

Laços de Repetição em JavaScript

**APRENDENDO A AUTOMATIZAR
TAREFAS COM FOR E WHILE**

Gregory Klaus

O que é um Laço de Repetição?

Laços de repetição (ou loops) são estruturas que permitem executar o mesmo bloco de código várias vezes automaticamente

Usamos quando queremos:

- Repetir ações várias vezes.
- Percorrer estruturas como arrays e objetos.
- Reduzir a quantidade de código repetido.

Existem três laços principais no JavaScript:

- for
- for...in
- while

Exemplo sem e com laço

Sem laço:

```
console.log(1);  
console.log(2);  
console.log(3);  
console.log(4);  
console.log(5);
```

Com laço for:

```
for (let i = 1; i <= 5; i++) {  
  console.log(i);  
}
```

O segundo exemplo faz a mesma coisa com apenas 3 linhas.
Mais limpo, reutilizável e fácil de manter.

O que é o for?

O laço for é ideal quando sabemos exatamente quantas vezes queremos repetir uma ação..

Ele é composto por 3 partes principais:

```
for (inicialização; condição;  
    incremento) {  
    // bloco de código a ser  
    executado  
}
```

Cada parte:

- Inicialização: onde criamos a variável de controle. Ex: let i = 0.
- Condição: determina até quando o laço vai continuar. Ex: i < 5.
- Incremento: altera a variável após cada repetição. Ex: i++.

Exemplo detalhado com for

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

O que está acontecendo:

1. `let i = 0` → o laço começa com `i` valendo 0.
2. `i < 5` → enquanto `i` for menor que 5, o bloco será executado.
3. `console.log(...)` → imprime o valor atual de `i`.
4. `i++` → depois de imprimir, `i` recebe mais 1 (vai para o próximo número).

Resultado no console:

Número: 0

Número: 1

Número: 2

Número: 3

Número: 4

Exercícios com for

1. Imprimir os números de 1 a 10 no console.
2. Imprimir os números pares de 0 a 20.
3. Somar os números de 1 a 100 e mostrar o resultado no console.
4. Fazer a tabuada do 5 até o 10.
Exemplo($5 \times 1 = 5$, $5 \times 2 = 10$)

Correção dos exercícios com for (Comentada)

Exercício 1: Imprimir números de 1 a 10

```
for (let i = 1; i <= 10; i++) {  
  console.log(i); // Exibe o valor de i no console  
}
```


Correção dos exercícios com for (Comentada)

Exercício 2: Imprimir números pares de 0 a 20

```
for (let i = 0; i <= 20; i += 2) {  
  console.log(i); // Mostra apenas os números  
pares  
}
```


Correção dos exercícios com for (Comentada)

Exercício 3: Somar os números de 1 a 100

```
let soma = 0; // variável acumuladora
```

```
for (let i = 1; i <= 100; i++) {  
  soma += i; // soma = soma + i  
}
```

```
console.log("Soma total: " + soma); // Mostra o resultado  
final
```


Correção dos exercícios com for (Comentada)

Exercício 3: Somar os números de 1 a 100

```
let soma = 0; // variável acumuladora
```

```
for (let i = 1; i <= 100; i++) {  
  soma += i; // soma = soma + i  
}
```

```
console.log("Soma total: " + soma); // Mostra o resultado  
final
```


O que é o while?

O laço while repete um bloco de código enquanto uma condição for verdadeira.

Diferente do for, o while é útil quando não sabemos quantas vezes vamos repetir.

Exemplo:

```
let i = 1;
while (i <= 5) {
  console.log("Número: " + i);
  i++; // Não esquecer de incrementar, ou vira loop infinito!
}
```

Sintaxe:

```
while (condição) {
  // código repetido
}
```


Exercícios com while

1. Imprimir os números de 1 a 10 com while.
2. Mostrar os múltiplos de 3 de 0 a 30.
3. Fazer uma contagem regressiva de 10 até 1.

Correção dos exercícios while (Comentada)

Exercício 1

```
let i = 1;  
while (i <= 10) {  
  console.log(i);  
  i++;  
}
```


Correção dos exercícios while (Comentada)

```
// Inicializa a variável j com o valor 0
```

```
let j = 0;
```

```
// Enquanto j for menor ou igual a 30, o
```

```
laço continua
```

```
while (j <= 30) {
```

```
    // Verifica se o valor de j é múltiplo de 3
```

```
    if (j % 3 === 0) {
```

```
        // Se for múltiplo de 3, imprime o valor  
no console
```

```
        console.log(j);
```

```
    }
```

```
    // Incrementa o valor de j em 1 para a  
próxima iteração
```

```
    j++;
```

```
}
```


Correção dos exercícios

while (Comentada)

// Inicializa a variável j com o valor 0

```
let j = 0;
```

// Enquanto j for menor ou igual a 30, o laço continua

```
while (j <= 30) {
```

// Verifica se o valor de j é múltiplo de 3

```
if (j % 3 === 0) {
```

// Se for múltiplo de 3, imprime o valor no console

```
  console.log(j);
```

```
}
```

// Incrementa o valor de j em 1 para a próxima iteração

```
  j++;
```

```
}
```

.....

