



Aula 06:

Estruturas de Repetição (2)

Programação de Computadores I

Túlio Toffolo

www.toffolo.com.br

Departamento de Computação
Universidade Federal de Ouro Preto

Agenda

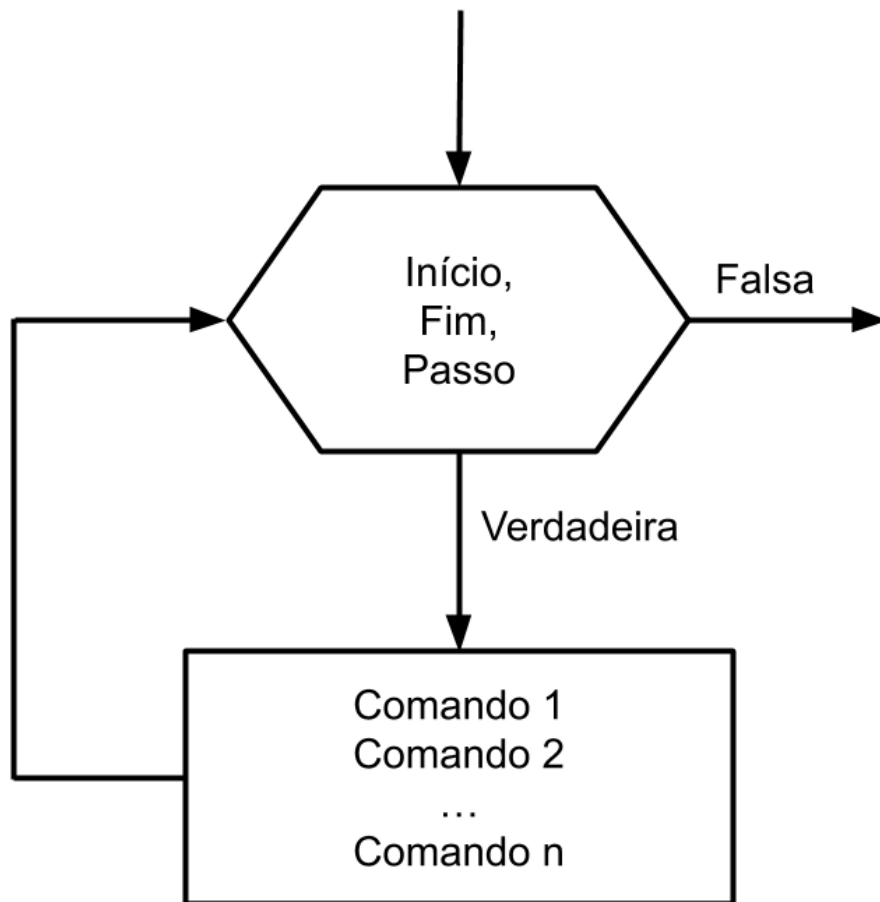
1. O comando for
2. A função range
3. Qual comando de repetição utilizar?
4. Exemplos e exercícios

