

## 1. Operadores Aritméticos (Matemáticos)

```
# Adição (+)
print(5 + 3) # Saída: 8

# Subtração (-)
print(10 - 4) # Saída: 6

# Multiplicação (*)
print(3 * 4) # Saída: 12

# Divisão (/)
print(10 / 2) # Saída: 5.0 (sempre retorna float)

# Divisão inteira (//)
print(10 // 3) # Saída: 3 (descarta a parte decimal)

# Módulo (resto da divisão) (%)
print(10 % 3) # Saída: 1 (resto de 10 dividido por 3)

# Exponenciação (**)
print(2 ** 3) # Saída: 8 (2 elevado a 3)
```

## 2. Operadores de Comparação (Retornam True/False)

```
# Igual (==)
print(5 == 5) # Saída: True

# Diferente (!=)
print(5 != 3) # Saída: True

# Maior que (>)
print(5 > 3) # Saída: True

# Menor que (<)
print(5 < 3) # Saída: False

# Maior ou igual (>=)
print(5 >= 5) # Saída: True

# Menor ou igual (<=)
print(5 <= 3) # Saída: False
```

## 3. Operadores de Atribuição

```
x = 5 # Atribuição simples
x += 3 # Equivalente a x = x + 3 (agora x vale 8)
x -= 2 # x = x - 2 (x vale 6)
x *= 4 # x = x * 4 (x vale 24)
x /= 3 # x = x / 3 (x vale 8.0)
x %= 5 # x = x % 5 (x vale 3.0)
x **= 2 # x = x ** 2 (x vale 9.0)
x //= 2 # x = x // 2 (x vale 4.0)
```

## 4. Operadores Lógicos (Booleanos)

```
# AND (e) - True se ambos forem True
print(True and False) # Saída: False

# OR (ou) - True se pelo menos um for True
print(True or False) # Saída: True

# NOT (negação)
print(not True) # Saída: False
```

## 5. Operadores de Identidade

```
a = [1, 2]
b = a
c = [1, 2]

# is (verifica se é o mesmo objeto na memória)
print(a is b) # Saída: True
print(a is c) # Saída: False (objetos diferentes)

# is not (verifica se não é o mesmo objeto)
print(a is not c) # Saída: True
```

## 6. Operadores de Associação (Para sequências)

```
lista = [1, 2, 3, 4]

# in (está contido)
print(3 in lista) # Saída: True

# not in (não está contido)
print(5 not in lista) # Saída: True
```

## 7. Operadores Bit a Bit (Para números binários)

```
a = 10 # 1010 em binário
b = 4 # 0100 em binário

# AND bit a bit (&)
print(a & b) # 0000 → 0

# OR bit a bit (|)
print(a | b) # 1110 → 14

# XOR bit a bit (^)
print(a ^ b) # 1110 → 14

# NOT bit a bit (~)
print(~a) # Inverte os bits → -11

# Deslocamento à esquerda (<<)
print(a << 1) # 10100 → 20

# Deslocamento à direita (>>)
print(a >> 1) # 0101 → 5
```

## Operador Ternário (Condicional em uma linha)

```
idade = 18
status = "Maior" if idade >= 18 else "Menor"
print(status) # Saída: "Maior"
```

## Operador Walrus (:=) - Python 3.8+

```
# Atribui e usa o valor na mesma expressão
if (n := len([1,2,3])) > 2:
    print(f"Lista tem {n} elementos") # Saída: "Lista tem 3 elementos"
```

Dica útil: Você pode ver a precedência completa dos operadores (quem é calculado primeiro) na documentação oficial do Python. Quando houver dúvidas, use parênteses para deixar explícito a ordem desejada!