



Universidade Federal do Rio Grande
Centro de Ciências Computacionais



Algoritmos e Estruturas de Dados I

Estruturas de Repetição

Profs. Drs. Cleo Billa, Rafael Penna e Thiago da Silveira

1º Semestre de 2020

Roteiro

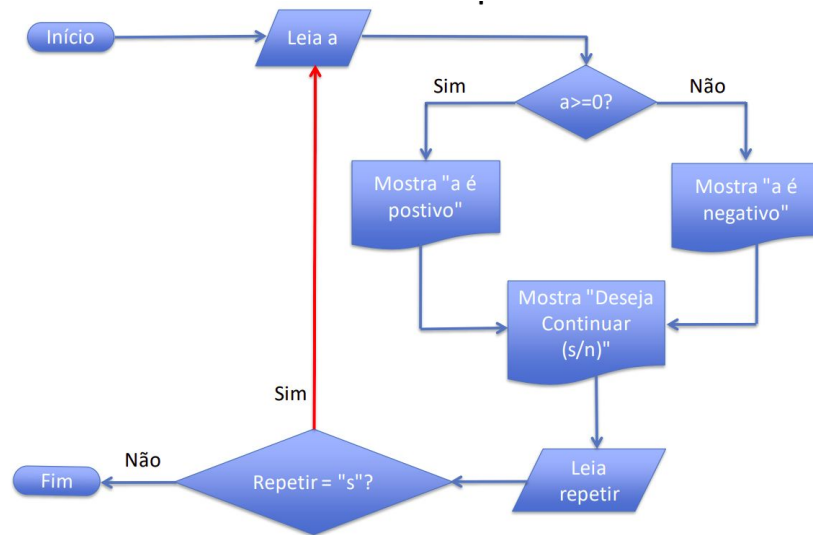
- O que é uma estrutura de repetição?
- Fluxograma vs. código em Python
- Comando `while`
- Comando `for`
- Estruturas de repetição aninhadas

O que é uma estrutura de repetição?

- São estruturas que permitem repetir blocos de comandos até que uma determinada situação ocorra;
- Situações de parada são descritas como condições (expressões) associadas a variáveis do programa (variáveis de controle);
- Repetição é fundamental para iterar sobre conjuntos de dados (como vamos ver em outras aulas).

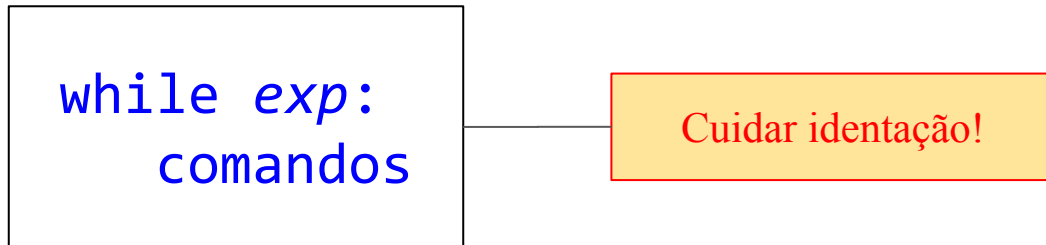
Fluxograma vs. código em Python

- Podemos utilizar **fluxogramas** para representar algoritmos;
- Repetição é representada por um losango (decisão) e *looping*;
 - Há uma condição (simples ou composta);
 - Há a definição dos caminhos possíveis;
 - Um deles forma um *looping*!



Comando while

- Operador é utilizado para executar um bloco de comandos várias vezes, enquanto uma determinada condição for satisfeita;



- Em que comandos pode ser um ou vários comandos em Python.

Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

Comando while

Exemplos:

```
valor = 0  
while valor < 10:  
    print(valor)
```

0, 0, 0, 0, ...

Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

0, 0, 0, 0, ...

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


Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

0, 0, 0, 0, ...

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

0, 1, 2, 3, ..., 9

Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

0, 0, 0, 0, ...

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

0, 1, 2, 3, ..., 9

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

Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

0, 0, 0, 0, ...

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

0, 1, 2, 3, ..., 9

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

1, 2, 3, ..., 10

Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

0, 0, 0, 0, ...

