

# Aulas Senac

Desenvolvimento Web para Adolescentes

Pablo Garcia

Laços de Repetição

## Laços de Repetição

Sabe quando você precisa repetir uma parte do seu código? E aí, você vai lá e copia e cola aquele trecho na quantidade de vezes que você precisa. Mas logo depois você olha para o que acabou de fazer e percebe que ficou estranho: Muitas linhas de código para fazer uma coisa muito pequena.

Deve ter um jeito melhor de fazer isso, né? SIM! Tem sim, são as **Estruturas de repetição**.

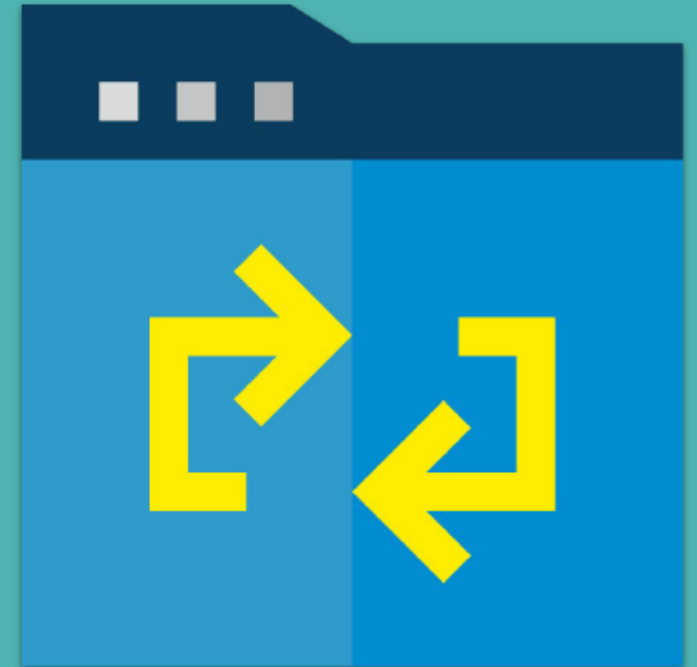


## Laços de Repetição

Então, o que é afinal estrutura de repetição? Basicamente é uma estrutura que permite repetir a execução de um conjunto de instruções por um número de vezes (fixo ou variável).

Por exemplo:

- Repetir um menu até o usuário digitar x;
- Somar os  $n$  primeiros números pares;
- Repetir o valor de 1 até 100, escrevendo cada vez que repetir, com o número da repetição.



# Laços de Repetição

## Definição

- Laços de repetição servem para **repetir um bloco de código** durante uma certa quantidade determinada ou indeterminada de vezes.
- Um laço de repetição conta com uma condição, assim como a estrutura condicional if/else.
- Enquanto a condição for satisfeita o bloco de código continua sendo executado.



# Laços de Repetição

## Definição

Existem três laços de repetição que podemos utilizar.

- O primeiro e segundo deles é o laço **WHILE** e **DO WHILE**, que executa um bloco de código enquanto uma condição for satisfeita.
- O terceiro laço é o **FOR**, que executa um laço iterando sobre valores pré determinados



# While

# Laços de Repetição

## While

### Sintaxe

```
while (condição) {  
    bloco de código  
}
```

**While** é a palavra chave reservada

**Condição** é o teste lógico que deve ser satisfeito para que o bloco de código do while seja executado

Quando o **bloco de código** termina, a condição é novamente testada, e assim o bloco de código continua sendo executado até que a condição seja falsa.



# Laços de Repetição

## While

```
let contador = 0

while(contador < 3){
  console.log('Contador: ' + contador)
  contador++
}
```

contador	saída



# Laços de Repetição

## While

```
let contador = 0
```

```
while(contador < 3){  
  console.log('Contador: ' + contador)  
  contador++  
}
```

contador	saída
0	

# Laços de Repetição

## While

VERDADEIRO

```
let contador = 0
```

```
while(contador < 3){  
  console.log('Contador: ' + contador)  
  contador++  
}
```

contador	saída
0	

# Laços de Repetição

## While

```
let contador = 0

while(contador < 3){
  console.log('Contador: ' + contador)
  contador++
}
```

contador	saída
0	Contador: 0

# Laços de Repetição

## While

```
let contador = 0

while(contador < 3){
  console.log('Contador: ' + contador)
  contador++
}
```

contador	saída
0	Contador: 0
1	

# Laços de Repetição

## While

VERDADEIRO

```
let contador = 0
```

```
while(contador < 3){  
  console.log('Contador: ' + contador)  
  contador++  
}
```

