

---

# Algoritmos e Técnicas de Programação

## Tema: Introdução a linguagem Python

Professor Wendell Bento Geraldes

Instituto Federal de Goiás  
campus Luziânia  
<wendell.geraldes@ifg.edu.br>



## 1 Estruturas de decisão no Python



# Estruturas de decisão no Python



## Condições lógicas

Python suporta as condições lógicas usuais da matemática:

- É igual a:  $a == b$
- Não é igual a:  $a != b$
- Menor que:  $a < b$
- Menor ou igual a:  $a \leq b$
- Maior que:  $a > b$
- Maior ou igual a:  $a \geq b$

Essas condições podem ser usadas de várias maneiras, mais comumente em "instruções if" e loops. Uma "instrução if" ela é escrita usando a palavra-chave **if** (W3SCHOOLS, 2024).



# Estruturas de decisão no Python

## Exemplo

```
1 a = 33
2 b = 200
3 if b > a:
4     print("b é maior que a")
```

Neste exemplo, usamos duas variáveis, a e b , que são usadas como parte da instrução if para testar se b é maior que a . Como a é 33 e b é 200 , sabemos que 200 é maior que 33 e, portanto, imprimimos na tela que "b é maior que a".



# Estruturas de decisão no Python

## Recuo

Python depende de recuo (espaço em branco no começo de uma linha) para definir o escopo do código. Outras linguagens de programação geralmente usam chaves para esse propósito.

```
5 a = 33
6 b = 200
7 if b > a:
8     print("b é maior que a") #aqui vai dar erro
```



# Estruturas de decisão no Python

## Elif

A palavra-chave **elif** é a maneira do Python dizer "se as condições anteriores não forem verdadeiras, então tente esta condição".

```
9 a = 33
10 b = 33
11 if b > a:
12     print("b é maior que a")
13 elif a == b:
14     print("a e b são iguais")
```

Neste exemplo, a é igual a b , então a primeira condição não é verdadeira, mas a condição elif é verdadeira, então imprimimos na tela que "a e b são iguais".



else

Você também pode ter um **else** sem o **elif**:

```
15 a = 33
16 b = 33
17 if b > a:
18     print("b é maior que a")
19 else a == b:
20     print("a e b são iguais")
```





# Estruturas de decisão no Python

## Modo curto

Se você tiver apenas uma instrução para executar, você pode colocá-la na mesma linha da instrução `if`.

```
21 if a > b: print("a is greater than b")
```

## Se..Senão

Se você tiver apenas uma instrução para executar, uma para `if` e uma para `else`, você pode colocar tudo na mesma linha.

```
22 a = 2  
23 b = 330  
24 print("A") if a > b else print("B")
```

Essa técnica é conhecida como Operadores Ternários ou Expressões Condicionais .



# Estruturas de decisão no Python

## Se..Senão

Você também pode ter várias instruções **else** na mesma linha:

```
25 a = 330
26 b = 330
27 print("A") if a > b else print("=") if a == b else print("B")
```

## and

A palavra-chave **and** é um operador lógico e é usada para combinar instruções condicionais:

```
28 a = 200
29 b = 33
30 c = 500
31 if a > b and c > a:
32     print("Ambas as condições são verdadeiras")
```



## or

A palavra-chave **or** é um operador lógico e é usada para combinar instruções condicionais:

```
33 a = 200
34 b = 33
35 c = 500
36 if a > b or a > c:
37     print("Pelo menos uma das condições é verdadeira")
```



# Estruturas de decisão no Python

---

## not

A palavra-chave **not** é um operador lógico e é usada para reverter o resultado da instrução condicional:

```
38 a = 33
39 b = 200
40 if not a > b:
41     print("a NÃO é maior que b")
```



## SE aninhado

Você pode ter instruções **if** dentro de instruções **if**, o que é chamado de instruções aninhadas **if**.

```
42 x = 41
43
44 if x > 10:
45     print("Acima de dez,")
46     if x > 20:
47         print("e também acima de 20!")
48     else:
49         print("mas não acima de 20.")
```



## A declaração pass

As instruções if não podem estar vazias, mas se por algum motivo você tiver uma instrução if sem conteúdo, insira a palavra **pass** para evitar erros.

```
50 a = 33
51 b = 200
52
53 if b > a:
54     pass
```



 W3SCHOOLS. **Python**. W3C, 2024. Disponível em: <<https://www.w3schools.com>>.

