




# Aprendendo LAÇOS DE REPETIÇÃO na prática



Uma forma mais intuitiva e eficiente usando teorias e exercícios para aprender Laços de Repetição mais rapidamente.

ALUNOS:  
José Roberto  
Vanessa  
Wallyson



# Os passos a serem seguidos no Aprendizado

---

## 1. Teoria:

- a. O que é estrutura de repetição;
- b. Para que são utilizadas as estruturas de repetição;
- c. Quais são os laços de repetição e como cada um funciona;

## 2. Prática:

- a. Fazer os exercícios presente no link: {LINK DA APP WEB COM EXERCICIOS}
- b. Treinar conhecimentos adquiridos

# O que é estrutura de repetição

---

Laços de repetição, também conhecidos como laços de iteração ou simplesmente loops, são comandos que permitem iteração de código, ou seja, que comandos presentes no bloco sejam repetidos diversas vezes. Através de laços de repetição é possível criar programas que percorram arrays, analisando individualmente cada elemento, e até mesmo criar trechos de código que sejam repetidos até que certa condição estabelecida seja cumprida.

Existem 4 estruturas de repetição básicas para praticamente todas as linguagens de programação, seja C ou Javascript.

# Para que são utilizadas as estruturas de repetição

Em muitos algoritmos, ocorre a necessidade de executar determinado comando (ou bloco de comandos) por mais de uma vez.

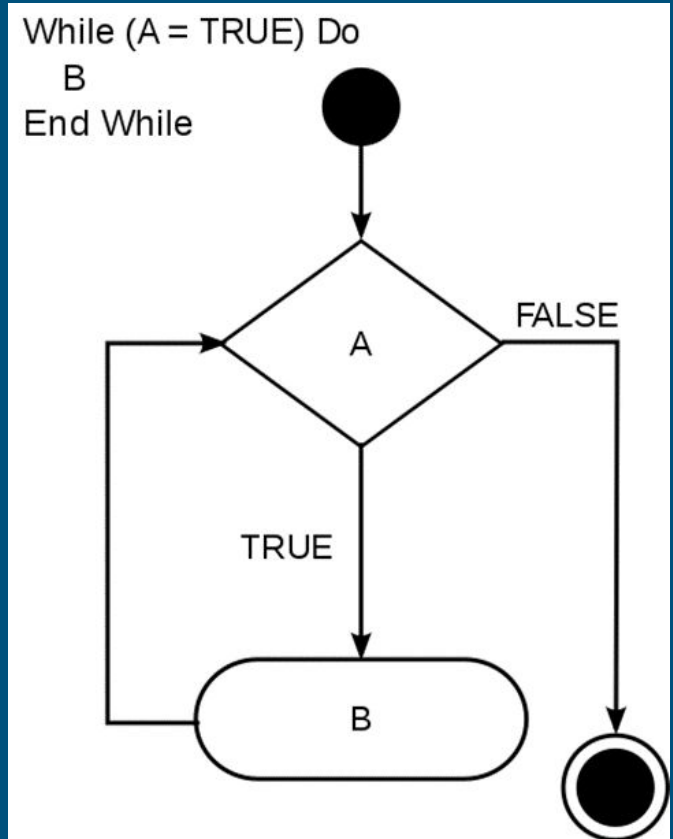
Para não precisar reescrever várias linhas em um algoritmo, pode-se utilizar as estruturas de repetição.

Ao lado um exemplo da estrutura while.



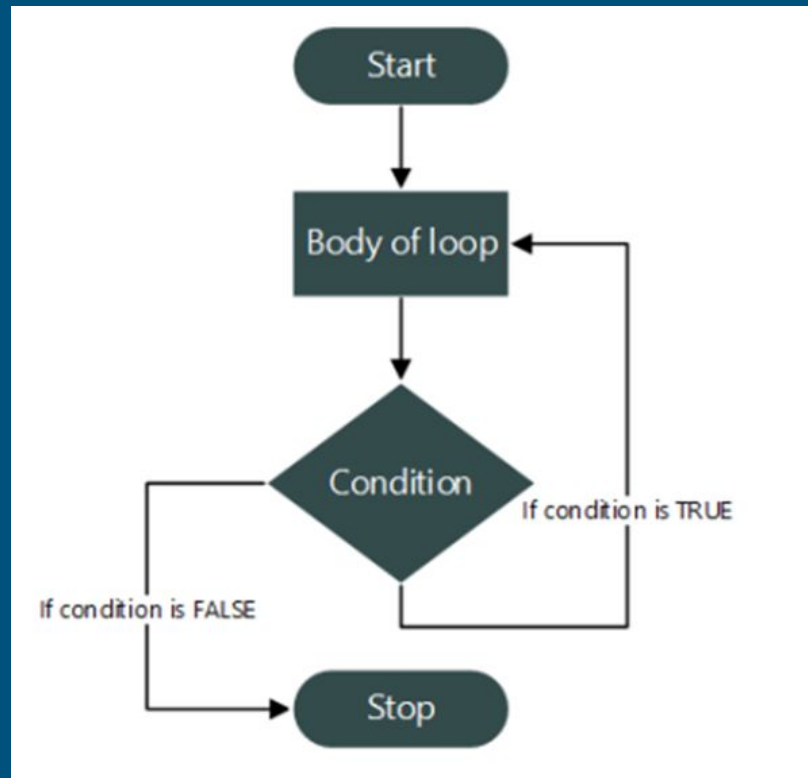
# 1- While (enquanto)

