



- Função Range
- FOR
- FOR com ELSE
- Exemplos com FOR
- WHILE

# Python

## Lógica de Programação

### Principais conceitos da Linguagem Python

## 01 Função Range

- A função range retorna uma sequência de números e é comumente utilizada com o comando FOR, que aprenderemos na sequência. Existe 3 opções (“sabores”) para a função range:
  - **range(n)** na qual gera números de 0 até n-1
    - Por exemplo, range(3) gera 0, 1 e 2
  - **range(a,n)** na qual gera números de a até n-1
    - Por exemplo, range(1,3) gera 1 e 2
  - **range(a,n,s)** na qual gera números de a até n-1, pulando de s em s
    - Por exemplo, range(0,5,2) em teoria geraria 0,1,2,3,4. Entretanto, o último parâmetro (s) que tem o valor 2 significa que ele pula de 2 em 2. Como assim? No 0, ele pula diretamente para 2 e do 2 exatamente para o 4, de modo que os 1 e 3 são desconsiderados.

```
1 print(range(5))
2 print(list(range(5)))
3 print(list(range(0,5)))
4 print(list(range(0,5,2)))
```

**Saída**

```
range(0, 5)
[0, 1, 2, 3, 4]
[0, 1, 2, 3, 4]
[0, 2, 4]
```

- Note que, no exemplo acima, diferente da folha anterior, não usamos for. A primeira linha não exibe o resultado esperado, que é uma lista. Para isto, usamos o método list(), que converte o range() em uma lista, tornando-o apresentável no print.

## 02 Operador FOR

- O operador FOR é usado para iterar sobre uma sequência (lista, dupla, String), entre outros objetos iteráveis. O formato é o seguinte.

```
for var in sequencia:
    corpo do loop (linha 1)
```

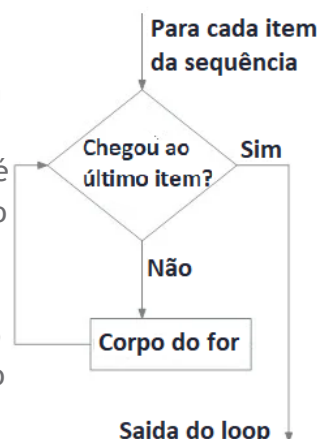
```
1 frutas = ['maça', 'banana', 'abacaxi']
2 for fruta in frutas:
3     print(fruta)
4     print('mais uma linha dentro do FOR')
5     print() #pula linha
6 print('fim do FOR')
```

**Saída**

```
maça
mais uma linha dentro do FOR
banana
mais uma linha dentro do FOR
abacaxi
mais uma linha dentro do FOR
fim do FOR
```

- Acima temos um exemplo comum do uso do FOR. Na linha 1, declaramos uma lista de `frutas`, contendo 3 frutas. Na linha 2, usamos o FOR, onde `fruta` é uma variável temporária, que guarda o valor de cada fruta da lista. Como temos 3 itens na lista `frutas`, o for executará 3 vezes, de modo que as linhas 3,4 e 5 serão executadas 3 vezes.
  - Na primeira iteração, a linha 3 imprimirá maçã, depois imprime a linha 4 e pula a linha (linha 5)
  - Na segunda iteração, a linha 3 imprimirá banana, depois imprime a linha 4 e pula a linha (linha 5)
  - a linha 3 imprimirá abacaxi, depois imprime a linha 4 e pula a linha (linha 5)

- O funcionamento do FOR é apresentado à direita. Para cada item da sequência (`frutas`), antes de executar os comandos, verifica-se se é o último elemento da sequência. Caso não seja, executa-se o corpo do for (linhas 3,4 e 5).
- Quando alcançar o fim da lista, pula o corpo do for e segue o funcionamento do programa (linha 6 e seguintes)
- Verifique o código a seguir



```
1 for n in range(0,5,2):
2     print(n)
3
4 for n in range(0,5):
5     print(n)
6
7 for n in range(5):
8     print(n)
```

