



IN

INFINITY SCHOOL

V I S U A L A R T C R E A T I V E C E N T E R

LP – Estrutura de Repetição: FOR

- 01 Estrutura de Repetição
- 02 Laço For
- 03 Percorrendo coleções com o For
- 04 Laço For e função range()



LP – Estrutura de Repetição: FOR

O que é um Laço de Repetição?

Um Laço de Repetição, ou *loop*, é uma estrutura em programação que repete uma sequência de instruções até que uma condição específica seja atendida. Nós programadores usamos *loops* para percorrer os valores, adicionar somas de números, repetir funções e muitas outras coisas. Em Python, existem dois tipos *loops*: o **FOR** e o **WHILE**. Nesta aula, entenderemos o funcionamento do laço **FOR**.



LP – Estrutura de Repetição: FOR



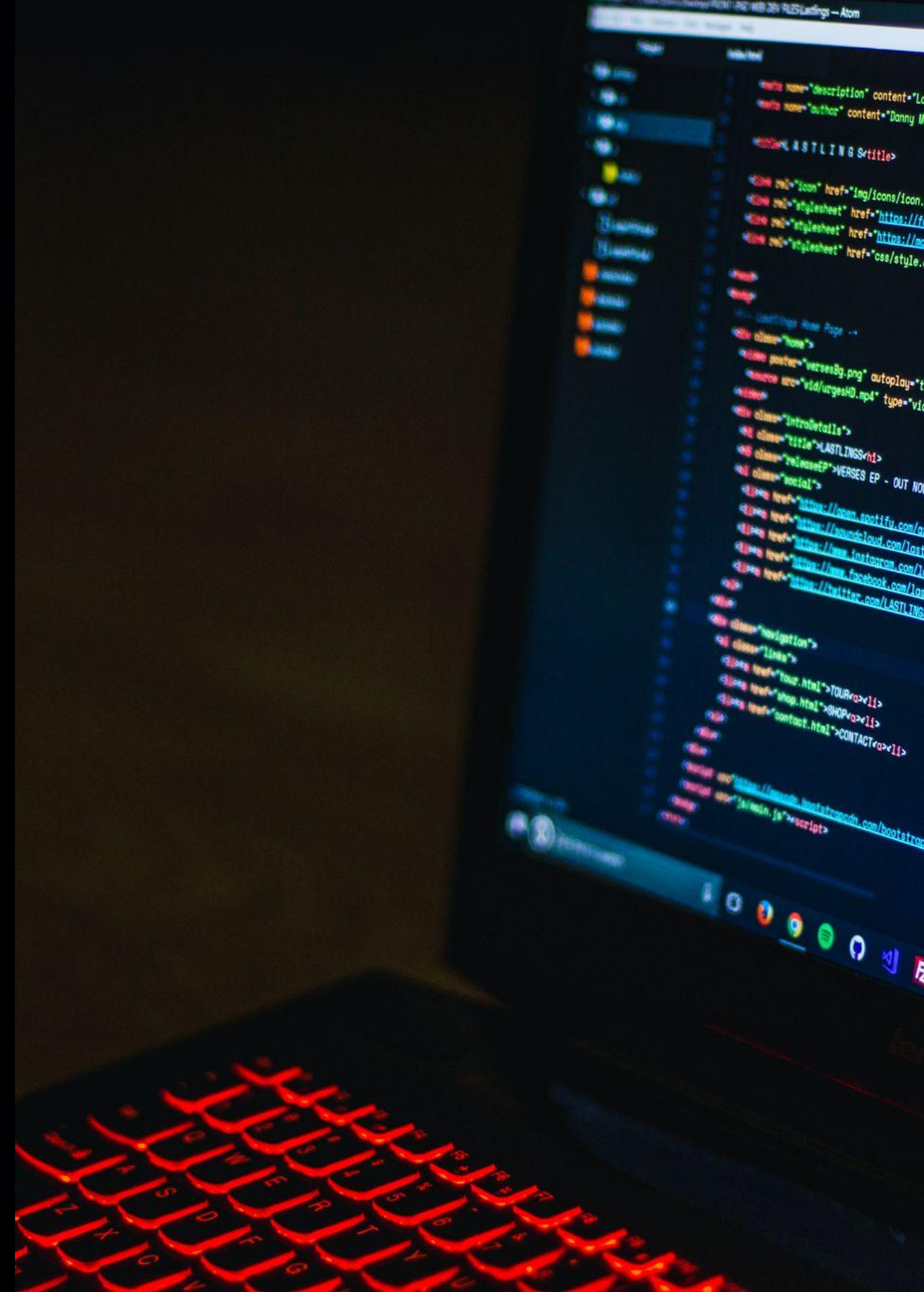
Características do Laço de Repetição FOR em Python

- Realiza iterações sobre coleções.
- Independe dos índices para seu funcionamento.
- A repetição termina quando chega ao fim da coleção.

