



**INSTITUTO FEDERAL**

Catarinense

Campus Camboriú

# **AULA X - Estruturas de Repetição**

**Professora: Lidiane Visintin**

*lidiane.visintin@ifc.edu.br*

**Professor: Rafael de Moura Speroni**

*rafael.speroni@ifc.edu.br*

# Objetivos da Aula:



- Compreender o conceito de estruturas de repetição
  - Para (for)

# Como imprimir os números de 1 até 3

## Opção 1

```
1 print(1)
2 print(2)
3 print(3)
```

## Opção 2

```
1 x = 1
2 print(x)
3 x = 2
4 print(x)
5 x = 3
6 print(x)
```

## Opção 3

```
1 x = 1
2 print(x)
3 x = x + 1
4 print(x)
5 x = x + 1
6 print(x)
```

# Repetição sem contador explícito.

---

Outra estrutura de repetição do Python é o **for**, possui comportamento similar ao **while**.

```
for <variável> in <estrutura_a_ser_percorrida>:  
# início para  
    <Comandos>  
    <Comandos>  
# fim para
```



será percorrida toda a estrutura

# Repetição sem contador explícito.

Exemplo:

```
Aula9 > ex1.py > ...
1  for k in 'abacate':
2  |   print(k)
```

PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE

```
PS C:\Users\lidia\Desktop\Algoritmos> & C:/Users/li
dia/AppData/Local/Programs/Python/Python39/python.e
xe c:/Users/lidia/Desktop/Algoritmos/Aula9/ex1.py
a
b
a
c
a
t
e
PS C:\Users\lidia\Desktop\Algoritmos> |
```

# Repetição sem contador explícito.

Exemplo:

```
Aula9 > ex1.py > ...
1  texto = 'abacate 123'
2
3  for k in texto:
4      print(k)
```

PROBLEMS   OUTPUT   TERMINAL   DEBUG CONSOLE

```
a
b
a
c
a
t
e

1
2
3
PS C:\Users\lidia\Desktop\Algoritmos>
```

# Repetição sem contador explícito.

---

- Embora o **for** facilite nosso trabalho, pois não há necessidade de fazer o uso de um contador, esta estrutura não substitui completamente o **while**.
- Dependendo do problema utilizaremos o **for** ou o **while**.
  - utilizamos **for**, normalmente quando queremos processar um a um dos elementos de uma lista;
  - **while** é indicado quando não sabemos quantas vezes vamos repetir ou caso

# Range

---

- Podemos utilizar a função range para gerar listas simples. A função range não retorna uma lista propriamente dita, mas um gerador ou *generator*.
  - Por enquanto basta entender que podemos usá-la, pois lista(conteúdo) será abordado em outra disciplina do curso(Algoritmos e Programação II).



# Range

- Exemplo
  - Se quisermos exibir os números de 0 a 9 na

```
Aula9 > ex1.py > ...  
1  for v in range(10):  
2  print(v)
```

```
PS C:\Users\lidia\Desktop\Algoritmos> & C:/Users/li  
dia/AppData/Local/Programs/Python/Python39/python.e  
xe c:/Users/lidia/Desktop/Algoritmos/Aula9/ex1.py  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
PS C:\Users\lidia\Desktop\Algoritmos>
```

Apenas informamos  
onde parar.  
Normalmente gera  
valores a partir de 0

# Range

- Exemplo
  - Se quisermos exibir os números de 5 a 9 na

```
Aula9 > ex1.py > ...  
1  for v in range(5, 10):  
2      print(v)
```

```
PS C:\Users\lidia\Desktop\Algoritmos> & C:/Users/li  
dia/AppData/Local/Programs/Python/Python39/python.e  
xe c:/Users/lidia/Desktop/Algoritmos/Aula9/ex1.py  
5  
6  
7  
8  
9  
PS C:\Users\lidia\Desktop\Algoritmos>
```

Podemos definir o  
valor de início.

# Range

- Exemplo
  - Se quisermos exibir os números pares de 0 a 9

```
Aula9 > ex1.py > ...  
1 for v in range(0, 10, 2):  
2     print(v)
```

```
PS C:\Users\lidia\Desktop\Algoritmos> & C:/Users/li  
dia/AppData/Local/Programs/Python/Python39/python.e  
xe c:/Users/lidia/Desktop/Algoritmos/Aula9/ex1.py  
0  
2  
4  
6  
8  
PS C:\Users\lidia\Desktop\Algoritmos>
```

Podemos definir o  
valor de início, valor  
de parada e o valor de  
acrécimo.

# Exercícios

---

1. Desenvolva um gerador de tabuada (**use o for**), capaz de gerar a tabuada de qualquer número inteiro entre 1 a 10. O usuário deve informar de qual número ele deseja ver a tabuada. A saída deve ser conforme o exemplo abaixo:

Tabuada de 5:

5 x 1 = 5

5 x 2 = 10

...

5 x 10 = 50

2. Faça um programa que imprima na tela apenas os números ímpares entre 1 e 50 (**use o for**).
3. Faça um programa que calcule o fatorial de um número inteiro fornecido pelo usuário. Ex.:  $5! = 5.4.3.2.1 = 120$  (**use o for**).

# Exercícios

---

4. Selecione e implemente 6 algoritmos da lista de exercícios fornecido na semana passada e os implemente com o uso do for.
  - Compare os códigos elaborados;

# Referências

---

## Referências Básicas

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. Pearson Prentice Hall. 2005  
MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de.. Algoritmos: lógica para desenvolvimento de programação de computadores.. 27. ed.. Érica. 2014

## Referências Complementares

DOWNEY, Allen B. **Pense em Python**. 2ª Ed. Novatec. 2016  
MENEZES, Nilo Ney de Coutinho. **Introdução a programação com Python**. 3ª Ed. Novatec. 2019  
CORMEN, Thomas H et al. **Algoritmos: teoria e prática**. 2. ed. Elsevier, Campus,. 2002

## Referências na Internet

<https://docs.python.org/3/>

<https://www.w3schools.com/python/default.asp>