

Programação em Python

Estruturas de repetição (for)

Prof. Daniel Di Domenico

https://github.com/danidomenico/gex003_algprog

Slides cortesia da profa. Andrea Charão (UFSM) e do prof. João V. F. Lima (UFSM)

Estruturas de repetição

- **Problema:** elabore um algoritmo para somar 10 números

```
n1 = int(input("Digite um numero: "))
n2 = int(input("Digite um numero: "))
n3 = int(input("Digite um numero: "))
n4 = int(input("Digite um numero: "))
n5 = int(input("Digite um numero: "))
n6 = int(input("Digite um numero: "))
n7 = int(input("Digite um numero: "))
n8 = int(input("Digite um numero: "))
n9 = int(input("Digite um numero: "))
n10 = int(input("Digite um numero: "))

soma = n1 + n2 + n3 + n4 + n5 + n6 + n7 + n8 + n9 + n10

print(soma)
```

Estruturas de repetição

- **Solução:** somar 10 números (**while**)

```
conta = 0
soma = 0
while conta < 10:
    n1 = int(input("Digite um numero: "))
    soma = soma + n1
    conta = conta + 1

print(soma, conta)
```

Estruturas de repetição

- **Solução:** somar 10 números (**for**)

```
soma = 0
for num in 0, 1, 2, 3, 4, 5, 6, 7, 8, 9:
    n1 = int(input("Digite um número: "))
    soma = soma + n1

print(soma)
```

Estruturas de repetição

- **for** em Python:
 - "para todos os valores em um conjunto **X**"
 - **valor** é uma variável que assumirá cada um dos valores existentes no conjunto **X**

```
for valor in X:  
    # bloco de comando(s)  
    # a repetir
```

- **Importante:**
 - **Evite** alterar a variável de controle do laço

```
for valor in X:  
    valor = 200
```



ERRADO!

Estruturas de repetição

- Atenção ao **recuo** (endentaçãoooo!!!)
- Usar sempre o mesmo recuo em todo programa

```
for valor in x:  
    # comando(s)
```

```
for valor in x:  
    # comando(s)
```

CERTO!

```
for valor in x:  
# comando(s)
```

```
for valor in x:  
    # comando(s)
```

ERRADO!
Recuos diferentes!

Estruturas de repetição

- O conjunto pode ser de **inteiros**
 - no futuro, o conceito de **listas** será discutido.
- Exemplo:
- Saída:

```
for num in 9, 12, 13, 23, 24:  
    print(num)
```

```
9  
12  
13  
23  
24
```

Estruturas de repetição

- O conjunto pode ser de **caracteres**
 - no futuro, o conceito de **listas** será discutido.
- Exemplo:
- Saída:

```
for c in "abcde":  
    print(c)
```

```
a  
b  
c  
d  
e
```


Estruturas de repetição

- O conjunto pode ser de **ambos**
 - mesmo sem fazer muito sentido
- Exemplo:

```
for valor in 2, 4, "toto", 5:  
    print(valor)
```

- Saída:

```
2  
4  
toto  
5
```

Estruturas de repetição

- **range(n)**
 - Cria um conjunto de valores inteiros (de **0** até **n-1**)
- Exemplo:
- Saída:

```
for num in range(10):  
    # repete os comandos 10 vezes  
    print(num)
```

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

Estruturas de repetição

- **range(a, n)**
 - Cria um conjunto de valores inteiros (de **a** até **n-1**)
- Exemplo:
- Saída:

```
for num in range(2, 11):  
    # repete os comandos 9 vezes  
    print(num)
```

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

Estruturas de repetição

- **range(a, n, i)**
 - Cria um conjunto de valores inteiros (de **a** até o máximo de **n-1**), em intervalos de **i**
- Exemplo:
- Saída:

```
for num in range(2, 12, 2):  
    print(num)
```

```
2  
4  
6  
8  
10
```

Estruturas de repetição

- **range(a, n, i)**
 - Cria um conjunto de valores inteiros (de **a** até o máximo de **n-1**), em intervalos de **i**
- Exemplo:
- Saída:

```
for num in range(9, -1, -1):  
    print(num)
```

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

Estruturas de repetição

- Laços equivalentes:

```
num = 0
while num < 10:
    print(num)
    num = num + 1
```

```
for num in range(10):
    print(num)
```

Estruturas de repetição

- **Problema:** elabore um programa que leia 10 valores e imprima **o maior deles** no final.
- Usar uma variável para guardar o maior número lido. Essa variável é testada e possivelmente atualizada a cada repetição.

Estruturas de repetição

- Programa em Python:

```
maior = int(input("Digite um valor: "))  
for i in range(9):  
    valor = int(input("Digite um valor: "))  
    if valor > maior:  
        maior = valor  
  
print("Maior valor =", maior)
```


Guia para programar repetição

- Determine as variáveis/valores/comandos que precisam estar **ANTES** do laço
 - é o caso sem repetição
- Determine os limites do laço
 - Início, fim, incremento
- Dentro do laço
 - Não esqueça do recuo à direita
 - Não modifique o valor da variável de controle do laço

Estruturas de repetição

- Quando costuma-se utilizar **FOR**:
 - Quando sabe-se o número de iterações do laço;
- Quando costuma-se utilizar **WHILE**:
 - Quando o número de iterações do laço é indefinido;

ATENÇÃO!

- **Isto não é regra.** Cabe ao programador definir a melhor estratégia.

Estruturas de repetição

- 1) Escreva um programa usando **for** que leia 5 valores e mostre a soma entre o maior e o menor deles.
- 2) Escreva um programa usando **for** que imprima os números pares entre 0 e 50.