0 comando for

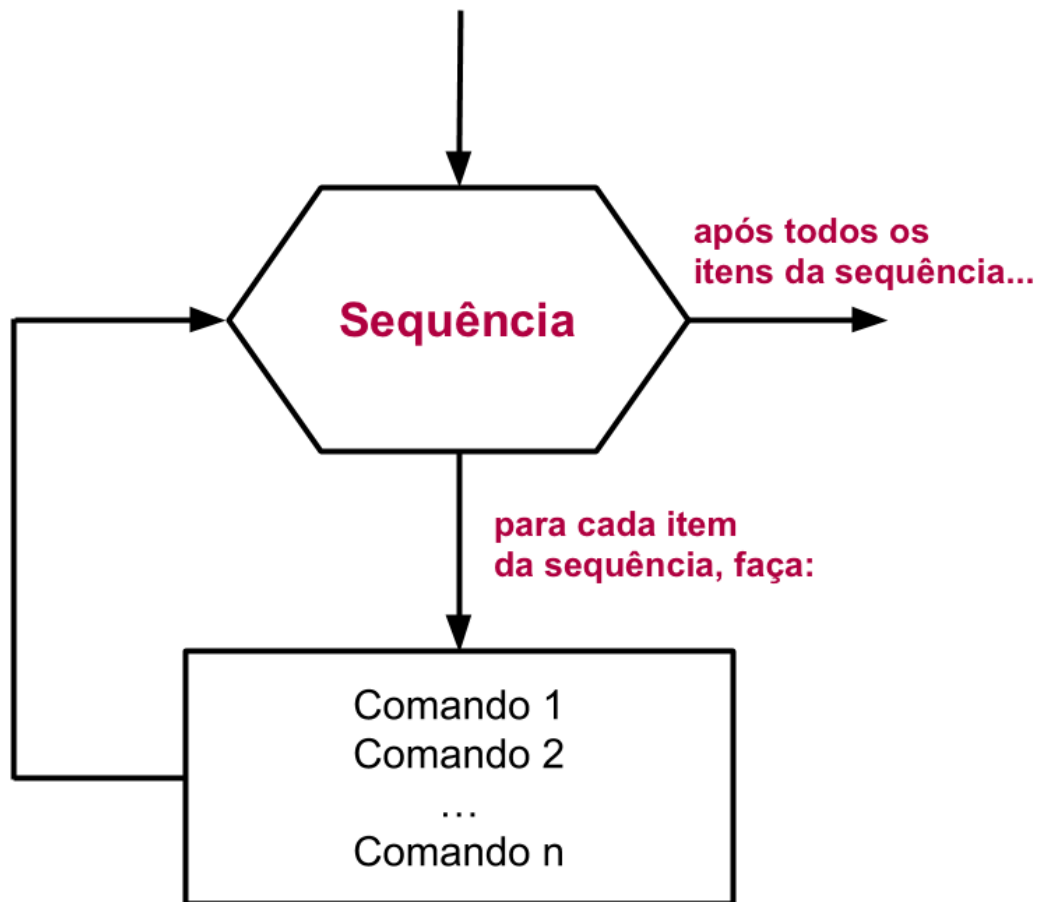
Laços de repetição

- Em um laço **controlado logicamente**, os comandos (corpo do laço) são repetidos enquanto uma **expressão lógica for verdadeira**.
 - `while`
 - **Em um laço controlado por contador ou por um iterador, os comandos (corpo do laço) são repetidos um determinado número de vezes.**
 - `for`
-
- Denomina-se **iteração** a repetição de um conjunto de comandos: cada execução do corpo do laço, juntamente com a condição de terminação do laço, é uma iteração.

O comando **for**



O comando **for**



O comando **for**

O `for` possui a seguinte estrutura:

```
1  ...  
2  for <Variável> in <Sequência>:  
3      <Comando 1>  
4      <Comando 2>  
5      ...  
6      <Comando n>  
7  ...
```

Observe a indentação (espaços no início da linha de código) e que:

- Devemos definir uma `<Sequência>`.
- No código devemos usar uma `<Variável>` para **iterar** os itens que estão em `<Sequência>`.

A função range

A função **range**

A função `range` permite definir sequências de forma simples e rápida. A função resulta em uma sequência de inteiros e pode ser usada de diferentes formas:

- `range(fim)`: sequência de `0` até `fim-1`.
- `range(ini, fim)`: sequência de `ini` até `fim-1`.
- `range(ini, fim, passo)`: sequência de `ini` até `fim-1` somando o valor de `passo` a cada iteração.

Se não especificarmos os valores de `ini` e `passo`, assume-se que:

- `ini=0`
- `passo=1`

Exemplos de uso:

```
1 # i assumirá os valores [0,1,2,3,4,5,6,7,8,9]
2 for i in range(10):
3     print(i)
```

Resulta em :

```
1 0
2 1
3 2
4 3
5 4
6 5
7 6
8 7
9 8
10 9
```

Exemplos de uso:

```
1 # i assumirá os valores [0,1,2,3,4,5,6,7,8,9]
2 for i in range(0,10):
3     print(i)
```

Resulta em :

```
1 0
2 1
3 2
4 3
5 4
6 5
7 6
8 7
9 8
10 9
```

Exemplos de uso:

```
1 # i assumirá os valores [10,11,12,13,14,15,16,17,18,19]
2 for i in range(10, 20):
3     print(i)
```

Resulta em :

```
1 10
2 11
3 12
4 13
5 14
6 15
7 16
8 17
9 18
10 19
```

Exemplos de uso:

```
1 # i assumirá os valores [1,4,7,10,13,16,19]
2 for i in range(1, 20, 3):
3     print(i)
```

Resulta em :

```
1 1
2 4
3 7
4 10
5 13
6 16
7 19
```

Note que começamos com o valor 1. Na iteração seguinte somamos 3 ao valor, ou seja, $1+3 = 4$. Repetimos até valor atingir (ou ultrapassar) o fim, que neste caso é igual a 20.

Exemplos de uso:

```
1 # i assumirá os valores [10,9,8,7,6,5,4,3,2,1]
2 for i in range(10, 0, -1):
3     print(i)
```

Resulta em :

```
1 10
2 9
3 8
4 7
5 6
6 5
7 4
8 3
9 2
10 1
```

Qual comando de repetição utilizar?

