

Módulo 4 – Laços de Repetição em JavaScript

Objetivo do módulo

Aprender a usar estruturas que repetem ações automaticamente, economizando trabalho e tornando o código mais eficiente.

1. Por que usar laços?

Muitas vezes precisamos repetir tarefas várias vezes. Em vez de escrever o mesmo código várias vezes, usamos **laços de repetição**.

Exemplo simples: imprimir números de 1 a 5.

Sem laço:

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

Com laço:

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

2. Laço for

O **for** é usado quando sabemos **quantas vezes** queremos repetir algo.

Sintaxe:

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

Exemplo:

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

3. Laço while

O **while** repete **enquanto** uma condição for verdadeira.

Sintaxe:

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

Exemplo:

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

4. Laço `do...while`

O `do...while` garante que o bloco **execute pelo menos uma vez**, mesmo que a condição seja falsa no início.

Sintaxe:

```
do {  
    // código a ser repetido  
} while(condição);
```

Exemplo:

```
let contador = 0;  
  
do {  
    console.log("Contador: " + contador);  
    contador++;  
} while(contador < 5);
```

5. Combinando laços com condicionais

É comum usar `if/else` dentro de um laço para tomar decisões em cada iteração.

Exemplo: imprimir apenas números pares de 1 a 10.

```
for(let i = 1; i <= 10; i++) {  
    if(i % 2 === 0) {  
        console.log(i + " é par");  
    }  
}
```

6. Boas práticas

- Evite laços infinitos, que nunca param:

```
while(true) {  
    // cuidado! isso não termina sozinho  
}
```

- Use variáveis claras para controle de laço.
- Indente corretamente para facilitar a leitura do código.

7. Próximos passos

No próximo módulo você aprenderá sobre **funções em JavaScript**, que permitem **organizar o código em blocos reutilizáveis**, tornando seus programas mais claros e fáceis de manter.

Pratique criando laços para automatizar tarefas simples, como somar números, imprimir listas ou verificar condições.