

Introdução à Programação para Sensoriamento Remoto

Aula 06 – Introdução à Programação com a Linguagem Python

**Gilberto Ribeiro de Queiroz
Thales Sehn Körting
Fabiano Morelli**



27 de Março de 2019

Tópicos

- Controlando o fluxo de um programa com estruturas de repetição.
- Laços do tipo `for` usando `range`.
- Laços do tipo `while`.

Estruturas de Repetição

Conversão entre Escalas de Temperatura: °F → °C

- **Problema:** Escrever um programa para converter a tabela abaixo de temperaturas em graus Fahrenheit na equivalente na escala Celsius.

temperatura inicial

Intervalo
(20 em 20)

temperatura final

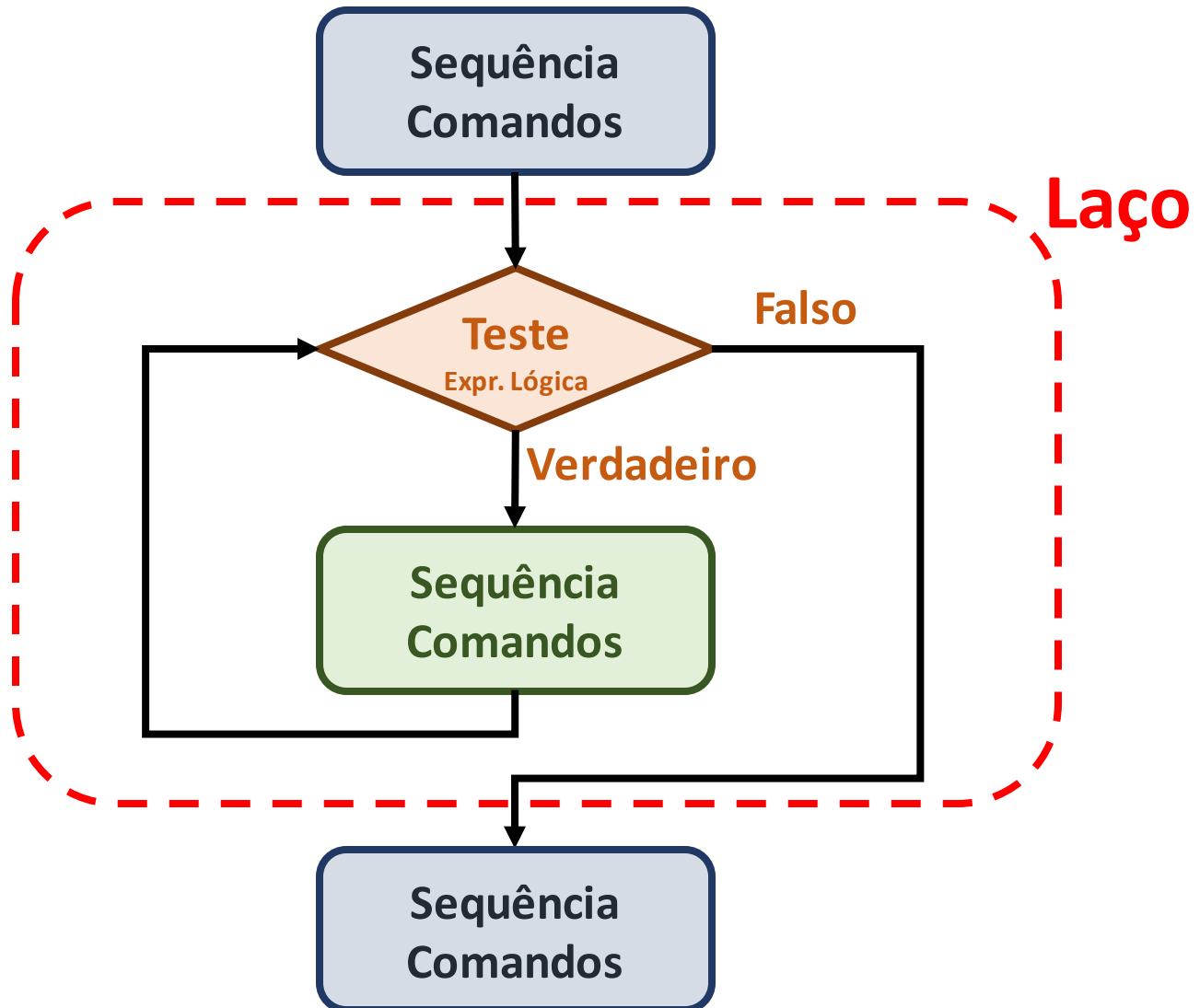
°F	°C
0	?
20	?
40	?
60	?
...	...
300	?