O laço while (na tradução literal para a língua portuguesa “enquanto”) determina que enquanto uma determinada condição for válida, o bloco de código será executado. O laço while testa a condição antes de executar o código, assim sendo, caso a condição seja inválida no primeiro teste o bloco nem é executado.



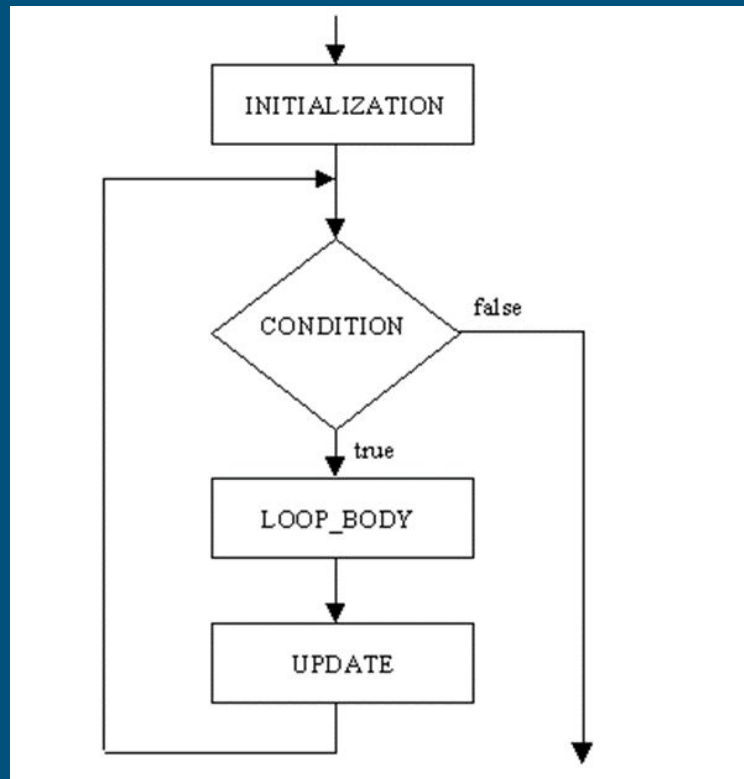
## 2 - Do... While (faça enquanto)

O laço do... while (na tradução literal para a língua portuguesa “faça... enquanto”), assim como o laço while, determina que enquanto uma determinada condição for válida o bloco de código será executado. Entretanto, do... While testa a condição após executar o código, assim sendo, mesmo que a condição seja considerada inválida no primeiro teste o bloco será executado pelo menos uma vez.



### 3- For (para)

O comando for (na tradução literal para a língua portuguesa “para”) permite que uma variável contadora seja testada e incrementada a cada iteração, sendo essas informações definidas na chamada do comando. O comando “for” recebe como entrada uma variável contadora, a condição e o valor de incremento.



## 4- Foreach (para cada)

- O FOREACH é uma simplificação do operador FOR.
- Permite acessar cada elemento individualmente iterando sobre toda a coleção sem a necessidade de informação de índices. Veja o exemplo:

```
$vetor = array(1 => 'a', 2 => 'b', 3 => 'c', 4 => 'd', 5 => 'e');
```

```
foreach($vetor as $key => $item){
```

```
    echo $key . ' = ' . $item;
```

```
}
```

Quando utilizar?

- Quando há uma coleção que você precisa acessar o valor ou dado que está no índice.

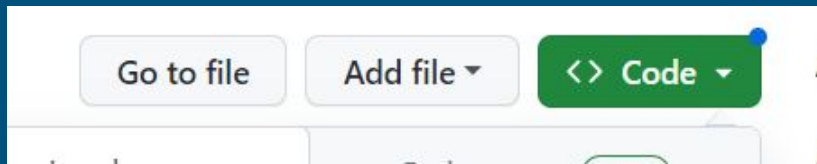


# Exercícios

---

<https://github.com/Wallysson-utfpr/projAlgoritmos>

1º Acesse o repositório e clona o projeto clicando na opção em verde e escolhendo o modo para clonar



2º Abra o projeto e encontre o arquivo "index.html"

3º Clique para abrir em algum navegador e faça os exercícios