



Introdução ao Processamento de Dados Turma (2025.1)



Estruturas de Repetição (Parte 2)

Tassio Sirqueira (IME/UERJ)

tassio.sirqueira@ime.uerj.br

Decisão e Repetição

- Estruturas fundamentais para a programação.
- Praticamente todo o programa contém tais estruturas.
- Elas interferem no **fluxo de programação**: ordem em que as instruções (linhas) de programa são executadas.
- **Decisão**: executa um conjunto de instruções *se (if)* uma determinada condição é atendida.
- **Repetição**: executa repetidas vezes um conjunto de instruções *enquanto (while)* uma condição é atendida, ou por um número fixo de **iterações**.

Fluxo de Processamento

- Já vimos como as **estruturas de decisão** podem alterar o fluxo de processamento.
- Nesta aula continuaremos a entender como as **estruturas de repetição** podem alterar o fluxo de processamento
- Existem duas estruturas de repetição no Python:
 - ***while***: executa repetidas vezes um conjunto de instruções **enquanto** uma condição é atendida (dada na aula passada).
 - ***for***: executa um conjunto de instruções repetidas vezes por um **número fixo de iterações** (nesta aula).

Estrutura de Repetição: Para-Faça

- Executa uma sequência (bloco) de instruções por um certo número de vezes (**iterações**).
- O número de iterações (repetições do bloco de instruções) é **fixo e conhecido**.
- Estrutura de repetição (Pseudocódigo):

```
...  
para variável ← valor_inicial até valor_final faça  
    instruções  
fim para  
...
```

Estrutura de Repetição: Para-Faça

- Estrutura de repetição (Pseudocódigo):

```
...  
para variável ← valor_inicial até valor_final faça  
    instruções  
fim para  
...
```

- Geralmente *variável*, *valor_inicial* e *valor_final* são inteiros.
- *valor_inicial* e *valor_final* podem ser variáveis ou constantes.
- Com frequência, o valor de *variável* é acessado/usado dentro do bloco de instruções.

Estrutura de Repetição: Para-Faça

...

para *variável* \leftarrow *valor_inicial* até *valor_final* faça
 instruções

fim para

...

- *variável* recebe o *valor_inicial*, e o bloco de instruções é executado.
- Soma 1 (incrementa) ao valor da *variável*.
- Se o valor da *variável* for menor que o *valor_final*: executa novamente o bloco de instruções.
- Se o valor da *variável* for maior que o *valor_final*: prossegue com o programa (não executa o bloco de instruções uma última vez!)

Estrutura de Repetição: Para-Faça

Exemplo: algoritmo que retorna o valor da soma dos 10 primeiros números inteiros.

```
algoritmo soma_10_inteiros
inicio
    soma ← 0
    para i ← 1 até 10 faça
        soma ← soma + i
    fim para
    escrever 'Soma dos 10 primeiros inteiros: ', soma
fim
```

Estrutura de Repetição: Para-Faça

Exemplo: algoritmo que retorna o valor da soma dos N primeiros números inteiros.

```
algoritmo soma_N_inteiros
inicio
    ler N
    soma ← 0
    para i ← 1 até N faça
        soma ← soma + i
    fim para
    escrever 'Soma dos', N, 'primeiros inteiros: ', soma
fim
```


Estrutura de Repetição: Para-Faça

Exemplo: algoritmo que lê um número (n) e calcula o fatorial ($n!$) deste número.

$$4! = 4 * 3 * 2 * 1 = 24$$

Estrutura de Repetição: Para-Faça

Exemplo: versão com estrutura Enquanto-Faça (*while*).

$$4! = 4 * 3 * 2 * 1 = 24$$

```
algoritmo fatorial_n
inicio
    ler n
    fatorial ← 1
    enquanto n > 0 faça
        fatorial ← fatorial * n
        n ← n - 1
    fim enquanto
    escrever 'O fatorial de', n, 'é:', fatorial
fim
```

Estrutura de Repetição: Para-Faça

Exemplo: versão com estrutura Para-Faça (*for*).

$$4! = 4 * 3 * 2 * 1 = 24$$

algoritmo **fatorial_n**

Início

ler n

fatorial \leftarrow 1

para i \leftarrow 1 até n faça

fatorial \leftarrow fatorial * i

fim para

escrever 'O fatorial de', n, 'é:', fatorial

fim

Estrutura de Repetição: Para-Faça (*for*)

Sintaxe em Python:

```
...  
for variavel in lista_valores:  
    instrucao_1  
    ...  
    instrucao_n  
...
```

- *lista_valores* é um **objeto sequencial** do Python (uma lista de valores).
- A cada iteração, *variavel* vai receber um valor de dentro (*in*) da *lista_valores* (de forma ordenada).
- Haverá tantas iterações quanto o número de valores na lista.

