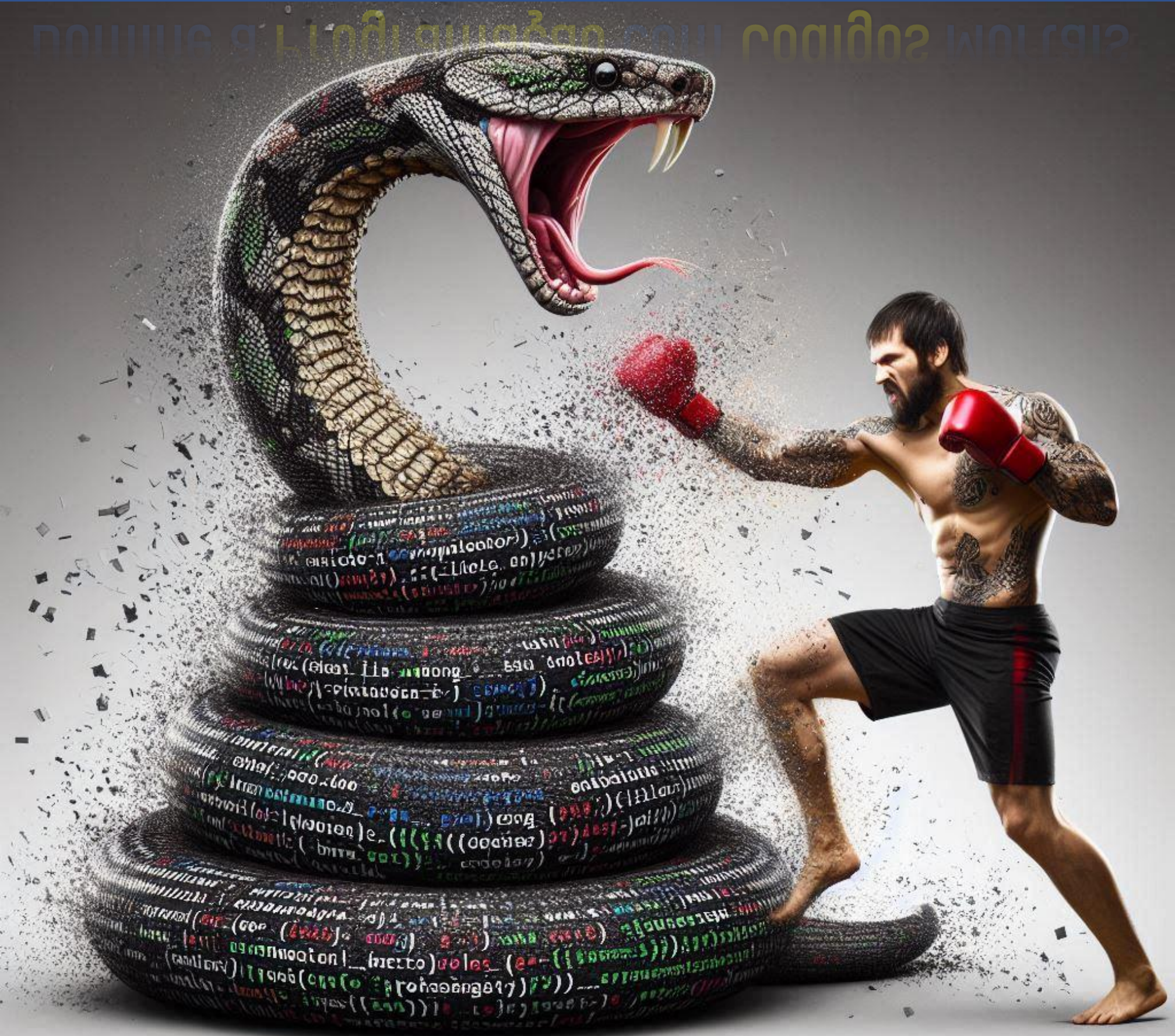


Python Kombat

Domine a Programação com Códigos Mortais



JONISSON AMARAL



Principais Tipos de Dados em Python

Introdução

Python é uma linguagem de programação versátil e poderosa, amplamente utilizada em diversas áreas como desenvolvimento web, ciência de dados, automação e muito mais. Conhecer os principais tipos de dados é fundamental para escrever códigos eficientes e compreensíveis. Neste ebook, abordaremos de maneira simples e prática os tipos de dados mais comuns em **Python**, acompanhados de exemplos reais.



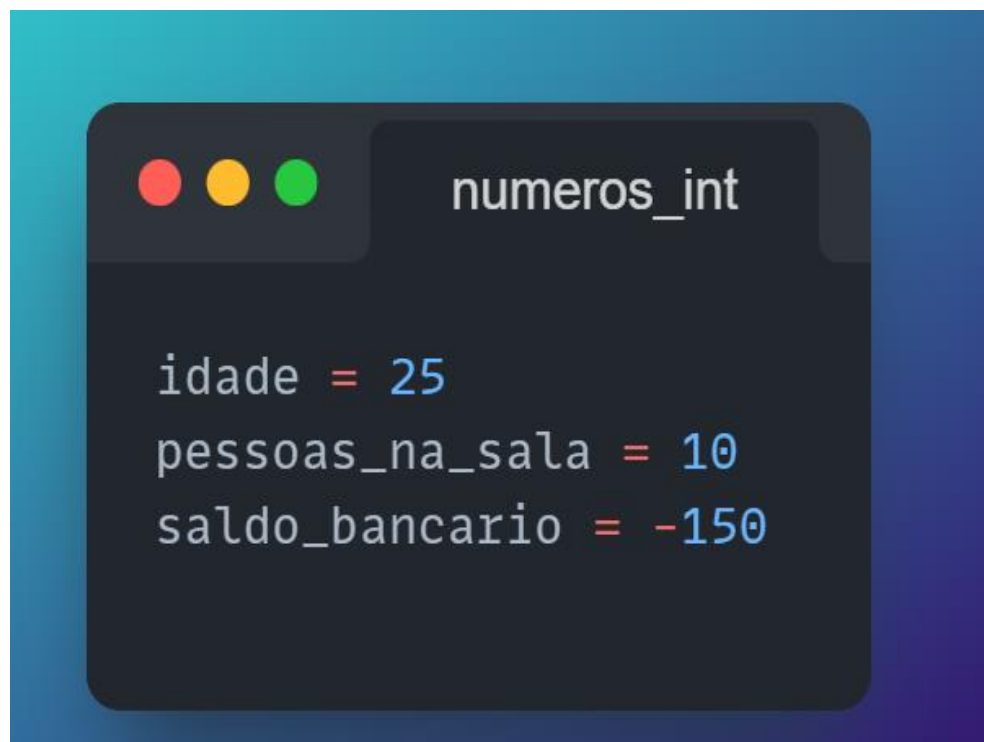
01

INTEIROS (int)

Os números inteiros são usados para representar valores numéricos sem casas decimais. Em Python, os inteiros podem ser positivos ou negativos.

