

Curso de JavaScript 2023



Aula 04 - Estruturas de Repetição: for, while, do-while



Instrutor: *Prof. Me. Mário Carvalho*

E-mail para contato: mario.carvalho@ufms.br

Realização: *UFMS e Semadur*





1. Sobre o curso

- Instrutor: **Prof. Me. Mário de Araújo Carvalho**
- E-mail para contato: mario.carvalho@ufms.br
- Sala Google Meet: <https://meet.google.com/fcq-djzs-dzd>
- Repositório oficial do curso:
- <https://github.com/MarioCarvalhoBr/semadur-curso-javascript-2023>
- **Ava:** <https://www.eadfapec.com.br/course/view.php?id=245§ion=4>
- Instituição: **UFMS e Semadur**
- Modalidade: **Online síncrono**
- Duração: 45 horas

Introdução

- - Definição de Estruturas de Controle

Fundamentos: As estruturas de repetição, também conhecidas como loops, são recursos importantes em programação que permitem executar um bloco de código repetidamente, enquanto uma determinada condição for verdadeira.

Propósito: Automatizar tarefas repetitivas, iteram sobre dados, condicionam a repetição com base em critérios e definem um número pré-determinado de execuções de um trecho de código.



Introdução

- - Definição de Estruturas de Controle

Tipos Principais de Estruturas de Repetição

Estruturas de Loop:

Executam um bloco de código repetidamente.

Exemplos: **for**, **while**, **do-while**.





Parte 1: Estruturas de Repetição (for)

- - Conceito e Sintaxe de for

Definição: Uma estrutura de controle que repete um bloco de código um número específico de vezes.

Componentes do for:

Inicialização: Define o ponto de partida do loop (ex: let i = 0).

Condição: Determina até quando o loop deve continuar (ex: i < 10).

Incremento/Decremento: Atualiza a variável de loop a cada iteração (ex: i++).

Exemplo:

```
for (inicialização; condição; incremento/decremento) {  
    // Bloco de código a ser repetido  
}
```



Parte 1: Estruturas de Repetição (for)

- - Conceito e Sintaxe de for

Uso: Executar repetidas vezes uma parte do código.

Funcionamento: Estrutura usada para repetir um bloco de código um número conhecido de vezes.

Exemplo 1:

```
for (let i = 0; i < 10; i++) {  
    console.log(i);  
}
```


Parte 1: Estruturas de Repetição (for)

- - Exemplos Práticos

Exemplo 2: Uso do for

- **Propósito:** Iterando sobre números.
- **Código:**

```
for (let i = 0; i < 5; i++) {  
    console.log(i); // Imprime números de 0 a 4  
}
```

Parte 1:

Estruturas de Repetição (for)

- - Exercícios e Discussão

Exercício 1. Contagem Regressiva: Use um loop for para imprimir uma contagem regressiva de 10 a 0.

Exercício 2. Soma de Números: Crie um loop for para somar os números de 1 a 10 e imprimir o resultado.

Exercício 3. Iteração em Arrays: Utilize um loop for para percorrer um array e imprimir cada elemento.

Exercício 4. Tabuada: Use um loop for para imprimir a tabuada de um número fornecido pelo usuário.



Parte 2: Estruturas de Repetição (while)

- - Conceito e Sintaxe de while

Definição: Uma estrutura de controle que repete um bloco de código enquanto uma condição especificada é verdadeira.

Componentes do while

Condição: Expressão booleana que determina a continuidade do loop.

Controle de Loop: Importante atualizar a variável de controle dentro do loop para evitar loops infinitos.

Exemplo:

```
while (condição) {  
    // Bloco de código a ser repetido  
}
```



Parte 2: Estruturas de Repetição (while)

- - Aplicações do while
- **Flexibilidade:** O loop while é ideal para situações onde o número de iterações não é conhecido antecipadamente.
- **Controle Cuidadoso:** Requer atenção especial para evitar loops infinitos.