Estrutura de Repetição: *for*

Exemplo: escrever os valores de um lista de valores (objeto sequencial).

```
for v in ['a','b','c','d','e']:  
    print(v)
```

```
for i in ['e','banana','1000',20,9.5]:  
    print(i)
```

```
for n in [1,2,3,4,5,6,7,8,9,10]:  
    print(n)
```

Estrutura de Repetição: *for*

Sintaxe em Python:

```
...  
for variavel in range(val_inicial, val_final, passo):  
    instrucao_1  
    ...  
    instrucao_n  
...
```

- Quando estamos interessados em iterar por uma lista de valores numéricos (inteiros) usamos uma construção com ***in range(...)***.
- O comando range constrói uma lista de valores inteiros de acordo com os parâmetros *val_inicial*, *val_final* e *passo*.

Estrutura de Repetição: *for*

Sintaxe em Python:

```
...  
for variavel in range(val_inicial, val_final, passo):  
    instrucao_1  
    ...  
    instrucao_n  
...
```

- A lista de valores produzidas pelo comando ***range*** começa com *val_inicial*, mas acaba com (*val_final* - 1) caso o *passo* seja 1 (o intervalo é fechado no início e aberto no final).
- O valor de *passo* pode ser qualquer número inteiro (positivo ou negativo).
- O *passo* é opcional, caso seja omitido a sequência vai de 1 em 1.

Estrutura de Repetição: comando *range(...)*

Exemplos de listas de valores produzidas pelo comando **range(...)**:

- `range(1,10,1)` → [1, 2, 3, 4, 5, 6, 7, 8, 9]
- `range(1,10)` → [1, 2, 3, 4, 5, 6, 7, 8, 9]
- `range(1,10,2)` → [1, 3, 5, 7, 9]
- `range(0,10,2)` → [0, 2, 4, 6, 8]
- `range(0,10)` → [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- `range(10)` → [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
- `range(10,0,-1)` → [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

Estrutura de Repetição: *for*

Exemplo: programa que calcula o fatorial de um número (*while*).

$$4! = 4 * 3 * 2 * 1 = 24$$

```
n = int(input('Entre com um número positivo: '))
fatorial = 1
while n > 0:
    fatorial = fatorial * n
    n = n - 1
print('Fatorial:', fatorial)
```

Estrutura de Repetição: *for*

Exemplo: programa que calcula o fatorial de um número (*for*).

$$4! = 4 * 3 * 2 * 1 = 24$$

```
N = int(input('Entre com um número inteiro positivo:'))  
fatorial = 1  
for i in range(1,N+1):  
    fatorial = fatorial * i  
print('O fatorial de', N, 'é:', fatorial)
```

Estrutura de Repetição: *for*

Exemplo: programa que calcula o fatorial de um número (*for*).

$$4! = 4 * 3 * 2 * 1 = 24$$

```
N = int(input('Entre com um número inteiro positivo:'))
fatorial = 1
for i in range(2,N+1):
    fatorial = fatorial * i
print('O fatorial de', N, 'é:', fatorial)
```

Dá no mesmo! Por quê?

Estrutura de Repetição: *for*

Exemplo: dado um número, dizer se ele é primo.

- Número primo: divisível somente por 1 e por ele mesmo.
- A ideia é partir do princípio que o número é primo (hipótese inicial).
- Depois podemos dividir o número por todos os números menores do que ele (excluindo o número 1).
- Se alguma das divisões der resto zero: ele é divisível por outro número, e não é primo (a hipótese inicial estava errada).