Qual comando de repetição utilizar?

while

- Ideal para laços **controlados logicamente**, em que os comandos (corpo do laço) são repetidos enquanto uma **expressão lógica** for verdadeira.

for

- Ideal para laços **controlados por contadores** ou por **iteradores**, em que os comandos (corpo do laço) são repetidos um **determinado número de vezes**.

Comandos **while** e **for**

*Resolvemos o problema a seguir usando **while**:*

Faça um programa em *Python* para calcular a média harmônica de n valores digitados pelo usuário, dada pela equação a seguir:

$$H = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}, \text{ ou seja: } H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$$

```
1 Cálculo da Média Harmônica.  
2 Quanto números você quer digitar? 2  
3 Número 1: 2  
4 Número 2: 4  
5  
6 Média Harmônica: 2.67
```

Comandos **while** e **for**

Usando o **for**, podemos simplificar o código:

```
1 # Lendo dados de entrada
2 print("Cálculo da Média Harmônica.")
3 n = int(input("Quanto números você quer digitar? "))
4
5 # Calculando o somatório
6 soma = 0
7 i = 0
8 while i < n:
9     x = float(input(f"Número {i+1}: "))
10    soma = soma + 1 / x
11    i += 1
12
13 # Calculando e imprimindo a média harmônica
14 media = n / soma
15 print(f"\nMédia Harmônica: {media:.2f}")
```

Comandos **while** e **for**

Usando o **for**, podemos simplificar o código:

```
1 # Lendo dados de entrada
2 print("Cálculo da Média Harmônica.")
3 n = int(input("Quanto números você quer digitar? "))
4
5 # Calculando o somatório
6 soma = 0
7 for i in range(n):
8     x = float(input(f"Número {i}: "))
9     soma = soma + 1 / x
10
11 # Calculando e imprimindo a média harmônica
12 media = n / soma
13 print(f"\nMédia Harmônica: {media:.2f}")
```

Comandos **while** e **for**

Observe, agora, o exemplo a seguir:

Crie um programa que calcula o IMC do usuário até ele informar que não quer mais calculá-lo. Exemplos de execução:

```
1 Deseja calcular o IMC? (S/N) S
2 Qual sua altura (em metros)? 1.9
3 Qual seu peso (em kg)? 90
4
5 Seu IMC é 24.93
6
7 Deseja calcular o IMC? (S/N) S
8 Qual sua altura (em metros)? 1.7
9 Qual seu peso (em kg)? 65
10
11 Seu IMC é 22.49
12
13 Deseja calcular o IMC? (S/N) N
```

Comandos **while** e **for**

```
1 resp = input('Deseja calcular o IMC? (S/N) ')
2 while resp == 'S':
3     # Capturando dados para cálculo do IMC
4     altura = float(input('Qual sua altura (em metros)? '))
5     peso = float(input('Qual seu peso (em kg)? '))
6     imc = peso / altura ** 2
7
8     # Imprimindo resultado
9     print(f'\nSeu IMC é {imc:.2f}\n')
10
11     # Verificando com usuário se execução deve continuar...
12     resp = input('Deseja calcular o IMC? (S/N) ')
```

Comandos **while** e **for**

```
1 resp = input('Deseja calcular o IMC? (S/N) ')
2 while resp == 'S':
3     # Capturando dados para cálculo do IMC
4     altura = float(input('Qual sua altura (em metros)? '))
5     peso = float(input('Qual seu peso (em kg)? '))
6     imc = peso / altura ** 2
7
8     # Imprimindo resultado
9     print(f'\nSeu IMC é {imc:.2f}\n')
10
11    # Verificando com usuário se execução deve continuar...
12    resp = input('Deseja calcular o IMC? (S/N) ')
```

Neste caso, poderíamos usar **for**?

- **Não**, pois não sabemos quantas vezes o laço será repetido. Neste caso, não há como definir uma <Sequência>.

Exemplos e exercícios

Exemplo 1

Um número natural é primo se ele possui apenas dois divisores distintos. Assim, um número maior que 1 é primo se for divisível apenas por si próprio e por 1. Crie uma programa que imprime se um número é primo ou não.

Exemplos de execução: (texto digitado pelo usuário em destaque)

```
1 Digite um número: 29881
2 29881 é um número primo!
```

```
1 Digite um número: 16
2 16 NÃO é um número primo!
```

```
1 Digite um número: 3
2 3 é um número primo!
```


Exemplo 1

Solução que segue o enunciado à risca:

```
1 n = int(input('Digite um número natural: '))
2 divisores = 0
3
4 # Verificando o número de divisores de n entre {1..n}
5 for i in range(1, n + 1):
6     if n % i == 0:
7         divisores += 1
8
9 if divisores == 2:
10     print(f'{n} é um número primo')
11 else:
12     print(f'{n} NÃO é um número primo')
```

Exemplo 1

Solução alternativa (um pouco mais elegante):

```
1 n = int(input('Digite um número natural: '))
2 primo = True
3
4 # Verificando se n tem algum divisor entre {2..n-1}
5 for i in range(2, n):
6     if n % i == 0:
7         primo = False
8
9 if primo:
10     print(f'{n} é um número primo')
11 else:
12     print(f'{n} NÃO é um número primo')
```

- Aprenderemos em breve formas de melhorar a implementação acima, incluindo o comando `break`.

