

IOS – Instituto de  
Oportunidade Social

## JS 05 - Laços de repetição



- Compreender o uso de **laços de repetição**;
- Diferenciar e entender quando aplicar os laços **for**, **while** e **do-while**;
- Combinar o uso de laços com **arrays**.

IOS – Instituto de  
Oportunidade Social

Laços de repetição



Os **laços de repetição** são comandos utilizados para **executar instruções mais de uma vez**, ou seja, cria **loops** de repetição de uma ou mais instruções com um número **limitado de vezes**. Eles possuem com variáveis de controle/acumuladora.

Para trabalhar com laços de repetições, sempre será necessário testar uma **condição** que irá resultar em um valor **booleano** (true ou false). Por isso, quase sempre, fazemos usar de operadores de **comparação** e/ou **lógicos**:

## Operadores de **Comparação**:

Categoria	Operador	Descrição
Operadores de comparação	==	Igual
	!=	Diferente
	<	Menor que
	<=	Menor ou igual
	>	Maior que
	>=	Maior ou igual
	===	Triplo igual

## Operadores de Lógicos:

Categoria	Operador	Descrição
Operadores lógicos	&&	Lógica “and” ou “e”, que retorna verdadeiro se todos os operandos forem verdadeiros.
		Lógica “or” ou “ou”, que retorna verdadeiro se pelo menos um operando for verdadeiro.
	!	Lógica “not” ou “não”, que inverte o valor lógico se é verdadeiro, retorna falso e se é falso retorna verdadeiro.

Uma estrutura de repetição permite especificar que um **BLOCO ( { } )** de instruções ou o programa todo deverá ser repetido **enquanto** alguma **condição** permanecer **verdadeira**. O número de repetições pode ser fixo ou atender alguma condição.

Vantagens: O algoritmo passa a ter um **número menor de linhas** e de instruções repetidas devido a estrutura de repetição e é possível aumentar a **amplitude de processamento** sem alterar o tamanho do código.

IOS – Instituto de  
Oportunidade Social

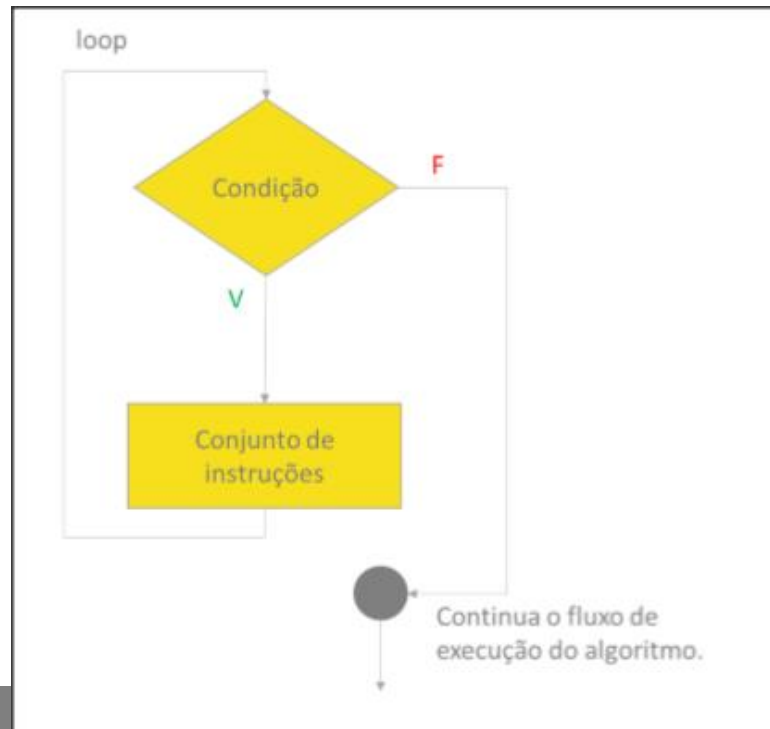
Laço white





# Laço while

O laço **while** é uma repetição com **teste no início do comando**. Assim, o bloco de instruções será repetidamente **executado enquanto a condição for verdadeira**. Quando a condição for falsa, a execução do bloco de comandos será interrompido:



A **sintaxe** da estrutura while é:

```
while ( condição ) {  
    instrucao1;  
    instrucao2;  
    instrucao3;  
    instrucaoN;  
}  
proximalInstrucao;
```

O **bloco de comandos** é executado enquanto a **condição** for **verdadeira**. As chaves podem ser ocultadas se o comando contiver apenas uma instrução: **while ( condição ) instrucao1;**