Como implementar esse
programa?

Comandos de Repetição

- Muitas das computações que realizamos em um programa são inerentemente **repetitivas**.
- Nas **linguagens imperativas**, encontramos estruturas específicas para essa finalidade, que são chamadas de **comandos de repetição** ou **laços** (*loops*) ou **estruturas de repetição**.
- Através desses comandos, podemos realizar uma computação até que uma certa condição seja satisfeita.

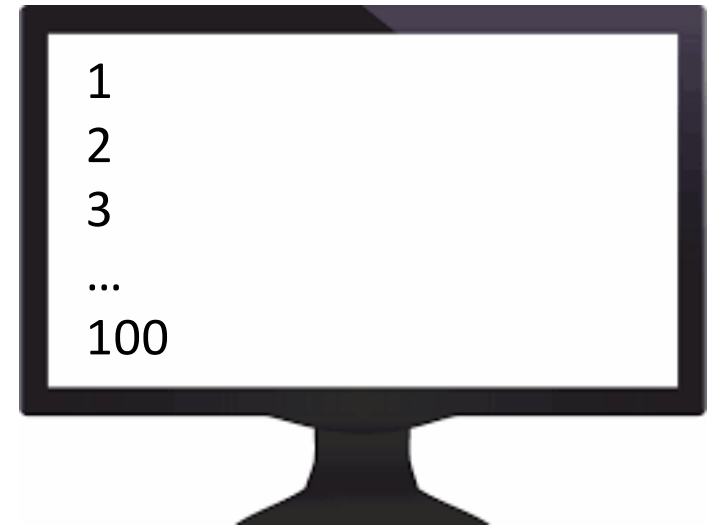
Estrutura de Repetição: Fluxograma



Vamos começar por um
problema mais simples...

Escrita repetitiva na tela

- **Problema:** Escreva um programa para escrever os números de 1 a 100 na tela.



```
for i in range(1, 101):  
    print(i)
```

Laços do tipo for

Iterando nos itens de um Conjunto

```
for variavel-it in iterável:  
    instrução-1  
    instrução-2  
    ...  
    instrução-n
```

Esse tipo de laço é muito útil quando estamos lidando com sequências (como strings, listas, e tuplas) ou iteráveis.

Somatório

- **Problema:** Escrever um programa para computar o seguinte somatório:

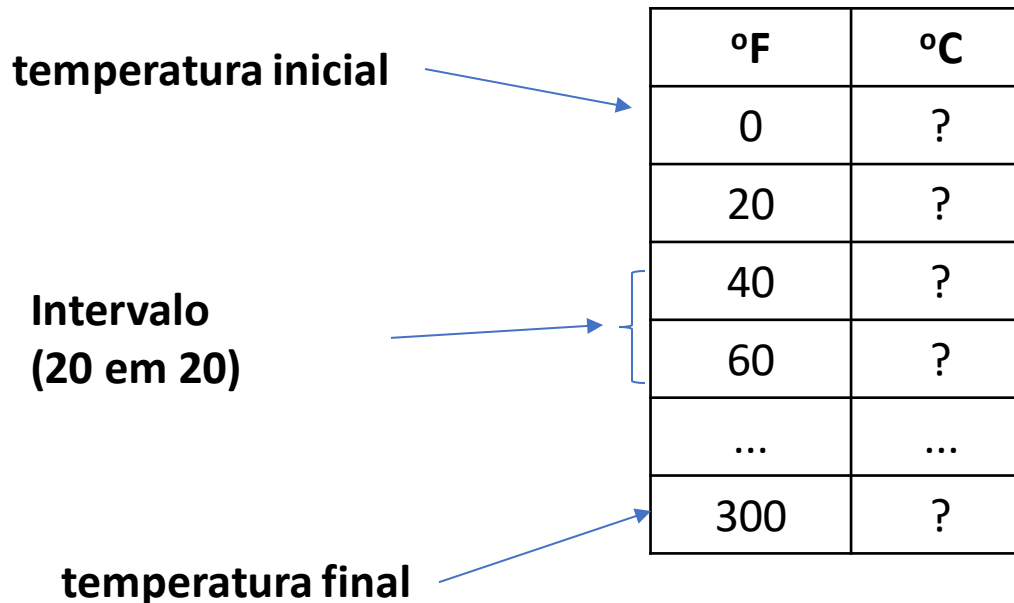
$$\sum_{i=1}^{i=5} i$$

```
01 soma = 0
02 for i in range(1, 6):
03     soma = soma + i
04 print(f"soma: {soma}")
```

Voltando ao nosso
problema...

