

Curso Preparatório

Semana 5:

Estruturas repetição - WHILE

Admissão 2025



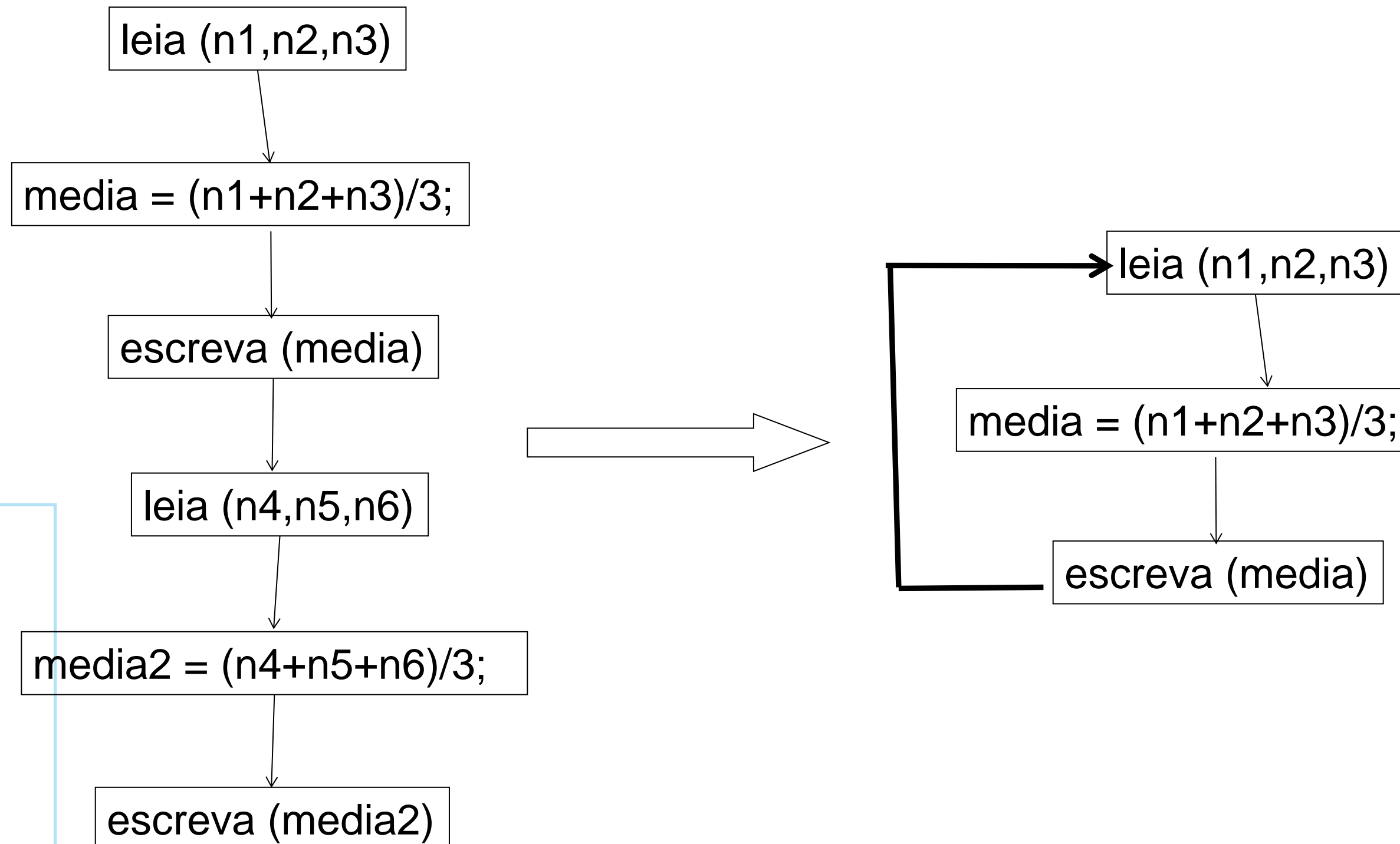


Estruturas de repetição

- Uma estrutura de repetição em Python é um recurso para desenvolver tarefas repetitivas em um loop contínuo. O loop funciona até uma condição ser satisfeita.
- É importante configurar devidamente essa condição para não cair em erros fatais em um programa.
 - O loop faz o que chamamos em programação de iteração, uma repetição que analisa alguma estrutura.
- A estrutura de repetição em Python funciona como um bloco de código ideal para executar uma única operação em todos os dados.
 - Contudo, se você precisa alterar o código a depender da informação lida, você pode estabelecer subcondições (com diferentes IFs) para verificações específicas.



Estruturas de repetição





Estruturas de repetição

- Imagina um programa para calcular a média de idade de um grupo de 10 pessoas

**E se fossem 50 pessoas ?
100 pessoas ?
200 pessoas ?**

```
algoritmo "media idades"
var
  id1, id2, id3, id4, id5, id6, id7, id8, id9, id10: inteiro
  media: real
inicio
  Escreva ("Digite a idade da pessoa 1: ")
  leia(id1)