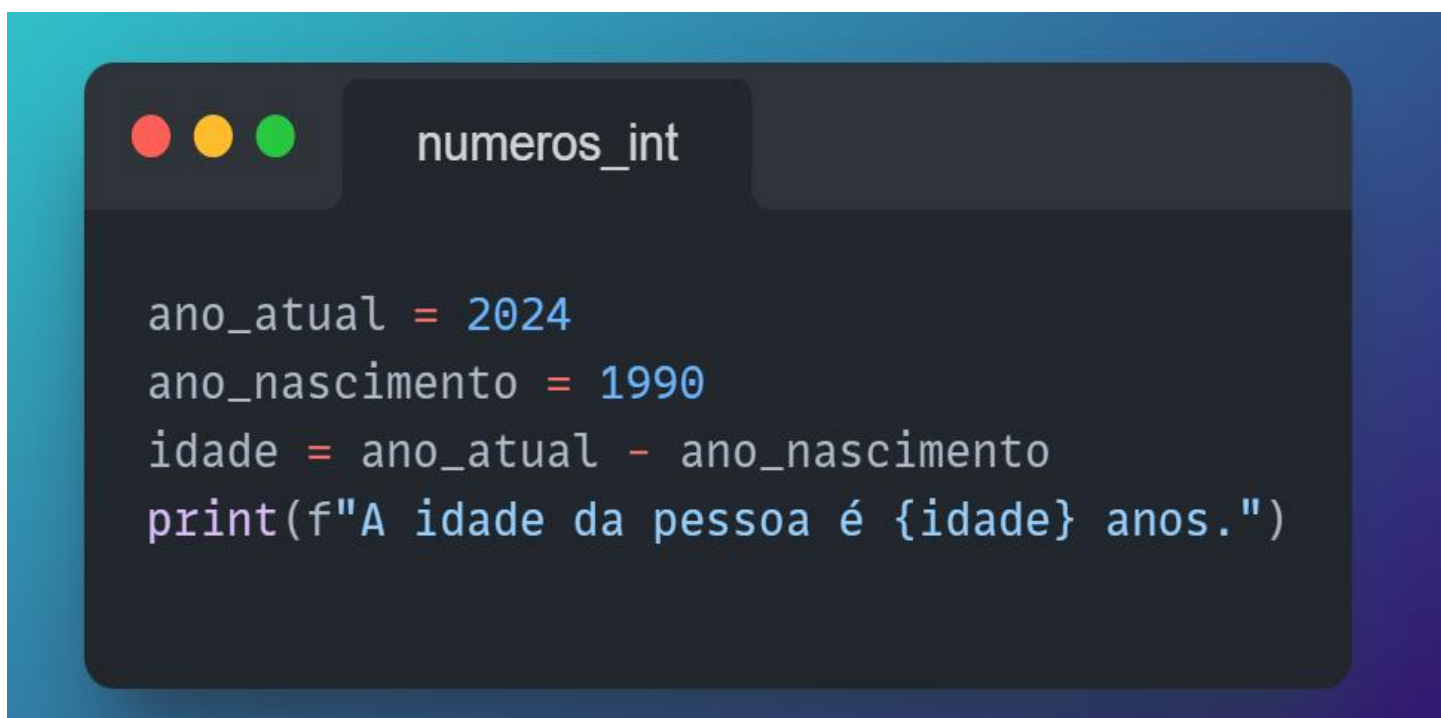
Python Kombat

Exemplo:

A screenshot of a terminal window with a dark background and light blue text. The window has a title bar with three colored circles (red, yellow, green) and the text 'numeros_int'. The code inside the terminal is:

```
idade = 25
pessoas_na_sala = 10
saldo_bancario = -150
```

Contexto real: Calcular a idade de uma pessoa a partir do ano de nascimento.

A screenshot of a terminal window with a dark background and light blue text. The window has a title bar with three colored circles (red, yellow, green) and the text 'numeros_int'. The code inside the terminal is:

```
ano_atual = 2024
ano_nascimento = 1990
idade = ano_atual - ano_nascimento
print(f"A idade da pessoa é {idade} anos.")
```

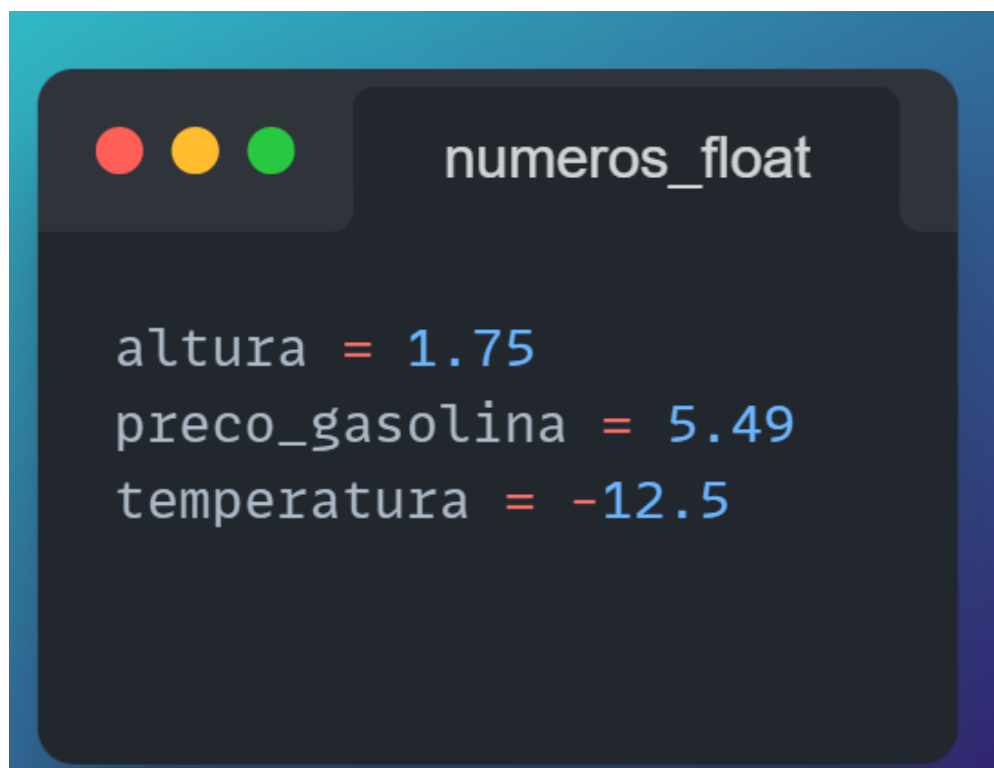
02

NÚMEROS de PONTO FLUTUANTE (float)

Os números de ponto flutuante representam valores numéricos com casas decimais.

