

Aula – Operadores em Python

1. Introdução

Operadores em Python são símbolos especiais usados para realizar operações em variáveis e valores.

Eles permitem desde cálculos matemáticos até comparações lógicas que controlam o fluxo do programa.

Neste módulo, estudaremos os Operadores Aritméticos e os Operadores de Comparação (Relacionais).

2. Operadores Aritméticos

São utilizados para realizar operações matemáticas básicas.

Principais operadores:

Adição (+): soma dois valores.

Exemplo: `20 + 30` resulta em 50.

Subtração (-): subtrai o valor da direita do valor da esquerda.

Exemplo: `45 - 15` resulta em 30.

Multiplicação (*): multiplica dois valores.

Exemplo: `5 * 2` resulta em 10.

Exponenciação (**): eleva um número a uma potência.

Exemplo: `5 ** 2` resulta em 25.

Uso de parênteses: define a ordem de execução das operações.

Exemplo: `(3 + 9) * (15 - 11)` resulta em 48.

Módulo (%): retorna o resto de uma divisão.

Exemplo: `10 % 3` resulta em 1; `15 % 4` resulta em 3.

O operador módulo é útil, por exemplo, para verificar se um número é par ou ímpar.

Exemplo: `x % 2 == 0` resulta em True se x for par.

Exemplo Prático com Operadores Aritméticos

```
minutos = 360

horas = minutos / 60

print(horas)    # Saída: 6
```

3. Operadores de Comparação (Relacionais)

São usados para comparar valores e retornar resultados lógicos (True ou False).

Igual (==): verifica se os valores são iguais.

Exemplo: 5 == 5 resulta em True; 5 == 3 resulta em False.

Diferente (!=): verifica se os valores são diferentes.

Exemplo: 5 != 3 resulta em True; 5 != 5 resulta em False.

Menor que (<): verifica se o valor da esquerda é menor que o da direita.

Exemplo: 3 < 5 resulta em True; 5 < 3 resulta em False.

Maior ou Igual (>=): verifica se o valor da esquerda é maior ou igual ao da direita.

Exemplo: 5 >= 3 resulta em True; 3 >= 5 resulta em False.

4. Aplicações dos Operadores de Comparação

São muito utilizados em estruturas de controle como:

- Instruções condicionais (if/else).
- Laços de repetição (for, while).
- Controle do fluxo lógico de programas.

5. Conclusão

Agora você é capaz de:

- Usar operadores aritméticos para realizar cálculos matemáticos em Python.
- Usar operadores de comparação para avaliar expressões lógicas que retornam True ou False.
- Aplicar operadores em contextos práticos, como verificações condicionais e cálculos de rotina.