contador	saída
0	Contador: 0
1	

# Laços de Repetição

## While

```
let contador = 0

while(contador < 3){
  console.log('Contador: ' + contador)
  contador++
}
```

contador	saída
0	Contador: 0
1	Contador: 1

# Laços de Repetição

## While

```
let contador = 0

while(contador < 3){
  console.log('Contador: ' + contador)
  contador++
}
```

contador	saída
0	Contador: 0
1	Contador: 1
2	



# Laços de Repetição

## While

VERDADEIRO

```
let contador = 0
```

```
while(contador < 3){  
    console.log('Contador: ' + contador)  
    contador++  
}
```

contador	saída
0	Contador: 0
1	Contador: 1
2	

# Laços de Repetição

## While

```
let contador = 0

while(contador < 3){
  console.log('Contador: ' + contador)
  contador++
}
```

contador	saída
0	Contador: 0
1	Contador: 1
2	Contador: 2

# Laços de Repetição

## While

```
let contador = 0

while(contador < 3){
  console.log('Contador: ' + contador)
  contador++
}
```

contador	saída
0	Contador: 0
1	Contador: 1
2	Contador: 2
3	

# Laços de Repetição

## While

Falso

```
let contador = 0  
  
while(contador < 3){  
    console.log('Contador: ' + contador)  
    contador++  
}
```

contador	saída
0	Contador: 0
1	Contador: 1
2	Contador: 2
3	

# Laços de Repetição

## While

Falso

```
let contador = 0  
  
while(contador < 3){  
    console.log('Contador: ' + contador)  
    contador++  
}
```

contador	saída
0	Contador: 0
1	Contador: 1
2	Contador: 2
3	

# Laços de Repetição

## While

```
let contador = 0

while(contador < 3){
  console.log('Contador: ' + contador)
  contador++
}
```

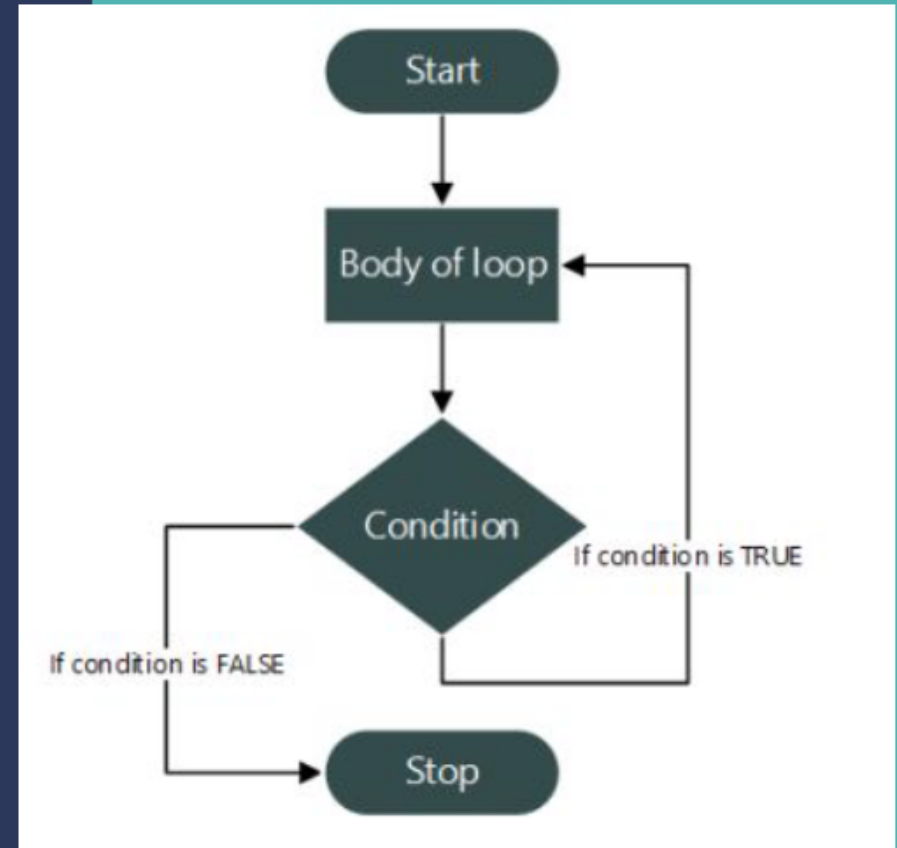
Programa Finalizado

contador	saída
0	Contador: 0
1	Contador: 1
2	Contador: 2
3	

# Laços de Repetição

## Do While (Faça Enquanto)

- Muito parecido com o while, porém tem uma diferença crucial: **a condição é verificada após executar o bloco de comandos.**
- Há um bloco de comandos e logo depois uma verificação. Assim, caso a variável condicional for alterada dentro do bloco de comandos, isso afetará a validação da condição.
- A escolha entre while e do while é mínima, então dependerá do bom senso do programador, que optará pela estrutura que deixar o algoritmo mais simples e legível.



