



Aula 07: **Laços aninhados**

Programação de Computadores I

Túlio Toffolo

www.toffolo.com.br

Departamento de Computação
Universidade Federal de Ouro Preto

Agenda

1. Laços aninhados
2. Comandos break e continue
3. Exemplos e exercícios

Laços aninhados

Laços aninhados

Crie um programa que imprime o seguinte:

```
1 1
2 22
3 333
4 4444
5 55555
6 666666
7 7777777
8 88888888
9 999999999
```

Utilizaremos laços aninhados:

- O laço externo será variar o valor de $n = 1, 2, \dots, 9$.
- O laço interno imprimirá n vezes o valor de n .

Laços aninhados

Resolvemos facilmente com **laços aninhados**:

```
1 for n in range(1, 10):  
2     for i in range(n):  
3         print(n, end=' ')  
4     print()
```

- **Repetição 1**: varia o valor de n .
- **Repetição 2**: imprime n vezes o valor de n .

Note o uso de `end=' '` na função `print`. Isto faz com que um `\n` não seja impresso no final, evitando a quebra de linha.

Laços aninhados

O código a seguir é muito útil para entender laços aninhados:

```
1 for i in range(3):  
2     print(f'{i+1}a iteração do laço externo')  
3     for j in range(3):  
4         print(f'    {j+1}a iteração do laço interno')
```

Algumas perguntas...

- Quantas vezes o `print` da **linha 2** será executado?
- E o `print` da **linha 4**? Quantas vezes será executado?
- Para entender melhor, vamos executar o código passo-a-passo!

```
1 for i in range(3):
2     print(f'{i+1}a iteração do laço externo')
3     for j in range(3):
4         print(f'    {j+1}a iteração do laço interno')
```

Assim, o código imprimirá:

```
1 1a iteração do laço externo
2   1a iteração do laço interno
3   2a iteração do laço interno
4   3a iteração do laço interno
5 2a iteração do laço externo
6   1a iteração do laço interno
7   2a iteração do laço interno
8   3a iteração do laço interno
9 3a iteração do laço externo
10  1a iteração do laço interno
11  2a iteração do laço interno
12  3a iteração do laço interno
```

Exemplo de uso (1)

Implemente um programa que exibe na saída a tabuada de multiplicação de 1 a 9, conforme o exemplo a seguir:

```
1 1 x 1 = 1
2 1 x 2 = 2
3 ...
4 9 x 8 = 72
5 9 x 9 = 81
```

Usaremos laços aninhados, é claro:

```
1 for i in range(1, 10):
2     for j in range(1, 10):
3         x = i * j
4         print(f"{i} x {j} = {x}")
```


Exemplo de uso (1)

Poderíamos resolver o exercício anterior usando `while`.

```
1 i = 1
2 while i < 10:
3     j = 1
4     while j < 10:
5         x = i * j
6         print(f"{i} x {j} = {x}")
7         j += 1
8     i += 1
```

Qual você prefere? (note o código com `for` a seguir)

```
1 for i in range(1, 10):
2     for j in range(1, 10):
3         x = i * j
4         print(f"{i} x {j} = {x}")
```

Exemplo de uso (2)

Faça um programa que imprime a tabuada de x até y.
Valores de x e y devem ser digitados pelo usuário.

```
1 Digite o valor de x: 5
2 Digite o valor de y: 14
3
4      5    6    7    8    9   10   11   12   13   14
5 -----
6      5    25   30   35   40   45   50   55   60   65   70
7      6    30   36   42   48   54   60   66   72   78   84
8      7    35   42   49   56   63   70   77   84   91   98
9      8    40   48   56   64   72   80   88   96  104  112
10     9    45   54   63   72   81   90   99  108  117  126
11    10    50   60   70   80   90  100  110  120  130  140
12    11    55   66   77   88   99  110  121  132  143  154
13    12    60   72   84   96  108  120  132  144  156  168
14    13    65   78   91  104  117  130  143  156  169  182
15    14    70   84   98  112  126  140  154  168  182  196
```

Exemplo de uso (2)

Passo 1:

- Ler os dados de entrada

Passo 2:

- Imprimir o cabeçalho (é a parte mais difícil)

Passo 3:

- Imprimir a tabela
- Atenção: não esquecer do cabeçalho de cada linha

Exemplo de uso (3)

Crie um programa que calcula o valor da expressão a seguir.
Considere que os valores de n e m serão fornecidos pelo usuário.

$$x = \sum_{i=1}^n \sum_{j=1}^m (i + j)$$

Exemplo de execução:

```
1 Valor de n: 10
2 Valor de m: 5
3
4 x = 425
```

Exemplo de uso (3)

```
1 n = int(input('Valor de n: '))
2 m = int(input('Valor de m: '))
3
4 soma = 0
5 for i in range(1, n+1):
6     for j in range(1, m+1):
7         soma += i + j
8
9 print(f'\nx = {soma}')
```

Comandos break e continue

Desvio em laços

Em diversos momentos queremos **encerrar** a execução de um laço de repetição. Um dos comandos utilizados para isso é o `break`

- Este comando permite encerrar o laço imediatamente.
- O comando `break` é particularmente útil para evitar uma quantidade excessiva de `if` aninhados.

Comando **break**

Uso de `break` em laço `while`:

```
1 ...  
2 while <Condição>:  
3     ...  
4     break  
5     ...  
6 ...
```

Uso de `break` em laço `for`:

```
1 ...  
2 for <Variável> in <Sequência>:  
3     ...  
4     break  
5     ...  
6 ...
```


Exemplo de uso: comando **break**

Faça um programa que imprime, entre 1 e 1 milhão, o primeiro número que é divisível por 11, 13 e 17.

