

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUÍ</p>	<p><b>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ</b></p> <p><b>Curso: TDS</b></p> <p><b>Disciplina: Programação para a WEB front-end</b></p> <p><b>Professor: Ely</b></p>
--	---

## Introdução ao JavaScript

### 1. O que é JavaScript e para que serve

- **Definição:** JavaScript é uma linguagem de programação interpretada, conhecida por sua capacidade de adicionar interatividade e dinamismo a páginas web.
- **Utilidade no Desenvolvimento Web:** Amplamente usada em desenvolvimento web, JavaScript permite a criação de conteúdo atualizado dinamicamente, controle de multimídia, animações gráficas, e praticamente qualquer outra coisa que possa ser executada em um navegador web.
- **Amplitude de Uso:** Além de seu papel fundamental no desenvolvimento de front-end, JavaScript também é usado no desenvolvimento de back-end (Node.js), em aplicativos móveis (React Native), e em muitos outros contextos.

### 2. Histórico do JavaScript

- **Criação e Evolução:**
  - 1995: Criado por Brendan Eich na Netscape Communications Corporation. Originalmente chamado de Mocha, depois renomeado para LiveScript, e finalmente para JavaScript.
  - Padronização ECMAScript: Em 1997, a linguagem foi padronizada pela Ecma International na especificação ECMA-262, conhecida como ECMAScript.
  - Evolução Contínua: Desde então, JavaScript continua evoluindo, com adições significativas e melhorias na linguagem, incluindo ECMAScript 6 (ES6) em 2015, que trouxe muitas mudanças, como classes, módulos, arrow functions, entre outras.
- **Expansão e Aceitação:** A expansão de JavaScript foi impulsionada por sua adoção por grandes navegadores e pelo surgimento de frameworks e bibliotecas como Angular, React e Vue, que simplificaram e ampliaram as possibilidades de desenvolvimento com JavaScript.

### 3. Principais Características

#### Características Técnicas

- **Linguagem de Alto Nível:** JavaScript é uma linguagem de alto nível, o que significa que é abstrata, focada na lógica de programação, e distante dos detalhes do hardware do computador.
- **Interpretada e Dinamicamente Tipada:** Não precisa ser compilada antes da execução e permite que tipos de dados sejam alterados dinamicamente.
- **Multiparadigma:** Suporta estilos de programação orientada a objetos, imperativa e funcional.
- **Execução no Lado do Cliente:** Executada principalmente no navegador do cliente, proporcionando uma experiência interativa sem a necessidade de recarregar a página.

#### Funcionalidades e Capacidades

- **Manipulação do DOM:** JavaScript pode acessar e modificar o Document Object Model (DOM), permitindo alterar conteúdos, estruturas e estilos de páginas web.
- **Eventos e Interação do Usuário:** Responde a ações do usuário, como cliques, movimentos do mouse, pressionamento de teclas, etc.
- **Assíncrona e Não-Bloqueante:** Com recursos como Promises e async/await, JavaScript pode realizar operações assíncronas, como buscar dados de um servidor, sem bloquear a execução do programa.
- **Compatibilidade:** É suportado por todos os navegadores modernos e é uma das poucas linguagens que podem ser executadas nativamente na web.

#### Impacto e Importância

- **Onipresença na Web:** JavaScript é essencial para a moderna web. A maioria dos sites usa JavaScript para melhorar a usabilidade e a experiência do usuário.
- **Ecossistema Rico:** Possui um ecossistema vasto e ativo, com inúmeras bibliotecas e frameworks, o que facilita o desenvolvimento de aplicações complexas e robustas.
- **Comunidade e Recursos:** Uma comunidade grande e ativa significa que existem muitos recursos, tutoriais e suporte disponíveis para desenvolvedores de todos os níveis.

JavaScript, desde a sua criação, se tornou uma das linguagens de programação mais influentes e amplamente usadas no mundo, essencial tanto para o desenvolvimento web quanto para muitas outras aplicações fora do navegador.