Exemplo 2

Crie um programa que lê n salários e, em seguida, informa o maior salário, o salário médio, e o menor salário.

Exemplo de execução: (texto digitado pelo usuário em destaque)

```
1  Quantos funcionários serão considerados? 4
2  Digite o salário do funcionário 1: 3000
3  Digite o salário do funcionário 2: 5200
4  Digite o salário do funcionário 3: 6500
5  Digite o salário do funcionário 4: 2300
6
7  Menor salário: 2300.00
8  Salário médio: 4250.00
9  Maior salário: 6500.00
```

Exemplo 2

```
1 import math
2
3 n = int(input('Quantos funcionários serão considerados? '))
4 menor = math.inf # representa 'infinito'
5 maior = 0
6 soma = 0
7
8 for i in range(n):
9     x = float(input(f'Digite o salário do funcionário {i+1}: '))
10    if x < menor:
11        menor = x
12    if x > maior:
13        maior = x
14    soma += x
15
16 print()
17 print(f'Menor salário: {menor:.2f}')
18 print(f'Salário medio: {soma/n:.2f}')
19 print(f'Maior salário: {maior:.2f}')
```

Exemplo 3

Antes de fazer o **Exemplo 3**, vamos revisar o operador `//` e aprender um "truque" para utilizar com a função `print`:

- `//` é o operador de **divisão inteira** em Python 3:
 - O resultado de `1 // 2` é igual a `0`
 - O resultado de `3 // 2` é igual a `1`
 - O resultado de `4 // 2` é igual a `2`
- A função `print` tem o argumento nomeado `end`, que indica o que é impresso ao final da *string*. O valor padrão é `\n`.
 - Exemplo: `print('Olá mundo!', end='')`
 - Exemplo: `print('Olá mundo!', end='\t')`

Faça um programa que desenha um trapézio retângulo de bases x e y usando asteriscos. O usuário deve informar os valores de x e y , que devem ser **pares** e tais que $x < y$. Certifique-se de que o usuário digitou valores válidos.

```
1 Digite o valor de x: 3
2 Digite o valor de y: 6
3 Os valores devem ser pares e tais que x < y.
4
5 Digite o valor de x: 10
6 Digite o valor de y: 8
7 Os valores devem ser pares e tais que x < y.
8
9 Digite o valor de x: 4
10 Digite o valor de y: 14
11
12      ****
13     *
14    *
15   *
16  *
17 *
18 *****
19 *****
20 *****
21 *****
22 *****
23 *****
24 *****
25 *****
26 *****
27 *****
28 *****
29 *****
30 *****
31 *****
32 *****
33 *****
34 *****
35 *****
36 *****
37 *****
38 *****
39 *****
40 *****
41 *****
42 *****
43 *****
44 *****
45 *****
46 *****
47 *****
48 *****
49 *****
50 *****
51 *****
52 *****
53 *****
54 *****
55 *****
56 *****
57 *****
58 *****
59 *****
60 *****
61 *****
62 *****
63 *****
64 *****
65 *****
66 *****
67 *****
68 *****
69 *****
70 *****
71 *****
72 *****
73 *****
74 *****
75 *****
76 *****
77 *****
78 *****
79 *****
80 *****
81 *****
82 *****
83 *****
84 *****
85 *****
86 *****
87 *****
88 *****
89 *****
90 *****
91 *****
92 *****
93 *****
94 *****
95 *****
96 *****
97 *****
98 *****
99 *****
100 *****
```

```

1  # Lendo valores de x e y
2  x = int(input("Digite o valor de x: "))
3  y = int(input("Digite o valor de y: "))
4
5  # Relendo valores de x e y até que eles sejam válidos
6  while x % 2 == 1 or y % 2 == 1 or x >= y:
7      print('Os valores devem ser pares e tais que x < y.\n')
8      x = int(input("Digite o valor de x: "))
9      y = int(input("Digite o valor de y: "))
10
11 print() # (linha em branco para saída ficar idêntica)
12
13 for i in range(x, y+1, 2):
14     espacos = (y - i) // 2
15
16     # Imprimindo os espaços em branco
17     for j in range(espacos):
18         print(' ', end='')
19
20     # Imprimindo os asteriscos
21     for j in range(i):
22         print('*', end='')
23
24     print() # (quebra de linha)

```

Perguntas?
