



Café com Python

DOMINANDO A LINGUAGEM

Capítulo 1:

Fala, devs! Bem-vindos ao mundo do Python, a linguagem que está dominando o universo da programação. Quer saber por que Python é a escolha certa? Porque é simples, poderoso e versátil. Seja você um desenvolvedor iniciante ou um veterano, Python tem algo para te oferecer. Bora mergulhar nessa?



O que é Python?

Python é uma linguagem de programação criada por Guido van Rossum nos anos 90. É como um bom vinho: só melhora com o tempo. E o nome? Uma homenagem ao Monty Python, porque programar também pode ser divertido!

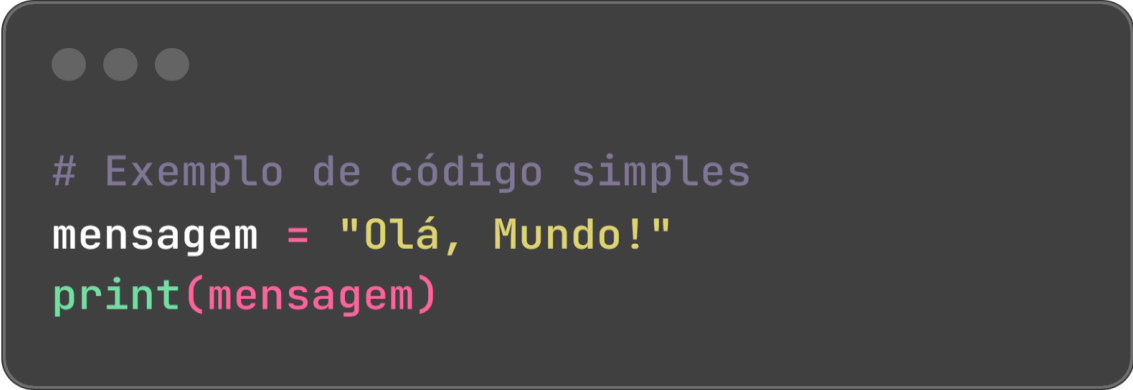
Capítulo 1: Introdução ao Python

Por Que Python é Tão Popular?

Python é uma linguagem de programação que conquistou muitos desenvolvedores pela sua simplicidade e versatilidade. Ela é fácil de aprender e usar, tornando-se uma excelente escolha para iniciantes e para projetos de todas as complexidades.

Simplicidade e Leitura Fácil

Python é conhecido por sua sintaxe limpa e legível. Isso facilita tanto a escrita quanto a leitura do código.

A dark-themed code editor window with three small circles in the top-left corner. It contains three lines of Python code: a comment, an assignment, and a print statement.

```
# Exemplo de código simples  
mensagem = "Olá, Mundo!"  
print(mensagem)
```

Capítulo 02:

Estruturas de Controle de Fluxo em Python

Condicionais:

Condicionais permitem que você execute diferentes blocos de código com base em certas condições. Em Python, usamos `if`, `elif` e `else`.

Capítulo 2: Estruturas de Controle de Fluxo em Python

Condicionais

Condicionais permitem que você execute diferentes blocos de código com base em certas condições. Em Python, usamos *if*, *elif* e *else*

```
# Exemplo de uso de if, elif e else
idade = 18

if idade < 18:
    print("Você é menor de idade.")
elif idade == 18:
    print("Você acabou de se tornar maior de idade.")
else:
    print("Você é maior de idade.")
```

Capítulo 03:

Loop, Funções e Listas



Loops

O loop *for* é usado para iterar sobre uma sequência (como uma lista, tupla ou string).

```
# Exemplo de loop while
contador = 0

while contador < 5:
    print(contador)
    contador += 1
```

O loop *while* continua executando enquanto uma condição for verdadeira.

```
# Exemplo de loop while
contador = 0

while contador < 5:
    print(contador)
    contador += 1
```

Funções

Funções são blocos de código reutilizáveis que executam uma tarefa específica. Em Python, você define uma função usando *def*.

```
# Exemplo de definição e chamada de função
def saudacao(nome):
    print(f"Olá, {nome}!")

saudacao("Maria")
```

Funções podem retornar valores usando a palavra-chave `return`.

```
# Exemplo de função com retorno
def soma(a, b):
    return a + b

resultado = soma(5, 3)
print(resultado)
```


List Comprehensions

List comprehensions são uma maneira concisa de criar listas.

```
# Exemplo de list comprehension
numeros = [1, 2, 3, 4, 5]
quadrados = [x**2 for x in numeros]
print(quadrados)
```

Estruturas de controle de fluxo são essenciais para escrever programas eficientes e organizados. Com condicionais, loops e funções, você pode controlar o fluxo do seu programa e reutilizar código de maneira eficaz.