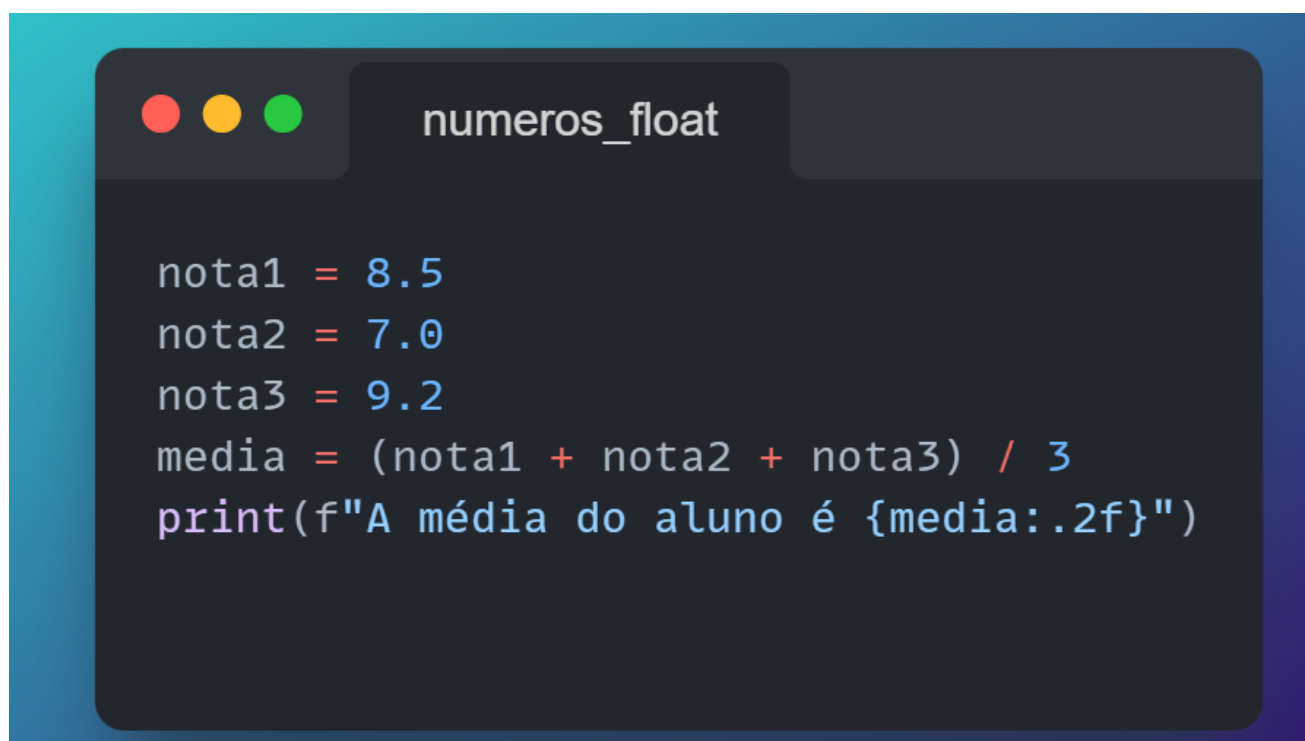
Python Kombat

Exemplo:



```
altura = 1.75
preco_gasolina = 5.49
temperatura = -12.5
```

Contexto real: Calcular a média de notas de um aluno.



```
nota1 = 8.5
nota2 = 7.0
nota3 = 9.2
media = (nota1 + nota2 + nota3) / 3
print(f"A média do aluno é {media:.2f}")
```