Estrutura de Repetição: *for*

Exemplo: dado um número, dizer se ele é primo.

```
numero = int(input('Entre com um número inteiro positivo:'))
primo = True
for i in range(2,numero):
    if numero%i == 0:
        primo = False
if primo:
    print('O número é primo!')
else:
    print('O número não é primo!')
```

Estrutura de Repetição: *for*

Exemplo: dado um número, dizer se ele é primo (*enquanto o usuário quiser*).

```
continua = 's'  
while (continua == 's'):  
    numero = int(input('Entre com um número inteiro positivo:'))  
    primo = True  
    for i in range(2,numero):  
        if numero%i == 0:  
            primo = False  
    if primo:  
        print('O número é primo!')  
    else:  
        print('O número não é primo!')  
continua = input("Pressione 's' para continuar.")
```

Estrutura de Repetição: *for*

Exemplo: dado um número, dizer se ele é primo (outra forma de fazer).

- Número primo: divisível somente por 1 e por ele mesmo.
- Podemos contar a quantidade de números menores do que o número testado que o dividem.
- Se esta quantidade for 1 (somente o número 1 divide o número testado), o número é primo.

Estrutura de Repetição: *for*

Exemplo:

```
numero = int(input('Entre com um número inteiro positivo:'))
contador = 0
for i in range(1,numero):
    if numero%i == 0:
        contador = contador + 1
if contador==1:
    print('O número é primo!')
else:
    print('O número não é primo!')
```

Este programa está errado: não funciona corretamente para o número 1!

Como corrigir?

Estrutura de Repetição: *for*

Exemplo:

```
numero = int(input('Entre com um número inteiro positivo:'))
contador = 0
for i in range(2,numero):
    if numero%i == 0:
        contador = contador + 1
if contador==0:
    print('O número é primo!')
else:
    print('O número não é primo!')
```

O que o comando range vai retornar se o número for 1?

Estrutura de Repetição: *for*

Exercício: escreva um programa que retorne os N primeiros números primos.

Estrutura de Repetição: *for*

Exercício (estatísticas):

- Fazer um programa para ler o sexo ('M' ou 'F') e a idade de 20 pessoas.
- Escrever a média de idades das mulheres e dos homens.
- Escrever as idades da mulher e do homem mais velhos.
- Escrever as idades da mulher e do homem mais novos.

Estrutura de Repetição: *for*

Exercício (estatísticas):

- Fazer um programa para ler o sexo ('M' ou 'F') e a idade de 20 pessoas.
- Escrever a média de idades das mulheres e dos homens.

Apesar de sabermos que temos 20 indivíduos, precisamos contar a quantidade de homens e de mulheres para fazer a média.

Também precisamos somar as idades dos homens e das mulheres.

Depois dividimos as somas pelas respectivas quantidades.

Estrutura de Repetição: *for*

```
for i in range(0,20):
    sexo = input('Entre com o sexo (M ou F):')
    idade = int(input('Entre com a idade:'))
    if sexo == 'M':
        qteM = qteM + 1
        somaM = somaM + idade
    else:
        qteF = qteF + 1
        somaF = somaF + idade
print('Média de idade dos homens:', somaM/qteM)
print('Média de idade das mulheres:', somaF/qteF)
```

Este programa não vai funcionar! Qual é o problema?

Estrutura de Repetição: *for*

```
qteM = qteF = somaM = somaF = 0
for i in range(0,20):
    sexo = input('Entre com o sexo (M ou F):')
    idade = int(input('Entre com a idade:'))
    if sexo == 'M':
        qteM = qteM + 1
        somaM = somaM + idade
    else:
        qteF = qteF + 1
        somaF = somaF + idade
print('Média de idade dos homens:', somaM/qteM)
print('Média de idade das mulheres:', somaF/qteF)
```

As variáveis que vão **agregar** (acumular) valores precisam ser **inicializadas**!

Estrutura de Repetição: *for*

Exercício (estatísticas):

- Fazer um programa para ler o sexo ('M' ou 'F') e a idade de 20 pessoas.
- Escrever a média de idades das mulheres e dos homens.

Apesar de sabermos que temos 20 indivíduos, precisamos contar a quantidade de homens e de mulheres para fazer a média.

