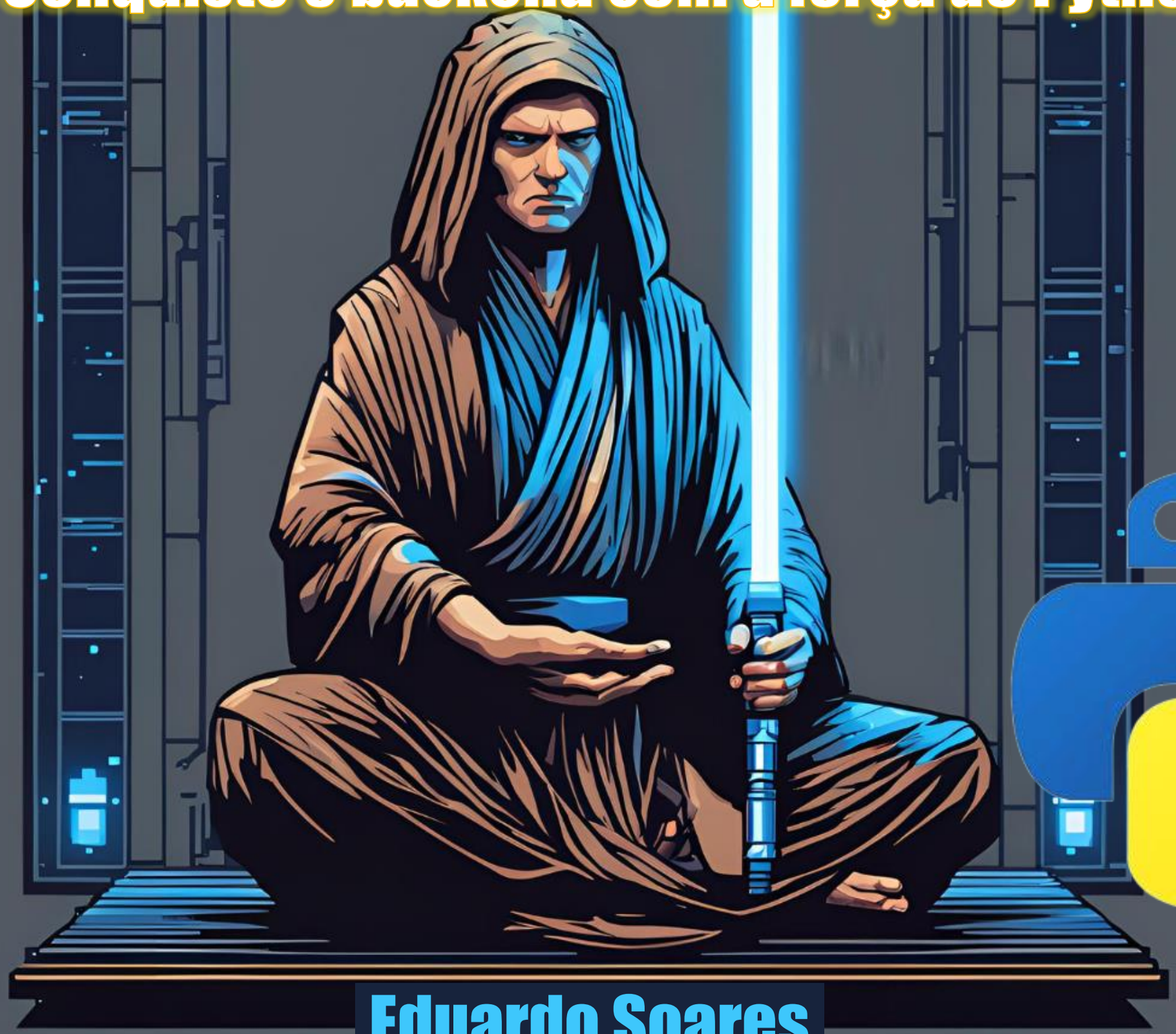
THE PYTHON FORCE

Conquiste o backend com a força do Python

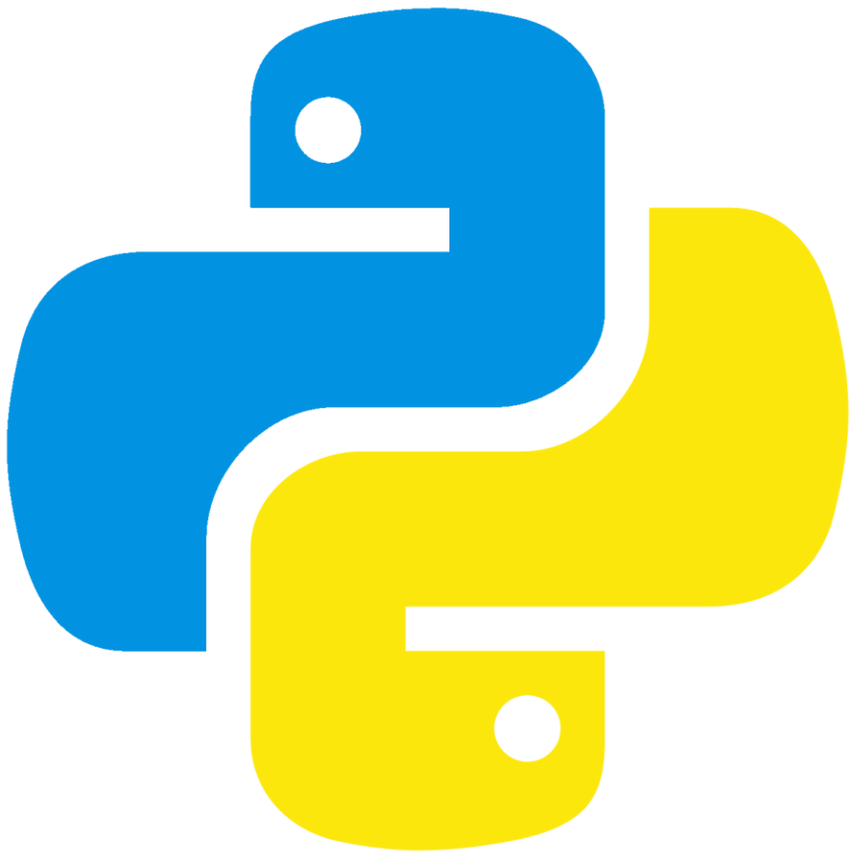


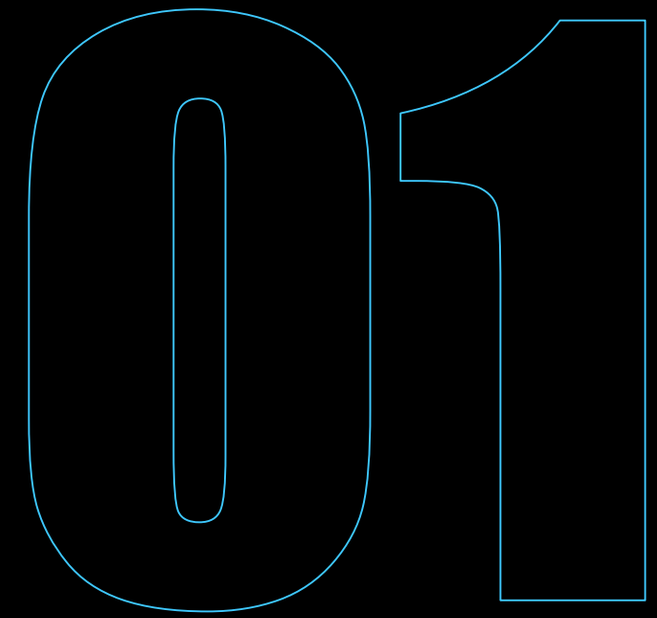
Eduardo Soares

Desbravando os Operadores Python

Introdução

Os operadores em Python são ferramentas essenciais que permitem manipular dados, fazer cálculos, e tomar decisões em seu código. Eles são fundamentais para qualquer programador, independentemente do nível de experiência. Neste capítulo, vamos explorar os principais operadores em Python, explicando-os de forma simples e com exemplos práticos de uso. Prepare-se para dominar esses operadores e tornar seu código mais eficiente e poderoso.