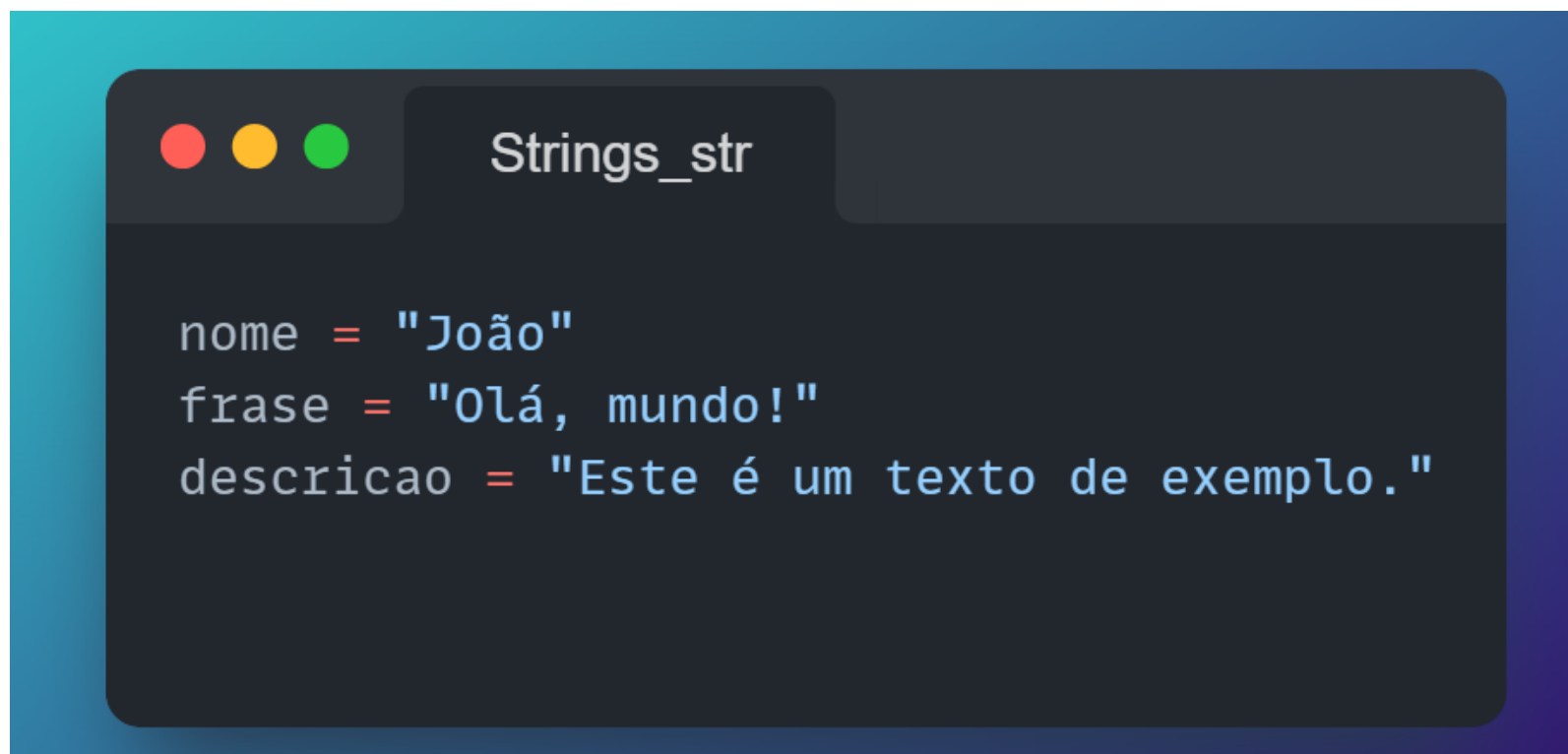
03

STRiNGS (STR)

Strings são usadas para representar sequências de caracteres, como palavras ou frases.

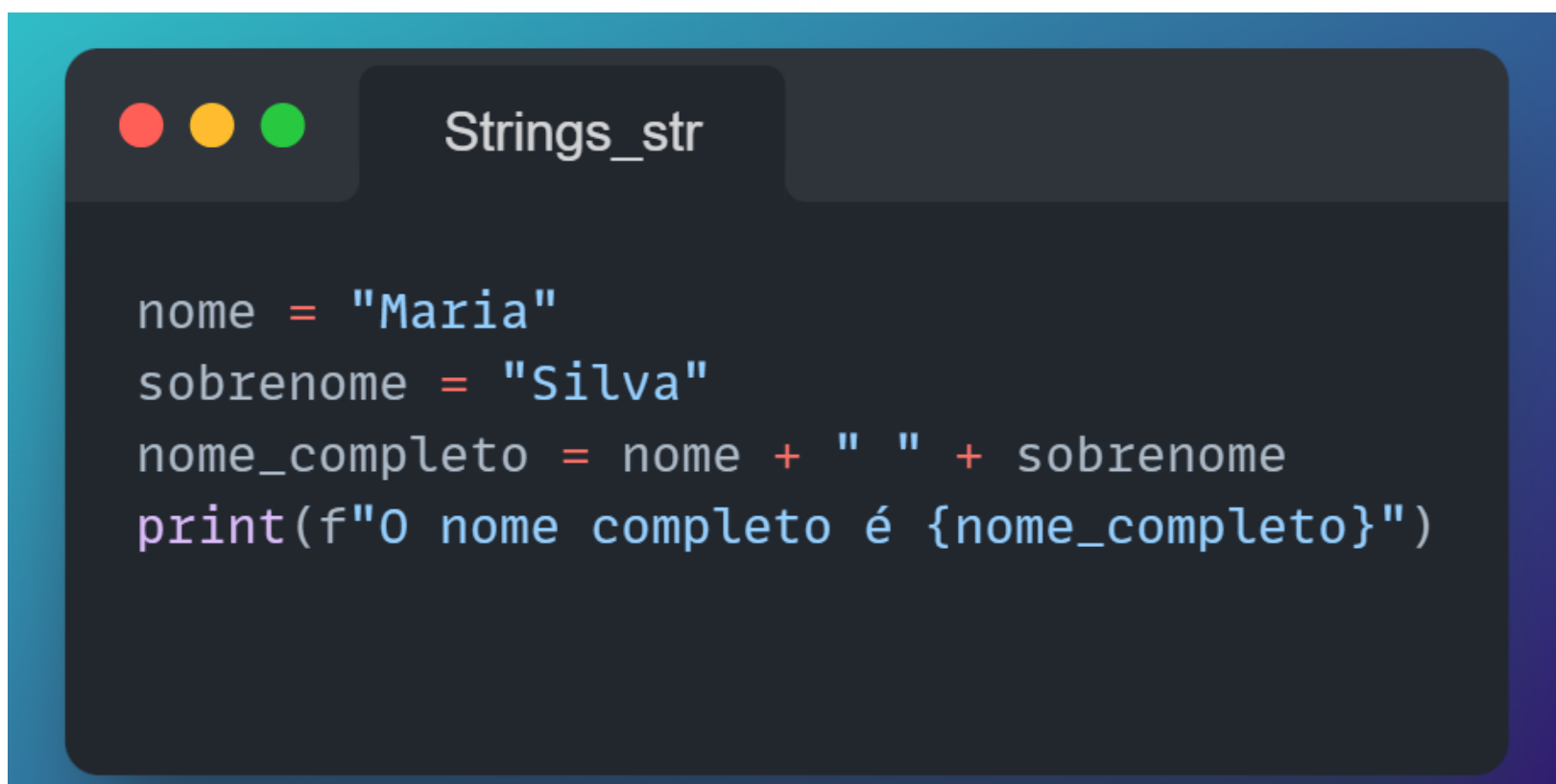
Python Kombat

Exemplo:



```
nome = "João"
frase = "Olá, mundo!"
descricao = "Este é um texto de exemplo."
```

Contexto real: Concatenar nome e sobrenome para criar um nome completo.



```
nome = "Maria"
sobrenome = "Silva"
nome_completo = nome + " " + sobrenome
print(f"O nome completo é {nome_completo}")
```

