

# JS



# JavaScript

JavaScript é uma **linguagem de programação de alto nível, orientada a objetos e interpretada**, amplamente utilizada no desenvolvimento web para tornar as páginas **dinâmicas e interativas**.

# JavaScript – Variáveis e Tipos de Dados

Variáveis são utilizadas para armazenar informações.

**JavaScript é dinamicamente tipado, o que significa que o tipo de uma variável pode ser alterado durante a execução do programa.**

## // Exemplo de variáveis

```
// Exemplo de variáveis  
let numero = 42;  
let texto = "Olá, mundo!";  
let booleano = true;
```

# JavaScript – Operadores

JavaScript suporta uma variedade de operadores para realizar operações aritméticas, lógicas e de comparação.

## // Exemplo de operadores

```
// Exemplo de operadores  
let soma = 5 + 3;  
let isEqual = (10 === 10); // Igualdade estrita  
let andOperator = (true && false);
```

# JavaScript – Estruturas de Controle de Fluxo

**if, else, switch** são usados para controle condicional. **for, while, do-while** são usados para loops.

## // Exemplo de estruturas de controle

```
// Exemplo de estruturas de controle
if (idade >= 18) {
    console.log("Pode dirigir");
} else {
    console.log("Não pode dirigir");
}

for (let i = 0; i < 5; i++) {
    console.log(i);
}
```



# JavaScript – Funções

Funções são blocos de código reutilizáveis. Podem receber parâmetros e retornar valores.

## // Exemplo de função

```
// Exemplo de função
function saudacao(nome) {
    return "Olá, " + nome + "!";
}

let mensagem = saudacao("João");
console.log(mensagem);
```

# JavaScript – Objetos e Arrays

Objetos e Arrays são estruturas de dados importantes em JavaScript.

## // Exemplo de objeto




```
// Exemplo de objeto
let pessoa = {
  nome: "Maria",
  idade: 25,
  profissao: "Engenheira"
};
```

```
// Exemplo de array
let frutas = ["maçã", "banana", "laranja"];
```



# JavaScript – DOM (Document Object Model),

O DOM representa a estrutura de uma página HTML como uma árvore de objetos. JavaScript é usado para manipular o DOM e interagir dinamicamente com elementos HTML.



## // Exemplo de manipulação do DOM

```
// Exemplo de manipulação do DOM
let paragrafo = document.createElement("p");
paragrafo.textContent = "Novo parágrafo";

document.body.appendChild(paragrafo);
```

# JavaScript – Eventos

JavaScript permite que os desenvolvedores respondam a eventos, como cliques do mouse, pressionamentos de teclas, etc.

## // Exemplo de evento

```
// Exemplo de evento
let botao = document.getElementById("meuBotao");
botao.addEventListener("click", function() {
    console.log("Botão clicado!");
});
```



# JavaScript – AJAX e Fetch

JavaScript é utilizado para fazer requisições assíncronas ao servidor, atualizando partes da página sem recarregá-la.

## // Exemplo de Fetch API

```
// Exemplo de Fetch API
fetch('https://api.exemplo.com/dados')
  .then(response => response.json())
  .then(data => console.log(data))
  .catch(error => console.error('Erro:', error));
```

# JavaScript – Promessas e Async/Await

Promessas são usadas para lidar com operações assíncronas. Async/Await simplifica a escrita de código assíncrono.

# // Exemplo de Promessa e Async/Await

```
// Exemplo de Promessa e Async/Await
function obterDados() {
  return new Promise(resolve => {
    setTimeout(() => {
      resolve("Dados obtidos com sucesso!");
    }, 1000);
  });
}

async function principal() {
  const resultado = await obterDados();
  console.log(resultado);
}

principal();
```



JavaScript é uma ferramenta essencial no arsenal de qualquer desenvolvedor web. Sua capacidade de manipular o DOM, realizar requisições assíncronas e interagir dinamicamente com o usuário contribui para a criação de experiências web ricas e interativas.




# Exercícios




Abaixo



## 1. Operações Básicas:


- 
- Crie duas variáveis, a e b, e atribua valores numéricos a elas.
  - Realize as operações básicas de adição, subtração, multiplicação e divisão entre essas variáveis.
  - Exiba os resultados no console.

## 2. Estruturas de Controle:




- Crie uma função que receba um número como parâmetro.
  - Dentro da função, use uma estrutura condicional para verificar se o número é par ou ímpar.
  - Retorne uma mensagem indicando se é par ou ímpar.
  - Chame a função com diferentes números e exiba os resultados.
- 
- 
- 



### 3. Funções e Arrays:

- 
- Crie uma função que receba um array de números como parâmetro.
  - Dentro da função, calcule a média dos números no array.
  - Retorne o resultado.
  - Teste a função com diferentes arrays.


### 4. Manipulação de Strings:

- Crie uma string e converta todas as letras para maiúsculas.
  - Separe a string em um array de palavras.
  - Substitua uma palavra por outra na string.
  - Exiba o resultado no console.
- 
- 
- 








## 5. Objetos e Métodos:

- 
- Crie um objeto representando uma pessoa com propriedades como nome, idade e profissão.
  - Adicione um método ao objeto que cumprimente a pessoa.
  - Chame o método e exiba a mensagem no console.

## 6. Manipulação do DOM:




- Crie uma página HTML simples com um botão.
  - Adicione um script JavaScript que, ao clicar no botão, altera o conteúdo de um elemento na página.
  - Utilize o `document.getElementById()` para selecionar o elemento.
- 
- 
- 



## 7. Requisições AJAX (usando Fetch API):

- Faça uma requisição a uma API pública (por exemplo, JSONPlaceholder) usando a Fetch API.
- Exiba os dados obtidos no console.

## 8. Manipulação de Eventos:

- Crie uma página HTML com um campo de entrada (`input`) e um botão.
  - Adicione um evento ao botão que captura o valor do campo de entrada quando o botão é clicado.
  - Exiba o valor capturado no console.
- 
- 
- 



Esses exercícios abrangem diferentes aspectos do JavaScript, desde operações básicas até manipulação de DOM e interações assíncronas. Lembre-se de praticar regularmente para fortalecer suas habilidades de programação em JavaScript.



**Até a próxima aula**