



# Estruturas condicionais

ICC - Introdução à Ciência da Computação

UFCG/UASC

Prof. Roberto M. de Faria

# Estruturas condicionais

## Conteúdo:

- Operadores e expressões relacionais
- Operadores lógicos expressões lógicas
- O tipo bool
- O comando condicional **if**
- O comando condicional **if-else**
- O comandos condicionais aninhados
- A instrução condicional **if-elif-else**

# Estruturas condicionais

- Tudo que aprendemos até agora está no conjunto das estruturas **sequenciais**;
- **Estrutura Sequencial:** *conjunto de instruções executadas em sequência linear, de cima para baixo*;
- *Além da estrutura sequencial, existem as estruturas condicionais e as estruturas de repetição.*
  - **Condicionais:** *desviam o código executado de acordo com condições;*
  - **Repetição:** *permitem que conjuntos de instruções sejam executados mais de uma vez de acordo com condições.*

# Operadores relacionais

Símbolo	Operador	Descrição
<	Menor que	A condição é verdadeira se o lado esquerdo for menor que o lado direito
>	Maior que	A condição é verdadeira se o lado esquerdo for maior que o lado direito
<=	Menor ou igual a	A condição é verdadeira se o lado esquerdo for menor ou igual ao lado direito
>=	Maior ou igual a	A condição é verdadeira se o lado esquerdo for maior ou igual ao lado direito
==	Igual a	A condição é verdadeira se o lado esquerdo for igual ao lado direito
!=	Diferente de	A condição é verdadeira se o lado esquerdo for diferente do lado direito

# Expressões relacionais

- Expressões que utilizam operadores relacionais e são avaliadas com os valores True (**verdadeiro**) ou False (**falso**)

Exemplos:

```
a = 5
```

```
b = 10
```

```
a == b           # valor False
```

```
a <= b           # valor True
```

```
10 > x%10        # valor True
```

```
a > 77//10       # valor False
```

```
a + b < 30//2    # valor False
```

# Operadores lógicos

Símbolo	Operador	Descrição
<b>and</b>	E	O operador e é um operador binário que avalia como verdadeiro (True), se e somente se os lados esquerdo e direito forem verdadeiros
<b>or</b>	Ou	O operador ou é um operador binário avaliado como verdadeiro (True) se pelo menos um de seus lados for verdadeiro
<b>not</b>	Não	O operador não é uma negação unária, é seguida por algum valor. É avaliado como verdadeiro (True) se esse valor for falso (False) e vice-versa

# Expressões lógicas

## Expressões Lógicas em Python:

`A > B`

`D <= C`

`X == Y`

`X >= 0 and X < 100`

`Opcao == "C" or Salario < 100`

Todas terão como resposta: **VERDADEIRO** ou **FALSO**

# Precedência de operadores

Da maior precedência para a menor (de cima para baixo):

- (operador unário)	
(...(...))	dos mais internos para os mais externos
**	
*, /, //, %	o que primeiro aparecer da esquerda para a direita
+ e -	o que primeiro aparecer da esquerda para a direita
<, <=, >, >=, ==, !=	o que primeiro aparecer da esquerda para a direita
not	
and	
or	



# O tipo **bool**

- Além dos tipos que já conhecemos na linguagem Python: **int** para inteiros, **float** para reais de ponto flutuante e **str** para cadeia de caracteres ou strings, temos agora o tipo **bool** para valores lógicos (**True** ou **False**)
- Uma variável do tipo **bool** só pode armazenar valores **True** ou **False**

## Exemplos de uso:

```
ehvalido = True
resultBool = a < b
val_logico = bool(inteiro)
n_eh_par = not(n%2)
falso = str("")
```

# Estruturas condicionais

---

- E em Python?
- Quais comandos/estruturas podemos usar para **codificar condições** em nossos algoritmos?

# Estruturas condicionais

---

- E em Python?
- Quais comandos/estruturas podemos usar para **codificar condições** em nossos algoritmos?
- Estruturas condicionais
  - if
  - if-else
  - If's aninhados
  - if-elif-else

# O comando `if`

*if condição:*

*bloco do verdadeiro:*

uma ou várias instruções que são executadas  
se a condição for avaliada como True

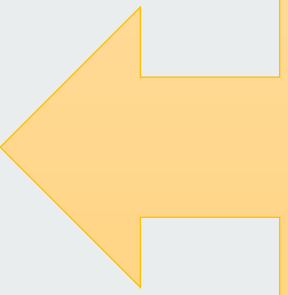
# O comando if

```
...  
if condição:  
    comando1  
    comando2  
    comando3  
    ...  
    comandoN  
...
```

# O comando if

```
...  
if condição:  
    comando1  
    comando2  
    comando3  
    ...  
    comandoN
```

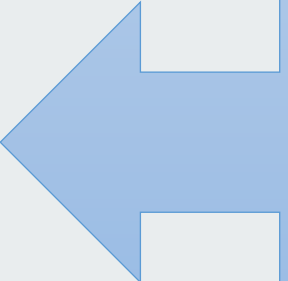
...