  Escreva ("Digite a idade da pessoa 2: ")
  leia(id2)
  Escreva ("Digite a idade da pessoa 3: ")
  leia(id3)
  Escreva ("Digite a idade da pessoa 4: ")
  leia(id4)
  Escreva ("Digite a idade da pessoa 5: ")
  leia(id5)
  Escreva ("Digite a idade da pessoa 6: ")
  leia(id6)
  Escreva ("Digite a idade da pessoa 7: ")
  leia(id7)
  Escreva ("Digite a idade da pessoa 8: ")
  leia(id8)
  Escreva ("Digite a idade da pessoa 9: ")
  leia(id9)
  Escreva ("Digite a idade da pessoa 10: ")
  leia(id10)
  media<-(id1+id2+id3+id4+id5+id6+id7+id8+id9+id10)/10
  Escreva("A media das idades é: ", media)
fimalgoritmo
```



Estruturas de repetição

Note que existe um “Padrão de Comportamento” nesse algoritmo:

```
Programa mediaidades
  Inteiro id1, id2, id3, id4, id5, id6, id7
  real media
  inicio
  Escreva ("Digite a idade da pessoa 1: ")
  leia(id1)

  Escreva ("Digite a idade da pessoa 2: ")
  leia(id2)

  Escreva ("Digite a idade da pessoa 3: ")
  leia(id3)

  Escreva ("Digite a idade da pessoa 4: ")
  leia(id4)

  Escreva ("Digite a idade da pessoa 5: ")
  leia(id5)

  Escreva ("Digite a idade da pessoa 6: ")
  leia(id6)

  Escreva ("Digite a idade da pessoa 7: ")
  leia(id7)
```

