

# Operadores Aritméticos e Estruturas Condicionais



# OPERADORES ARITMÉTICOS

- +** Adição
- Subtração
- \*** Multiplicação
- /** Divisão
- //** Divisão Inteiro
- %** Módulo (*resto da divisão*)
- \*\*** Exponenciação (*ao cubo*)

```
> 5 % 2  
< 1  
-----  
> 6 % 2  
< 0
```

```
> 2 ** 3  
< 8  
-----  
> 3 ** 2  
< 9
```

# Operadores Aritméticos

**Operadores aritméticos** são símbolos que permitem realizar operações matemáticas básicas.

Em Python, esses operadores são usados para manipular valores numéricos.

operação	símbolo	exemplo	ordem de x e y faz diferença?
<b>Soma</b>	+	$x + y$	<b>não</b>
<b>Subtração</b>	-	$x - y$	<b>sim</b>
<b>Multiplicação</b>	*	$x * y$	<b>não</b>
<b>Divisão</b>	/	$x / y$	<b>sim</b>

# Operadores Aritméticos

## Soma (+)

```
a = 5  
b = 3  
resultado = a + b # resultado é 8
```

## Subtração (-)

```
a = 10  
b = 4  
resultado = a - b # resultado é 6
```

## Multiplicação (\*)

```
a = 7  
b = 3  
resultado = a * b # resultado é 21
```

## Divisão (/)

```
a = 10  
b = 2  
resultado = a / b # resultado é 5.0
```

# Operadores Aritméticos

## Divisão Inteira (//)

```
a = 10
b = 3
resultado = a // b # resultado é 3
```

O operador // realiza a divisão inteira, descartando o resto.

## Módulo (% Resto da Divisão)

```
a = 10
b = 4
resultado = a % b # resultado é 2
```

## Exponenciação (\*\*)

```
a = 2
b = 3
resultado = a ** b # resultado é 8
```

O operador \*\* é usado para elevar um valor a uma potência.

# Operadores de atribuição

Os operadores de atribuição são usados para atribuir valores a variáveis. Além do simples operador de atribuição =, Python oferece operadores compostos que combinam uma operação com uma atribuição.

```
x = 10 # x recebe o valor 10
```



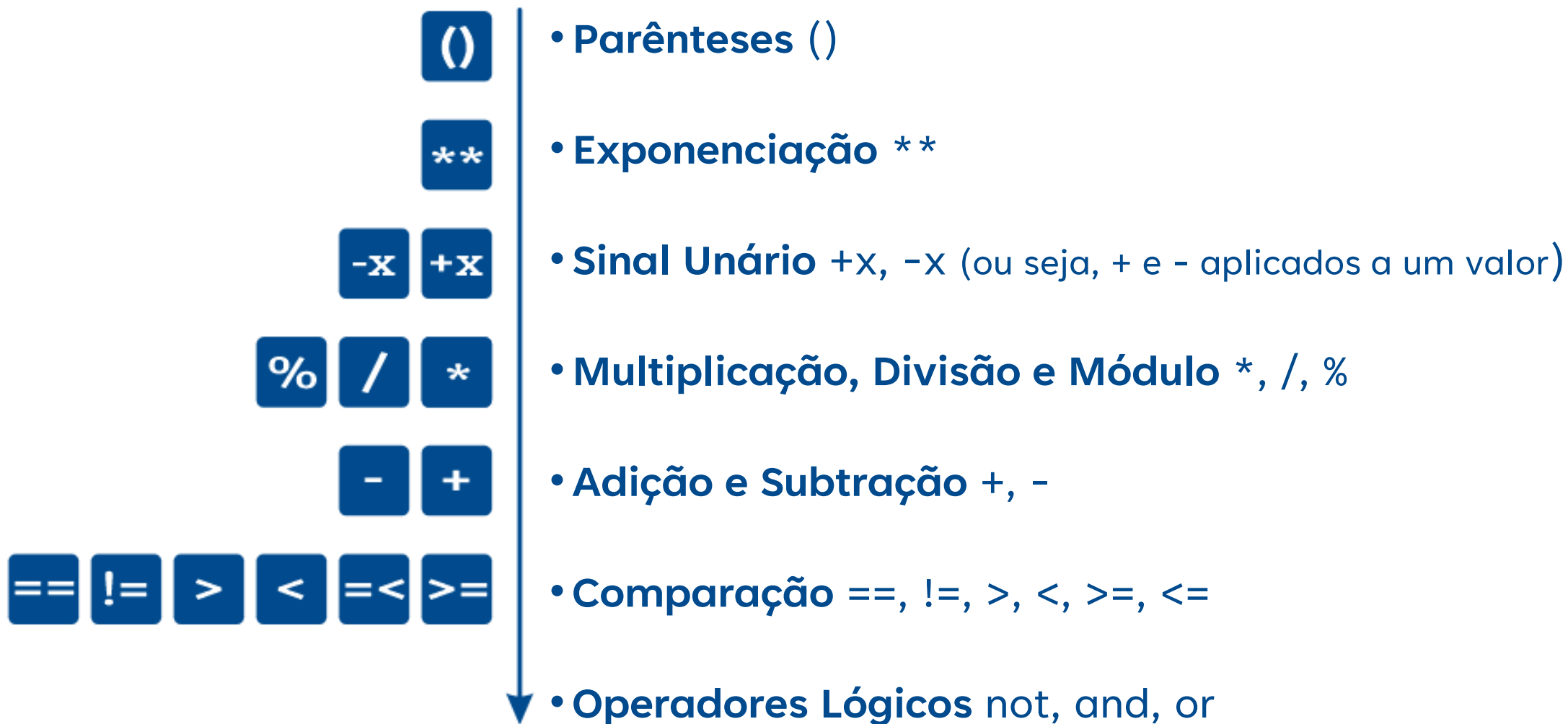
```
x = 5  
x += 3 # equivalente a x = x + 3; x agora é 8
```

```
x = 7  
x *= 2 # equivalente a x = x * 2; x agora é 14
```

```
x = 10  
x -= 4 # equivalente a x = x - 4; x agora é 6
```

```
x = 20  
x /= 4 # equivalente a x = x / 4; x agora é 5.0
```

