

LÓGICA E LINGUAGEM DE PROGRAMAÇÃO

Curso Técnico Subsequente em Informática
Lucas Sampaio Leite



Estruturas de repetição

- As linguagens de programação oferecem mecanismos que automatizam repetições, conhecidos como estruturas de repetição ou, em inglês, loops.
- No Python, contamos com duas principais formas de criar repetições:
 - `while` → ideal para quando não sabemos quantas vezes o código deve se repetir.
 - `for` → perfeito para repetições com quantidade conhecida ou ao percorrer coleções (como listas).

Estruturas de repetição (for)

- O for é uma estrutura de repetição, assim como o while, mas costuma ser mais usada quando se sabe exatamente quantas vezes o bloco de código deve ser executado.
- Sintaxe do for com a função range():

```
for i in range(<n>):  
    comandos
```
- A função range() gera uma sequência de números inteiros e é muito usada com o for para repetir algo várias vezes ou iterar com base em contadores.
- Ao utilizar range(n), a variável de controle (geralmente chamada de i) começa com o valor 0 e é incrementada de 1 em 1 até atingir n - 1.

Estruturas de repetição (for)

- Sintaxe do for com a função range():

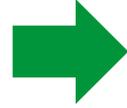
```
for i in range(<n>):  
    comandos
```

- Para usar o for, é sempre necessário indicar uma variável iteradora, que assumirá um valor diferente a cada repetição do laço.
- Também é necessário indicar os limites de iteração, ou seja, de onde a contagem começa, onde termina e, se desejado, o valor do passo (incremento ou decremento).

Estruturas de repetição (for)

- Exemplo:

```
for i in range(5):  
    print(i)
```



```
0  
1  
2  
3  
4
```

Estruturas de repetição (for)

- Exemplo:

```
for i in range(5):  
    print(i)
```



```
0  
1  
2  
3  
4
```

Neste caso, o laço for imprime na tela o valor atual de *i* a cada iteração. A variável *i* começa em 0 e é incrementada de um em um até atingir o valor 4.

Estruturas de repetição (for)

- É possível definir um valor inicial diferente de 0 para a contagem.
- Para isso, basta utilizar a função `range()` com dois parâmetros: o valor inicial e o valor final.
- Sintaxe do `for` com a função `range()`:

```
for i in range(<início>, <fim>):  
    comandos
```

Estruturas de repetição (for)

- É possível definir um valor inicial diferente de 0 para a contagem.
- Para isso, basta utilizar a função `range()` com dois parâmetros: o valor inicial e o valor final.
- Sintaxe do for com a função `range()`:

```
for i in range(<início>, <fim>):  
    comandos
```

- Exemplo:

```
for i in range(2, 6):  
    print(i)
```



```
2  
3  
4  
5
```

Estruturas de repetição (for)

- É possível definir um valor inicial diferente de 0 para a contagem.
- Para isso, basta utilizar a função range() com dois parâmetros: o valor inicial e o valor final.
- Sintaxe do for com a função range():

```
for i in range(<início>, <fim>):  
    comandos
```

- Exemplo:

```
for i in range(2, 6):  
    print(i)
```



```
2  
3  
4  
5
```

Neste caso, o laço for imprime na tela o valor de i a cada iteração. A variável i inicia em 2 e é incrementada de um em um até atingir o valor 5.

Estruturas de repetição (for)

- A função `range()` também pode ser usada para controlar a execução do laço `for`, utilizando três parâmetros: `range(m, n, p)`.
- Essa forma gera uma sequência de números inteiros que começa em `m`, vai até `n - 1` e é incrementada de `p` em `p`.

```
m = 1
n = 100
p = 2
for i in range(m, n, p):
    print(i)
```



```
for i in range(1, 100, 2):
    print(i)
```

O que será impresso?

Estruturas de repetição (for)

- Também é possível utilizar um passo negativo na função range() para fazer a contagem regressiva, ou seja, contar de trás para frente.

- Exemplo:

```
for i in range(5, 0, -1):  
    print(i)
```

O que será impresso?

Estruturas de repetição (for)

- Também é possível utilizar um passo negativo na função range() para fazer a contagem regressiva, ou seja, contar de trás para frente.

- Exemplo:

```
for i in range(5, 0, -1):  
    print(i)
```



```
5  
4  
3  
2  
1
```

Estruturas de repetição (for)

```
for x in range (0, 5, 1) :  
    print(x)
```

**início (opcional) –
quando omitido,
início = 0**

fim (obrigatório)

**incremento (opcional) –
quando omitido,
incremento = 1**

Estruturas de repetição (for)

- A variável iteradora também pode assumir diretamente os valores de uma string ou dos elementos de uma lista:

- Exemplo:

```
string = "Programação é 10"  
  
for char in string:  
    print(char)
```



P
r
o
g
r
a
m
a
ç
ã
o

é

1
0

Estruturas de repetição (for)

- O laço for pode ser controlado pelas instruções break e continue.
 - A instrução break interrompe o laço completamente, terminando sua execução.
 - A instrução continue faz com que o laço pule para a próxima iteração, sem finalizar sua execução.

```
for i in range(5):  
    if i == 0:  
        print(f"i = 0, então {i}")  
    elif i == 1:  
        continue  
    elif 1 < i < 3:  
        print(f"A variável i é {i}")  
    elif i == 3:  
        print(f"i = 3, então break")  
        break  
    else:  
        print(f"i > 3, então {i}")
```

O que será impresso?