### Estudo prático

#### 4. Criar uma tag <script> ao final do documento HTML

Crie um documento HTML e adicione uma tag <script> no final do corpo do documento. Utilize alert() para exibir uma mensagem indicando que o script foi carregado corretamente.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Exemplo de JavaScript</title>
</head>
<body>
  <h1>Bem-vindo ao estudo de JavaScript!</h1>

  <script>
    alert("Página carregada e script executado.");
  </script>
</body>
</html>
```

## 5. Escrever no documento com o método document.write()

Imagine uma página HTML como um livro. O objeto document é o livro inteiro. O objeto document serve como ponto de entrada para o conteúdo da página web, permitindo o acesso e a manipulação da estrutura, estilo e conteúdo da página.

Assim, dentro da tag <script>, utilize document.write() para adicionar um parágrafo no corpo da página HTML:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Escrevendo com JavaScript</title>
</head>
<body>
  <h1>JavaScript em Ação</h1>

  <script>
    document.write("<p>Este parágrafo foi adicionado pelo JavaScript.</p>");
  </script>
</body>
</html>
```

## 6. Criar variáveis com a palavra-chave var

O que são variáveis e como declará-las em JavaScript? As variáveis em JavaScript armazenam dados que podem ser usados mais tarde. JavaScript usa as palavras var, let ou const para declaração de variáveis:

```
var numero = 10;  
const PI = 3.14;
```

Crie duas variáveis usando a palavra-chave var para armazenar o nome e a idade de uma pessoa e exiba essas informações na página usando document.write.

```
<!DOCTYPE html>  
<html lang="pt-BR">  
<head>  
  <meta charset="UTF-8">  
  <title>Variáveis em JavaScript</title>  
</head>  
<body>  
  <script>  
    var nome = "Maria";  
    var idade = 25;  
    document.write("<p>Nome: " + nome + "</p>");  
    document.write("<p>Idade: " + idade + "</p>");  
  </script>  
</body>  
</html>
```

## 7. Acessar tags do documento com getElementById()

A função **document.getElementById()** é usada para selecionar um elemento HTML pelo seu atributo **id** e retorna um objeto que representa o elemento HTML com o **id** especificado. Se não encontrar nenhum elemento, retorna **null**. Já a propriedade **innerHTML** é usada em JavaScript para acessar ou modificar o conteúdo HTML de um elemento. Ela permite tanto ler quanto alterar o conteúdo que está dentro de uma tag específica.

Adicione um parágrafo com um id específico e exiba seu conteúdo usando **document.write()**.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Acessando Elementos</title>
</head>
<body>
  <p id="meuParagrafo">Este é um parágrafo.</p>

  <script>
    var paragrafo = document.getElementById("meuParagrafo");
    document.write("<p>Conteúdo acessado: " + paragrafo.innerHTML + "</p>");
  </script>
</body>
</html>
```

## 8. Alterar o conteúdo de uma tag com innerHTML

Use a propriedade innerHTML para alterar o conteúdo do parágrafo criado anteriormente e exiba o novo conteúdo usando document.write().

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <title>Alterando Conteúdo</title>
</head>
<body>
  <p id="meuParagrafo">Este é um parágrafo.</p>

  <script>
    document.getElementById("meuParagrafo").innerHTML = "O conteúdo foi
alterado!";
    var novoConteudo = document.getElementById("meuParagrafo").innerHTML;
    document.write("<p>Novo conteúdo: " + novoConteudo + "</p>");
  </script>
</body>
</html>
```

## Exercícios propostos

9. Declarar Variáveis e Exibir a Soma.

Crie um arquivo HTML que declare duas variáveis usando `var`. As variáveis devem armazenar dois números inteiros. Calcule a soma dessas variáveis e exiba o resultado em um parágrafo usando `document.write()`.

10. Trocar o Conteúdo de um `<h1>` com o de um Parágrafo

Crie uma página com um `<h1>` e um `<p>` contendo textos diferentes. Use `getElementById()` e `innerHTML` para trocar os conteúdos entre o `<h1>` e o parágrafo.

11. Exibir Endereços em um Único Parágrafo

Crie uma página com vários parágrafos. Cada um deve conter uma informação sobre o endereço: Logradouro, Número, Bairro, Complemento, e Cidade. Use `getElementById()` e `innerHTML` para exibir todas essas informações em um único parágrafo, separadas por vírgulas.