OPERADORES ARITMÉTICOS

Os operadores aritméticos são a base de muitas operações em Python. Eles permitem realizar cálculos matemáticos simples, que são essenciais para qualquer programador.

Operadores Aritméticos: Somando, Subtraindo, Multiplicando e Dividindo

Os operadores aritméticos são a base de muitas operações em Python. Eles permitem realizar cálculos matemáticos simples, que são essenciais para qualquer programador.
Adição (+): Soma dois valores.
Subtração (-): Subtrai o segundo valor do primeiro.
Multiplicação (*): Multiplica dois valores.
Divisão (/): Divide o primeiro valor pelo segundo, resultando em um número de ponto flutuante.
Divisão Inteira (//): Divide o primeiro valor pelo segundo, resultando em um número inteiro.
Módulo (%): Retorna o resto da divisão do primeiro valor pelo segundo.
Exponenciação (**): Eleva um número à potência de outro.
Exemplo em código:

The Python Force

```python
# Exemplo prático: Calculando o custo total de um pedido
preco_unitario = 49.99
quantidade = 3
desconto = 10

custo_bruto = preco_unitario * quantidade
custo_com_desconto = custo_bruto - desconto
custo_final = custo_com_desconto + (custo_com_desconto * 0.1)
#imposto de 10%
print(f"Custo final do pedido: R${custo_final:.2f}")
```

02

OPERADORES DE COMPARAÇÃO

Os operadores de comparação permitem comparar dois valores e retornam ‘True’ ou ‘False’ com base na condição

Operadores de Comparação: Avaliando Condições

Os operadores de comparação permitem comparar dois valores e retornam True ou False com base na condição.

•**Igual (==)**: Verifica se dois valores são iguais.
•**Diferente (!=)**: Verifica se dois valores são diferentes.
•**Maior que (>)**: Verifica se o primeiro valor é maior que o segundo.
•**Menor que (<)**: Verifica se o primeiro valor é menor que o segundo.
•**Maior ou Igual (>=)**: Verifica se o primeiro valor é maior ou igual ao segundo.
•**Menor ou Igual (<=)**: Verifica se o primeiro valor é menor ou igual ao segundo.

Exemplo em código:

The Python Force

```
# Exemplo prático: Verificando a elegibilidade para um desconto
idade = 21

if idade ⩾ 18:
    print("Você é elegível para o desconto.")
else:
    print("Desculpe, o desconto é apenas para maiores de 18 anos.")
```

OPERADORES LÓGICOS

Os operadores lógicos são utilizados para combinar várias condições

Operadores Lógicos: Combinando Condições

Os operadores lógicos são utilizados para combinar várias condições.
- **E (and)**: Retorna True se ambas as condições forem verdadeiras.
- **Ou (or)**: Retorna True se pelo menos uma das condições for verdadeira.
- **Não (not)**: Inverte o valor booleano da condição.

Exemplo em código:

The Python Force

```
# Exemplo prático: Verificando o acesso a uma área restrita
usuario_ativo = True
usuario_admin = False

if usuario_ativo and usuario_admin:
    print("Acesso concedido.")
else:
    print("Acesso negado. Você precisa ser um administrador ativo.")
```

OPERADORES DE ATRIBUIÇÃO

Os operadores de atribuição são usados para atribuir valores a variáveis.

Operadores de Atribuição: Atribuindo Valores a Variáveis

Os operadores de atribuição são usados para atribuir valores a variáveis.
Atribuição (=): Atribui um valor à variável.
Atribuição com Adição (+=): Soma o valor ao existente na variável.
Atribuição com Subtração (-=): Subtrai o valor do existente na variável.
Atribuição com Multiplicação (*=): Multiplica o valor pelo existente na variável.
Atribuição com Divisão (/=): Divide o valor pelo existente na variável.

