Operadores em Python

***Adaptado por José Alfredo Costa - Jun / 24***

Data: 16-6-2024

**Então, o que são operadores?**

Operadores são símbolos especiais que ajudam a realizar operações de atribuição, aritméticas ou lógicas.

**Existem muitos tipos de operadores:**

* **Operadores Aritméticos:** utilizados para cálculos matemáticos.
* **Operadores de Comparação:** utilizados para comparações lógicas.
* **Operadores Lógicos:** utilizados para combinar múltiplas condições.

Veremos todos eles um a um neste vídeo. Também veremos os diferentes tipos de operadores e a precedência entre eles.

**O que são operandos?**

Operandos são os valores sobre os quais o operador opera. Ou seja, são os valores entre os quais aplicamos o operador para realizar uma operação.

**Vamos começar com os tipos de operadores em Python:**

Em Python, existem sete tipos principais de operadores:

1. **Aritméticos**
2. **Atribuição**
3. **Comparação**
4. **Lógicos**
5. **Membro**
6. **Identidade**
7. **Bit a bit**

**Vamos começar com os operadores aritméticos:**

Os operadores aritméticos incluem:

* **Adição (+)**
* **Subtração (-)**
* **Multiplicação (\*)**
* **Divisão (/)**
* **Módulo (%)**
* **Divisão de Piso (//)**
* **Exponenciação (**)\*\*

Esses operadores são utilizados para realizar cálculos matemáticos.

**Vamos ver a hierarquia dos operadores aritméticos:**

A hierarquia dos operadores aritméticos é em ordem decrescente de precedência:

1. **Parênteses ()**
2. **Exponenciação (**)\*\*
3. **Divisão (/)**
4. **Multiplicação (\*)**
5. **Adição (+)**
6. **Subtração (-)**

**Vamos testar cada um deles e como utilizá-los:**

Suponha que temos dois valores: a = 5 e b = 6.

* **Subtração:** a - b dará como resultado -1.
* **Divisão:** a / b dará como resultado 0.8333333333333334.
* **Módulo:** a % b dará como resultado 5.
* **Divisão de Piso:** a // b dará como resultado 0.
* **Exponenciação:** b \*\* a dará como resultado 7776.

**Precedência:**

Vamos verificar a precedência dos operadores com um exemplo:

a = 7 - 2 \* (27 / 3 \*\* 2) + 4

Nesse caso, o resultado é 5.

1. Primeiro, os parênteses são avaliados.
2. Dentro dos parênteses, a exponenciação é avaliada primeiro.
3. Depois, a divisão é avaliada.
4. Em seguida, a multiplicação.
5. Depois, a subtração.
6. Por fim, a adição.

**Vamos agora analisar o próximo operador: operador de atribuição:**

O operador de atribuição mais comum é o sinal de igual (=), utilizado para atribuir valores a variáveis. Outros operadores de atribuição incluem:

* **+=** (Adição e atribuição)
* **-=** (Subtração e atribuição)
* **\*=** (Multiplicação e atribuição)
* **/=** (Divisão e atribuição)

**Exemplo:**

* a = 5 atribui o valor 5 à variável a.
* a += b é equivalente a a = a + b.

**O operador de atribuição exige que a variável esteja sempre à esquerda do operador e o valor à direita.**

**Vamos agora analisar os operadores de comparação:**

Os operadores de comparação são:

* **<** (Menor que)
* **<=** (Menor ou igual a)
* **>** (Maior que)
* **>=** (Maior ou igual a)
* **==** (Igual a)

Os operadores de comparação retornam um valor booleano (True ou False).

**Exemplo:**

* x = 5
* y = 7
* x < y retornará True, pois x é menor que y.
* x > y retornará False, pois x é menor que y.
* x == y retornará False, pois x é diferente de y.

**Operadores Lógicos:**

Os operadores lógicos são:

* **and** (E lógico)
* **or** (Ou lógico)
* **not** (Negação)

Os operadores lógicos são utilizados para combinar condições e retornam um valor booleano.

**Exemplo:**

* **and:** Retorna True se ambas as condições forem True.
* **or:** Retorna True se pelo menos uma condição for True.
* **not:** Inverte o valor booleano.

**Operador de Membro:**

Os operadores de membro são:

* **in** (Pertence a)
* **not in** (Não pertence a)

Os operadores de membro são utilizados para verificar se um elemento está presente em uma sequência (como uma lista ou uma string).