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

0, 1, 2, 3, ..., 9

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

1, 2, 3, ..., 10

```
valor = 0
while valor < 10:
    valor = valor+2
    print(valor)
```

Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

0, 0, 0, 0, ...

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

0, 1, 2, 3, ..., 9

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

1, 2, 3, ..., 10

```
valor = 0
while valor < 10:
    valor = valor+2
    print(valor)
```

2, 4, 6,..., 10

Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

0, 0, 0, 0, ...

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

0, 1, 2, 3, ..., 9

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

1, 2, 3, ..., 10

```
valor = 0
while valor < 10:
    valor = valor+2
    print(valor)
```

2, 4, 6,..., 10

```
valor = 0
while valor != 10:
    valor = valor+2
    print(valor)
```

Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

0, 0, 0, 0, ...

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

0, 1, 2, 3, ..., 9

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

1, 2, 3, ..., 10

```
valor = 0
while valor < 10:
    valor = valor+2
    print(valor)
```

2, 4, 6,..., 10

```
valor = 0
while valor != 10:
    valor = valor+2
    print(valor)
```

2, 4, 6,..., 10

Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

0, 0, 0, 0, ...

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

0, 1, 2, 3, ..., 9

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

1, 2, 3, ..., 10

```
valor = 0
while valor < 10:
    valor = valor+2
    print(valor)
```

2, 4, 6,..., 10

```
valor = 0
while valor != 10:
    valor = valor+2
    print(valor)
```

2, 4, 6,..., 10

```
valor = 0
while valor != 10:
    valor = valor+3
    print(valor)
```


Comando while

Exemplos:

```
valor = 0
while valor < 10:
    print(valor)
```

0, 0, 0, 0, ...

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

0, 1, 2, 3, ..., 9

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

1, 2, 3, ..., 10

```
valor = 0
while valor < 10:
    valor = valor+2
    print(valor)
```

2, 4, 6,..., 10

```
valor = 0
while valor != 10:
    valor = valor+2
    print(valor)
```

2, 4, 6,..., 10

```
valor = 0
while valor != 10:
    valor = valor+3
    print(valor)
```

3, 6, 9, ...

Comando for

- Operador é utilizado para executar um bloco de código por um número definido de vezes (definido por um iterável);
- A variável `var` assume cada um dos possíveis valores de `iteravel`;

```
for var in iteravel:  
    comandos
```

Cuidar indentação!

- Em que comandos pode ser um ou vários comandos em Python.

O iterável range

- O iterável `range` define uma **sequência** de valores;

`range(stop)`

Inteiros em `[0, stop)`

`range(start, stop [,step])`

Inteiros em `[start, stop)`

- Os parâmetros `start`, `stop` e `step` (opcional) devem ser inteiros;
- O parâmetro `step` assume o valor 1 caso não seja especificado;
 - A sequência resultante é equiespaçada por `step` unidades.

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

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

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

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

0, 1, 2, ..., 9

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

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

0, 1, 2, ..., 9

```
for i in range(11):  
    print(i)
```


Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

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

0, 1, 2, ..., 9

```
for i in range(11):  
    print(i)
```

0, 1, 2, ..., 10

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

```
for i in range(-1,11):  
    print(i)
```

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

0, 1, 2, ..., 9

```
for i in range(11):  
    print(i)
```

0, 1, 2, ..., 10

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

```
for i in range(-1,11):  
    print(i)
```

-1, 0, 1, ..., 10

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

0, 1, 2, ..., 9

```
for i in range(11):  
    print(i)
```

0, 1, 2, ..., 10

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

```
for i in range(-1,11):  
    print(i)
```

-1, 0, 1, ..., 10

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

0, 1, 2, ..., 9

```
for i in range(10,0):  
    print(i)
```

```
for i in range(11):  
    print(i)
```

0, 1, 2, ..., 10

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

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

0, 1, 2, ..., 9

```
for i in range(11):  
    print(i)
```

0, 1, 2, ..., 10

```
for i in range(-1,11):  
    print(i)
```

-1, 0, 1, ..., 10

```
for i in range(10,0):  
    print(i)
```

?