LP – Estrutura de Repetição: FOR

Como o laço FOR funciona?

O laço de repetição **FOR** no Python percorre uma **coleção** até o seu último item. Uma coleção, por sua vez, pode ser uma **string**, uma **lista**, uma **tupla**, um **dicionário** ou outra coleção. Assim, se temos a palavra '**amor**' o **FOR** acessará cada **letra** dessa palavra sequencialmente, começando pela letra '**a**' e terminando com a letra '**r**': '**a**', '**m**', '**o**', '**r**'.



LP – Estrutura de Repetição: FOR



```
1  # Declaramos uma variável e atribuímos a ela
2  # um tipo de dado string (str)
3  palavra = 'amor'
4
5  # Implementamos o laço for e declaramos a
6  # variável 'letra' para percorrer a coleção
7  # de caracteres da string 'palavra'
8  for letra in palavra:
9      # Exibimos cada letra da palavra 'amor'
10     print(letra)
```

CADA LETRA DA PALAVRA

Note que, após a declaração da variável **palavra**, utilizamos o laço **FOR** para percorrer suas letras. Essa repetição acontecerá até que todas as **letras** da coleção **palavra** tenham sido percorridas. Para isso, declaramos a variável **letra** que será responsável por acessar cada item da coleção **palavra**.

LP – Estrutura de Repetição: FOR

LETRAS DO ALFABETO

Após definirmos uma *string* com as letras do alfabeto, pedimos ao laço FOR que percorra essa **coleção** (*string*) e exiba cada uma das letras sequencialmente, ou seja, na ordem em que elas foram definidas. Em outras palavras, o **FOR** printa na tela cada letra atribuída ao nosso '**alfabeto**'.



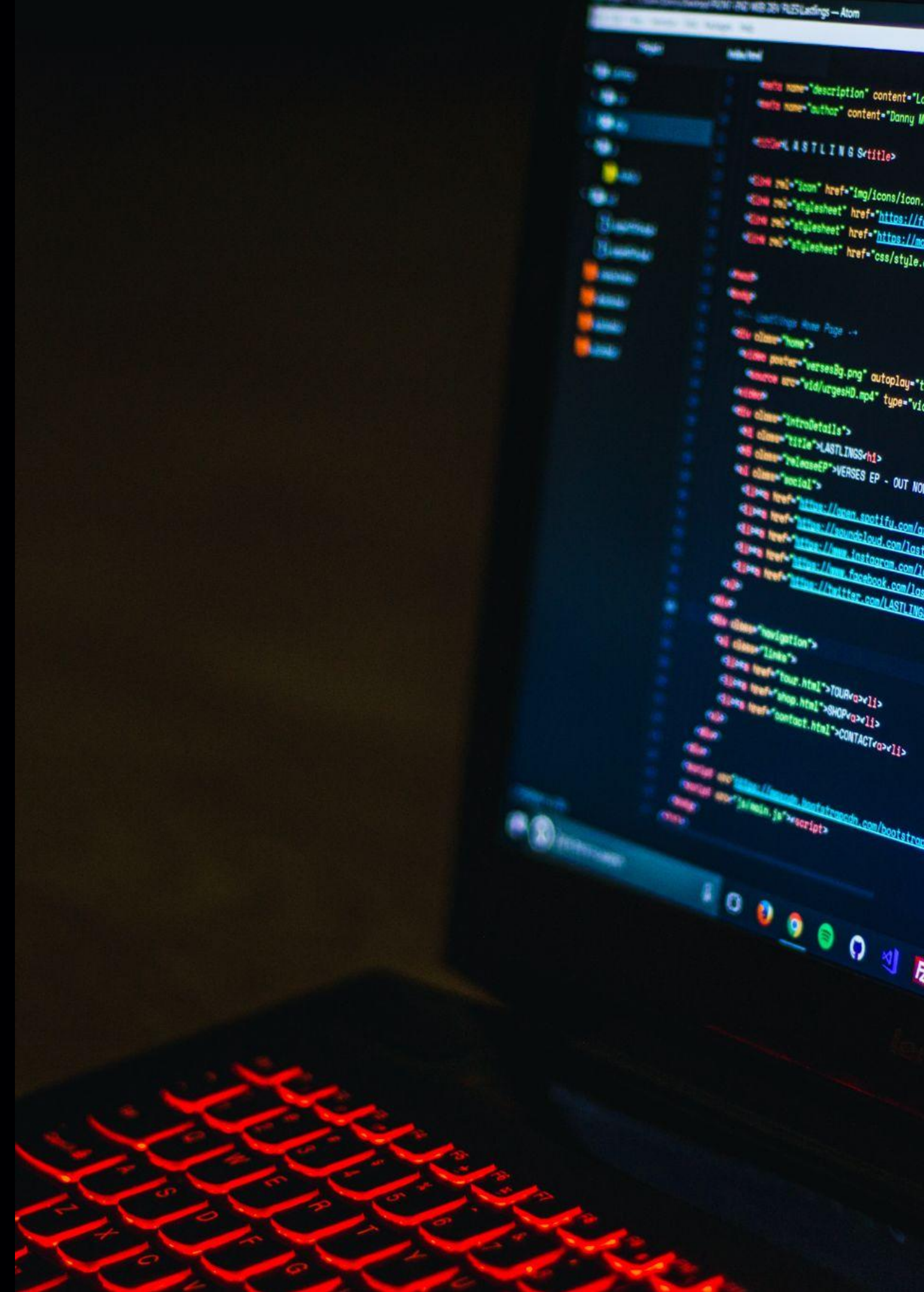
```
1  # Declaramos uma variável e atribuímos a ela
2  # um tipo de dado string (str)
3  alfabeto = 'abcdefghijklmnopqrstuvwxyz'
4
5  # Implementamos o laço for e declaramos a
6  # variável 'letra' para percorrer a coleção
7  # de caracteres da string 'alfabeto'
8  for letra in alfabeto:
9      # Exibimos cada letra definida na variável
10     # alfabeto
11     print(letra)
```

