

# Estruturas de controle

**Professor:** Aldecir de Almeida Fonseca

**Contato:** [aldecirfonseca@hotmail.com](mailto:aldecirfonseca@hotmail.com)

## REPETIÇÃO (laço) incondicional - for

- **b) Repetição incondicional:** mais conhecido como **for**, é a estrutura de repetição mais utilizada no Python para percorrer sequências ou processar iteradores.
- A instrução for aceita não só sequências estáticas, mas também sequências geradas por iteradores. Iterador é uma estrutura que permite iterações, ou seja, acesso aos itens de uma coleção de elementos, de forma sequencial.

## ...REPETIÇÃO (laço) incondicional - for

- Durante a execução de um laço for, a referência aponta para um elemento da sequência. A cada iteração, a referência é atualizada, para que o bloco de código do for processe o elemento correspondente.

Sintaxe:

```
for referência in sequencia:  
    bloco de código  
    continue  
    break  
else:  
    bloco de código
```

## ... REPETIÇÃO (laço) incondicional - for

### ➤ Características:

- Repete o bloco de comandos ( $\text{valor\_final} - \text{valor\_inicial} + 1$ ) vezes.
- Incrementa (atualiza) automaticamente a variável de controle cada vez que o bloco é executado (incremento “default” de 1 até alcançar o valor final).
- Se o valor final definido for menor que o valor inicial, o **laço** (*loop*) de repetição não é executado nenhuma vez.
- A variável de controle deve ser do tipo **inteiro**.
- A variável usada como controle da estrutura não pode ser modificada dentro do bloco.

## ...REPETIÇÃO (laço) incondicional – classe range

➤ O tipo da classe range representa uma sequência imutável de números e é comumente usada para fazer um looping um número determinado de vezes por for.

➤ `class range(stop)`

➤ `class range(start, stop, [step])`

➤ Os argumentos para o construtor de intervalo devem ser inteiros.

➤ Se o argumento start for omitido será usado 0 (zero) como padrão.

➤ Se o argumento **step** for omitido, será usado 1 como padrão.

## ...REPETIÇÃO (laço) incondicional– Exemplos: for usando range

➤ Exemplo de iterações por quantidades definidas.

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

- Observe no exemplo que não foi passado o parâmetro **start** com isso o intervalo sempre se inicia em 0 (zero).
- Outro ponto importante que o ponto de parada fornecido nunca é incluído no intervalo da lista.

## ...REPETIÇÃO (laço) incondicional– Exemplos: for usando range

- Exemplo de iterações por intervalos definidos.

```
for x in range(1, 11):  
    print( x )
```

- Observe que para exibir de 1 a 10, precisamos colocar o ponto de parada como 11, pois o mesmo não é incluído na lista.

## ...REPETIÇÃO (laço) incondicional– Exemplos: for usando range

- Exemplo de iterações por intervalos definidos com incrementos fora do padrão.

```
for x in range(1, 11, 2):  
    print( x )
```

- Observe que definimos 2 como intervalo e com isso teremos um salto de 2 em 2 em nossas iterações.



## ...REPETIÇÃO (laço) incondicional– Exemplos: for usando range

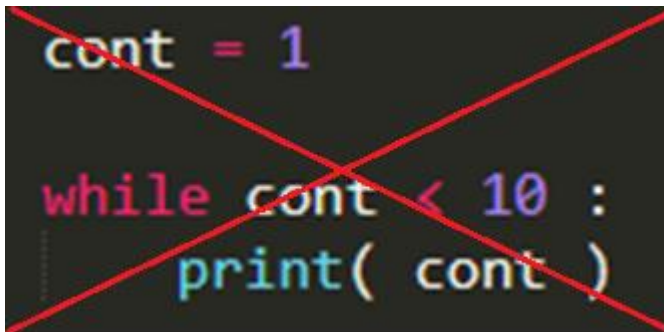
- Exemplo de iterações por intervalos definidos em ordem decrescente.

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

- Observe que definimos o start com o maior valor e o stop com o menor valor, no step adicionamos o sinal de menos (-) para diminuir o índice e não aumentar o mesmo.

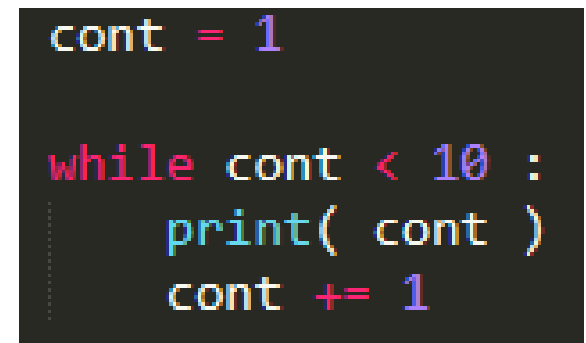
## LOOP INFINITO

- Ao se usar estruturas de repetições deve-se tomar muito cuidado para garantir que as repetições tenham fim, ou seja, a condição de parada seja alcançada.
- Caso contrário, o algoritmo ficará preso dentro da repetição, e nunca irá sair, causando um travamento do computador.
- Esse problema é muito comum, e é conhecido como *loop infinito*.



```
cont = 1  
while cont < 10 :  
    print( cont )
```

The code snippet is crossed out with a large red 'X', indicating it is an incorrect example of a loop.



```
cont = 1  
while cont < 10 :  
    print( cont )  
    cont += 1
```

The code snippet is a correct example of a loop with a termination condition.