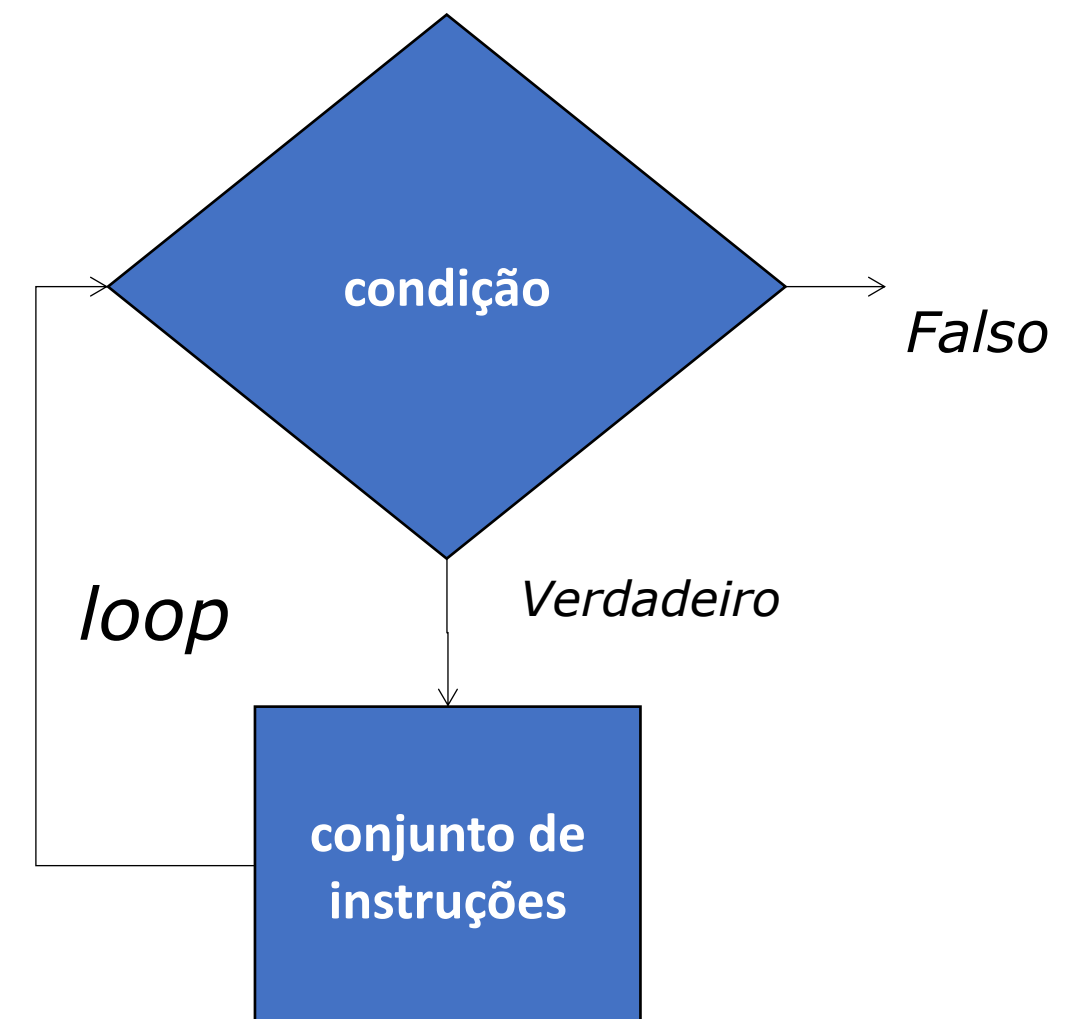
Padrão de Comportamento

REPETIÇÃO !



Estruturas de repetição

- Quando uma sequência de comandos deve ser executada **repetidas vezes** usamos uma estrutura de repetição.
- A estrutura de repetição, assim como a de decisão, envolve sempre a avaliação de uma **condição**.
- Também conhecidas como laços de repetição ou *loops*.