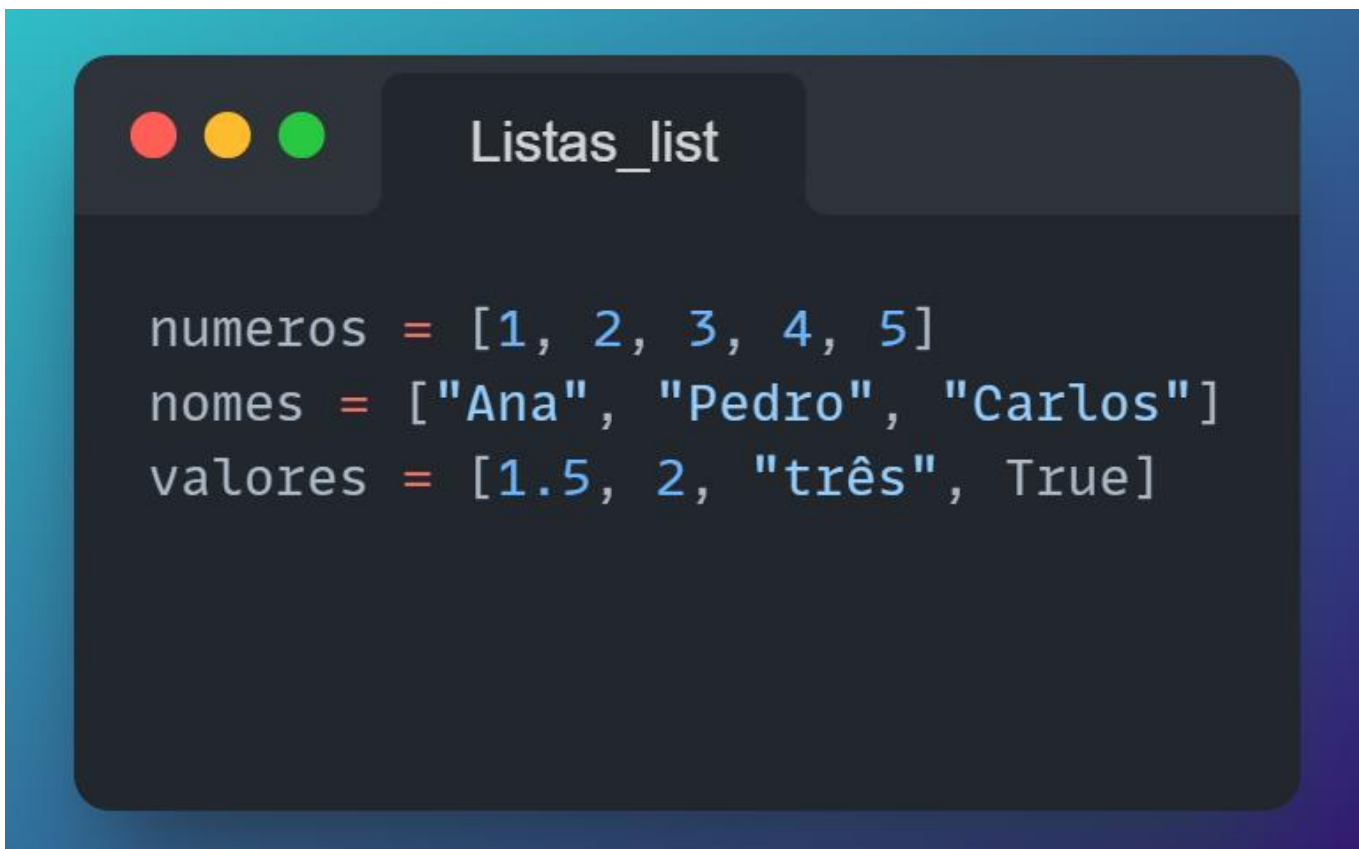

04

LISTAS (list)

Listas são coleções ordenadas e mutáveis de itens, que podem ser de diferentes tipos de dados.

