



# Estruturas de Repetição em Python

**Prof. Fábio Procópio**

**Prof. João Nascimento**



2

# Relembrando...

- Na [aula passada](#), além dos principais operadores usados pelos Python e do comportamento da Tabela Verdade, vimos as **Estruturas Condicionais**
  - Simples, representada por **if**
  - Composta, representada por **if..else**
  - Encadeada, representada por **if..elif..else**
- Nesta aula, falaremos sobre como fazer com que um programa execute um mesmo trecho de código mais de uma vez. Para isso, usaremos as estruturas de repetição.





# Introdução

- Em um programa, existem situações onde se faz necessário executar um trecho de código mais de uma vez;
- Para casos como esses, são usados os comandos de **iteração**, também chamados de **repetição** ou de **loop**
  - **Iteração** é o processo em que um bloco de instrução é executado repetidamente enquanto uma condição é verdadeira
- Python possui duas estruturas nativas de repetição:
  - while
  - for



# Função range()

- Seu principal objetivo é retornar uma lista de números inteiros;
- A forma mais simples de usá-la é **range(numero)**
  - O primeiro número gerado na lista é 0
  - O último elemento da lista é o “numero”-1
- Outra maneira de utilizá-la é **range(inicio, fim)**
  - O primeiro número gerado na lista é “inicio”
  - O último elemento da lista é o “fim”-1
- **range()** pode ser utilizada para criar estruturas de repetição **for**, cujas sintaxes são:

```
for var in range(numero):  
    executa comandos
```

```
for var in range(inicio, fim):  
    executa comandos
```

```
for x in range(55):  
    print("ABC campeão!")
```

```
for x in range(1, 56):  
    if x == 1:  
        print("ABC campeão 1 vez.")  
    else:  
        print("ABC campeão %d vezes." %x)
```



# Estrutura while

- Executa um conjunto de instruções desde que a condição avaliada seja verdadeira;
- Sintaxe básica:

```
while <condição é True>:  
    comandos a serem executados quando a condição é True
```

- Exemplos:

```
cont = 1  
while cont <= 55:  
    print("ABC campeão!")  
    cont += 1 #cont = cont + 1
```

```
cont = 0  
while cont < 55:  
    print("ABC campeão!")  
    cont += 1 #cont = cont + 1
```

```
cont = 1  
while cont <= 5:  
    if cont % 2 == 0:  
        print("%d é par." %cont)  
    cont += 1 #cont = cont + 1
```



6

# Comando break

- A instrução **break** termina a estrutura de repetição que a contém
  - O controle do programa flui para a instrução imediatamente posterior ao corpo do loop
  - Caso o break esteja dentro de um loop aninhado (loop dentro de outro loop), será encerrado o loop mais interno

- Estrutura:

```
for var in sequence:  
    # codes inside for loop  
    if condition:  
        break  
    # codes inside for loop  
# codes outside for loop
```

```
while test expression:  
    # codes inside while loop  
    if condition:  
        break  
    # codes inside while loop  
# codes outside while loop
```

Exemplo

```
for val in "Procópio":  
    if val == "c":  
        break  
    print(val)  
print("Fim")
```

P  
r  
o  
c  
ó  
p  
i  
o  
Fim



7

# Comando continue

- A instrução **continue** é usada para “pular” o código dentro de uma estrutura de repetição apenas na iteração atual
  - O controle do programa não é finalizado, mas continua na próxima iteração
- Estrutura:

```
for var in sequence:  
    # codes inside for loop  
    if condition:  
        continue  
    # codes inside for loop
```

```
# codes outside for loop
```

```
while test expression:  
    # codes inside while loop  
    if condition:  
        continue  
    # codes inside while loop
```

```
# codes outside while loop
```

Exemplo

```
for val in "Procópio":  
    if val == "c":  
        continue  
    print(val)  
  
print("Fim")
```

P  
r  
o  
ó  
p  
i  
o  
  
Fim



8

# Exercícios de Fixação

- 1) Crie um programa no qual o usuário informe um valor N. Armazene esse valor em um variável chamada N. Em seguida, o usuário deve digitar N números. Ao fim, o programa deve imprimir a média aritmética dos N números digitados.
- 2) Para construir esse programa, considere que usuário só informará números inteiros positivos. Crie um programa que receba 5 números digitados e, ao fim, exibir quantos números são pares.
- 3) Construa um programa que receba o sexo e a idade de alunos de uma turma. Considere que o usuário não sabe quantos alunos existem na turma. Ao fim, o programa deve exibir a quantidade de homens acima de 18 anos e a quantidade de mulheres de qualquer idade. Para que o programa seja encerrado, o usuário deve informar uma idade negativa.





# Referências

- 1) EXCRIPT. **Curso de Python - [Aula 34] – Iteração.** Disponível em <https://www.youtube.com/watch?v=XNjaWMqJ9ng&list=PLesCEcYj003QxPQ4vTXkt22-E11aQvoVj&index=36>. Acessado em 29 jan. 2019.
- 2) EXCRIPT. **Curso de Python - [Aula 35] – Laços de Repetição.** Disponível em <https://www.youtube.com/watch?v=-zEl8Jsl0A4&list=PLesCEcYj003QxPQ4vTXkt22-E11aQvoVj&index=37>. Acessado em 29 jan. 2019.
- 3) EXCRIPT. **Curso de Python - [Aula 36] – while else.** Disponível em <https://www.youtube.com/watch?v=ymXN3n9ek2c&list=PLesCEcYj003QxPQ4vTXkt22-E11aQvoVj&index=38>. Acessado em 29 jan. 2019.
- 4) EXCRIPT. **Curso de Python - [Aula 37] – for loop.** Disponível em <https://www.youtube.com/watch?v=xLdj9rxspuA&list=PLesCEcYj003QxPQ4vTXkt22-E11aQvoVj&index=39>. Acessado em 29 jan. 2019.