Estruturas de repetição (for)

- O laço for pode ser controlado pelas instruções break e continue.
 - A instrução break interrompe o laço completamente, terminando sua execução.
 - A instrução continue faz com que o laço pule para a próxima iteração, sem finalizar sua execução.

```
for i in range(5):  
    if i == 0:  
        print(f"i = 0, então {i}")  
    elif i == 1:  
        continue  
    elif 1 < i < 3:  
        print(f"A variável i é {i}")  
    elif i == 3:  
        print(f"i = 3, então break")  
        break  
    else:  
        print(f"i > 3, então {i}")
```



```
i = 0, então 0  
A variável i é 2  
i = 3, então break
```

Atenção!!!

- **A Lista 02 está disponível na plataforma Beecrowd:**
 - Escolha 10 problemas dentro os 11 disponíveis.
 - A atividade deve ser realizada individualmente.
 - Deve ser respondida exclusivamente com base no conteúdo abordado em sala de aula.
 - A correção será feita automaticamente pela plataforma, com detecção de plágio.
 - Prazo final: até 18 de maio de 2025, às 23h59.
- **Avaliação 01 - 19 de maio de 2025 (formato papel)**

Exercícios (utilizando for)

1. Escreva um programa que imprime todos os numeros de 0 até 50, incluindo-os.
2. Modifique o programa anterior de forma que este imprima apenas os números que são pares.
3. Escreva um programa para contar a quantidade de números pares entre dois números quaisquer fornecidos pelo usuário?
4. Escreva um programa para calcular o fatorial de um número fornecido pelo usuário.

Exercícios (utilizando for)

5. Faça um programa que peça dois números, base e expoente, calcule e mostre o primeiro número elevado ao segundo número. Não utilize a função de potência da linguagem ou o operador de exponenciação.
6. Desenvolva um gerador de tabuada, capaz de gerar a tabuada de qualquer número inteiro entre 1 a 10. O usuário deve informar de qual numero 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
```

Exercícios (utilizando for ou while)

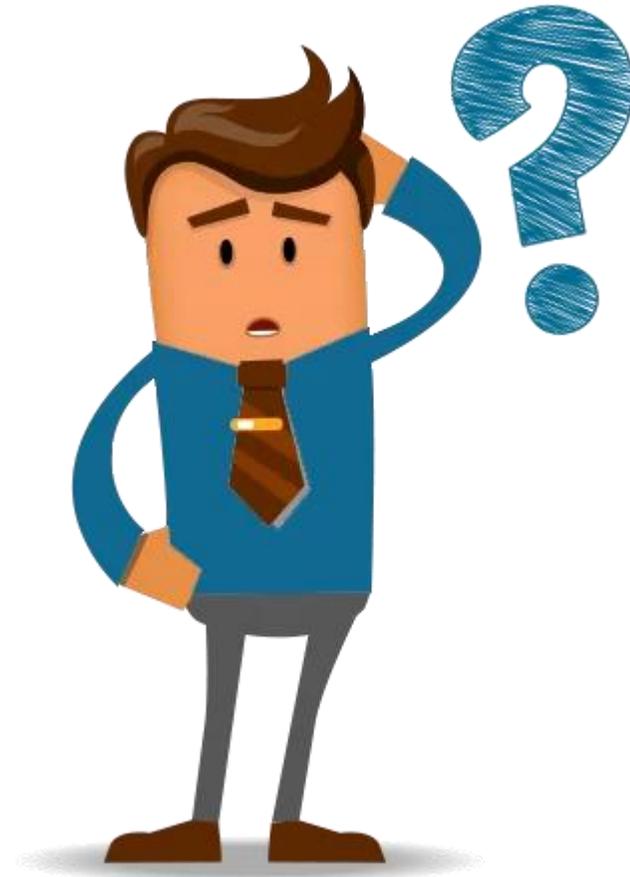
7. Escreva um programa que leia um número inteiro e calcule a soma de todos os divisores desse número, com exceção dele próprio. Ex: a soma dos divisores do número 66 é $1 + 2 + 3 + 6 + 11 + 22 + 33 = 78$
8. Em Matemática, o número harmônico designado por $H(n)$ define-se como sendo a soma da série harmônica: $H(n) = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$
Faça um programa que leia um valor n inteiro e positivo e apresente o valor de $H(n)$.

Exercícios (utilizando for ou while)

9. A série de Fibonacci é formada pela seqüência 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... Faça um programa capaz de gerar a série até o n-ésimo termo.
10. A série de Fibonacci é formada pela seqüência 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ... Faça um programa que gere a série até que o valor seja maior que 500.
11. Faça um programa que calcule o fatorial de um número inteiro fornecido pelo usuário. Ex.: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$. A saída deve ser conforme o exemplo abaixo:

```
Fatorial de: 5  
5! = 5 . 4 . 3 . 2 . 1 = 120
```

Dúvidas



LÓGICA E LINGUAGEM DE PROGRAMAÇÃO

Curso Técnico Subsequente em Informática
Lucas Sampaio Leite