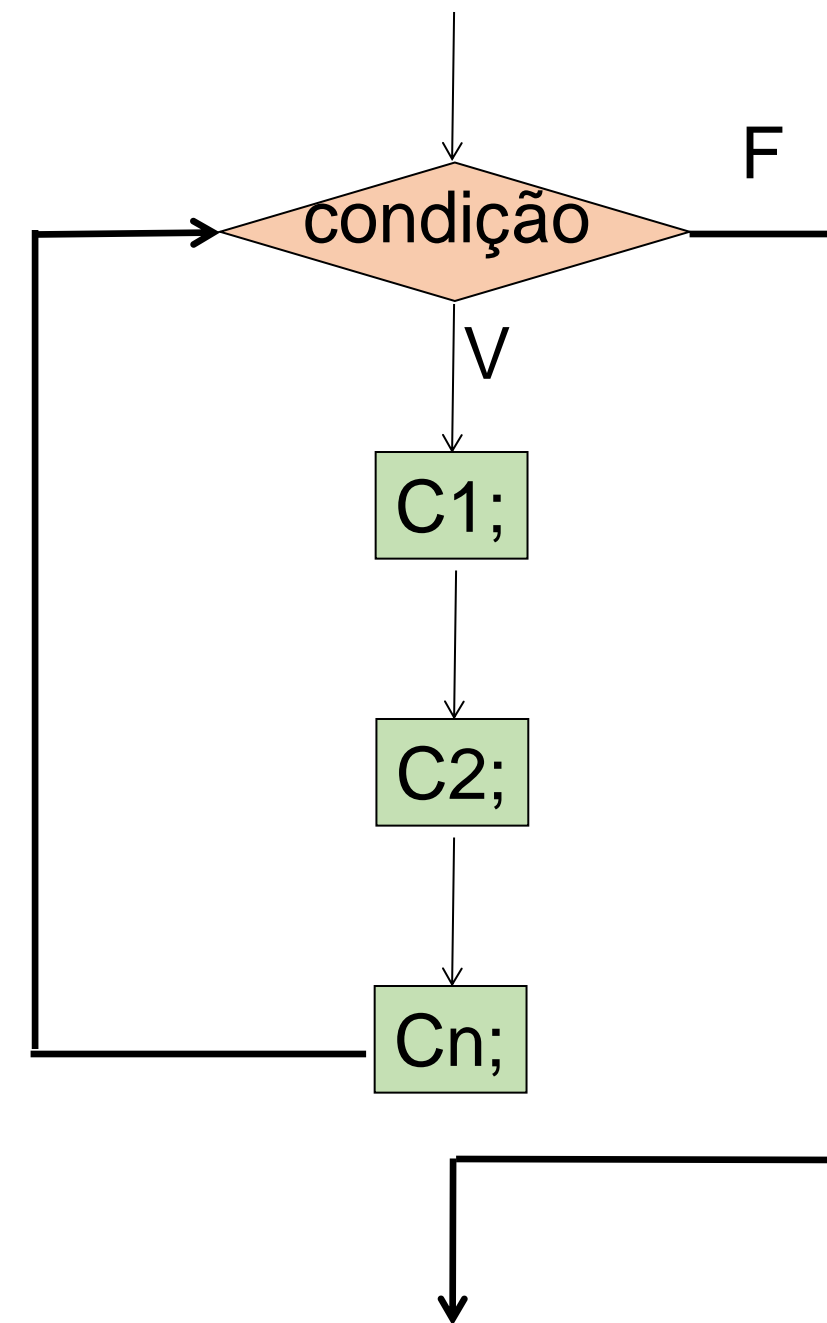
Estruturas de repetição

- Em Python as repetições podem ser realizadas com:
 - Teste no início;
ENQUANTO (WHILE)
 - Laços de repetição
 - *Loops ou looping*
PARA (FOR)



Estruturas de repetição

- Quando o resultado da condição for **falso**, o comando de repetição é abandonado.
- Se o resultado já é inicialmente falso, o bloco de repetição **NÃO** é executado nenhuma vez.





Estruturas de repetição

- Assim como no IF, o que marca o while é a **indentação**

Em JAVA

```
while (<condição>)  
{  
    C1;  
    C2;  
    Cn;  
}
```

Em Python

```
while (condição):  
    C1  
    C2  
    Cn
```



Estruturas de repetição

- O que faz o programa abaixo?
- Vamos fazer o **teste de mesa**!?

```
1  num1 = 5
2  num2 = 10
3  ✓ while (num1 < num2):
4      soma = num1+num2
5      print("O resultado da soma é",soma)
6      num2 -= 2
7  print("O Valor final de num2 é",num2)
```

```
O resultado da soma é 15
O resultado da soma é 13
O resultado da soma é 11
O valor final de num2 é 4
```



Estruturas de repetição

NADA!