Comandos a  
serem executados  
**caso** a condição  
seja **VERDADEIRA**

# O comando if

```
...  
if condição:  
    comando1  
    comando2  
    comando3  
    ...  
    comandoN  
...
```



Cuidado com a  
**INDENTAÇÃO**

# O comando if

```
...  
if condição:  
    comando1  
    comando2  
    comando3  
    ...  
    comandoN  
...
```

Cuidado com a  
**INDENTAÇÃO**

Comandos **dentro de um bloco if** precisam estar com um **TAB** de espaçamento horizontal (indentação) em relação ao comando if



# O comando if

```
...  
if condição:  
    comando1  
    comando2  
    comando3  
    ...  
    comandoN  
...
```

Cuidado com a  
**INDENTAÇÃO**

O primeiro comando  
escrito **sem o TAB** de  
espaçamento será  
interpretado como **fora do  
bloco if**

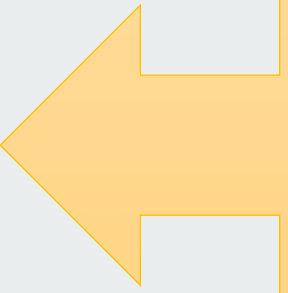
# O comando if

...

```
if condição:  
    comando1  
    comando2  
    comando3  
    ...  
    comandoN
```

...

Como seriam essas condições?



Comandos a serem executados caso a condição seja VERDADEIRA

# O comando if

...

**if** **condição**:

comando1

comando2

comando3

...

comandoN

...

Expressões Lógicas

Comandos a  
serem executados  
**caso** a condição  
seja **VERDADEIRA**

# O comando if

Exemplo:

```
...  
if A > B:  
    print("A eh maior que B")  
...
```

# O comando if

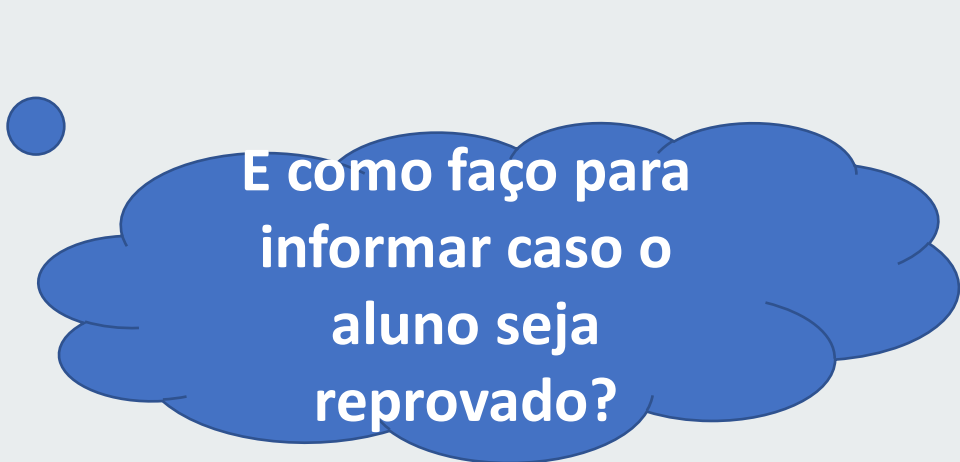
## Exemplo:

```
n1 = float(input("Digite Nota 1: "))
n2 = float(input("Digite Nota 2: "))
media = (n1+n2)/2
if media >= 5:
    print("Aluno APROVADO")
```

# O comando if

Exemplo:

```
...  
if media >= 5:  
    print("Aluno APROVADO")  
...
```



E como faço para  
informar caso o  
aluno seja  
reprovado?

# O comando if

Uma alternativa seria:

```
n1 = float(input("Digite Nota 1: "))
n2 = float(input("Digite Nota 2: "))
media = (n1+n2)/2
if media >= 5:
    print("Aluno APROVADO")
if media < 5:
    print("Aluno REPROVADO")
```

# O comando if

Uma alternativa seria:

```
n1 = float(input("Digite Nota 1: "))
n2 = float(input("Digite Nota 2: "))
media = (n1 + n2) / 2
if media >= 5:
    print("Aluno APROVADO")
if media < 5 :
    print("Aluno REPROVADO")
```

**Tem uma maneira melhor**



# O comando if-else

if condição:

*bloco do verdadeiro:*

uma ou várias instruções que são executadas  
se a condição for avaliada como True

else:

