

Curso Preparatório

Semana 8:

Repetições: FOR

Admissão 2025



FOR

- Assim como no IF e While, o que marca o For é a **indentação**

JavaScript

```
for (var i=1; i<=10; i++)  
{  
    C1;  
    C2;  
    C3;  
    Cn;  
}
```

Python

```
for n in range(1,10):  
    C1  
    C2  
    C3  
    Cn
```



FOR

```
for <item> in <conjunto_de_itens>:  
    __<bloco_de_codigo>
```

- **item**: corresponde a cada elemento presente na variável que permite a iteração;
- **conjunto_de_itens**: pode ser uma lista, uma string, uma tupla, um dicionário ou um objeto que permita iterações.

Incremento é por padrão em 1

FOR



Exemplo em Python

Nome da variável de controle

Valor final da variável de controle-1 (porque é < que 10)

```
for n in range(1,10)
```

Valor inicial da variável de controle

Valor final da variável de controle-1

```
for n in range(10)
```

Se colocar so um numero, começa em 0

Se não for definido um terceiro elemento, o Incremento é padrão em 1



FOR

- No for, você pode usar uma String já para ser o ***range***, nesse caso a variável de controle (**n** no exemplo abaixo) busca o valor.

```
1 nome = "marcelo"  
2 for n in nome:  
3     print(n)
```

Saída:

```
m  
a  
r  
c  
e  
l  
o
```



FOR encadeado

- Muitas vezes é preciso usar o FOR composto, um FOR dentro do outros, como fazíamos os IF aninhados, lembra?
- ***Vamos fazer um programa que faz uma contagem regressiva de um valor informado pelo usuário:***

```
1  min = int(input())
2  print(f"{min:02}:00")
3  for i in range(min-1, -1, -1):
4      for j in range(59, -1, -1):
5          if i != 0 or j != 0:
6              print(f"{i:02}:{j:02}")
7          else:
8              print("FIM")
```




FOR com o break

- **Configurando onde o loop deve parar: break**
- Por padrão, a estrutura de repetição só termina depois de ler o último elemento da variável iterável. Entretanto, é possível modificar essa condição e interromper o loop no meio do caminho. Para isso, utilizamos a instrução **break** (*lembra do switch-case de java?*), que encerra a execução do loop ao encontrar uma condição específica.
- Devemos utilizar a instrução **break** em conjunto com uma estrutura condicional, como a **if/else** ou até mesmo com outro laço de repetição **for**. Veja como fica a sintaxe da estrutura de repetição quando utilizamos o break:

```
for <item> in <conjunto_de_itens>:  
    <bloco_de_codigo>  
    if <condicao_verdadeira>:  
        <outras_instrucoes>  
        break
```

```
1  vetor = [1,2,3,4,5,6,7,8,9,10]  
2  for i in vetor:  
3      print(i)  
4      if (i==3):  
5          print("Chegou em 3. PARE!")  
6          break
```

```
1  
2  
3  
Chegou em 3. PARE!
```





Repetições - Práticas

- Triângulo com Python. O objetivo é desenhar um triângulo como na imagem abaixo, onde o usuário informa o tamanho desse triângulo:

```
Qual o tamanho do triângulo? 10
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

```
1 linhas = int (input("Qual o tamanho do triângulo? "))
2 for i in range(0, linhas):
3     for j in range(i, linhas):
4         print("*", end=" ")
5     print()
```




Repetições - Práticas

- Tabuada de 5 com While e For em Python

```
1 contador = 1
2 while (contador <=10):
3     print(contador, "x 5 =", (contador*5))
4     contador+=1
```

```
1 for contador in range(1,11):
2     print(contador, "x 5 =", (contador*5))
```



Repetições - Práticas

Sequência de Fibonacci para 10 termos, em JAVA e em Python

Java

```
1 public class Main{
2     public static void main(String[] args) {
3         int temp1, temp2, aux;
4         temp1=0;
5         temp2=1;
6         for (int i = 1; i < 10; i++) {
7             System.out.println("(" + i + "):" + temp2);
8             aux=temp2;
9             temp2=temp1+temp2;
10            temp1=aux;
11        }
12    }
13 }
```

Python

```
1 temp1 = 0
2 temp2 = 1
3 for i in range(1, 11):
4     print("(" + str(i) + "):" + str(temp2))
5     aux=temp2
6     temp2 = temp1 + temp2
7     temp1=aux
```