



Notas de Aula #09: Estrutura De Repetição (FOR)

FOR

A estrutura de repetição **FOR** é utilizada para percorrer uma variável do tipo iterável¹ (listas, dicionários, sets, strings, conjuntos, dicionários, tuplas), onde em cada iteração podemos executar um conjunto de instruções.

Em algumas situações, a estrutura **FOR** se assemelha a estrutura **WHILE**.

Sintaxe :

```
for <variável> in <objeto iterável>:  
    bloco_de_comandos  
else:  
    bloco_comandos_false
```

Exemplo 1:

```
print('Imprimindo todas as letras de uma string')  
texto = 'PYTHON'  
for letra in texto:  
    print('Exibindo: ', letra)
```

Exemplo 2:²

```
print('Imprimindo todos os itens de uma lista')  
linguagens_programacao = ['PYTHON', 'C++', 'RUBY', 'PHP', 'JAVA']  
for linguagem in linguagens_programacao:  
    print('Exibindo: ', linguagem)
```

A instrução **ELSE** tem utilização semelhante quando utilizada na instrução **WHILE**.

Exemplo 3:

```
print('Imprimindo todas as letras de uma string')  
texto = 'PYTHON'  
for letra in texto:  
    print('Exibindo: ', letra)  
else:  
    print('Última letra da string', texto, 'exibida.')
```

¹ Objeto capaz retornar um elemento de cada vez de uma sequência ou coleção, quando solicitado. Por definição, é considerado um iterável todo objeto que pode ser percorrido, um a um, por um laço de repetição (WHILE ou FOR).



Pode-se utilizar a estrutura **FOR** para fazer um contador. Nesse caso devemos utilizar a função **RANGE()**.

A função **RANGE()** retorna uma sequência de números, iniciando em um determinado valor (**start**) , cujo valor default é 0 e finalizando em um determinado (**stop**) e um valor de incremento (**step**), cujo valor padrão é 1.

Sintaxe :

```
range( start, stop, step )
```

Exemplo 4:

```
print('Imprimindo a tabuada de multiplicação')
valor = int(input('Informe o valor: '))
for multiplicador in range(1, 11, 1):
    print(valor, ' x ', multiplicador, ' = ', valor * multiplicador)
else:
    print('Fim da tabuada')
print('Fim do programa')
```

A instrução **BREAK** força a interrupção do laço de repetição da mesma forma que na estrutura **WHILE**.

Exemplo 5:

```
print('Imprimindo a tabuada de multiplicação')
valor = int(input('Informe o valor: '))
for multiplicador in range(1, 11, 1):
    # Interrompe o laço caso tenha sido informado um valor <= 0
    if (valor<=0): break
    print(valor, ' x ', multiplicador, ' = ', valor * multiplicador)
else:
    print('Fim da tabuada')
print('Fim do programa')
```

A instrução **CONTINUE** força o retorno da execução para o início do bloco de comandos do laço de repetição da mesma forma que na estrutura **WHILE**.

Exemplo 6:

```
print('Imprimindo a tabuada de multiplicação')
valor = int(input('Informe o valor: '))
for multiplicador in range(1, 11, 1):
    # Interrompe o laço caso tenha sido informado um valor <= 0
    if (valor<=0): break
    # Retorna ao início do laço (próxima iteração) caso multiplicador seja ímpar
    if (multiplicador % 2 != 0): continue
    print(valor, ' x ', multiplicador, ' = ', valor * multiplicador)
else:
    print('Fim da tabuada')
print('Fim do programa')
```