

Programação para Web I

Aula - 09

Loops em JavaScript

- Loops permitem executar um bloco de código várias vezes.
- São úteis para automatizar tarefas repetitivas, especialmente ao trabalhar com arrays ou listas de dados.

Loops em JavaScript

- **For:** Itera por um bloco de código um número específico de vezes.
- **While:** Executa o código enquanto a condição for verdadeira.
- **Do while:** Executa o código ao menos uma vez e continua enquanto a condição for verdadeira.

Loop For

- O loop for é composto por três expressões opcionais:

```
for (expressão 1; expressão 2; expressão 3) {  
    // bloco de código a ser executado  
}
```

- **Expressão 1:** Executada uma vez, antes do início do loop (inicializa uma variável).
- **Expressão 2:** Define a condição para continuar a execução do loop.
- **Expressão 3:** Executada ao final de cada iteração (normalmente para incrementar a variável).

Exemplo de Loop For

```
for (let i = 0; i < 5; i++) {  
  text += "O número é " + i + "<br>";  
}
```

- **Expressão 1:** `let i = 0;` (inicializa `i` com 0)
- **Expressão 2:** `i < 5;` (loop roda enquanto `i` for menor que 5)
- **Expressão 3:** `i++` (incrementa `i` a cada iteração)

Escopo da Variável no Loop

- Usando **var**:

```
var i = 5;  
for (var i = 0; i < 10; i++) {  
    // código  
}  
// Aqui, i é 10
```

- Usando **let**:

```
let i = 5;  
for (let i = 0; i < 10; i++) {  
    // código  
}  
// Aqui, i ainda é 5
```

Loop While

- O while executa um bloco de código enquanto a condição especificada for verdadeira.
- Útil quando não sabemos o número exato de iterações antecipadamente.
- Verifica a condição antes de executar o bloco de código.
- Sintaxe:

```
while (condição) {  
    // bloco de código a ser executado  
}
```

Exemplo Loop While

- Vamos criar um loop que conta de 0 a 5.

```
let i = 0;  
while (i <= 5) {  
  console.log("Número: " + i);  
  i++;  
}
```


Loop Do-While

- O **do-while** garante que o bloco de código seja executado pelo menos uma vez, mesmo que a condição seja falsa.
- Executa o código primeiro, depois verifica a condição.

```
do {  
    // bloco de código a ser executado  
} while (condição);
```

Loop Do-While

- Vamos contar de 1 a 3 com um **do-while**.

```
let i = 1;  
do {  
    console.log("Número: " + i);  
    i++;  
} while (i <= 3);
```

JavaScript Arrays

- Um array é uma variável especial que pode armazenar múltiplos valores.
 - Exemplo:
`const frutas = ["Manga", "Abacaxi", "Morango"];`
- Vantagem: Armazenar uma lista de itens com um único nome.
- Exemplo com múltiplos itens sem array:
`let fruta1 = "Manga";`
`let fruta2 = "Abacaxi";`
`let fruta3 = "Morango";`

JavaScript Arrays

- Como Criar um Array

- Usando a notação literal:

```
const animais = ["Cachorro", "Gato", "Pássaro"];
```

- Declarar e adicionar elementos posteriormente:

```
const animais = [];  
animais[0] = "Cachorro";  
animais[1] = "Gato";  
animais[2] = "Pássaro";
```

JavaScript Arrays

- Acesso a Elementos

- Acessar elementos por índice:

```
const cores = ["Azul", "Verde", "Vermelho"];
```

```
let primeiraCor = cores[0]; // Azul
```

JavaScript Arrays

- Alterar Valores
 - Modificar um elemento específico:
const cores = ["Azul", "Verde", "Vermelho"];
cores[1] = "Amarelo";

JavaScript Arrays

- Propriedades e métodos úteis:
 - **length**: número de elementos.
 - **push()**: adiciona um novo elemento.
 - **sort()**: ordena o array.

```
const numeros = [5, 3, 8];  
numeros.push(1);    // Adiciona 1  
numeros.sort();     // Ordena: [1, 3, 5, 8]
```

