

# MC-102 — Aula 04

## Expressões Relacionais, Lógicas e Comandos Condicionais

Eduardo C. Xavier

Instituto de Computação – Unicamp

8 de Março de 2018

# Roteiro

- 1 Expressões relacionais
- 2 Expressões lógicas
- 3 Comandos condicionais
- 4 Exercícios

# O Tipo bool

Em python o tipo **bool** especifica os valores booleanos falso (`False`) e verdadeiro (`True`).

- Podemos criar variáveis associadas a booleanos mas o uso mais comum é na verificação de resultados de expressões relacionais e lógicas.

```
>>> a = True
>>> type(a)
<class 'bool'>
```

# Expressão

- Já vimos que constantes e variáveis são expressões.

## Exemplo

```
a = 10  
a = b
```

- Vimos também que operações aritméticas também são expressões.

## Exemplo

```
a = 2 + 2  
a = 10 / 3  
a = a + 1
```

# Expressões relacionais

Expressões relacionais são aquelas que realizam uma **comparação** entre duas expressões e retornam

- ❶ **False**, se o resultado é falso
- ❷ **True**, se o resultado é verdadeiro.

# Operadores Relacionais

Os operadores relacionais da linguagem Python são:

- `==` : igualdade.
- `!=` : diferente.
- `>` : maior que.
- `<` : menor que.
- `>=` : maior ou igual que.
- `<=` : menor ou igual que.

# Expressões relacionais

- *expressão* == *expressão* : Retorna verdadeiro quando as expressões forem iguais.

```
>>> 9 == 9
True
>>> 9 == 10
False
>>>
```

- *expressão* != *expressão* : Retorna verdadeiro quando as expressões forem diferentes.

```
>>> 9 != 9
False
>>> 9 != 10
True
>>>
```

# Expressões relacionais

- *expressão* > *expressão* : Retorna verdadeiro quando a expressão da esquerda tiver valor maior que a expressão da direita.

```
>>> 9 > 5  
True
```

- *expressão* < *expressão* : Retorna verdadeiro quando a expressão da esquerda tiver valor menor que a expressão da direita.

```
>>> 9 < 5  
False  
>>>
```



# Expressões relacionais

- *expressão*  $\geq$  *expressão* : Retorna verdadeiro quando a expressão da esquerda tiver valor maior ou igual que a expressão da direita.

```
>>> 9 >= 5
True
>>>
```

- *expressão*  $\leq$  *expressão* : Retorna verdadeiro quando a expressão da esquerda tiver valor menor ou igual que a expressão da direita.

```
>>> 9 <= 5
False
>>>
```

O que será impresso pelo programa?

```
print(9 > 3)
```

```
print((3*4)/2 != (2*3) )
```

```
a = 1
```

```
b = -1
```

```
print(a != b)
```

# Expressões lógicas

Expressões lógicas são aquelas que realizam uma operação lógica (**ou**, **e**, **não**, etc...) e retornam verdadeiro ou falso (como as expressões relacionais).

# Operadores Lógicos

Na linguagem Python temos os seguintes operadores lógicos:

- and: operador E.
- or: operador OU.
- not: operador NÃO.

# Expressões lógicas

- *expressão* and *expressão* : Retorna verdadeiro quando ambas as expressões são verdadeiras. Sua tabela verdade é:

$Op_1$	$Op_2$	$Ret$
V	V	V
V	F	F
F	V	F
F	F	F

Qual o resultado da expressão lógica abaixo?

a = 0

b = 0

(a == 0 and b == 0)

# Expressões lógicas

- *expressão* or *expressão* : Retorna verdadeiro quando pelo menos uma das expressões é verdadeira. Sua tabela verdade é:

$Op_1$	$Op_2$	$Ret$
V	V	V
V	F	V
F	V	V
F	F	F

Qual o resultado da expressão lógica abaixo?

a = 0

b = 1

(a == 0 or b == 0)

# Expressões lógicas

- `not expressão` : Retorna verdadeiro quando a expressão é falsa e vice-versa. Sua tabela verdade é:

$Op_1$	$Ret$
V	F
F	V

Qual o resultado da expressão lógica abaixo?

`a = 0`

`b = 1`

`not (a != b)`

O que será impresso pelo programa?

```
print(8>9 and 10!=2)
```

```
print(14 > 100 or 2>1)
```

```
print(not (14>100) and not (1>2) )
```



# Comandos condicionais

Um comando condicional é aquele que permite decidir se um determinado bloco de comandos deve ou não ser executado, a partir do resultado de uma expressão relacional ou lógica.



# Bloco de comandos

- É um conjunto de instruções agrupadas.
- Os comandos agrupados do bloco devem estar **indentados** dentro de um comando anterior seguido de **dois pontos**.
- A indentação é feita em geral com 2 espaços em branco antes de cada comando que deve estar dentro do bloco.

# Comandos condicionais

- O principal comando condicional é o **if**, cuja sintaxe é:  
`if expressão relacional ou lógica :`  
comandos executados se a expressão é verdadeira
- Os comandos são executados somente se a expressão relacional/lógica for verdadeira.

