



Estácio

ARA0066 - PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

PROFESSOR: LUCAS SAMPAIO LEITE

Agenda

- ❑ Estruturas de repetição
 - ❑ While
 - ❑ **For**

Estruturas de repetição (for)

- ❑ Laço de repetição, assim como o while.
- ❑ A sintaxe da utilização do for com a função range():

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

- ❑ Neste caso, o primeiro valor de i é 0 e segue sendo incrementado de um em um até o número n-1.
- ❑ Para utilizar o for é sempre necessário indicar uma variável iteradora que irá assumir um valor diferente para cada iteração do laço, e é sempre necessário indicar os limites de iteração.

Estruturas de repetição (for)

- ❑ Laço de repetição, assim como o while.
- ❑ A sintaxe da utilização do for com a função range():

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

- ❑ Exemplo:

```
main.py  
1  for i in range(5):  
2      print(i)  
3
```

Neste caso, o laço for imprime na tela o respectivo valor de i para cada iteração.


O primeiro valor de i é 0 e segue sendo incrementado de um em um até o número 4.

Estruturas de repetição (for)

- ❑ É possível incluir um valor de início da contagem diferente de 0;
- ❑ Para isso devemos usar a função com dois parâmetros:
- ❑ Exemplo:

```
main.py > ...  
1  for i in range(2,6):  
2      print(i)  
3
```

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



Neste caso, o laço for imprime na tela o respectivo valor de i para cada iteração.
O primeiro valor de i é 2 e segue sendo incrementado de um em um até o número 5.

Estruturas de repetição (for)

- ❑ A função range também pode ser utilizada para limitar a execução do laço de repetição for, como a função range(m, n, p), que cria uma lista de inteiros começando em m e terminando em n-1 sendo incrementada de p em p.

```
main.py > ...  
1 m=1  
2 n=100  
3 p=2  
4 for i in range (m,n,p):  
5     print(i)
```

```
main.py > ...  
1 for i in range(1,100,2):  
2     print(i)  
3
```

O que seria impresso?

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 pode assumir o valor de uma string ou elemento de uma lista:

```
 main.py > ...  
1   x = 'Lógica de programação'  
2   for i in x:  
3       |   print (i)
```


Estruturas de repetição (for)

- ❑ O laço for também é utilizado para percorrer listas.
- ❑ Escrever **for i in [0, 1, 2, 3]:** é o mesmo que representar:
 - ❑ “Para cada valor de “i” dentro dos valores (0, 1, 2 e 3), faça:”.

```
main.py > ...  
1  ∨ for i in [0, 1, 2, 3]:  
2    |     print(i)  
3
```

Estruturas de repetição (for)

- ❑ A função `len()` é utilizada quando se quer percorrer uma lista, de tal forma que o valor da variável iteradora não assuma cada valor da lista e sim cada posição da lista.
- ❑ Ou seja, não importa o conteúdo de cada posição da lista, a variável iteradora irá assumir o valor de cada posição.

```
main.py > ...  
1  x = ['Lógica de programação', 'é', 'nota', 10.0, '\n']  
2  
3  for i in range(len(x)):  
4      |    print (i)  
5  
6  for i in x:  
7      |    print (i)
```

Estruturas de repetição (for)

- O laço for pode ser utilizado através das instruções break e continue.
- A instrução break interrompe o laço (terminando-o por completo) e a instrução continue pula para a próxima iteração imediatamente (não termina o laço, apenas passa à próxima iteração).

```
main.py > ...  
1  for i in range(5):  
2      if i == 0:  
3          print('\ni = 0, Então: ', i)  
4      elif i == 1:  
5          print('\ni = 1, Então: continue')  
6          continue  
7      elif 1 < i < 3:  
8          print('\nA variável i, é: ', i)  
9      elif i == 3:  
10         print('\ni = 3, Então: break')  
11         break  
12     else:  
13         print('\ni > 3, Então: ', i)
```

O que será impresso quando i=4?

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.

Mais 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
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
```

Mais exercícios (usando 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 harmónica:

$$H(n) = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n$$

Faça um programa que leia um valor n inteiro e positivo e apresente o valor de $H(n)$

Mais exercícios (usando for ou while)

9. A série de Fibonacci é formada pela seqüência 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.: $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$. A saída deve ser conforme o exemplo abaixo:

```
Fatorial de: 5
```

```
5! = 5 . 4 . 3 . 2 . 1 = 120
```

Dúvidas???

