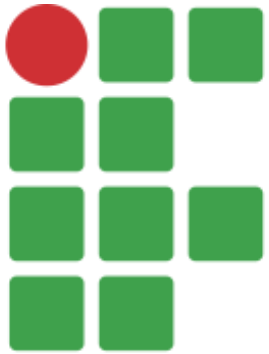


INSTITUTO FEDERAL
Paraíba

Campus Cajazeiras

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

PROFESSORES LEANDRO, MICHEL
CAJAZEIRAS / IFPB

Repetição While

OBJETIVOS

- Aprender sobre os estruturas de repetição em Python;
- Estudar a estrutura While;

ROTEIRO

- Estruturas de Repetição
- Repetição condicional

Estruturas de Repetição

- Em certas situações precisamos repetir as mesmas instruções várias vezes, por exemplo, suponha que precisamos ler as notas de um aluno e calcular sua respectiva média, teríamos o seguinte algoritmo:

main.py

```
1  #
2  #entrada de dados
3  nota1 = int(input("informe a primeira nota: "))
4  nota2 = int(input("informe a segunda nota: "))
5  #processamento
6  media = (nota1 + nota2) / 2
7  #saída de dados
8  print(f"a média calculada foi de {media}")
9
```

Console Shell

```
informe a primeira nota: 75
informe a segunda nota: 85
a média calculada foi de 80.0
>
```

Estruturas de Repetição

- Agora vamos calcular a média para 3 alunos:

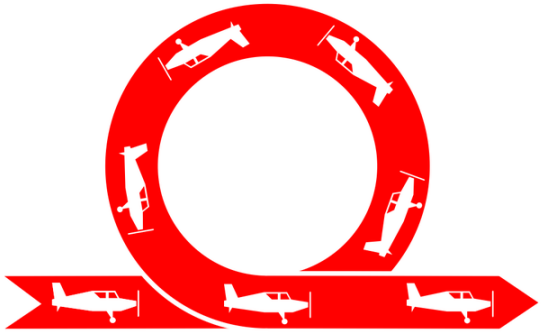
Agora, imagine uma turma com 30 alunos

```
#
#entrada de dados primeiro aluno
nota1Aluno1 = int(input("informe a primeira nota do primeiro aluno: "))
nota2Aluno1 = int(input("informe a segunda nota do primeiro aluno: "))
#processamento do primeiro aluno
mediaAluno1 = (nota1Aluno1 + nota2Aluno1) / 2
#saída de dados
print(f"a média calculada do primeiro aluno foi de {mediaAluno1}")

#entrada de dados segundo aluno
nota1Aluno2 = int(input("informe a primeira nota do segundo aluno: "))
nota2Aluno2 = int(input("informe a segunda nota do segundo aluno: "))
#processamento do segundo aluno
mediaAluno2 = (nota1Aluno2 + nota2Aluno2) / 2
#saída de dados
print(f"a média calculada do segundo aluno foi de {mediaAluno2}")

#entrada de dados terceiro aluno
nota1Aluno3 = int(input("informe a primeira nota do terceiro aluno: "))
nota2Aluno3 = int(input("informe a segunda nota do terceiro aluno: "))
#processamento do terceiro aluno
mediaAluno3 = (nota1Aluno3 + nota2Aluno3) / 2
#saída de dados
print(f"a média calculada do terceiro aluno foi de {mediaAluno3}")
```

Estruturas de Repetição

Permitem que um bloco de comandos seja executado diversas vezes

- Dois tipos de Repetição:
  - Repetição condicional: executa um bloco de código enquanto uma condição lógica for verdadeira (**while**)
  - Repetição contável: executa um bloco de código um número predeterminado de vezes (**for**)



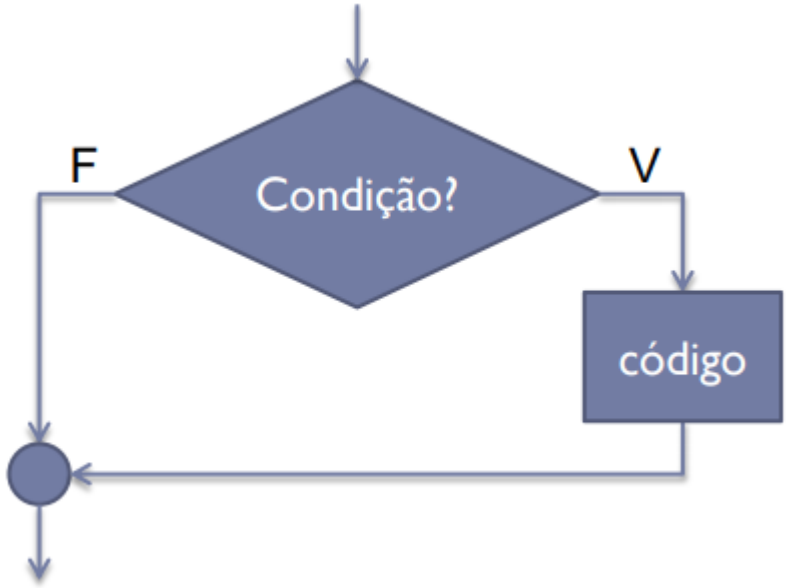
https://upload.wikimedia.org/wikipedia/commons/thumb/e/e5/Looping.png/640px-Looping.png