Também precisamos somar as idades dos homens e das mulheres.

Depois dividimos as somas pelas respectivas quantidades.

Podemos forçar que o sexo seja digitado corretamente.

Estrutura de Repetição: *for*

```
qteM = qteF = somaM = somaF = 0
for i in range(0,20):
    sexo_correto = False
    while not sexo_correto:
        sexo = input('Entre com o sexo (M ou F):')
        if (sexo != 'M') and (sexo != 'F'):
            print('Sexo incorreto! Por favor digite M ou F.')
            sexo_correto = False
        else:
            sexo_correto = True
    idade = int(input('Entre com a idade:'))
...
```


Estrutura de Repetição: *for*

Exercício (estatísticas):

- Fazer um programa para ler o sexo ('M' ou 'F') e a idade de 20 pessoas.
- Escrever a média de idades das mulheres e dos homens.
- Escrever as idades da mulher e do homem mais velhos.

Precisamos guardar a idade do mais velho.

A cada nova idade digitada, temos que comparar se a idade guardada para o mais velho é menor que a nova idade.

Estrutura de Repetição: *for*

```
qteM = qteF = somaM = somaF = 0
idade_mais_velho = idade_mais_velha = 0
for i in range(0,20):
    sexo = input('Entre com o sexo (M ou F):')
    idade = int(input('Entre com a idade:'))
    if sexo == 'M':
        if idade > idade_mais_velho:
            idade_mais_velho = idade
    else:
        if idade > idade_mais_velha:
            idade_mais_velha = idade
...
```

Estrutura de Repetição: *for*

Exercício (estatísticas):

- Fazer um programa para ler o sexo ('M' ou 'F') e a idade de 20 pessoas.
- Escrever a média de idades das mulheres e dos homens.
- Escrever as idades da mulher e do homem mais velhos.
- Escrever as idades da mulher e do homem mais novos.

Precisamos guardar a idade do mais novo.

A cada nova idade digitada, temos que comparar se a idade guardada para o mais novo é maior que a nova idade.

Como inicializar a variável que vai guardar a idade do mais novo?

Estrutura de Repetição: *for*

```
qteM = qteF = somaM = somaF = 0
idade_mais_novo = idade_mais_nova = 1000
for i in range(0,20):
    sexo = input('Entre com o sexo (M ou F):')
    idade = int(input('Entre com a idade:'))
    if sexo == 'M':
        if idade < idade_mais_novo:
            idade_mais_novo = idade
    else:
        if idade < idade_mais_nova:
            idade_mais_nova = idade
...
```

Estrutura de Repetição: *for*

Colocando tudo junto...

```
qteM = qteF = somaM = somaF = 0
idade_mais_velho = idade_mais_velha = 0
idade_mais_novo = idade_mais_nova = 1000
for i in range(0,20):
    sexo_correto = False
    while not sexo_correto:
        sexo = input('Entre com o sexo (M ou F):')
        if (sexo != 'M') and (sexo != 'F'):
            print('Sexo incorreto! Por favor digite M ou F.')
            sexo_correto = False
        else:
            sexo_correto = True
    idade = int(input('Entre com a idade:'))
```

...

...

```
    if sexo == 'M':
        qteM = qteM + 1
        somaM = somaM + idade
        if idade > idade_mais_velho:
            idade_mais_velho = idade
        if idade < idade_mais_novo:
            idade_mais_novo = idade
    else:
        qteF = qteF + 1
        somaF = somaF + idade
        if idade > idade_mais_velha:
            idade_mais_velha = idade
        if idade < idade_mais_nova:
            idade_mais_nova = idade
print('Média de idade dos homens:', somaM/qteM)
print('Idade do homem mais velho:', idade_mais_velho)
print('Idade do homem mais novo:', idade_mais_novo)
print('Média de idade das mulheres:', somaF/qteF)
print('Idade do mulher mais velha:', idade_mais_velha)
print('Idade do mulher mais nova:', idade_mais_nova)
```



Introdução ao Processamento de Dados Turma (2025.1)



Estruturas de Repetição (Parte 2)

Tassio Sirqueira (IME/UERJ)

tassio.sirqueira@ime.uerj.br