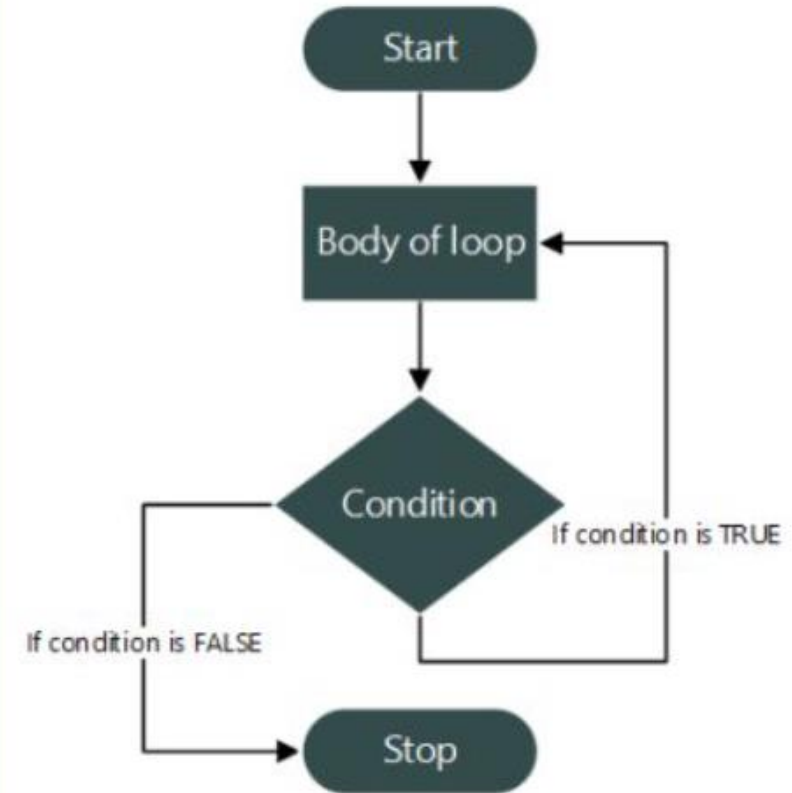


# Laços de Repetição

## Do While

```
let contador = 0

do {
  console.log('Contador: ' + contador)
  contador++
} while(contador < 3)
```



## Vamos praticar?

Faça um algoritmo que pergunte ao usuário um número inteiro e positivo qualquer e mostre uma contagem até esse valor:

Exemplo:

Input: 35

Output: 1 2 3 4 5 6 7 ... 33 34 35 Acabou!



## Vamos praticar?

Faça um programa que leia a idade de várias pessoas. A cada laço, você deverá perguntar para o usuário se ele quer ou não continuar a digitar dados. No final, quando o usuário decidir parar, mostre na tela:

- a) Quantas idades foram digitadas
- b) Qual é a média entre as idades digitadas
- c) Quantas pessoas têm 21 anos ou mais.



# For



# Laços de Repetição

## For

### Sintaxe

```
for (contador; condicao; incremento) {  
    bloco de código  
}
```

**for** é a palavra chave reservada.

**contador** é a variável que irá receber o valor inicial

**Condição** é o teste lógico que deve ser satisfeito para que o bloco de código do while seja executado

O **bloco de código** será executado tantas vezes quantos forem os elementos da lista.



# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída

# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	



# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	

# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	Contador: 1

# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	Contador: 1
2	

# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	Contador: 1
2	

# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	Contador: 1
2	Contador: 2

# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	Contador: 1
2	Contador: 2
3	

# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	Contador: 1
2	Contador: 2
3	



# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	Contador: 1
2	Contador: 2
3	Contador: 3

# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	Contador: 1
2	Contador: 2
3	Contador: 3

# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Contador	Saída
1	Contador: 1
2	Contador: 2
3	Contador: 3
4	

# Laços de Repetição

## For

```
for(let contador = 1; contador <= 3; i++) {  
  console.log('Contador: ' + contador)  
}
```

Programa Finalizado

Contador	Saída
1	Contador: 1
2	Contador: 2
3	Contador: 3
4	

## Vamos praticar?

Faça um algoritmo que pergunte ao usuário um número inteiro e positivo qualquer e mostre uma contagem até esse valor:

Exemplo:

Input: 35

Output: 1 2 3 4 5 6 7 ... 33 34 35 Acabou!





## Vamos praticar?

Escreva um programa que leia um número qualquer e mostre a tabuada desse número.

Exemplo:

Input: 5

Output:

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

...

$$5 \times 10 = 50$$