Atenção:

se esquecer de mudar o valor da variável de controle (i, j, k), o laço nunca para (loop infinito).

Desafio final (integrador)

Desafio:

- Peça ao usuário um número com `prompt()`
- Mostre todos os números de 1 até o número digitado
- Diga se cada número é par ou ímpar
- Use `for` ou `while`

Dica: use `Number(prompt(...))` para converter o dado de string para número.

Correção do desafio final

(Comentada)

```
let numero = Number(prompt("Digite um número:")); // Pede um  
número ao usuário
```

```
for (let i = 1; i <= numero; i++) { // Vai de 1 até o número informado  
  if (i % 2 === 0) {  
    console.log(i + " é par"); // Se for divisível por 2, é par  
  } else {  
    console.log(i + " é ímpar"); // Caso contrário, é ímpar  
  }  
}
```




O que é uma função?

Uma função é um bloco de código nomeado que executa uma tarefa específica.

- Você cria a função para organizar seu código, deixando ele mais limpo.
- Depois, pode usar essa função quantas vezes quiser.
- Pode receber informações para trabalhar (chamadas parâmetros).
- Pode devolver um resultado para ser usado em outra parte do programa.

Analogia: Pense na função como uma máquina de café: você coloca o café e a água (entrada), ela prepara o café (processo) e te entrega a bebida pronta (saída).



Como criar uma função

```
function soma(numero1, numero2):{  
    return numero1 + numero2  
}
```

- A palavra function indica que você está criando uma função.
- soma é o nome da função.
- O código indentado abaixo é o que a função faz.

Importante: Criar a função não executa ela. Para rodar o código, você precisa chamar a função:

```
soma() "
```




Parâmetros e Argumentos

O que são?

- Parâmetros são variáveis que ficam entre parênteses na definição da função.
 - Elas servem para receber informações quando a função for chamada.
 - Argumentos são os valores que você passa para esses parâmetros quando usa a função.

Analogia: A função é uma receita de bolo. Os parâmetros são os ingredientes que a receita pede, e os argumentos são os ingredientes que você coloca de verdade na hora de fazer o bolo.



Parâmetros e Argumentos

exemplos:

```
soma(5, 3); // Soma: 8  
soma(10, 7); // Soma: 17
```

```
function saudacao(nome) {  
  console.log("Olá,", nome);  
}
```

```
saudacao("Ana"); // Olá, Ana  
saudacao("João"); // Olá, João
```

```
function soma(a, b) {  
  console.log("Soma:", a + b);  
}
```

```
function repetirMensagem(msg,  
  vezes) {  
  for (let i = 0; i < vezes; i++) {  
    console.log(msg);  
  }  
}
```

```
repetirMensagem("Oi!", 3);  
// Oi!  
// Oi!  
// Oi!
```


Obrigado!

O que é o for...in?

O for...in é usado para percorrer todas as chaves (propriedades) de um objeto.

Sintaxe:

```
for (let chave in objeto) {  
  // código com objeto[chave]  
}
```

Aqui:

- info representa a chave (ex: "nome", "idade").
- aluno[info] acessa o valor da chave.

Exemplo:

```
const aluno = { nome: "João",  
  idade: 17, curso: "JS" };  
for (let info in aluno) {  
  console.log(info + ": " +  
    aluno[info]);  
}
```


Exercícios com for...in

1. Criar um objeto com informações de um carro e imprimir as chaves e valores.
2. Criar um objeto de aluno com nome, nota e presença, e exibir tudo.
3. Criar um objeto com 4 filmes favoritos e mostrar os nomes no console.

Correção dos exercícios for...in (Comentada)

Exercício 1

```
const carro = { marca: "Toyota", modelo: "Corolla",  
ano: 2020 };  
for (let propriedade in carro) {  
  console.log(propriedade + ": " +  
carro[propriedade]);  
}
```


Correção dos exercícios for...in (Comentada)

Exercício 2

```
const aluno = { nome: "Ana", nota: 9.2, presente:  
true };
```

```
for (let dado in aluno) {  
  console.log(dado + ": " + aluno[dado]);  
}
```


Correção dos exercícios for...in (Comentada)

Exercício 2

```
const filmes = { filme1: "Matrix", filme2: "Interestelar",  
filme3: "Inception", filme4: "Shrek" };
```

```
for (let key in filmes) {  
  console.log(filmes[key]);  
}
```


Importante:

- for...in não serve para arrays (para isso usamos for ou forEach).
- Sempre usamos objeto[chave] para acessar valores dentro de um for...in.