*bloco do falso:*

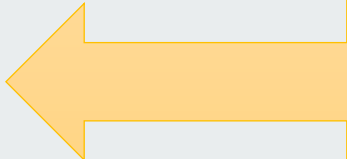
uma ou várias instruções que são executadas  
se a condição for avaliada como False

# O comando if-else

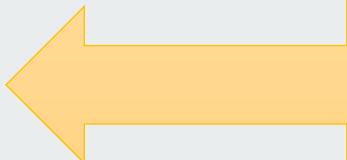
```
...  
if condição :  
    comandoV1  
    comandoV2  
    ...  
    comandoVN  
else:  
    comandoF1  
    comandoF2  
    ...  
    comandoFN  
...
```

# O comando if-else

```
...  
if condição :  
    comandoV1  
    comandoV2  
    ...  
    comandoVN  
else:  
    comandoF1  
    comandoF2  
    ...  
    comandoFN  
...
```



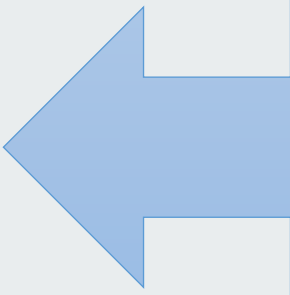
Comandos a serem executados **caso** a condição seja **VERDADEIRA**



Comandos a serem executados **caso** a condição seja **FALSA**

# O comando if-else

```
...  
if condição:  
    comandoV1  
    comandoV2  
    ...  
    comandoVN  
else:  
    comandoF1  
    comandoF2  
    ...  
    comandoFN  
...
```



Mais uma vez, é  
preciso que se  
tome cuidado com  
a **INDENTAÇÃO**

# O comando if-else

```
n1 = float(input("Digite Nota 1: "))
n2 = float(input("Digite Nota 2: "))
media = (n1 + n2) / 2
if media >= 5:
    print("Aluno APROVADO")
else:
    print("Aluno REPROVADO")
```

# O comando if-else

```
...  
if media >= 5:  
    print("Aluno APROVADO")  
else:  
    print("Aluno REPROVADO")  
...
```

# Comandos condicionais aninhados



- Podemos ter blocos **if** e/ou **if-else** combinados, um dentro do outro – aninhados.
- Servem para representar algoritmos que possuem estruturas condicionais em sequência.

# Comandos condicionais aninhados

...

**if** condição1:

    comando1           // 0-n comandos

**if** condição2:

        comando2       // 0-m comandos

        ...

...



# Comandos condicionais aninhados

Em Python:

...

```
if condição1:
```

```
    comando1                // 0-n comandos
```

```
    if condição2:
```

```
        comando2            // 0-m comandos
```

```
        ...
```

```
...
```

A indentação é obrigatória!

# Comandos condicionais aninhados

```
...  
if condição1:  
    if condição2:  
        comando1  
    ...  
else:  
    if condição3:  
        comando2  
    ...  
...
```

# Comandos condicionais aninhados

...

```
if condição1:
```

```
    if condição2:
```

```
        comando1
```

```
    ...
```

```
else:
```

```
    if condição3:
```

```
        comando2
```

```
    ...
```

```
...
```

A indentação é obrigatória!

# Comandos condicionais aninhados

```
if X1 >= X2:
    comando1
else:
    if X1%2 == 0:
        comando1
    else:
        comando3
...
```

# Comandos condicionais aninhados

```
if N > 50:
    print("É MAIOR")
else:
    if N < 50:
        print("É MENOR")
    else:
        print("É IGUAL")
```

# O comando if-else

```
opc = int(input("Digite uma opção entre 1 e 3"))
```

```
if opc == 1:
```

```
    print("Vc selecionou opção 1")
```

```
else:
```

```
    if opc == 2:
```

```
        print("Vc selecionou opção 2")
```

```
    else:
```

```
        if opc == 3:
```

```
            print("Vc selecionou opção 3")
```

```
        else:
```

```
            print("Opção invalida! ")
```

# O comando if-else

```
opc = int(input("Digite uma opção entre 1 e 3"))
```

```
if opc == 1:
```

```
    print("Vc selecionou opção 1")
```

```
else:
```

```
    if opc == 2:
```

```
        print("Vc selecionou opção 2")
```

```
    else:
```

```
        if opc == 3:
```

```
            print("Vc selecionou opção 3")
```

```
        else:
```

```
            print("Opção invalida! ")
```

Código começa a deslocar-se muito. Imagine 9 opções?!

# O comando if-else

```
opc = int(input('Digite uma opção entre 1 e 3'))
```

```
if opc == 1:
```

```
    print('Vc selecionou opção 1')
```

```
else:
```

```
    if opc == 2:
```

```
        print('Vc selecionou opção 2')
```

```
    else:
```

```
        if opc == 3:
```

```
            print('Vc selecionou opção 3')
```

```
        else:
```

```
            print('Opção invalida!')
```

Código começa a deslocar-se muito. Imagine 9 opções?!