# Precedência dos Operadores



# Precedência dos Operadores

```
resultado = (2 + 3) * 4 # resultado é 20
```

```
resultado = 5 + 3 - 2 # resultado é 6
```

```
resultado = -5 + 3 # resultado é -2
```

```
resultado = 5 > 3 # resultado é True
```

```
resultado = 5 == 5 # resultado é True
```

```
resultado = 10 / 2 * 3 # resultado é 15.0
```

```
resultado = 10 // 3 # resultado é 3
```

```
resultado = 10 % 3 # resultado é 1
```

```
resultado = True and False # resultado é False
```

```
resultado = not True # resultado é False
```

```
resultado = True or False # resultado é True
```

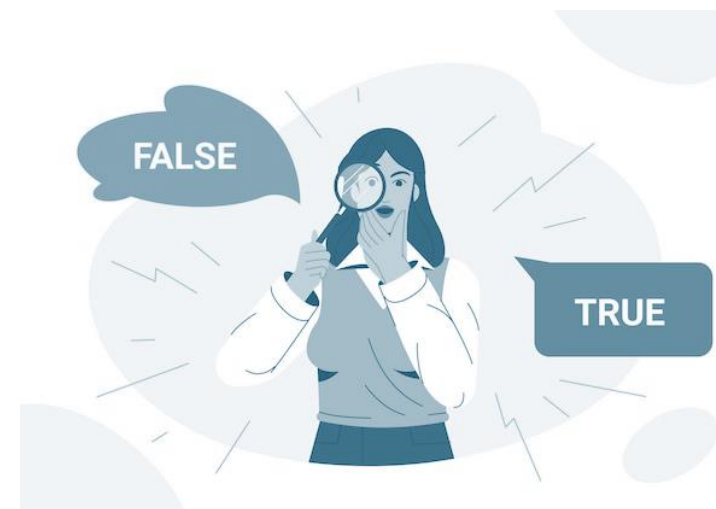


# Operadores Relacionais

Os **operadores relacionais** são usados para comparar dois valores e retornar um valor booleano (**True** ou **False**). Eles ajudam a determinar a relação entre dois operandos.

**==** Igualdade (valor)  
**!=** Diferente/não igual (valor)

**>** Maior  
**<=** Maior ou igual  
**<** Menor  
**>=** Menor ou igual



# Operadores Relacionais

Os **operadores relacionais** são usados para comparar dois valores e retornar um valor booleano (**True** ou **False**). Eles ajudam a determinar a relação entre dois operandos.

```
a = 5  
b = 5  
resultado = (a == b) # resultado é True
```

```
a = 5  
b = 3  
resultado = (a != b) # resultado é True
```

```
a = 7  
b = 3  
resultado = (a > b) # resultado é True
```

```
a = 5  
b = 8  
resultado = (a < b) # resultado é True
```

```
a = 5  
b = 5  
resultado = (a >= b) # resultado é True
```

```
a = 3  
b = 5  
resultado = (a <= b) # resultado é True
```

# Estruturas Condicionais – if, else

## Objetivo

Aprender a usar as estruturas condicionais if e else em Python para controlar o fluxo do seu programa com base em condições.