# Comandos condicionais

O programa abaixo determina se um valor é par.

```
a = int(input())  
if a%2 == 0:  
    print("O número digitado é par")
```

# Comandos condicionais

- Uma variação do comando **if** é o **if/else**, cuja sintaxe é:

```
if expressão relacional ou lógica :  
    comandos executados se a expressão é verdadeira  
else :  
    comandos executados se a expressão é falsa
```

# Comandos condicionais

Exemplo: Determinando o menor de dois números:

```
a = int(input(" Digite um número: "))
b = int(input(" Digite um número: "))
if a < b:
    print("O menor número é: ", a )
else:
    print("O menor número é: ", b )
```

# Comandos condicionais

- Note que o **if** é um comando, e como tal pode aparecer dentro do bloco de comandos de outro **if**.

Exemplo: Usando apenas **operadores relacionais e aritméticos**, vamos escrever um programa que lê um número e verifica em qual dos seguintes casos o número se enquadra:

- Par e menor que 100.
- Par e maior ou igual a 100.
- Ímpar e menor que 100.
- Ímpar e maior ou igual a 100.

# Comandos condicionais

```
a = int(input("Digite um número:"))
if (a % 2 == 0):    #<— Se número for par, executa bloco abaixo
    if (a<100):
        print("O número é par e menor do que 100")
    else:
        print("O número é par e maior ou igual que 100")
else:              #<— Se número for ímpar, executa bloco abaixo
    if (a<100):
        print("O número é ímpar e menor do que 100")
    else:
        print("O número é ímpar e maior ou igual que 100")
```

Se você pudesse usar operadores lógicos, como você poderia refazer este programa?



# Comandos condicionais

```
print("Digite um número:")
a = int(input())
if (a % 2 == 0) and (a < 100):
    print("O número é par e menor do que 100")
if (a % 2 == 0) and (a >= 100):
    print("O número é par e maior ou igual que 100")
if (a % 2 != 0) and (a < 100):
    print("O número é ímpar e menor do que 100")
if (a % 2 != 0) and (a >= 100):
    print("O número é ímpar e maior ou igual que 100")
```

# Comandos condicionais

Lembre-se que o que define a qual bloco de comandos um comando pertence é a sua indentação!

```
if (cond1):  
    if (cond2):  
        comando1  
else:  
    comando2
```

Quando o **comando2** é executado?

# Comandos condicionais

Lembre-se que o que define a qual bloco de comandos um comando pertence é a sua indentação!

```
if (cond1):  
    if (cond2):  
        comando1  
else:  
    comando2
```

Quando o **comando2** é executado?

Resposta: quando **cond1** for falsa.

# Comandos condicionais

```
if (cond1):  
    if (cond2):  
        comando1  
    else:  
        comando2
```

Quando o comando2 é executado?

# Comandos condicionais

```
if (cond1):  
    if (cond2):  
        comando1  
    else:  
        comando2
```

Quando o comando2 é executado?

Resposta: quando a cond1 for verdadeira e cond2 for falsa.

# Comandos condicionais

```
if (cond1):  
    if (cond2):  
        comando1  
    else:  
        comando2  
else:  
    if (cond3):  
        comando3  
    else:  
        comando4
```

Quando o **comando4** é executado?

# Comandos condicionais

```
if (cond1):  
    if (cond2):  
        comando1  
    else:  
        comando2  
else:  
    if (cond3):  
        comando3  
    else:  
        comando4
```

Quando o **comando4** é executado?

Resposta: quando a **cond1** for falsa e **cond3** for falsa.

# Comandos condicionais

```
a = 5
if (a > 3):
    if (a < 7):
        print("a")
    else:
        if (a > -10):
            print("b")
        else:
            print("c")
```

O que será impresso?



# Comandos condicionais

```
a = -12
if (a > 3):
    if (a < 7):
        print("a")
else:
    if (a > -10):
        print("b")
    else:
        print("c")
```

O que será impresso?

# Comandos condicionais

```
a = 9
if (a > 3):
    if (a < 7):
        print("a")
    else:
        if (a > -10):
            print("b")
        else:
            print("c")
```

O que será impresso?

# Exercícios

A solução abaixo está correta para classificar um número como par e menor que 100, ou par e maior ou igual a 100, etc, como no exemplo visto anteriormente?

```
print("Digite um número:")
a = int(input())
if (a % 2 == 0) and (a < 100):
    print("O número é par e menor do que 100")
else:
    if (a >= 100):
        print("O número é par e maior ou igual que 100")
if (a % 2 != 0) and (a < 100):
    print("O número é ímpar e menor do que 100")
else:
    if (a >= 100):
        print("O número é ímpar e maior ou igual que 100")
```

# Exercícios

- Escreva um programa que lê um número inteiro do teclado e imprime "SIM" se o número for par e maior do que 10, ou for ímpar e menor do que 50. Caso contrário o programa deve imprimir "NAO".

# Exercícios

- Escreva um programa lê três números e imprime o maior deles.

# Exercícios

- Escreva um programa lê três números distintos e os imprime em ordem (ordem decrescente).