Repetição condicional

- Executa o bloco de instruções **enquanto a condição for verdadeira**
- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores
- O bloco de código pode conter um ou mais comandos
- O início e o fim do bloco são definidos de acordo com a endentação

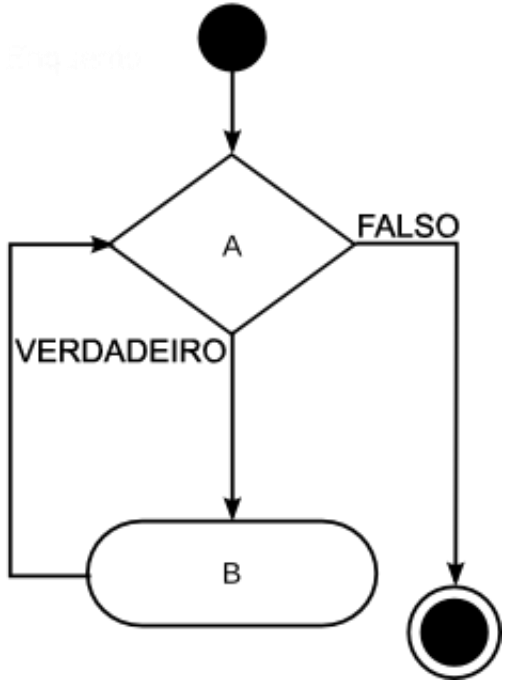
Repetição condicional

Se




```
...
if Condição:
    instrução 1
    instrução 2
    instrução N
    ...
```

Enquanto




```
...
while Condição:
    instrução 1
    instrução 2
    instrução N
    ...
```

Repetição condicional

O bloco de código dentro do laço **while** é repetido enquanto a condição do laço estiver sendo avaliada como verdadeira.

Sintaxe

```
while <condição>:
    <bloco de código>
    continue
    break
else:
    <bloco de código>
```

Executa um bloco de código atendendo a uma condição.

A instrução **continue** ignora a iteração atual de um loop e continua com a próxima iteração.

A instrução **break** do Python é usada para sair do loop imediatamente. Ele simplesmente salta completamente do loop, e o programa continua após o loop.

Python permite um **else** opcional no final de um **while**. O **else** será executada se o loop terminar naturalmente.

Repetição condicional

Variáveis:
Acumuladora
contadora

Exemplos

main.py

```
1  # Soma de 0 a 99
2  s = 0
3  x = 1
4  while x < 100:
5      s = s + x
6      x = x + 1
7  print(x)
8  print(s)
9
```

```
100
4950
```

main.py

```
1  # subitrair 1, até que x se torne 0
2  x = 6
3  while x:
4      print(x)
5      x -= 1
6  # Imprime 6 5 4 3 2 1
```

```
6
5
4
3
2
1
```

Repetição condicional

A instrução break é usada para sair do loop imediatamente. Ele salta completamente do loop, e o programa continua após o loop.

A instrução continue ignora a iteração atual de um loop e continua com a próxima interação.

Exemplos (break e continue)

main.py

```python
# Saia quando x se tornar 3
x = 6
while x:
    print(x)
    x -= 1
    if x == 3:
        break
# Imprime 6 5 4
```

```
6
5
4
```

main.py

```python
# Pular números ímpares
x = 6
while x:
    x -= 1
    if (x % 2) != 0:
        continue
    print(x)
# Imprime 4 2 0
```

```
4
2
0
```

```python
#continue
#loop pula para a proxima interação
#evitando ficar procesando o continue
x = 0
while x < 10:
    if x == 5:
        x +=1
        continue
    print(x)
    x +=1

print("fim")
```

Repetição condicional

Python permite o **else** opcional no final de um **while**. o **else** será executada quando o loop terminar normalmente (a condição se torna falsa).

O **else** ainda será executada se a condição for falsa no início.

Exemplos (Else)

main.py

```
1  x = 6
2  while x:
3      print(x)
4      x -= 1
5  else:
6      print("Concluído!")
7  # Imprime 6 5 4 3 2 1
8  # Imprime Concluído!
```

```
6
5
4
3
2
1
Concluído!
```

main.py

```
1  x = 6
2  while x:
3      print(x)
4      x -= 1
5      if x == 3:
6          break
7  else:
8      print("Concluído!")
9  # Imprime 6 5 4
10
```

```
6
5
4
```

Exercícios

Usando o laço de repetição while, resolva os exercícios abaixo.

1 - Programa que imprime a quantidade de números pares de 100 até 200, incluindo-os

2 -programa para contar a quantidade de números pares entre dois números quaisquer?

3 - Programa que imprime a soma de todos os números pares entre dois números quaisquer, incluindo-os.

4 - Programa que calcule o fatorial de um número.