentus: 2020.

FLATSCHART, F.; BACHINI, C.; CUSIN, C. Open Web Platform. Brasport: 2013.

NEVES, M. C. B. de A. Sites de Alta Performance. Contentus: 2020

PAGE-JONES, M. Fundamentos do desenho orientado a objeto com UML. Pearson: 2001.

PUGA, S.; RISSETTI, G. Lógica de programação e estruturas de dados, com aplicações em Java. Pearson: 2016.

SEGURADO, V. S. (Org.). Projeto de interface com o usuário. Pearson: 2016.

## Atividade Prática 1 – Introdução a JavaScript

Título da Prática: Apresentação da linguagem Javascript

Objetivos: Entender conceitos iniciais da linguagem JavaScript

Materiais, Métodos e Ferramentas: Computador com uma IDE instalada (recomendado: VS Code) e Javascript instalado via node/npm e o entendimento do contexto abaixo

## Atividade Prática

JavaScript é uma linguagem de programação muito valiosa no desenvolvimento de aplicações web interativas e dinâmicas. Ela permite que desenvolvedores criem funcionalidades e comportamentos complexos nas páginas web, tornando-as mais atraentes e interativas para os usuários. Como uma linguagem de script executada no lado do cliente, o JavaScript

roda diretamente no navegador do usuário, possibilitando a criação de páginas mais rápidas e com melhor experiência de uso.

Para dominar o JavaScript, precisamos entender seus conceitos básicos, como o uso de variáveis, tipos de dados, operadores, estruturas de controle, funções e objetos. Esses fundamentos são a base para criar bons códigos e soluções. Além disso, o JavaScript oferece diferentes formas de declarar variáveis e usar comentários, o que contribui para um código mais organizado e fácil de entender.

Para compreender e escrever códigos eficazes em JavaScript, é fundamental entender seus conceitos básicos.

Entre os principais fundamentos estão:

**Variáveis e Tipos de Dados:** é possível declarar variáveis utilizando as palavras-chave `var`, `let` ou `const`. Cada uma delas possui características específicas de escopo e comportamento, permitindo ao programador controlar melhor o uso de valores em diferentes partes do código.

**Operadores:** incluem operadores aritméticos, de atribuição, de comparação, e lógicos, fundamentais para manipular dados e controlar o fluxo do programa.

**Estruturas de Controle:** estruturas de controle como `if`, `else`, `switch`, `for`, `while`, e `do-while`, definem condições e repetições que determinam o fluxo de execução do código, permitindo decisões baseadas em diferentes situações.

**Funções:** são blocos de código que podem ser chamados para realizar uma tarefa específica. Elas permitem a reutilização do código e facilitam a organização e modularização das operações.

**Objetos:** são coleções de pares chave-valor, fundamentais para representar e manipular entidades abstraídas no mundo real, como uma pessoa com propriedades como nome, idade e endereço.

Além disso, JavaScript, como muitas outras linguagens de programação, oferece o uso de comentários. Os comentários podem ser de uma linha, iniciados com `//`, ou de várias linhas, delimitados por `/*` e `*/`. Comentários

ajudam a tornar o código mais legível e facilmente documentável.

Esse foi um resumo de conceitos importantes para a linguagem, a partir deles e da vivência de sala de aula, responda as perguntas abaixo:

1. Qual é a principal função de um comentário em JavaScript?

- a. Ajudar a tornar o código mais eficiente.
- b. Explicar o código e torná-lo mais legível.
- c. Modificar a execução do código.
- d. Acelerar a execução do código.

2. O que são objetos em JavaScript?

- a. São blocos de código que podem ser reutilizados.
- b. São coleções de pares chave-valor e representam entidades do mundo real.
- c. São variáveis que não podem ser alteradas.
- d. São tipos de dados primitivos como números e strings.

3. Qual é a principal vantagem de o JavaScript ser executado no navegador do usuário?

- a. Ele permite que o servidor controle a execução do código.
- b. Ele torna a execução da página mais lenta e menos interativa.
- c. Ele possibilita a criação de páginas mais rápidas e com melhor experiência de uso.
- d. Ele torna as páginas menos dinâmicas, focando apenas na estrutura HTML.

4. Qual é o principal objetivo das estruturas de controle for e while em JavaScript?

- a. Definir uma condição para determinar o tipo de variável a ser utilizada.
- b. Realizar iteração, ou seja, repetir um bloco de código várias vezes, enquanto uma condição for verdadeira.
- c. Definir funções que podem ser chamadas em outros pontos do código.
- d. Comparar dois valores e retornar um valor booleano.

### Gabarito Atividade Prática

1. b. Explicar o código e torná-lo mais legível.

Comentário: Comentários são usados para explicar o código de forma que ele se torne mais legível e compreensível para outras pessoas desenvolvedoras que irão utilizar o código – sem alterar a execução do código.

2. b. São coleções de pares chave-valor e representam entidades do mundo real.

Comentário: Objetos em JavaScript são estruturas que armazenam dados em pares de chave-valor, permitindo representar entidades complexas abstraindo a realidade.

3. c. Ele possibilita a criação de páginas mais rápidas e com melhor experiência de uso.

Comentário: Como o JavaScript é executado diretamente no navegador do usuário, ele permite que as páginas sejam mais rápidas e interativas, já que o código é processado no lado do cliente, sem precisar fazer requisições constantes ao servidor.

4. b. Realizar iteração, ou seja, repetir um bloco de código várias vezes, enquanto uma condição for verdadeira.

Comentário: As estruturas de controle for e while são usadas para repetir um bloco de código várias vezes, com base em uma condição. O for é normalmente usado quando o número de iterações é conhecido, enquanto o while continua a execução até que a condição especificada se torne falsa.

**Ir para exercício**