## O que são Condicionais?

Condicionais permitem que você execute diferentes blocos de código dependendo de certas condições. Com o if e else, você pode criar ramificações na execução do seu código.



# Principais tipos de estruturas condicionais



# Estruturas Condicionais – if, else

## Objetivo

Aprender a usar as estruturas condicionais if e else em Python para controlar o fluxo do seu programa com base em condições.

if condição

true

else

false

```
if condição:
    # código a ser executado se a condição for verdadeira
else:
    # código a ser executado se a condição for falsa
```

# Condicional Simples (if)

A **estrutura condicional simples** é a forma mais básica de condicional. Ela executa um bloco de código se a condição **especificada for verdadeira**.

Simples

if condição

true

```
if condição:  
    # código a ser executado se a condição for verdadeira
```

```
condicional.py ×  
1  idade = 20  
2  if idade >= 18:  
3      print("Você é maior de idade.")
```

# Condicional Composta (if-else)

A **condicional composta** inclui uma cláusula **else**, que fornece um bloco de código alternativo a ser executado se a condição if não for verdadeira.

Composta

if condição

true

else

false

```
if condição:
    # código a ser executado se a condição for verdadeira
else:
    # código a ser executado se a condição for falsa
```

condicional.py ×

```
1  idade = 16
2  if idade >= 18:
3      print("Você é maior de idade.")
4  else:
5      print("Você é menor de idade.")
```

# Estruturas Condicionais – if, else

Neste exemplo, a variável `idade` é comparada a 18. Se `idade` for maior ou igual a 18, a mensagem "Você é maior de idade." será impressa. Caso contrário, a mensagem "Você é menor de idade." será impressa.

## Como Funciona

- 1.**Condição:** É a expressão que é avaliada como True (verdadeira) ou False (falsa).
- 2.**Bloco if:** O código dentro do bloco if é executado se a condição for verdadeira.
- 3.**Bloco else:** O código dentro do bloco else é executado se a condição for falsa.

```
if.py x
1  idade = 18
2
3  if idade >= 18:
4      print("Você é maior de idade.")
5  else:
6      print("Você é menor de idade.")
7
```



# Condicional Encadeada (if-elif-else)

A condicional **encadeada** permite verificar várias condições diferentes, em sequência, e executar o código correspondente à primeira condição verdadeira.

Encadeada

if condição

true

elif condição

true

else

false

```
condicional.py x
1  nota = 85
2
3  if nota >= 90:
4      print("Excelente")
5  elif nota >= 70:
6      print("Bom")
7  elif nota >= 50:
8      print("Suficiente")
9  else:
10     print("Insuficiente")
```

```
if condição1:
    # código a ser executado se condição1 for verdadeira
elif condição2:
    # código a ser executado se condição2 for verdadeira
elif condição3:
    # código a ser executado se condição3 for verdadeira
else:
    # código a ser executado se nenhuma das condições acima for verdadeira
```

# Estruturas Condicionais – elif

```
elif.py x
1  nota = 75
2
3  if nota >= 90:
4      print("Excelente")
5  elif nota >= 70:
6      print("Bom")
7  elif nota >= 50:
8      print("Suficiente")
9  else:
10     print("Insuficiente")
```

O elif (abreviação de "else if") permite verificar múltiplas condições.

**Sintaxe do if, elif, e else:**

```
if condição1:
    # código para condição1
elif condição2:
    # código para condição2
else:
    # código se nenhuma condição acima for verdadeira
```

**Explicação:**

**nota** >= 90: Primeira condição. Se verdadeira, imprime "Excelente".

**nota** >= 70: Segunda condição. Se a primeira for falsa e essa for verdadeira, imprime "Bom".

**nota** >= 50: Terceira condição. Se as duas anteriores forem falsas e essa for verdadeira, imprime "Suficiente".

# Operadores lógicos

```
operadores.py x
1  idade = 16
2  carta_conducao = False
3
4  if idade >= 18 and carta_conducao:
5      print("Você pode dirigir.")
6  else:
7      print("Você não pode dirigir.")
```

```
operadores.py x
1  hora = 15
2  tem_ingresso = True
3
4  if hora < 18 or tem_ingresso:
5      print("Você pode entrar no evento.")
6  else:
7      print("Você não pode entrar no evento.")
```

## Explicação:

`idade >= 18 and carta_conducao`: A condição precisa ser verdadeira para ambas as partes. A pessoa deve ter 18 anos ou mais e possuir a carta de condução.

