

# JavaScript

**Tecnologia em Análise e Desenvolvimento de Sistemas**

Senac - Portão - Curitiba/PR

Prof. Msc. Guilherme Matos Barbosa

# JavaScript

## Estrutura de repetição for

- A estrutura de repetição ``for`` é uma maneira eficiente de executar um bloco de código várias vezes em JavaScript;
- É particularmente útil quando você sabe exatamente quantas vezes deseja repetir uma ação.

Estrutura base de um laço for:

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



# JavaScript

## Estrutura base de um laço for:

**Inicialização:** É onde você define e/ou inicializa a variável de controle do loop. Isso geralmente é feito atribuindo um valor inicial a uma variável.

**Condição:** É a expressão que é avaliada antes de cada iteração do loop. Se a condição for verdadeira, o bloco de código dentro do loop é executado. Se for falsa, o loop é interrompido.

**Incremento:** É a operação que é realizada após cada iteração do loop. Geralmente, é usada para aumentar ou diminuir o valor da variável de controle do loop.

# JavaScript

## Estrutura de repetição for

Vamos ver um exemplo simples de um loop for que imprime os números de 1 a 5 no console:

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



# JavaScript

## Condições de Parada:

- Além das condições de inicialização, condição e incremento, você também pode adicionar condições de parada dentro do bloco de código do loop;
- Isso permite interromper o loop antes de ele ser concluído, com base em certas condições.



# JavaScript

## Exemplo com Condição de Parada usando *`continue`*:

- Suponha que queremos imprimir apenas os números pares de 1 a 10;
- Podemos adicionar uma condição de parada dentro do loop para verificar se o número atual é par;
- Se não for, podemos pular para a próxima iteração do loop.

# JavaScript

Exemplo com Condição de Parada usando ``continue``:

```
for (let i = 1; i <= 10; i++) {  
  if (i % 2 !== 0) {  
    continue; // pula para a próxima iteração se o número não for par  
  }  
  console.log(i);  
}
```



# JavaScript

## Exemplo com Condição de Parada usando *break*:

- Agora, vamos considerar um exemplo onde queremos imprimir os números de 1 até encontrar um número maior que 5;
- Nesse caso, podemos usar `break` para sair do loop assim que encontrarmos o número maior que 5.



# JavaScript

Exemplo com Condição de Parada usando *break*:

```
for (let i = 1; i <= 10; i++) {  
    console.log(i);  
    if (i > 5) {  
        break; // sai do loop quando o número for maior que 5  
    }  
}
```

# JavaScript

## Estrutura de repetição while

Além do `for`, JavaScript também oferece o laço de repetição `while`, que permite executar um bloco de código enquanto uma condição específica for verdadeira.

A sintaxe básica do `while` é a seguinte:

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



# JavaScript

## Funcionamento do while:

- A condição é avaliada antes de cada iteração do loop;
- Se a condição for verdadeira, o bloco de código dentro do while é executado;
- Se a condição for falsa, o loop é interrompido e a execução continua após o bloco while.

# JavaScript

## Estrutura de repetição while

Vamos ver um exemplo simples de um loop while que imprime os números de 1 a 5 no console:

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



# JavaScript

## Comparação entre for e while:

- O for é frequentemente usado quando você sabe exatamente quantas vezes deseja repetir uma ação;
- O while é mais adequado quando você precisa repetir uma ação, mas não sabe exatamente quantas vezes ela será repetida.