

Objetivo:

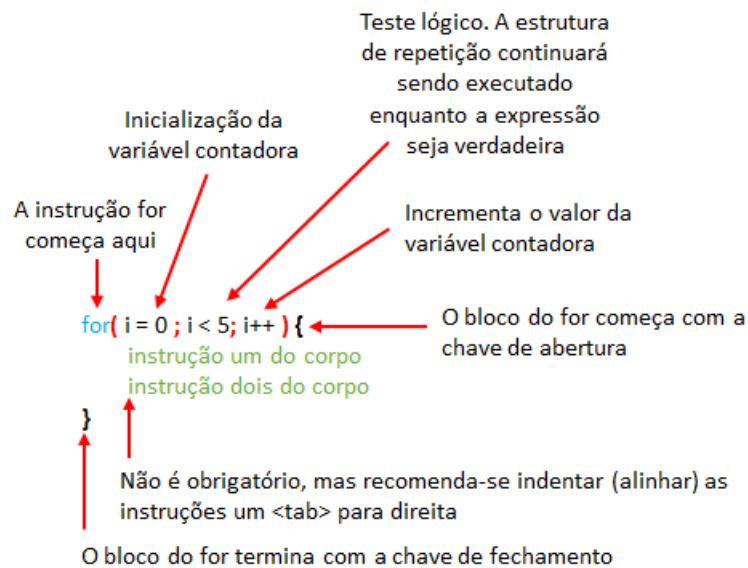
- I. Estrutura de repetição for.

Observação: Para fazer os exercícios e exemplos recomenda-se o uso da interface de programação online <https://replit.com/>.

I. Estrutura de repetição for

Ela é usada para iterar sobre os itens de uma sequência, assim como strings e arrays.

A instrução **for** possui a seguinte notação:



A instrução `i++` é equivalente a `i = i + 1`. Considere os seguintes exemplos.

```
txt = "Boa noite"
for( i = 0; i < txt.length; i++ ){
  console.log( txt[i] )
}
```

B
o
a
n
o
i
t
e

```
vetor = [4, 16, 2, 10]
for( i = 0; i < vetor.length; i++ ){
  console.log( vetor[i] )
}
```

4
16
2
10

No exemplo da esquerda, a instrução `txt[i]` acessará cada caractere da string Boa noite. Na estrutura de repetição a variável contadora `i` assumirá os valores no 0, 1, 2, A repetição continuará enquanto `i < txt.length` for verdadeiro.

No exemplo da direita, a instrução `vetor[i]` acessará cada elemento do array.

Nos exemplos a seguir, não confunda o contador de posição (variável `i`) do **for** com os caracteres da string e elementos do array.

```
txt = "Boa noite"
for( i = 0; i < txt.length; i++ ){
  console.log( i, txt[i])
}
```

```
0 B
1 o
2 a
3 n
4 o
5 i
6 t
7 e
```

```
vetor = [4, 16, 2, 10]
for( i = 0; i < vetor.length; i++ ){
  console.log( i, vetor[i] )
}
```

```
0 4
1 16
2 2
3 10
```

As estruturas de repetição **while** e **for** são equivalentes. Os exemplos anteriores podem ser codificados usando **while**, assim como é mostrado a seguir. Na estrutura **while** a variável contadora é inicializada antes da estrutura **while** e ela é incrementada dentro do corpo da estrutura de repetição.

```
txt = "Boa noite"
i = 0
while( i < txt.length ){
  console.log( i, txt[i] )
  i++
}
```

```
0 B
1 o
2 a
3 n
4 o
5 i
6 t
7 e
```

```
vetor = [4, 16, 2, 10]
i = 0
while( i < vetor.length ){
  console.log(i, vetor[i] )
  i++
}
```

```
0 4
1 16
2 2
3 10
```

A variável contadora da estrutura de repetição **for** pode ser incrementada ou decrementada, assim como pode ser observado nos exemplos a seguir. Porém, vale lembrar que neste caso ela precisa ser inicializada com o maior valor.

```
txt = "Boa noite"
for( i = txt.length-1; i >= 0; i-- ){
  console.log( i, txt[i])
}
```

```
8 e
7 t
6 i
5 o
4 n
3 a
2 o
1 B
0
```

```
vetor = [4, 16, 2, 10]
for( i = vetor.length-1; i >= 0 ; i-- ){
  console.log(i, vetor[i] )
}
```

```
3 10
2 2
1 16
0 4
```

A variável contadora da estrutura de repetição **for** pode ser incrementada ou decrementada por qualquer valor, nos exemplos a seguir os valores foram incrementados de 2. A instrução **i+=2** é equivalente a **i = i + 2**.

```
txt = "Boa noite"
for( i = 0; i < txt.length; i+=2 ){
  console.log( i, txt[i])
}
```

```
0 B
2 a
4 n
6 i
8 e
```

```
vetor = [4, 16, 2, 10]
for( i = 0; i < vetor.length; i+=2 ){
  console.log(i, vetor[i] )
}
```

```
0 4
2 2
```

Exercícios

Exercício 1: Fazer um programa que pede para o usuário entrar com o nome dele. Na sequência o programa imprime cada letra do nome numa linha. Use a estrutura de repetição for.

Dica: use a função prompt para ler o teclado e a função console.log para imprimir no console.

Exemplo de saída:

```
Entre com o seu nome:> Maria
M
a
r
i
a
```

Exercício 2: Alterar o programa do Exercício 1 para imprimir na tela somente as letras que estão em posições pares.

Dica: adicione 2 no incremento da variável contadora.

Exemplo de saída:

```
Entre com o seu nome:> Pedro Silva
P
d
o
S
l
a
```

Exercício 3: Completar o código a seguir para imprimir na tela o somatório dos elementos da variável nros. Use a estrutura de repetição for para percorrer os elementos da lista.

nros = [5,8,4,2,9,6,1,7,3]

Dica: crie uma variável para acumular a soma e inicialize ela com valor zero.

Exemplo de saída:

```
Somatório: 45
```

Exercício 4: Fazer um programa que pede para o usuário entrar com alguns números separados por vírgula. Na sequência, imprimir na tela os números fornecidos pelo usuário. Use a estrutura de repetição for.

Dica: use o método split para transformar a string numa lista.

Exemplo de saída:

```
Entre com os números separados por vírgula:> 8,2,5,3,9
8
2
5
3
9
```

Exercício 5: Alterar o programa do Exercício 4 para imprimir na tela o somatório dos valores fornecidos.

Dica: será necessário converter cada elemento da lista de string para inteiro, então use a função parseInt.

Exemplo de saída:

```
Entre com os números separados por vírgula:> 8,2,5,3,9
Somatório: 27
```

Exercício 6: Alterar o programa do

Exemplo de saída:

Exercício 4 para imprimir na tela somente os elementos que possuem valor par.

Dica: use if para testar o valor. Use a função parseInt para converter o item para inteiro.

```
Entre com os números separados por vírgula:> 8,2,5,3,9
8
2
```

Exercício 7: Alterar o programa do Exercício 4 para imprimir na tela somente os elementos que estão em posições pares.

Dica: adicione 2 no incremento da variável contadora.

Exemplo de saída:

```
Entre com os números separados por vírgula:> 8,2,3,5,9
8
3
9
```

Exercício 8: Alterar o programa do Exercício 7 para imprimir na tela o somatório dos elementos que estão em posições pares.

Dica: use a função parseInt para converter o elemento da lista para inteiro. Use uma variável para acumular o somatório dos elementos do array.

Exemplo de saída:

```
Entre com os números separados por vírgula:> 8,2,3,5,9
Somatório: 20
```