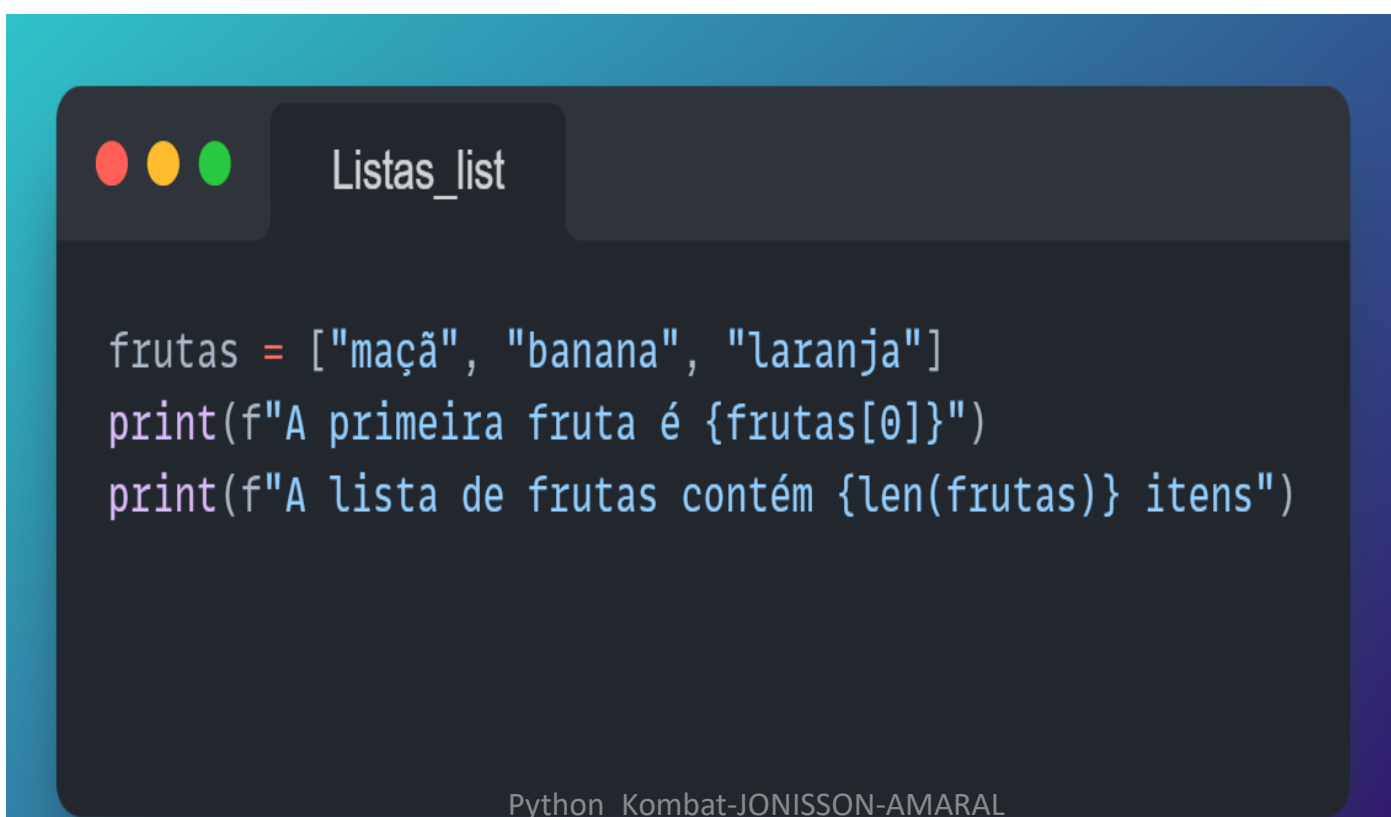
Python Kombat

Exemplo:



```
numeros = [1, 2, 3, 4, 5]
nomes = ["Ana", "Pedro", "Carlos"]
valores = [1.5, 2, "três", True]
```

Contexto real: Armazenar e acessar uma lista de frutas.



```
frutas = ["maçã", "banana", "laranja"]
print(f"A primeira fruta é {frutas[0]}")
print(f"A lista de frutas contém {len(frutas)} itens")
```

Python_Kombat-JONISSON-AMARAL

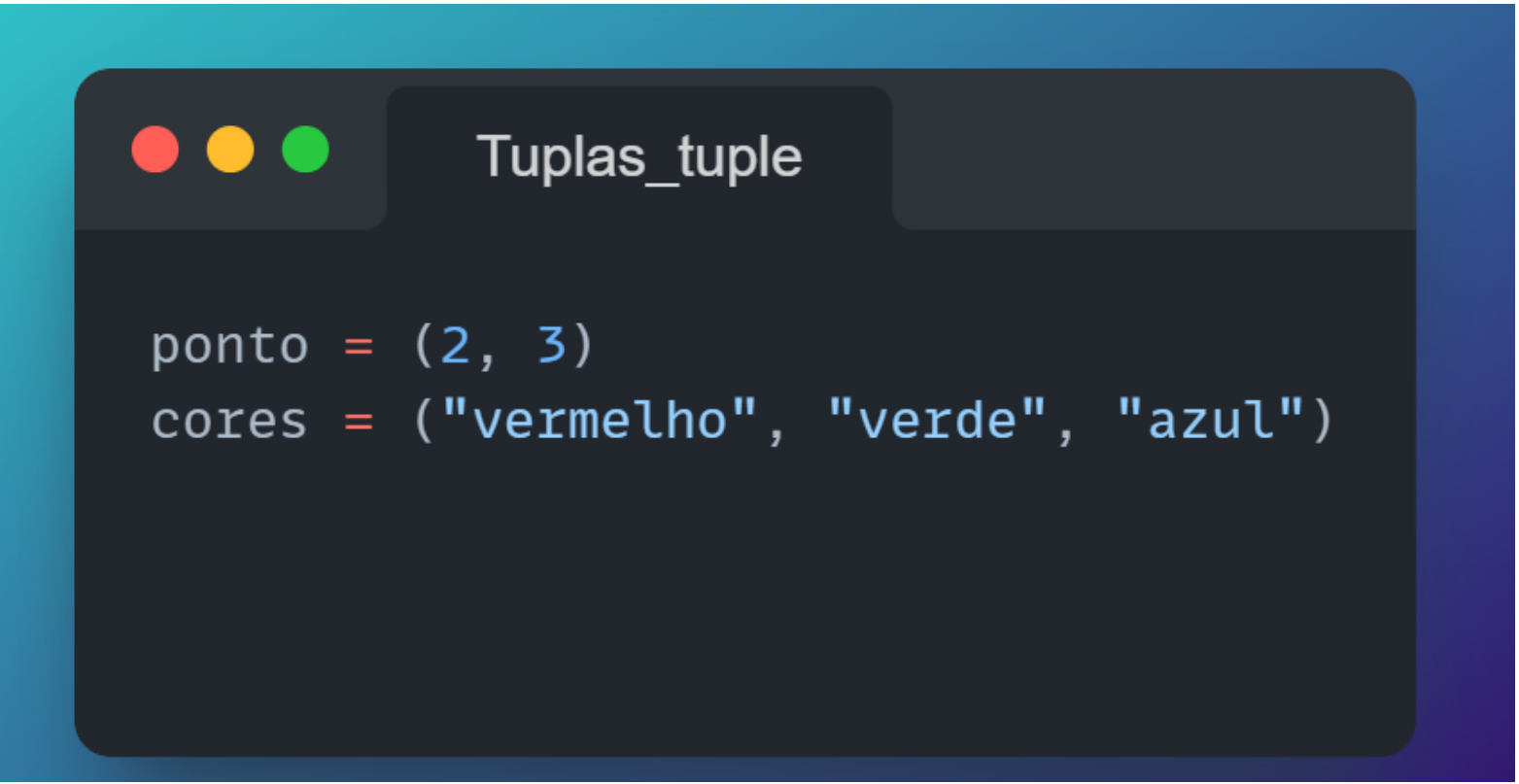
05

Tuplas (tuple)

Tuplas são coleções ordenadas e imutáveis de itens. Uma vez criadas, não podem ser modificadas.

Python Kombat

Exemplo:



```
ponto = (2, 3)
cores = ("vermelho", "verde", "azul")
```

Contexto real: Armazenar coordenadas de um ponto em um plano cartesiano.



```
ponto = (10, 20)
print(f"As coordenadas do ponto são {ponto}")
```