## 03 Operador FOR com ELSE

- Em um FOR é possível (mas não muito comum) termos um ELSE, como exemplificado na imagem a seguir. Quando terminamos o FOR das linhas 2 a 5, o ELSE da linha 6 (e consequentemente a linha 7) serão executados.
- Ao término da execução da linha 7, a linha 8 é executada.

```
1 frutas = ['maçã', 'banana', 'abacaxi']
2 for fruta in frutas:
3     print(fruta)
4     print('mais uma linha dentro do FOR')
5     print() #pula linha
6 else:
7     print('else do FOR')
8 print('fim do FOR')
```

## 04 Mais sobre FOR (Exemplos)

- Exemplo #1:** Na linha 1, declaramos uma lista com 4 elementos, enquanto na linha seguinte, declaramos uma variável soma, que vai armazenar a soma de todos os elementos de uma lista.

```
1 numeros = [6, 5, 3, 8]
2 soma = 0
3
4 for num in numeros:
5     soma = soma+num
6
7 print('A soma é', soma)
```

- Na linha 4, iniciamos o for e para cada um dos números da lista, executamos a linha `soma = soma+num` que basicamente:
  - Pega o valor que tem em soma e acresce a um número da lista
  - Ao término do for, a variável soma terá o valor da soma de todas as variáveis.
- Exemplo #2:** No exemplo a seguir, usamos o FOR para imprimir a palavra banana, letra a letra.

```
1 for letra in "banana":
2     print(letra)
```

- Exemplo #3:** Com o `continue` podemos parar a iteração atual do loop e continuar com a próxima. No exemplo a seguir, quando ele encontrar banana (linha 3), vai pular para o próximo elemento.

```
1 frutas = ['maçã', 'banana', 'abacaxi']
2 for fruta in frutas:
3     if fruta == 'banana':
4         continue
5     print('a fruta é',fruta)
6     print('mais uma linha dentro do FOR')
7     print() #pula linha
```

- Exemplo #4:** Com a instrução `break` podemos interromper o loop antes que ele tenha percorrido todos os itens, como no exemplo a seguir. Exibimos maçã e depois banana. Abacaxi não será exibido.

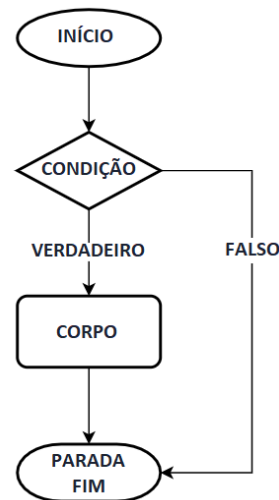
```
1 frutas = ['maçã', 'banana', 'abacaxi']
2 for fruta in frutas:
3     print('a fruta é',fruta)
4     print('mais uma linha dentro do FOR')
5     print() #pula linha
6     if fruta == 'banana':
7         break
8     print('outra linha')
```

## 05 Operador WHILE

- WHILE (Enquanto, em português) permite que você execute um bloco de código (corpo) repetidamente enquanto uma condição for True (verdadeira), conforme exemplo a seguir.

```
1 while condicao
2     corpo - linha de código 1
3     corpo - linha de código 2
```

- A direita temos o funcionamento do WHILE, que verifica a condição no início de cada iteração.
- Executará o corpo enquanto a condição for True. No corpo do loop, você precisa fazer algo para interromper o loop em algum momento.
- Senão, você entra em um LOOP infinito e o aplicativo não fecha mais.



- Exemplo #1:** Enquanto o número for menor que 6, vai imprimir o número (linha 3). Entretanto, na linha 4, incrementamos (somamos 1) o número de modo que, ele vai imprimir até 5 e para (critério de parada - fim).

```
1 numero = 1
2 while numero < 6:
3     print(numero)
4     numero = numero+1
5 print("Terminou o loop")
```

- Exemplo #2:** Este se refere a um dos usos comuns do WHILE, que é de apresentar menus de opções ao usuário. No exemplo a seguir, ele vai apresentar as opções 1) Saque 2) Extrato 3) Sair enquanto o usuário não digitar 3, que é Sair da aplicação.

```
1 print('Bem-vindo do Banco Abella')
2 command = '0'
3 while command != '3':
4     print('1) Saque 2) Extrato 3) Sair')
5     command = input('Digite sua opção ')
6     #funcionamento do programa
```