Conversão entre Escalas de Temperatura: °F → °C

- **Problema:** Escrever um programa para converter a tabela abaixo de temperaturas em graus Fahrenheit na equivalente na escala Celsius.



°F	°C
0	?
20	?
40	?
60	?
...	...
300	?

temperatura inicial

Intervalo
(20 em 20)

temperatura final

temperatura inicial



°F	°C
0	?
20	?
40	?
60	?
...	...
300	?

Intervalo
(20 em 20)



temperatura final



```
01 for fahr in range(0, 320, 20):  
02     celsius = 5 * (fahr - 32) / 9  
03     print(fahr, celsius)
```

Laços do tipo `while`

Laços com Interrupção no Início

```
while <condição>:  
    instrução-1  
    instrução-2  
    ...  
    instrução-n
```

Nesse tipo de laço, se a expressão lógica ou condição de repetição no início do laço for verdadeira, os comandos (ou instruções) dentro da estrutura de repetição são executados de maneira sequencial.

Ao final da execução dos comandos, internos ao laço, o fluxo de controle do programa volta ao início, para nova avaliação da expressão lógica.

Se a expressão for satisfeita novamente (verdadeira), o corpo do laço é novamente executado, até que a repetição seja interrompida quando a expressão resultar em um valor falso.

Exemplo:

$$\sum_{i=1}^{i=5} i$$

```
01 i = 1
02 soma = 0

03 while i <= 5:
04     soma = soma + i
05     print(i, soma)
06     i = i + 1

07 print("Soma Final:", soma)
08 print("Valor final de i:", i)
```

ser347 [D:\Devel\ser347] - ...Aula-03\ahrenheit-to-celsius-3.py [ser347] - PyCharm

File Edit View Navigate Code Refactor Run Tools VCS Window Help

ser347 Aula-03 aahrenheit-to-celsius-3.py

fahrenheit-to-celsius-3

Project

ser347 D:\Devel\ser347

Aula-01

fahrenheit-to-celsius-1.py

fahrenheit-to-celsius-2.py

input.py

ola-mundo.py

Aula-02

ndvi-1.py

Aula-03

bissexto-01.py

bissexto-2.py

fahrenheit-to-celsius-3.py

line-equation-1.py

line-equation-2.py

.gitignore

External Libraries

1 t_min = 0 # temperatura mínima

2 t_max = 300 # temperatra máxima

3 delta_t = 20 # delta de temperatura a cada passo

4

5 fahr = t_min # temperatura Fahrenheit inicial

6

7 while fahr <= t_max:

8 celsius = 5 * (fahr - 32) / 9

9

10 print(fahr, celsius)

11

12 fahr = fahr + delta_t

13

Run: bissexto-2 fahrenheit-to-celsius-3

C:\Users\gribeiro\anaconda\envs\ser347\python.exe D:/Devel/ser347/Aula-03

0 -17.77777777777778

20 -6.666666666666667

40 4.4444444444444445

60 15.555555555555555

80 26.666666666666668

100 37.77777777777778

120 48.888888888888886

4 files committed: going on with class #3 (26 minutes ago)

13:1 n/a UTF-8 Git: master

Considerações Finais

Considerações Finais

- O tipo lógico e as expressões lógicas são muito utilizadas na construção de programas.
- As estruturas condicionais e de repetição são importante para controlar o fluxo de execução de um programa.
- Laços do tipo **for** são indicados quando termos que iterar (acessar ou atravessar) por uma coleção de objetos (ou itens). Veremos esse tópico mais adiante no curso.
- Laços do tipo **while** são indicados quando temos uma condição a ser verificada a cada iteração.

Considerações Finais

- Veremos que podemos modificar o fluxo de um laço através dos comandos `break` e `continue`.
- Nas próximas aulas iremos reforçar o uso das estruturas condicionais e de repetição.

Referências Bibliográficas

Exercícios