

# JavaScript: Fundamentos para Desenvolvimento Web Interativo - Turma 2025B

## 2.9.2 Laços de repetição (loops)

Os laços permitem repetir um bloco de código várias vezes, geralmente com uma condição de término.

### for

O laço de repetição **for** é usado para executar um bloco de código várias vezes. Ele oferece uma maneira conveniente de iterar sobre uma sequência de valores ou realizar uma ação um número específico de vezes.

A sintaxe do **for** é composta por três partes principais, conforme quadro abaixo:

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

Onde:

- **Inicialização:** É executada apenas uma vez antes do início do loop. É usada para inicializar o contador ou variável de controle do loop.
- **Condição:** É avaliada antes de cada iteração do loop. Se for verdadeira, o bloco de código é executado. Se for falsa, o loop é encerrado.
- **Incremento/Decremento:** É executado após cada iteração do loop. É usado para alterar o valor da variável de controle do loop.

Veja no abaixo um exemplo:

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

Observe neste exemplo que:

- **let i = 0:** Inicializamos a variável **i** com o valor 0.
- **i < 5:** A condição verifica se **i** é menor que 5. Se for verdadeira, o loop continua.
- **i++:** Incrementamos **i** após cada iteração.

O **for** é frequentemente usado para iterar sobre os elementos de um array. Veja um exemplo no quadro abaixo:

```
let frutas = ["Maçã", "Banana", "Laranja"];  
for (let i = 0; i < frutas.length; i++) {  
  console.log(frutas[i]); // Imprime cada elemento do array  
}
```

Neste exemplo, **frutas.length** fornece o número de elementos no array, permitindo que o loop itere sobre todos os elementos.

Você pode personalizar o incremento/decremento conforme necessário. Veja um exemplo no quadro abaixo:

```
for (let i = 0; i < 10; i += 2) {  
  console.log(i); // Imprime os valores de 0 a 8 (pulando de 2 em 2)  
}
```

Também é possível fazer uma contagem regressiva usando decremento. Veja um exemplo no quadro abaixo:

```
for (let i = 5; i > 0; i--) {  
  console.log(i); // Imprime os valores de 5 a 1 (em ordem decrescente)  
}
```

while

A estrutura de controle **while** é usada para executar um bloco de código repetidamente enquanto uma condição específica for verdadeira. É útil quando o número de iterações não é conhecido antecipadamente e depende de uma condição dinâmica.

A estrutura básica do **while** é mostrada no quadro abaixo:

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

Observe que **condição** é uma expressão avaliada antes de cada iteração do loop. Se for verdadeira, o bloco de código é executado. Se for falsa, o loop é encerrado.

Para exemplificar, vamos criar um loop usando o **while** para imprimir os números de 1 a 5. Veja este exemplo no quadro abaixo:

```
let contador = 1;  
while (contador <= 5) {  
    console.log(contador); // Imprime os números de 1 a 5  
    contador++;  
}
```

Neste exemplo:

- **contador** é inicializado com o valor 1.
- **contador <= 5** é a condição que verifica se o **contador** é menor ou igual a 5. Enquanto essa condição for verdadeira, o bloco de código dentro do **while** será executado.
- **contador++** é usado para incrementar o contador após cada iteração.

**OBS:** É importante garantir que a condição do **while** eventualmente se torne falsa para evitar loops infinitos. Por exemplo, se a condição nunca for falsa, o loop continuará para sempre. Veja no quadro abaixo um exemplo:

```
let x = 0;  
while (x < 10) {  
    console.log(x); // Este loop é infinito, pois x nunca é incrementado  
}
```

## do-while

A estrutura de controle **do-while** é semelhante ao **while**, mas com uma diferença importante: no **do-while**, o bloco de código é executado pelo menos uma vez, mesmo se a condição for falsa. Depois da primeira execução, a condição é verificada, e se for verdadeira, o bloco de código é executado novamente. Essa estrutura é útil quando você precisa garantir que o bloco de código seja executado pelo menos uma vez, independentemente da condição.

A estrutura básica do **do-while** é a mostrada no quadro abaixo:

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

Onde:

- **Bloco de código:** É o bloco de código a ser executado repetidamente.
- **condição:** É uma expressão avaliada após a execução do bloco de código. Se for verdadeira, o bloco de código é executado novamente. Se for falsa, o loop é encerrado.

Veja um exemplo no quadro abaixo:

```
let contador = 0;  
do {  
    console.log(contador); // Imprime 0 a 2  
    contador++;  
} while (contador < 3);
```

Seguir para...

2.9.3 Controle de fluxo ►

[Baixar o aplicativo móvel.](#)