**Exemplo:**

* a = [1, 2, 3]
* 2 in a retornará True, pois 2 está presente em a.
* 4 in a retornará False, pois 4 não está presente em a.

**Operador de Identidade:**

Os operadores de identidade são:

* **is** (É o mesmo objeto)
* **is not** (Não é o mesmo objeto)

Os operadores de identidade verificam se dois objetos são o mesmo objeto na memória.

**Exemplo:**

* a = 5
* b = 5
* a is b retornará True, pois a e b referenciam o mesmo objeto na memória.
* a is not b retornará False.

**Operadores Bit a Bit:**

Os operadores bit a bit são:

* **&** (E bit a bit)
* **|** (Ou bit a bit)

Os operadores bit a bit são utilizados para realizar operações lógicas bit a bit em inteiros.

**Exemplo:**

* a = 6
* b = 3
* a & b retornará 2, pois o operador & realiza a operação "E" bit a bit.
* a | b retornará 7, pois o operador | realiza a operação "Ou" bit a bit.

**Precedência dos Operadores:**

A precedência dos operadores é a ordem em que as operações são avaliadas em uma expressão.

A precedência dos operadores em ordem decrescente é:

1. **Parênteses ()**
2. **Exponenciação (**)\*\*
3. **Multiplicação (\*), Divisão (/), Módulo (%), Divisão de Piso (//)**
4. **Adição (+), Subtração (-)**
5. **Operadores Bit a Bit (&, |)**
6. **Operadores de Comparação (==, !=, <, <=, >, >=)**
7. **Operadores de Identidade (is, is not)**
8. **Operadores de Membro (in, not in)**
9. **Operador Lógico Not (not)**
10. **Operadores Lógicos And (and)**
11. **Operadores Lógicos Or (or)**

Códigos Python - Operadores

Aqui estão os códigos Python apresentados no vídeo, junto com comentários explicativos:

**1. Operadores Aritméticos**

# Atribuindo valores às variáveis

a = 5

b = 6

# Subtração

print(a - b) # Saída: -1

# Divisão

print(a / b) # Saída: 0.8333333333333334

# Módulo

print(a % b) # Saída: 5

# Divisão de Piso

print(a // b) # Saída: 0

# Exponenciação

print(b \*\* a) # Saída: 7776

**2. Operadores de Atribuição**

# Atribuindo valor à variável

a = 5

# Adição e atribuição

a += 2 # Equivalente a **a = a + 2**

print(a) # Saída: 7

# Subtração e atribuição

a -= 3 # Equivalente a **a = a - 3**

print(a) # Saída: 4

**3. Operadores de Comparação**

# Atribuindo valores às variáveis

x = 5

y = 7

# Menor que

print(x < y) # Saída: True

# Maior que

print(x > y) # Saída: False

# Igual a

print(x == y) # Saída: False

**4. Operadores Lógicos**

# Atribuindo valores às variáveis

x = 7

y = 17

# Operador AND (ambas as condições devem ser True)

print(x <= y and x >= y) # Saída: False

# Operador OR (pelo menos uma condição deve ser True)

print(x <= y or x >= y) # Saída: True

# Operador NOT (inverte o valor booleano)

print(not (x <= y)) # Saída: False

**5. Operadores de Membro**

# Criando uma lista

a = [1, 2, 3]

# Verificando se um elemento está presente na lista

print(2 in a) # Saída: True

# Verificando se um elemento NÃO está presente na lista

print(5 in a) # Saída: False

**6. Operadores de Identidade**

# Atribuindo valores às variáveis

a = 5

b = 5

# Verificando se as variáveis referenciam o mesmo objeto

print(a is b) # Saída: True

# Criando listas

p = [1, 2, 3]

q = [1, 2, 3]

# Verificando se as listas referenciam o mesmo objeto

print(p is q) # Saída: False

**7. Operadores Bit a Bit**

# Atribuindo valores às variáveis

a = 6

b = 3

# Operador AND bit a bit

print(a & b) # Saída: 2

# Operador OR bit a bit

print(a | b) # Saída: 7

**8. Precedência de Operadores**

# Exemplo de precedência de operadores

a = 7 - 2 \* (27 / 3 \*\* 2) + 4

print(a) # Saída: 5

**Observação:** Os comentários no código explicam o que cada linha faz e o que é esperado como saída. Você pode executar esses códigos no seu interpretador Python para verificar os resultados.