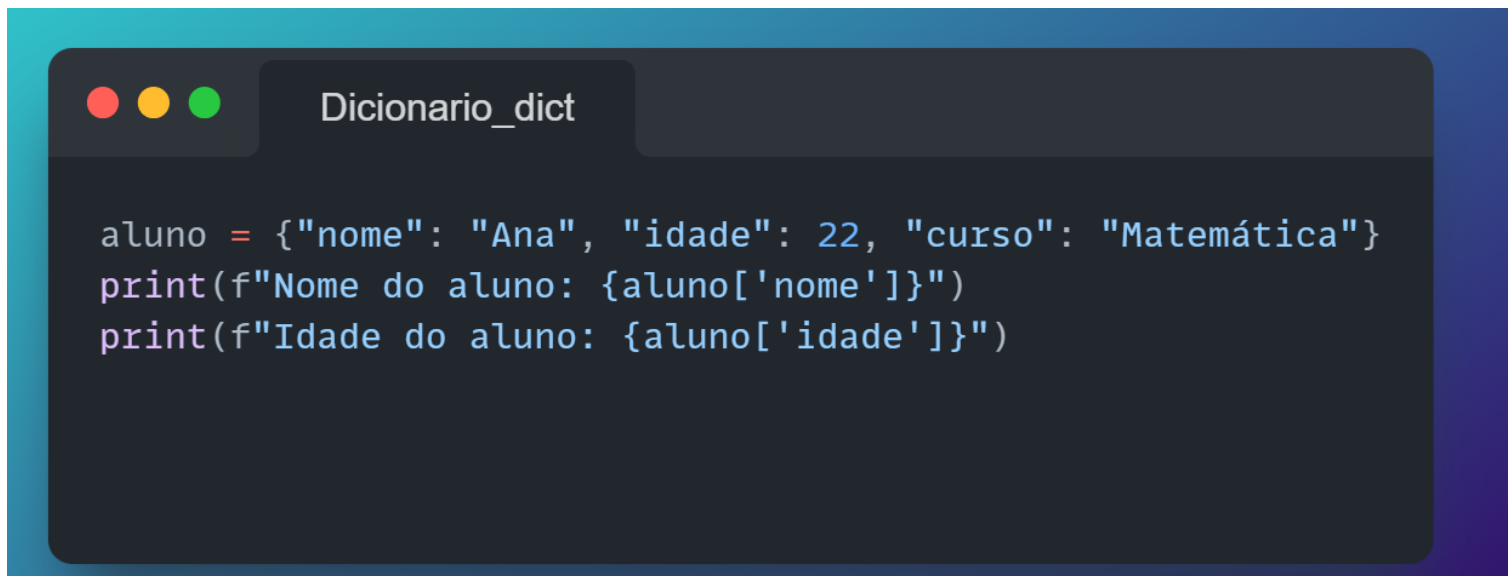
06

Dicionários (dict)

Dicionários são coleções desordenadas de pares chave-valor, onde cada chave é única.

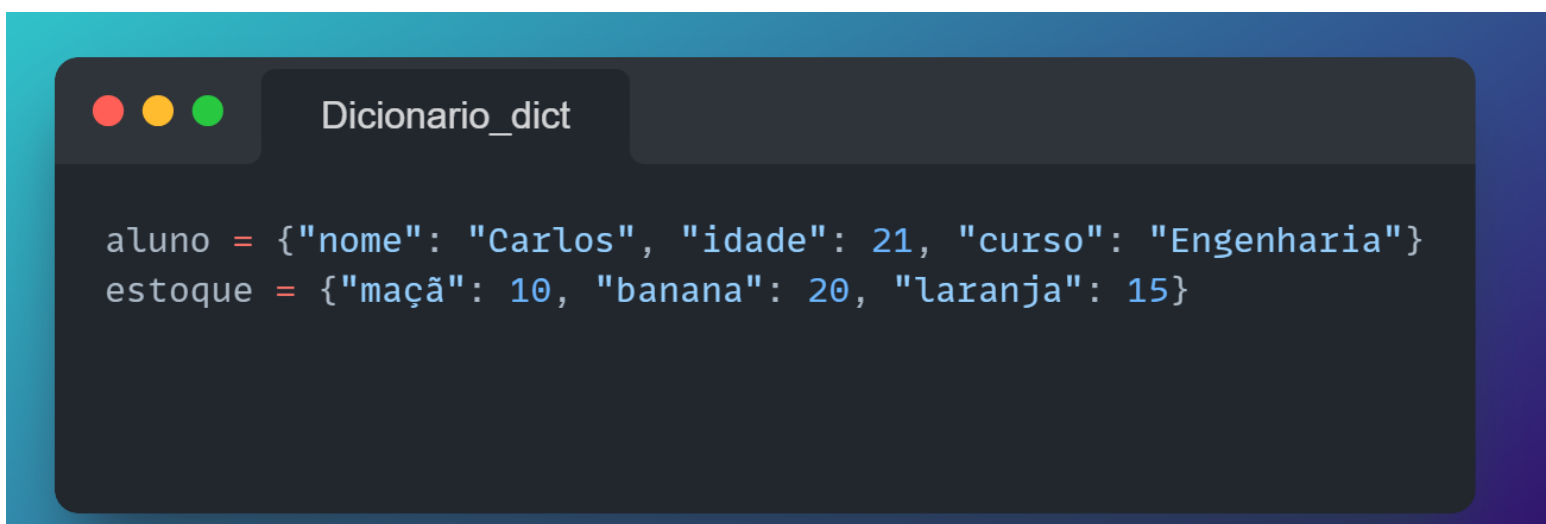
Python Kombat

Exemplo:



```
aluno = {"nome": "Ana", "idade": 22, "curso": "Matemática"}
print(f"Nome do aluno: {aluno['nome']}")
print(f"Idade do aluno: {aluno['idade']}")
```

Contexto real: Armazenar informações de um aluno e acessá-las.



```
aluno = {"nome": "Carlos", "idade": 21, "curso": "Engenharia"}
estoque = {"maçã": 10, "banana": 20, "laranja": 15}
```