- Precisamos do `break` para criar tal programa?
 - Definitivamente **não**!
 - Mas o `break` neste caso simplifica o código.

```
1 for i in range(1, 1000000+1):  
2     if i % 11 == 0 and i % 13 == 0 and i % 17 == 0:  
3         print(f'O número é {i}')4         break
```

Desvio em laços

Em diversos momentos queremos **alterar o fluxo** de um laço de repetição. Um dos comandos utilizados para isso é o `continue`

- Este comando permite alterar o fluxo do laço, fazendo-o retornar ao início.
- Assim como o `break`, o comando `continue` é útil para evitar `if` aninhados em alguns casos.

Comando **continue**

Uso de `continue` em laço `while`:

```
1 ...  
2 while <Condição>:  
3     ...  
4     continue  
5     ...  
6 ...
```

Uso de `continue` em laço `for`:

```
1 ...  
2 for <Variável> in <Sequência>:  
3     ...  
4     continue  
5     ...  
6 ...
```

Exemplo de uso: comando **continue**

Faça um aplicativo que soma todos os números de 1 até 1000, exceto os múltiplos de 5, 7 ou 13.

- Precisamos do `continue` para criar tal programa?
 - Definitivamente **não** (ele inclusive nem ajuda neste exemplo).
 - Mas o `continue` é uma alternativa válida que (em alguns casos) simplifica o código.

```
1 soma = 0
2 for i in range(1000+1):
3     if i % 5 == 0 or i % 7 == 0 or i % 13 == 0:
4         continue
5     soma += i
6
7 print(f"Soma = {soma}")
```

Exemplos e exercícios

Exemplo 1

Crie um programa em Python que imprime uma tabela contendo a tabuada de multiplicação de 1 a 20 ignorando os **números pares**.

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

1		1	3	5	7	9	11	13	15	17	19
2	-----										
3	1	1	3	5	7	9	11	13	15	17	19
4	3	3	9	15	21	27	33	39	45	51	57
5	5	5	15	25	35	45	55	65	75	85	95
6	7	7	21	35	49	63	77	91	105	119	133
7	9	9	27	45	63	81	99	117	135	153	171
8	11	11	33	55	77	99	121	143	165	187	209
9	13	13	39	65	91	117	143	169	195	221	247
10	15	15	45	75	105	135	165	195	225	255	285
11	17	17	51	85	119	153	187	221	255	289	323
12	19	19	57	95	133	171	209	247	285	323	361

Exemplo 1

```
1 # imprimindo cabeçalho
2 print('      ', end='')
3 for col in range(1, 21, 2):
4     print(f'{col:3d} ', end='')
5 print()
6 print('-' * 45)
7
8 # imprimindo a tabela
9 for lin in range(1, 21, 2):
10     print(f'{lin:3d} | ', end='')
11     for col in range(1, 21, 2):
12         produto = lin * col
13         print(f'{produto:3d} ', end='')
14     print()
```

Exemplo 2

Bart Simpson está aprendendo a jogar xadrez, mas tem dificuldade em saber para qual direção ele pode mover sua **Torre**.

Escreva um programa que solicite ao *Bart* o número da linha e da coluna que indicam a posição de sua **Torre**. O programa deve imprimir quais são os possíveis movimentos da **Torre**.

- Utilize  para indicar uma casa para a qual a Torre não pode ser movida e  para indicar uma casa para a qual ela pode ser movida.
- Sabemos que um tabuleiro de xadrez é composto por 8 linhas e 8 colunas, e que a **Torre** se move ortogonalmente, ou seja, pelas linhas (horizontais) e pelas colunas (verticais).

Exemplo 2

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

1 Digite a linha em que a Torre se encontra: 6

2 Digite a coluna em que a Torre se encontra: 3

3

4 Movimentos possíveis:

5

6 1 2 3 4 5 6 7 8

7 1 - - X - - - - -

8 2 - - X - - - - -

9 3 - - X - - - - -

10 4 - - X - - - - -

11 5 - - X - - - - -

12 6 X X X X X X X X

13 7 - - X - - - - -

14 8 - - X - - - - -

Exemplo 2

```
1  lin = int(input('Digite a linha em que a Torre se encontra: '))
2  col = int(input('Digite a coluna em que a Torre se encontra: '))
3
4  # imprimindo o cabeçalho
5  print('\nMovimentos possíveis:\n')
6  print('    1  2  3  4  5  6  7  8')
7
8  # imprimindo o tabuleiro
9  for i in range(1, 9):
10     print(f'{i}    ', end='')
11
12     for j in range(1, 9):
13         if lin == i or col == j:
14             print('x  ', end='')
15         else:
16             print('-  ', end='')
17
18     print()
```

Exemplo 3

Calcule o somatório a seguir, assumindo que x e y são valores dados pelo usuário. O resultado deve ser impresso com duas casas decimais.

$$S = \sum_{i=1}^{x-1} \sum_{j=1}^{y-1} \sqrt{i \times j}$$

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

```
1 Valor de x: 8
2 Valor de y: 5
3
4 S = 82.84
```

Exemplo 3

```
1 import math
2
3 x = int(input('Valor de x: '))
4 y = int(input('Valor de y: '))
5
6 S = 0
7 for i in range(1, x):
8     for j in range(1, y):
9         S += math.sqrt(i * j)
10
11 print(f'\nS = {S:.2f}')
```

Perguntas?