- E esse?
- Vamos fazer o **teste de mesa**!?

```
1  A = 0
2  B = 5
3  i = 0
4  while ( (A>B) and (i<100) ):
5      C = A + B
6      i = i + 1
7      print("O Valor de C é",C)
```



Contador

- A estrutura “enquanto” não oferece um recurso para indicar **quantas iterações já foram realizadas**. Portanto, precisamos estabelecer um modo de contagem → **contador**;
- A cada iteração, o valor do contador é atualizado ou incrementado;

```
inteiro CON;  
CON ← 0;  
...  
CON ← CON+1;  
...
```

CONTADORES geralmente são inicializados com 0 (ZERO) ou com 1



While - PRÁTICA

- Exemplo: um programa que calcule a média de 5 alunos e informe se eles foram aprovados ou reprovados.

```
1  contador = 1
2  while (contador <= 5):
3      nota1 = float(input("Digite a primeira nota: "))
4      nota2 = float(input("Digite a segunda nota: "))
5      nota3 = float(input("Digite a terceira nota: "))
6      media = (nota1+nota2+nota3)/3
7      print(f"A média desse aluno ficou {media:.1f}")
8      if (media >= 6):
9          print("Aluno APROVADO")
10     else:
11         print("Aluno REPROVADO")
12     contador = contador + 1
```

CONTADOR aqui



Acumulador

- Um outro “**padrão de instrução**” bastante utilizado em programas que requerem o uso de REPETIÇÃO é a figura do **ACUMULADOR**
- Um “**ACUMULADOR**” geralmente tem a seguinte forma
 - **var = var + X** ou seja, a variável “recebe” o valor dela somado com mais X ... e assim por diante ...

```
1 contador = 1
2 acumulador = 0
3 while (contador <= 5):
4     x = int(input("Digite um valor: "))
5     acumulador = acumulador + x
6     contador += 1
7
8 print("Acumulador =", acumulador)
```

O que esse trecho de programa faz ?

ACUMULADOR é inicializado com 0 (ZERO)



Acumulador

- Se quiser saber a média de uma turma com X alunos.

CONTADOR aqui

ACUMULADOR aqui

- Nesse caso, preciso ir somando tudo (acumulando) as médias.

- No final, basta eu dividir o acumulador pela quantidade de alunos e terei a média da turma

ACUMULADOR aqui

CONTADOR aqui

ACUMULADOR aqui

```
1 alunos = int (input ("Quantos alunos a turma tem? "))
2 contador = 1
3 acumulador = 0
4 while (contador <= alunos):
5     nota1 = float(input("Digite a primeira nota "))
6     nota2 = float(input("Digite a primeira nota "))
7     nota3 = float(input("Digite a primeira nota "))
8     media = (nota1+nota2+nota3)/3
9     print("Media =",media)
10    acumulador = acumulador+media
11    if (media > 6):
12        print("Aluno Aprovado")
13    else:
14        print("Reprovado")
15    contador+=1
16
17 #saida do while
18 media = acumulador/alunos
19 print("A média da turma foi",media)
```



REPETIÇÃO - WHILE (enquanto)

- Programa para imprimir a tabuada de conversão de centímetros para polegadas em Python

```
1  cm=1
2  while (cm<=10):
3      pol=cm*2.54
4      if (cm==1):
5          print(cm,"centimetro =",pol,"polegadas")
6      else:
7          print(cm,"centimetros =",pol,"polegadas")
8      cm+=1
```




DO-WHILE

- Não existe em **PYTHON**

JAVA

```
do
{
    comandos;
} while ( condição ) ;
```



Validação de dados

- Usamos as estruturas de repetições para validar algum dado digitado pelo usuário.
- Saiba que você pode tratar/validar praticamente tudo o que o usuário vai digitar, lembra da frase? **O USUÁRIO É BURRO!** Então, muitas vezes precisamos validar até mesmo o que ele vai digitar.
- Quero aceitar que o usuário digite apenas um número entre 1 e 10:

```
1  numero = 0
2  while (numero < 1 or numero > 10):
3      numero = int(input("Digite um numero entre 1 e 10: "))
4
5  print("Fim do Programa")
```

- Quero aceitar que o usuário digite apenas o nome “Roldão”:

```
1  nome = " "
2  while (nome.lower() != "roldão"):
3      nome = input("Digite a palavra 'Roldão': ")
4
5  print("Fim do Programa")
```