Exemplo 1:

```
let i = 0;
while (i < 5) {
    console.log(i); // Imprime números de 0 a 4
    i++;
}
```



Parte 2: Estruturas de Repetição (while)

- - Exercícios e Discussão

Exercício 1. Contador com Condição: Utilize um loop while para imprimir números de 1 a 10, mas interrompe o loop se o número for 5.

Exercício 2. Loop de Menu: Crie um menu de opções básico onde o usuário pode escolher uma opção ou sair. Use while para manter o menu ativo.

Exercício 3. Busca em Array: Dado um array, use um loop while para procurar um elemento específico e imprimir sua posição.

Exercício 4. Acumulador: Inicialize uma variável com 0 e use while para acumular a soma de números até 100.

Parte 3: Estruturas de Repetição (do-while)

- - Conceito e Sintaxe de do-while

Definição: Uma variação do loop while que garante a execução do bloco de código pelo menos uma vez antes de verificar a condição.

Componentes do do-while

Bloco de Código: Executado antes da avaliação da condição.

Condição: Avaliada após a execução do bloco de código.

Exemplo:

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



Parte 3: Estruturas de Repetição (do-while)

- - Aplicações do do-while
- **Garantia de Execução:** Útil em situações onde o bloco de código precisa ser executado pelo menos uma vez.
- **Cuidado com a Condição:** Importante para evitar loops infinitos.

```
let continuar;  
do {  
    console.log("Executando uma vez");  
    continuar = confirm("Deseja continuar?");  
} while (continuar);
```



Parte 3: Estruturas de Repetição (do-while)

● - Exercícios e Discussão

Exercício 1. Repetir Nome: Peça ao usuário para digitar seu nome e use o loop do-while para repetir a pergunta até que um nome seja inserido.

Exercício 2. Jogo de Adivinhação: Crie um jogo simples onde o usuário deve adivinhar um número. Use do-while para repetir a pergunta até que o número correto seja adivinhado.

Exercício 3. Validação de Entrada: Solicite um número ao usuário e use do-while para garantir que uma entrada válida seja fornecida antes de prosseguir.

Exercício 4. Menu Interativo: Implemente um menu de opções que se repete com do-while até que o usuário escolha a opção de sair.

Parte 4: Loop Infinito

- O que é um Loop Infinito?
- **Definição:** Um loop que nunca atinge sua condição de término, resultando em uma execução contínua.
- **Causa Comum:** *Erros na lógica de controle de fluxo.*
- **Exemplo genérico:**

```
while (true) {  
    // Código que é executado  
    //repetidamente sem fim  
}
```



Parte 4: Loop Infinito

- O que é um Loop Infinito?
- **Definição:** Um loop que nunca atinge sua condição de término, resultando em uma execução contínua.
- **Causa Comum:** *Erros na lógica de controle de fluxo.*
- **Exemplo genérico:**

```
while (true) {  
    // Código que é executado  
    //repetidamente sem fim  
}
```

Nota: Em muitos casos, um loop infinito é indesejado e resulta de um erro no código.



Parte 4: Loop Infinito

Exemplos

- Intencional

// Um servidor que escuta indefinidamente por requisições

```
while (true) {  
    escutarRequisicoes();  
}
```

- Erro comum

```
let i = 0;  
while (i < 5) {  
    // Esquecido de incrementar i  
    console.log(i);  
}
```

Parte 4: Loop Infinito

Conclusão sobre Loops Infinitos

Atenção Necessária: Loops infinitos devem ser evitados, a menos que sejam intencionais e controlados.

Importância do Controle de Fluxo:
Compreender e controlar o fluxo de execução é essencial para evitar loops infinitos não intencionais.

Parte 5. Encerramento

- - Resumo dos tópicos abordados e orientações sobre recursos para continuar aprendendo JavaScript:
 - MDN Web Docs:
<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>
- - Próximos passos e recursos adicionais para aprendizado autônomo:
 - HTML: <https://www.w3schools.com/html/default.asp>
 - JavaScript: <https://www.w3schools.com/js/default.asp>

Dúvidas?



Obrigado!

Mário Carvalho

mario.carvalho@ufms.br