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

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

0, 1, 2, ..., 9

```
for i in range(11):  
    print(i)
```

0, 1, 2, ..., 10

```
for i in range(-1,11):  
    print(i)
```

-1, 0, 1, ..., 10

```
for i in range(10,0):  
    print(i)
```

?

```
for i in range(10,0,-1):  
    print(i)
```

Comando for

Exemplos:

```
for i in range(0):  
    print(i)
```

?

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

0, 1, 2, ..., 9

```
for i in range(11):  
    print(i)
```

0, 1, 2, ..., 10

```
for i in range(-1,11):  
    print(i)
```

-1, 0, 1, ..., 10

```
for i in range(10,0):  
    print(i)
```

?

```
for i in range(10,0,-1):  
    print(i)
```

10, 9, 8, ..., 1

Estruturas de repetição aninhadas

- Podemos ter estruturas de seleção aninhadas;
- Redobrar cuidado com a indentação!

```
for var1 in iter1:  
    bloco 1  
    while exp_1:  
        bloco 1.1  
        while exp_2:  
            bloco 1.1.1
```


Estruturas de repetição e seleção aninhadas

- A imensa maioria dos nossos programas
- Podemos ter estruturas de seleção aninhadas;
- **Redobrar** cuidado com a indentação!

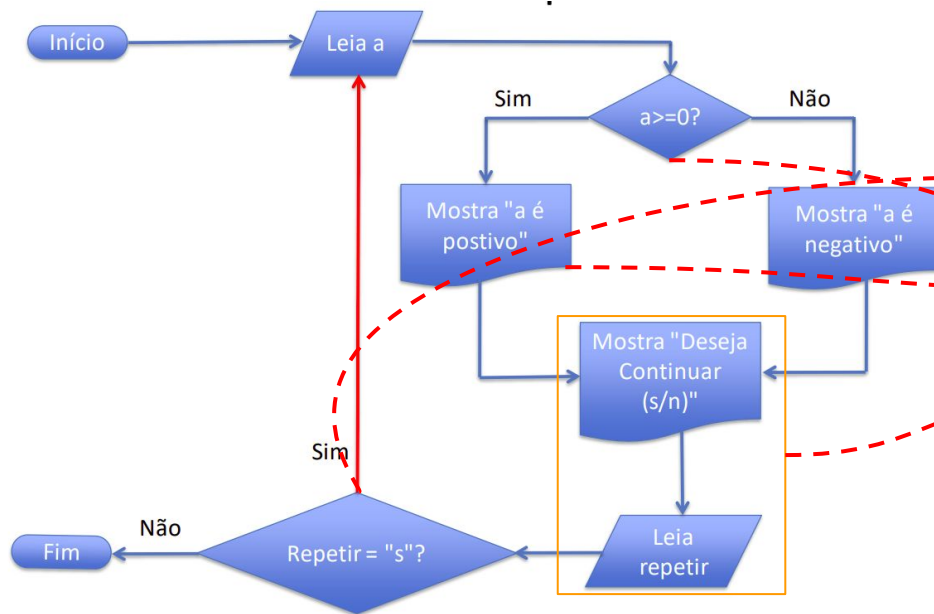
```
if exp_1:
    bloco 1
    while exp_2:
        bloco 1.1
        if exp_3:
            bloco 1.1.1
else:
    bloco 2
    while exp_4:
        bloco 2.1
```

Estruturas de repetição aninhadas

Exemplo:

```
numNotas = 2
numAlunos = 10
for aluno in range(numAlunos):
    nome = input('Digite seu nome: ')
    soma = 0
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        soma = soma + valor
    print(nome, 'teve média', soma/numNotas)
```

Fluxograma vs. código em Python (novamente)



```
Repetir = 's'
while Repetir == 's':
    a = float(input('Digite a: '))
    if a >= 0:
        print(a, 'é positivo')
    else:
        print(a, 'é negativo')
    Repetir = input('Deseja Continuar (s/n)? ')
```



Universidade Federal do Rio Grande
Centro de Ciências Computacionais



Algoritmos e Estruturas de Dados I

Estruturas de Repetição

Profs. Drs. Cleo Billa, Rafael Penna e Thiago da Silveira

1º Semestre de 2020