A **sintaxe** da estrutura while é:

```
while ( condição ) {  
    instrucao1;  
    instrucao2;  
    instrucao3;  
    instrucaoN;  
}  
proximalInstrucao;
```

O **bloco de comandos** é executado enquanto a **condição** for **verdadeira**. As chaves podem ser ocultadas se o comando contiver apenas uma instrução: **while ( condição ) instrucao1;**

# Laço while

Todo laço de repetição pode ter um contador com **variável de controle** da condição. Um **contador** é uma **variável do tipo inteiro** usada para contar a quantidade de vezes que um bloco de instruções é repetido e deve ser inicializado antes de ser utilizado.

Inicialização;

```
while(condição){  
  Bloco de instruções;  
  incremento ou decremento;  
}
```

```
let contador = 1;
```

```
while(contador <= 10){  
  console.log(`Valor = ${contador}`);  
  contador++;  
}
```

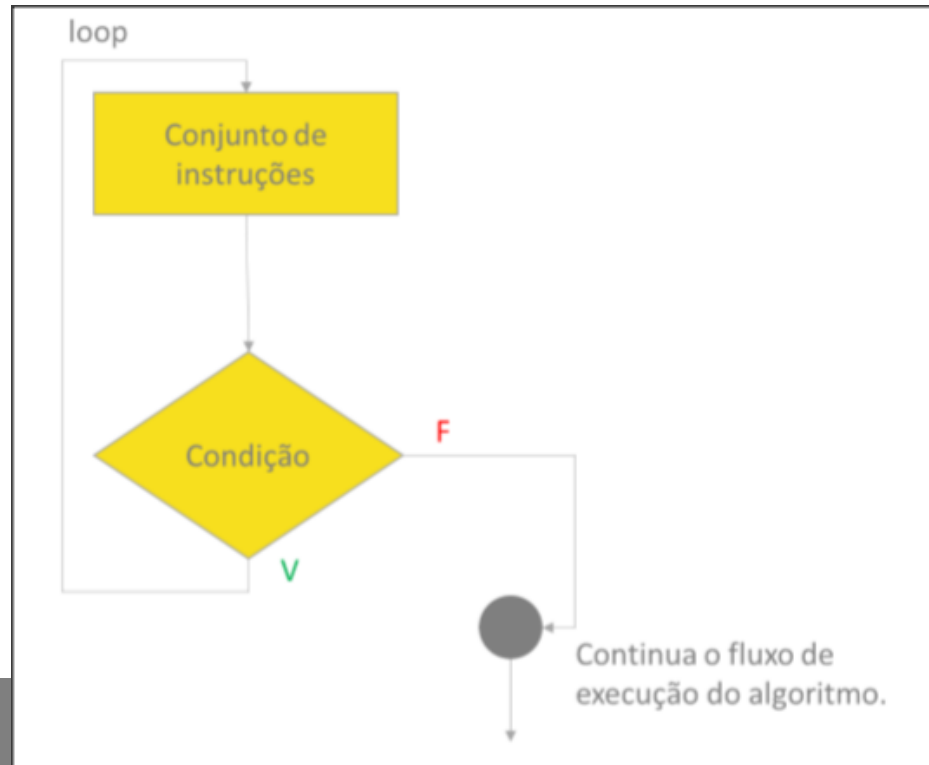
IOS – Instituto de  
Oportunidade Social

Laço do-while



# Laço do-while

Esse tipo de estrutura de repetição é caracterizado por fazer o **teste de controle no final do bloco de comandos**. Os **comandos** repetidos são **executados pelos menos uma vez** antes da condição ser testada. A condição é sempre testada no final do laço e após a execução do bloco de comandos dele.



A **diferença básica** para o comando while é que o bloco a ser repetido sempre **executa ao menos uma vez**. A **sintaxe** do-while é:

```
do{  
    instrucao1;  
    instrucao2;  
    instrucao3;  
    instrucaoN;  
} while( condição );  
proximalInstrucao;
```

O bloco de comandos é executado enquanto a condição for verdadeira.