LP – Estrutura de Repetição: FOR

Laço FOR e a função range()

Anteriormente, você entendeu que o FOR percorre uma coleção, certo? Nesse caso, imagine que quiséssemos percorrer uma coleção de número, de 0 até 9. Como poderíamos fazer isso?

Em Python, é possível criar uma variável e atribuir a ela a com os números: `numeros = '0123456789'`. No entanto, é possível utilizar o comando `range()` para fazer isso, o que facilitará a criação de coleções muito maiores do que os número de 0 a 9.



LP – Estrutura de Repetição: FOR



```
1 # Criamos o laço FOR e a variável que
2 # percorrerá a coleção criada pelo range()
3 for numero in range(0, 9):
4     # Exibimos cada um dos itens da coleção
5     print(numero)
```

Abra o seu VS Code e teste o código acima. O que aconteceu? Quantos números foram exibidos? Faltou alguns?

COLEÇÕES COM A FUNÇÃO RANGE()

Note que pedimos à função **range()** para que ela criasse a coleção. Para isso, precisamos informá-la duas coisas: onde a coleção iniciará (o início) e onde ela terminará (o fim): **range(0, 9)**. Após, o FOR fará todo o resto, percorrendo cada um dos itens criados pela função **range()**.

LP – Estrutura de Repetição: FOR

1. Faça um programa para imprimir os números de 1 a 10. Utilize a função `range()` para criar a coleção de números.
2. Faça um programa para imprimir os números pares de 1 a 20. Utilize a função `range()` para criar a coleção de números.
3. Faça um programa para imprimir os números ímpares de 1 a 19. Utilize a função `range()` para criar a coleção de números.
4. Faça um programa para calcular a soma dos números de 1 a 100. Utilize a função `range()` para criar a coleção de números.
5. Faça um programa para calcular a média de uma lista de números.
6. Faça um programa para verificar se um número é primo. Utilize a condicional IF dentro do laço FOR.
7. Faça um programa para Imprimir os caracteres de uma string separadamente.
8. Faça um programa para contar quantas vogais existem em uma palavra. Utilize a condicional IF dentro do laço FOR.
9. Faça um programa para contar quantas consoantes existem em uma palavra. Utilize a condicional IF dentro do laço FOR.
10. Faça um programa para verificar se uma palavra é um palíndromo. Exemplo: 'amor' = 'roma' (NÃO É) / 'arara' = 'arara' (É PALÍNDROMO).



**I AM A PROGRAMMER,
WHAT IS YOUR SUPER POWER**