**Tem uma maneira melhor**



# O comando **if-elif-else**

- Solução para o problema de múltiplos blocos **if** aninhados;
- **elif** substitui um par **else if** sem criar outro nível de estrutura;
- Evita deslocamentos desnecessários à direita;
- Simplifica o código e melhora a redigibilidade e a legibilidade.

# O comando if-elif-else

```
if condição_1:  
    bloco verdadeiro_1:  
        uma ou várias instruções que são executadas  
        se a condição_1 for avaliada como True  
elif condição_2:  
    bloco do verdadeiro 2:  
        uma ou várias instruções que são executadas  
        se a condição_2 for avaliada como True  
...  
elif condição_n:  
    bloco do verdadeiro n:  
        uma ou várias instruções que são executadas  
        se a condição_n for avaliada como True  
else:  
    bloco do falso (opcional):  
        uma ou várias instruções que são executadas  
        se todas as condições forem avaliadas como False
```

# O comando if-elif-else

...

**if** condição1:

comando1

**elif** condição2:

comando2

**elif** condição3:

comando3

**else:**

comando4

...

# O comando if-elif-else

```
...  
if condição1:  
    comando1  
elif condição2:  
    comando2  
elif condição3:  
    comando3  
else:  
    comando4  
...
```

Elif proporciona menos  
indentação obrigatória  
=  
menos deslocamento  
horizontal

# O comando if-elif-else

```
opc = int(input("Digite uma opção entre 1 e 3"))
```

```
if opc == 1:
    print("Vc selecionou opção 1")
elif opc == 2:
    print("Vc selecionou opção 2")
elif opc == 3:
    print("Vc selecionou opção 3")
else:
    print("Opção invalida! ")
```

# O comando if-elif-else

```
if N > 50:  
    print("É MAIOR")  
elif N < 50:  
    print("É MENOR")  
else:  
    print("É IGUAL")
```

# Mais exemplos

- Seleção simples ( **if** )

```
a = float(input("Digite a: "))  
b = float(input('Digite b: '))  
if a > b:  
    print("a eh maior do que b")
```

# Mais exemplos

- Seleção composta ( **if-else** )

```
n1 = float(input("Digite sua primeira nota: "))
n2 = float(input("Digite sua segunda nota: "))
media = (n1+n2)/2.0

if media >= 5:
    print("Sua media eh %.1f e voce esta APROVADO! " % media)
else:
    print("Sua media eh %.1f e voce esta REPROVADO! " % media)
```



# Mais exemplos

- Seleção múltipla ( **if-elif-else** )

```
# Entrada : Ler um numero
a = float(input("Digite um numero : "))
# Processamento e saída
if a > 0:
    print("POSITIVO")
elif a < 0 :
    print("NEGATIVO")
else:
    print("NULO")
```

# Mais exemplos

- Escreva um programa que verifique se um caractere digitado é ou não uma vogal.

```
# Entrada : Ler um caractere
c = input("Digite um caractere: ")
# Processamento e saída
if (c == "a" or c == "e" or c == "i" or
    c == "o" or c == "u") :
    print("Eh vogal ")
else:
    print("Nao eh vogal! ")
```

# Mais exemplos

- Escreva um Programa que pergunte ao usuário que horas são (apenas a hora) e posteriormente mostre na tela “Bom dia”, “Boa tarde” ou “Boa noite”.

```
# Entrada
hora = int(input("Que horas são? [0-23] "))
# Processamento e saída
if hora > 3 and hora < 12:
    print("Bom dia")
elif hora >= 12 and hora < 18:
    print("Boa tarde")
else:
    print("Boa noite")
```

# Mais exemplos

- Escreva um Programa que a **hora atual do sistema** e posteriormente mostre na tela “Bom dia”, “Boa tarde” ou “Boa noite”.

```
exemplo-hora-sistema.py x
1 from datetime import datetime
2 # Entrada
3 datahora_atual = datetime.now()
4 hora = int(datahora_atual.strftime('%H'))
5 # Processamento e saída
6 if hora > 3 and hora < 12:
7     print('Bom dia')
8 elif hora >= 12 and hora < 18:
9     print('Boa tarde')
10 else:
11     print('Boa noite')
```

# Referências e créditos

---

- Estes slides foram compilados a partir dos trabalhos:
  - do Prof. Maxwell Guimarães na disciplina Introdução à Programação/UASC/UFCG
  - do Prof. Roberto Faria nas disciplinas Introdução à Ciência da Computação e Introdução à Programação/UASC/UFCG