JavaScript Arrays

- Percorrendo Arrays

- Usando for:

```
const frutas = ["Maçã", "Banana", "Laranja"];  
for (let i = 0; i < frutas.length; i++) {  
  console.log(frutas[i]);  
}
```

- Usando forEach():

```
frutas.forEach(fruta => console.log(fruta));
```


JavaScript Objetos

- Objetos são coleções de valores. Cada objeto tem propriedades e métodos.
- Objetos na Vida Real
 - Computador: marca, modelo, processador e memória RAM.
 - Pessoa: nome, idade e profissão.

JavaScript Objetos

- Objeto Computador
 - **Propriedades:**
 - computador.marca = "Dell"
 - computador.modelo = "XPS 13"
 - computador.processador = "Intel i7"
 - computador.memoria = "16GB"
 - **Métodos:**
 - computador.ligar()
 - computador.desligar()

JavaScript Objetos

- Definição de Objetos em JavaScript
 - **Literal de Objeto:** {propriedade: valor}
 - **Construtor de Objeto:** new Object()
- Exemplo de literal de objeto em JavaScript

```
const computador = {  
  marca: "Dell",  
  modelo: "XPS 13",  
  processador: "Intel i7",  
  memoria: "16GB"  
};
```

JavaScript Objetos

- Exemplo com Construtor

```
const computador = new Object();  
computador.marca = "Dell";  
computador.modelo = "XPS 13";  
computador.processador = "Intel i7";  
computador.memoria = "16GB";
```

JavaScript Objetos

- Acesso a Propriedades

computador.marca; // "Dell"

computador["modelo"]; // "XPS 13"

JavaScript Objetos

- Métodos de Objetos

```
const computador = {  
  marca: "Dell",  
  modelo: "XPS 13",  
  ligar: function() {  
    return "Computador ligado!";  
  }  
};
```

```
computador.ligar(); // "Computador ligado!"
```

JavaScript Classes

- São templates para criar objetos em JavaScript.

- Sintaxe Básica das Classes:

```
class NomeDaClasse {  
    constructor() { ... }  
}
```

JavaScript Classes

- Exemplo.

```
class Carro {  
  constructor(marca, modelo) {  
    this.marca = marca;  
    this.modelo = modelo;  
  }  
  detalhes() { return `Marca: ${this.marca} e Modelo:  
    ${this.modelo}`; }  
}
```

- A classe `Carro` possui duas propriedades iniciais: `marca` e `modelo`.
- O método construtor inicializa as propriedades do objeto

JavaScript Classes

- Criando Objetos com uma Classe.

```
const carro1 = new Carro("Toyota", "Corolla");  
const carro2 = new Carro("Honda", "Civic");
```

- O método `constructor` é chamado automaticamente ao criar um novo objeto.

Exercícios

- 1) Crie uma página HTML com um botão que, ao ser clicado, exibe os números de 1 a 10 em uma lista não ordenada na página usando um loop for.
- 2) Crie uma página HTML com um botão que, ao ser clicado, exibe uma lista de frutas (usando um array) em uma lista ordenada.
- 3) Crie uma página HTML com um botão que, ao ser clicado, exibe as propriedades de um objeto carro (marca, modelo, ano) em um parágrafo. Implemente um método ao objeto carro que exiba uma mensagem "Este carro é um [marca] [modelo] de [ano]" quando o botão for clicado.

Exercícios

- 4) Crie uma página HTML com dois campos de entrada para o usuário informar o nome e a idade. Adicione um botão que, ao ser clicado, crie um objeto de uma classe Pessoa utilizando os valores inseridos pelo usuário e exiba o nome e a idade em um parágrafo. Implemente um método na classe Pessoa que exiba em um segundo parágrafo a mensagem 'Olá, meu nome é [nome]' quando outro botão for clicado.