Exemplo em código:

The Python Force

```python
# Exemplo prático: Atualizando a pontuação de um jogador
pontuacao = 50

# O jogador ganha 10 pontos
pontuacao += 10
print(f"Pontuação atual: {pontuacao}")

# O jogador perde 5 pontos
pontuacao -= 5
print(f"Pontuação atual: {pontuacao}")
```

OPERADORES DE ASSOCIAÇÃO

Os operadores de associação verificam se um valor está presente em uma sequência (como listas, tuplas, ou strings).

Operadores de Associação: Verificando Pertinência

Os operadores de associação verificam se um valor está presente em uma sequência (como listas, tuplas, ou strings).

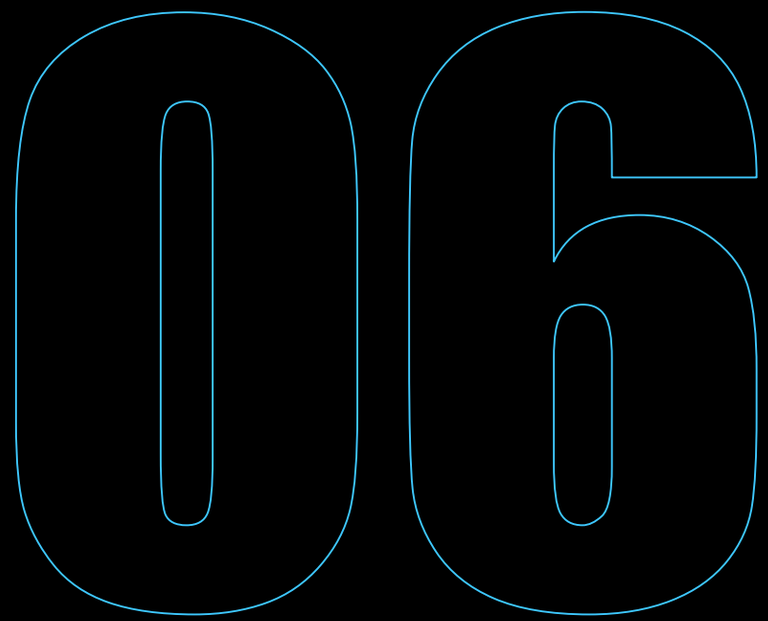
- **Em (in)**: Retorna True se o valor estiver presente na sequência.
- **Não em (not in)**: Retorna True se o valor não estiver presente na sequência.

Exemplo em código:

The Python Force

```python
# Exemplo prático: Verificando a presença de um item no carrinho de compras
carrinho_de_compras = ["maçã", "banana", "laranja"]

if "banana" in carrinho_de_compras:
    print("A banana está no carrinho de compras.")
else:
    print("A banana não está no carrinho de compras.")
```



OPERADORES DE IDENTIDADE

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

Operadores de Identidade: Verificando Identidade de Objetos

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

- **É (is)**: Retorna True se as variáveis comparadas referem-se ao mesmo objeto.
- **Não é (is not)**: Retorna True se as variáveis comparadas referem-se a objetos diferentes.

Exemplo em código:

The Python Force

```
# Exemplo prático: Verificando se duas variáveis apontam para o mesmo objeto
a = [1, 2, 3]
b = a
c = [1, 2, 3]

print(a is b)  # True, pois b é uma referência ao mesmo objeto que a
print(a is c)  # False, pois c é um novo objeto, mesmo com o mesmo conteúdo
```

Conclusões

Encerramento

Compreender os operadores do Python é fundamental para escrever código eficiente e funcional. Usando esses operadores de maneira eficaz, você pode manipular dados e tomar decisões baseadas em condições de forma mais intuitiva e poderosa.

Esse Ebook foi gerado por IA, e montado por humano.
O passo a passo se encontra no meu Github .

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizado uma revisão cuidadosa humana no conteúdo e pode conter falhas geradas por uma IA.

The Python Force

OBRIGADO POR LER ATE AQUI

Boa sorte nesta jornada no mundo da programação!