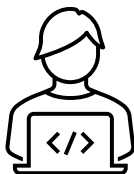


Prof. Lucas Teixeira

Repositório de aulas:  
[bit.ly/4bHLaad](https://bit.ly/4bHLaad)



# Front-End



# Loops

Loops, ou laços de repetição, são estruturas de controle que permitem executar um bloco de código repetidamente enquanto uma condição específica for verdadeira. Eles são usados para automatizar tarefas repetitivas e manipular dados de maneira eficiente.

# Tipos de Loops em JavaScript

**for:** O loop for é usado quando sabemos com antecedência quantas vezes queremos repetir um bloco de código. Ele é composto por três expressões:

- **Inicialização:** Executada uma vez no início do loop.
- **Condição:** Avaliada antes de cada iteração; o loop continua enquanto a condição for verdadeira.
- **Incremento/Decremento:** Executado no final de cada iteração.

# Exemplos for

```
for (inicialização; condição; incremento) {  
    // Código a ser executado  
}
```

```
for (let i = 1; i ≤ 10; i++) {  
    console.log(i);  
}
```

# Tipos de Loops em JavaScript

**while:** O loop while é usado quando não sabemos exatamente quantas vezes o bloco de código deve ser repetido. Ele continua a execução enquanto a condição for verdadeira.

# Exemplos while

```
while (condição) {  
    // Código a ser executado  
}
```

```
let numero = 2;  
  
while (numero % 2 === 0) {  
    console.log(numero, "é par");  
    numero++;  
}  
  
console.log(numero, "é ímpar");
```

# Tipos de Loops em JavaScript

**do...while:** O loop do...while é semelhante ao while, mas a condição é avaliada após a execução do bloco de código. Isso garante que o bloco de código seja executado pelo menos uma vez.

# Exemplos do...while

```
do {  
    // Código a ser executado  
} while (condição);
```

```
let contador = 0;  
  
do {  
    console.log("Mensagem ", contador + 1);  
    contador++;  
} while (contador < 10);
```



# Tipos de Loops em JavaScript

**do...while:** O loop do...while é semelhante ao while, mas a condição é avaliada após a execução do bloco de código. Isso garante que o bloco de código seja executado pelo menos uma vez.

# Atividades

1. Crie um script que exiba os números de 1 a 10 no console usando um loop **for**
2. Crie um script que exiba os números de 1 a 10 no console usando um loop **while**.
3. Crie um script que exiba os números de 1 a 10 no console usando um loop **do...while**.

# Array

## O que são Arrays?

Imaginem um armário com diversas gavetas numeradas. Cada gaveta guarda um item, e você pode acessar cada item facilmente pelo seu número. Essa é a ideia básica por trás dos arrays em JavaScript: coleções ordenadas de valores que podem ser acessados e manipulados individualmente.

# Criando e Acessando Arrays

Criar um array em JavaScript é tão fácil quanto abrir um armário.

```
const meuArray = []; // Um array vazio  
const frutas = ["Maçã", "Banana", "Laranja"]; // Um array com frutas
```

# Criando e Acessando Arrays

Para acessar um item do **array**, basta usar seu índice entre colchetes

```
console.log(frutas[0]);  
console.log(frutas[2]);
```

# Tamanho do Array

Saber quantos itens seu array mágico possui é essencial. Utilize a propriedade `length`.

```
console.log(frutas.length); // Exibe 3
```

# O que são Objetos em JavaScript?

Imagine um objeto do mundo real, como um carro. Ele possui diversos atributos (chave-valor) que o definem, como:

- **Marca:** Fiat
- **Modelo:** Argo
- **Ano:** 2024
- **Cor:** Vermelho
- **Placa:** ABC-1234

# O que são Objetos em JavaScript?

Em JavaScript, podemos representar esse carro utilizando um objeto:

```
const carro = {  
  marca: "Fiat",  
  modelo: "Argo",  
  ano: 2024,  
  cor: "Vermelho",  
  placa: "ABC-1234"  
};
```



# O que são Objetos em JavaScript?

Nesse objeto, cada par chave-valor é representado por uma propriedade:

- **Propriedade:** marca
- **Valor:** "Fiat"

# Acessando Propriedades de Objetos

Para acessar o valor de uma propriedade, utilizamos o operador ponto ( **.** )

```
console.log(carro.marca);  
console.log(carro["modelo"]);
```

# Acessando Propriedades de Objetos

Também podemos utilizar variáveis para acessar propriedades:

```
const marcaDoCarro = carro.marca;  
console.log(marcaDoCarro);
```

# Bibliografia

## Online

**Mozilla:** <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

**W3C Schools:**  
<https://www.w3schools.com/html/>

## Livros

**Html 5: Entendendo e Executando:**

<https://a.co/d/9o3XsHS>

**CSS Cookbook:** <https://a.co/d/6eU6USL>

**JavaScript: O Guia Definitivo:** <https://a.co/d/342B0rJ>

**HTML5 e CSS3: Guia Prático e Visual:**

<https://a.co/d/jg3ccdP>

**Use a Cabeça! HTML e CSS:** <https://a.co/d/66caZxW>

**A psicologia das cores:** <https://a.co/d/1jVHsnO>



# Por Hoje é Só

Continuamos na próxima aula

**Repositório de aulas:**  
<https://bit.ly/4bHLaad>