## ...REPETIÇÃO incondicional (laço) – continue

- A instrução **continue** interrompe a execução do ciclo sem interromper a execução do laço de repetição, fazendo com que o cursor de execução vá para o cabeçalho (início) da estrutura de repetição.

```
cont = 0
print("iremos entrar no while")
while cont < 10:
    cont += 1
    # verifica se cont é par, caso seja não efetua a exibição do mesmo
    if ( cont % 2 ) == 0:
        continue
    print(cont)
print("saímos do while")
```

## ...REPETIÇÃO incondicional (laço) – break

- A instrução **break** finaliza a iteração e o Script continua a execução normalmente. O objetivo dessa instrução é fornecer a capacidade de forçar a interrupção da iteração.
- A instrução break interrompe não somente o ciclo em execução, mas sim, todo o laço.

```
cont = 0
print("iremos entrar no while")
while cont < 10:
    cont += 1
    # verifica se cont é igual a 5, caso seja interrompe a execução do while
    if cont == 5:
        break
    print(cont)
print("saimos do while")
```

## VARIÁVEIS ESPECIAIS

- As variáveis representam a informação manipulada pelo algoritmo, e portanto, pode aparecer em diversas situações.
- Além das já apresentadas, existem três casos especiais:
  - Variáveis acumuladoras: acumula valores.
  - Variáveis contadoras: conta valores.
  - Variáveis sinalizadoras ou *flags*: sinaliza erros, condições de falha de teste, etc. São variáveis que informam que uma determinada situação foi alcançada. Normalmente, são usadas variáveis do tipo booleano como sinalizadores.

## ...VARIÁVEIS ESPECIAIS

➤ Variáveis acumuladoras:

```
acumula = 0

while acumula < 100:
    num = int(input("Digite um número: "))
    acumula = acumula + num
```

➤ Variáveis contadoras:

```
conta = 0

while conta < 30:
    print(conta)
    conta = conta + 1
```

## ...VARIÁVEIS ESPECIAIS

➤ Variáveis sinalizadoras (*flags*)

```
dia = int(input("Dia: "))
mes = int(input("Mês: "))
ano = int(input("Ano: "))
invalida = False

# Muda o valor do flag para sinalizar um erro ou mudança de estado

if (dia < 1 ) or (dia>31):
    invalida = True
else:
    if (mes < 1 ) or (mes > 12 ):
        invalida = True

if invalida:
    print("Erro na data - dia invalido ou mes invalido")
```

## ...VARIÁVEIS ESPECIAIS

- No último “if” não usa operadores lógicos do tipo **invalida == True**, pois como a variável **invalida** já é booleana, ou seja, só pode apresentar os valores **True** ou **False**, ela já apresenta um resultado lógico.
- Em geral, *flags* são utilizadas quando não conhecemos o número de repetições e este for determinado por um valor que será lido.
- Assim, deve-se utilizar um controle de repetições por entrada, também conhecido como controle por sentinela.
- Para controle por *flag*, a melhor estratégia é utilizar um “loop infinito” com a condição de parada sendo aplicada através do comando **break**, dentro de um comando **if** logo após a leitura da variável a ser comparada com o *flag*.



## ...VARIÁVEIS ESPECIAIS

```
# Laço com Flag

# -*- coding: utf-8 -*-

num = 0
cont = 0
soma = 0
media = 0.0

while True:          # Loop infinito
    num = int(input("Digite um número inteiro (0 para encerrar): "))

    if num == 0:
        break

    soma = soma + num
    cont = cont + 1;

media = soma / cont

print( "Média = ", media )
```

## EXERCICIOS

33) Faça um algoritmo que determine o maior entre N números lidos.

- A condição de parada é a entrada de um valor 0.
- Ou seja, o algoritmo deve ficar calculando o maior até que a entrada seja igual a 0 (ZERO).

## EXERCICIOS

34) Leia 10 valores inteiros e determine:

- Qual é o menor valor informado;
- Qual é o maior valor informado;
- A soma dos valores informados;
- A média dos valores informados.

## ...EXERCICIOS

- 35) Ler 20 números e ao final informar quantos número(s) est(á)ão no intervalo entre 10 (inclusive) e 50 (inclusive).
- 36) Faça um algoritmo que receba a idade de 15 pessoas e mostre mensagem informando “maior de idade” e “menor de idade” para cada pessoa. Considere a idade a partir de 18 anos como maior de idade. Ao final apresente quantos % das pessoas informadas são “maior de idade” e “menor de idade”.

## ...EXERCICIOS

37) O cardápio de uma lancheira é o seguinte

Código	Descrição	Preço
100	Cachorro Quente	5,00
101	Hamburger	12,00
102	xBurguer	14,50
103	xEggBurguer	16,00
104	Cachorrão de Salsicha	11,50
105	xEtudo	18,00

Escrever um algoritmo que leia o código do item pedido, a quantidade e calcule o valor a ser pago por aquele lanche. O usuário poderá pedir quantos lanches desejar. Se informado quantidade zero interrompe a execução e exibe o total do pedido.

## ...EXERCICIOS

- 38) Uma empresa concederá um aumento de salário aos seus funcionários, variável de acordo com o cargo, conforme a tabela abaixo. Faça um algoritmo que leia o salário e o cargo de 5 funcionário e calcule o novo salário. Se o cargo do funcionário não estiver na tabela, ele deverá, então, receber 40% de aumento. Mostre o salário antigo, o novo salário e a diferença para cada funcionário. No final apresente o valor total da folha antes do reajuste e depois do reajuste.

Código	Cargo	Percentual
101	Gerente	10%
102	Engenheiro	20%
103	Técnico	30%

## ...EXERCICIOS

39) Escrever um algoritmo que leia o nome e o sexo de 15 pessoas, ao final determine:

- Total de homens e de mulheres.
- A mulher mais nova
- O homem mais velho
- A média de idade de todas as pessoas