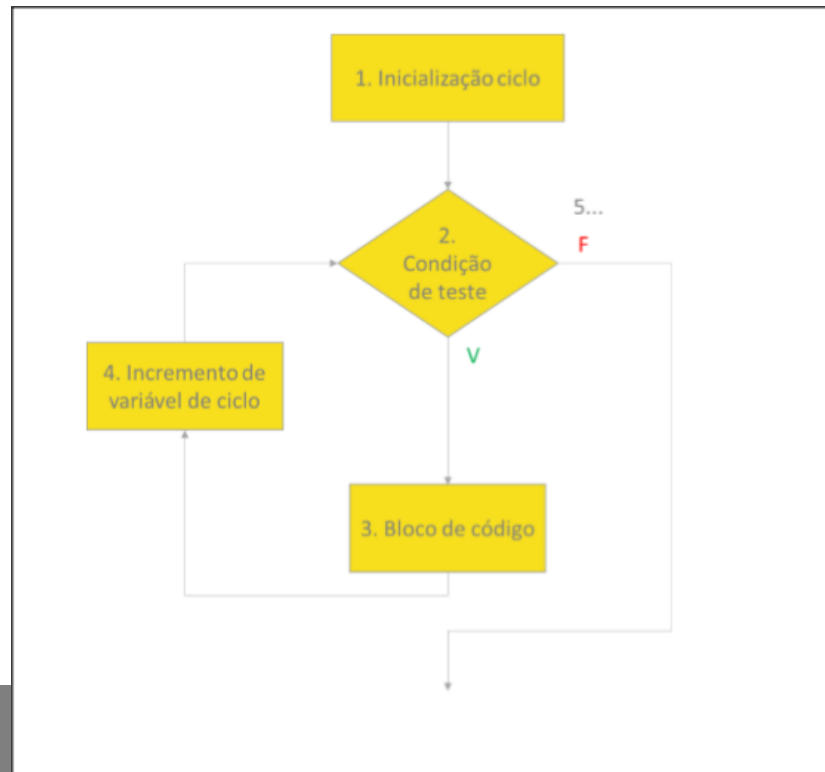
IOS – Instituto de  
Oportunidade Social

Laço For





O comando **for** também **realiza o teste lógico no início do laço**. Ele tem uma estrutura um pouco diferente do **while**, mas sua execução é similar. O programa não executará nenhuma repetição (ações programadas) sem antes testar a condição:



# Laço For

A sintaxe do comando for é:

```
1)          2)          3)  
for(inicialização; condição; incremento){  
    instrucao1;  
    instrucao2;  
    instrucaoN;  
}  
proximaInstrucao;
```

- 1) A inicialização é uma atribuição executada uma única vez antes
- 2) Condição que controla o laço. Repete o **bloco ({} enquanto a condição for verdadeira**
- 3) O incremento é sempre a última instrução do laço (ex: cont++)

As chaves podem ser ocultadas se o comando contiver apenas uma instrução: **for**(inicializacao; condicao; incremento ou decremento)  
**instrução;**

# Laço For

Comando for com incremento:

```
for (let i = 0; i < 10; i++)
```

The code snippet is enclosed in a box. Above the code, three labels are positioned: 'Inicialização' (Initialization) in blue, 'Condição' (Condition) in green, and 'Incremento' (Increment) in orange. The code itself is color-coded to match these labels: 'let i = 0' is blue, 'i < 10' is green, and 'i++' is orange.

Comando for com decremento:

```
for (let i = 10; i > 0; i--)
```

The code snippet is enclosed in a box. Above the code, three labels are positioned: 'Inicialização' (Initialization) in blue, 'Condição' (Condition) in green, and 'Decremento' (Decrement) in orange. The code itself is color-coded to match these labels: 'let i = 10' is blue, 'i > 0' is green, and 'i--' is orange.

IOS – Instituto de  
Oportunidade Social

Vamos Praticar



Apostila de JS

04.JavaScript

Páginas 68 a 79

OBS: Acompanhar o passo a passo com o instrutor

IOS – Instituto de  
Oportunidade Social

Conclusão



Procure sempre fazer mais do que é passado para você em sala de aula. Por exemplo, você pode **alterar as condições**, utilizar **comparações múltiplas nos laços**, criar **novos laços** para ver o resultado que será gerado.

Seguem alguns links para você estudar e aprender mais:

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/while>

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/do...while>

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/for>

IOS – Instituto de  
Oportunidade Social

## Exercícios





Criar uma página **lacosRepeticao.html** com seu respectivo arquivo JS com os exercícios abaixo, mostrando **cada passo** no **console** do navegador:

- 1) Mostrar no console os números de 1 ao 50. (**while**)
- 2) Mostrar no console contagem regressiva de 10 até 0. (**do-while**)
- 3) Mostrar no console os números pares de 1 a 100. (**for**)