

# Resumo – Operações no Python

Operadores no Python

Operadores aritméticos

Operadores relacionais (ou de comparação)

Operadores lógicos (*booleanos*)

Lendo valores do usuário

## Operadores no Python

Divididos entre:

- Aritméticos: `+ - * / ...`
- Relacionais: `== != < > ...`
- Lógicos: `and or not`

## Operadores aritméticos

- Soma: `2 + 2 = 4`
- Subtração: `2 - 1 = 1`
- Multiplicação: `2 * 3 = 6`
- Divisão: `3 / 2 = 1.5` (resultado é sempre um *float*)
- Exponenciação: `2 ** 3 = 8` (equivalente a `2 * 2 * 2`)
- Divisão inteira: `3 // 2 = 1` (remove a parte decimal - depois do `.`, resultado é sempre um *int*)
- Módulo (resto da divisão): `4 % 2 = 0` e `5 % 2 = 1`

## Operadores relacionais (ou de comparação)

- Igualdade: `10 == 10` → `True`
- Diferença: `10 != 10` → `False`

- Maior: `10 > 5` → `True`
- Maior ou igual: `5 >= 5` → `True`
- Menor: `5 < 5` → `False`
- Menor ou igual: `5 <= 5` → `True`

## Operadores lógicos (*booleanos*)

- E-lógico (*and*): `(5 == 10) and (5 < 10)` → `False`
- Ou-lógico (*or*): `(5 == 10) or (5 < 10)` → `True`
- Não-lógico (*not*): `not ((5 == 10) or (5 < 10))` → `False`

## Lendo valores do usuário

Utilizamos a função `input("Algum texto...")` para exibir uma mensagem ao usuário (o que estiver entre as aspas) e aguardar o usuário digitar algum valor. Podemos salvar esse valor em uma variável. Exemplo: `x = input("Digite o valor de x")`



Lembre que sempre que o usuário insere um valor ele vem como **string**. Se quisermos transformar em *int* ou *float* temos que usar as funções `int(valor)` ou `float(valor)`.