## Operadores Lógicos

**and**: Retorna True se ambas as condições forem verdadeiras.

**or**: Retorna True se pelo menos uma das condições for verdadeira.

**not**: Inverte o valor lógico da condição.

`hora < 18 or tem_ingresso`: Se a hora for antes das 18 ou a pessoa tiver ingresso, ela pode entrar.

# Condicional Aninhada

Além dessas três formas principais, as condicionais podem ser aninhadas dentro de outras condicionais, permitindo verificar múltiplas condições de forma mais complexa.

## Condições Aninhadas

if condição1

if condição2

true

else

false

else

false

```
if condição1:
    if condição2:
        # código a ser executado se condição1 e condição2 forem verdadeiras
    else:
        # código a ser executado se condição1 for verdadeira e condição2 for falsa
else:
    # código a ser executado se condição1 for falsa
```

```
condicional.py ×
1  idade = 25
2  tem_documento = True
3
4  if idade >= 18:
5      if tem_documento:
6          print("Você pode votar.")
7      else:
8          print("Você não tem o documento necessário para votar.")
9  else:
10     print("Você é menor de idade e não pode votar.")
```

# Prática



# Prática - Operadores de Comparação

```
x = 5
if x == 5:
    print("x é igual a 5")
```

**== (igual a):** Verifica se dois valores são iguais.

```
y = 10
if y != 5:
    print("y é diferente de 5")
```

**!= (diferente de):** Verifica se dois valores são diferentes.

```
a = 7
if a > 5:
    print("a é maior que 5")
```

**> (maior que):** Verifica se o valor da esquerda é maior que o valor da direita.

```
b = 3
if b < 5:
    print("b é menor que 5")
```

**< (menor que):** Verifica se o valor da esquerda é menor que o valor da direita.

```
c = 10
if c >= 10:
    print("c é maior ou igual a 10")
```

**>= (maior ou igual a):** Verifica se o valor da esquerda é maior ou igual ao valor da direita.

```
d = 4
if d <= 5:
    print("d é menor ou igual a 5")
```

**<= (menor ou igual a):** Verifica se o valor da esquerda é menor ou igual ao valor da direita.

# EXERCÍCIOS



## Exercício 1: Condicional Simples

**Objetivo:** Verificar se uma pessoa pode entrar em uma sessão de cinema com base na idade.

**Descrição:** Escreva um programa que peça a idade do usuário e imprima "Você pode assistir ao filme" se a idade for 12 anos ou mais. Caso contrário, imprima "Você não pode assistir ao filme".



# Exercício

## Exercício 1:

```
exercicios.py ×  
1  idade = int(input("Digite sua idade: "))  
2  
3  if idade >= 12:  
4      print("Você pode assistir ao filme.")
```

## Exercício 2: Condicional Composta

**Objetivo:** Verificar se um número é positivo ou negativo.

**Descrição:** Escreva um programa que peça um número do usuário e imprima "Número positivo" se o número for maior que zero. Caso contrário, imprima "Número negativo".

# Exercício

## Exercício 2:

 exercicios.py x

```
1  numero = float(input("Digite um número: "))
2
3  if numero > 0:
4      print("Número positivo")
5  else:
6      print("Número negativo")
```

## Exercício 03: Condicional Encadeada

**Objetivo:** Classificar uma nota em categorias de desempenho.

**Descrição:** Escreva um programa que peça a nota de um aluno (entre 0 e 100) e classifique a nota como "Excelente", "Bom", "Regular" ou "Insuficiente". Use as seguintes faixas:

- "Excelente" para notas 90 ou mais
- "Bom" para notas entre 70 e 89
- "Regular" para notas entre 50 e 69
- "Insuficiente" para notas abaixo de 50

# Exercício

## Exercício 3:

```
exercicios.py x
1  nota = float(input("Digite a nota do aluno: "))
2
3  if nota >= 90:
4      print("Excelente")
5  elif nota >= 70:
6      print("Bom")
7  elif nota >= 50:
8      print("Regular")
9  else:
10     print("Insuficiente")
```