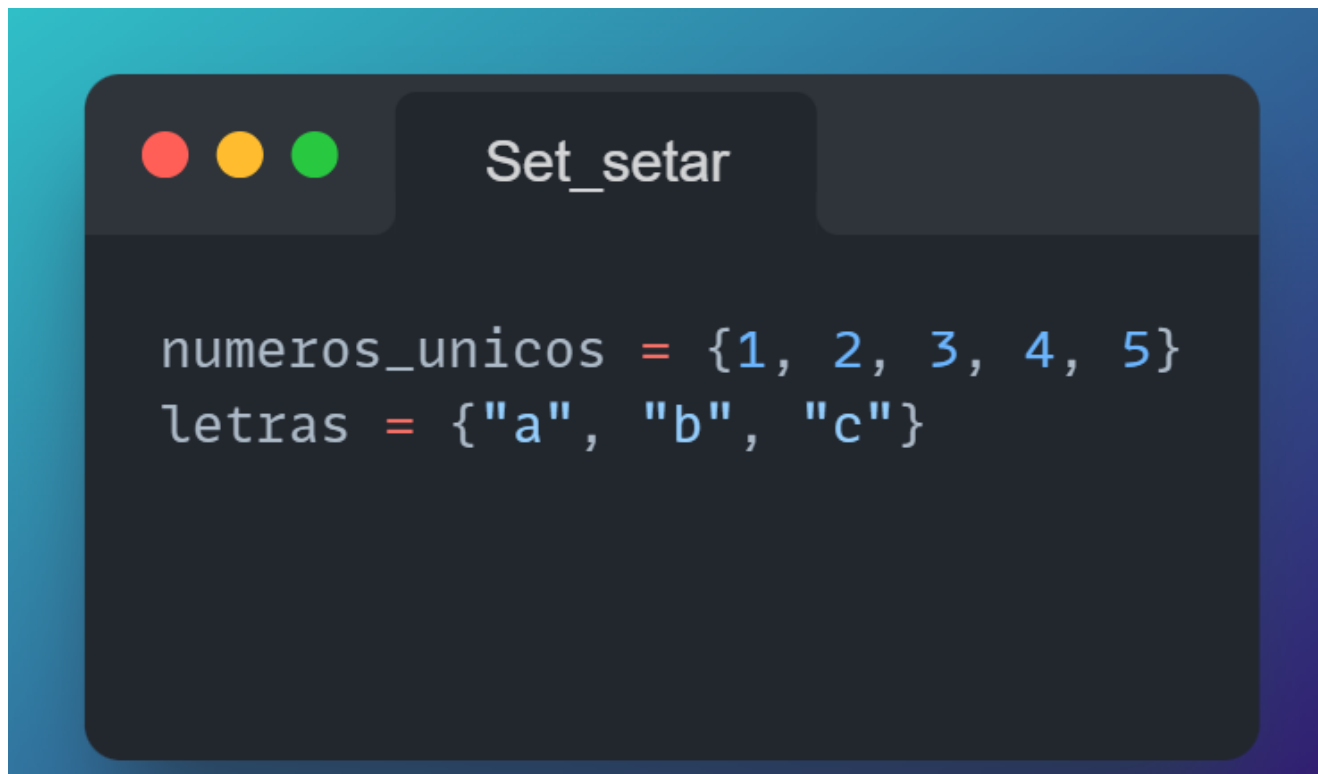
07

Conjuntos (SET)

Conjuntos são coleções desordenadas de itens únicos, sem valores duplicados.

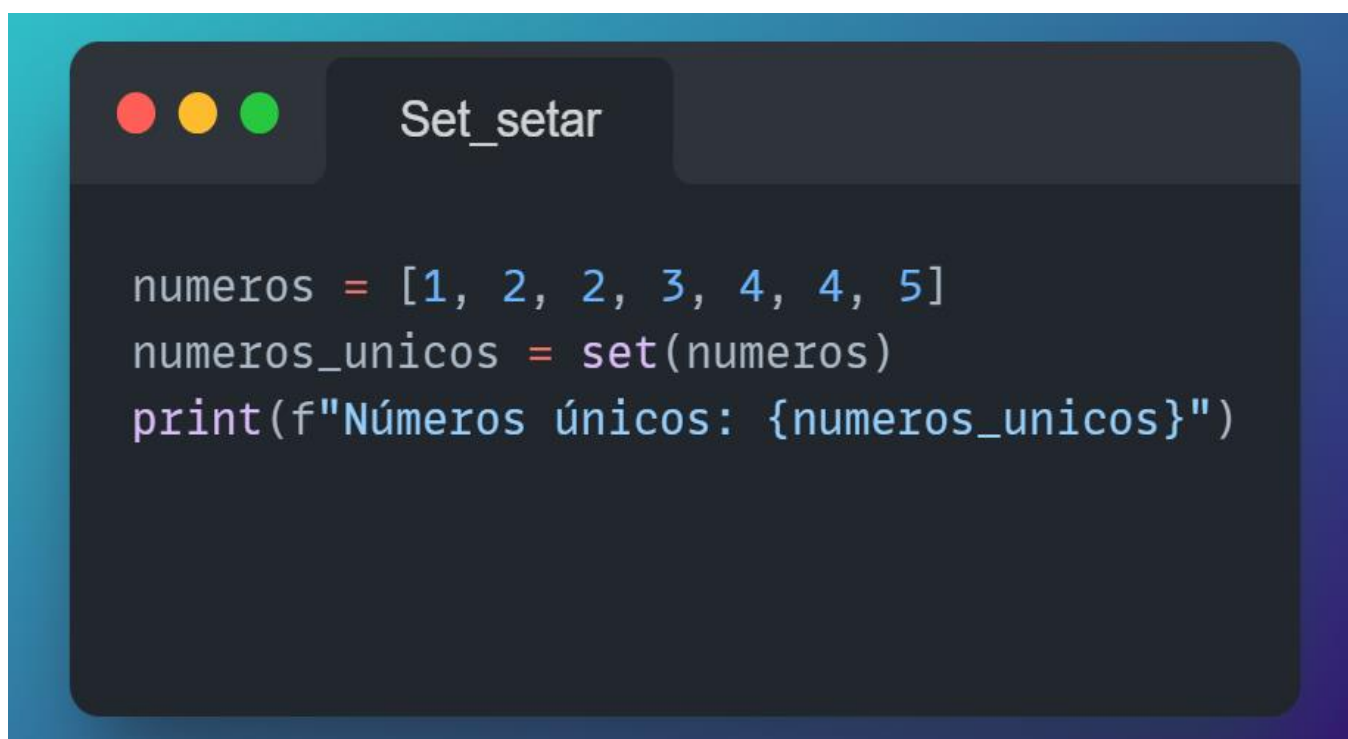
Python Kombat

Exemplo:



```
numeros_unicos = {1, 2, 3, 4, 5}
letras = {"a", "b", "c"}
```

Contexto real: Remover duplicatas de uma lista de números.



```
numeros = [1, 2, 2, 3, 4, 4, 5]
numeros_unicos = set(numeros)
print(f"Números únicos: {numeros_unicos}")
```

Python Kombat

Conclusão:

Conhecer os principais tipos de dados em Python é essencial para qualquer programador. Compreender como e quando usá-los facilita a escrita de códigos mais eficientes e organizados. Pratique utilizando esses tipos de dados em seus projetos e veja como eles podem simplificar suas tarefas diárias.



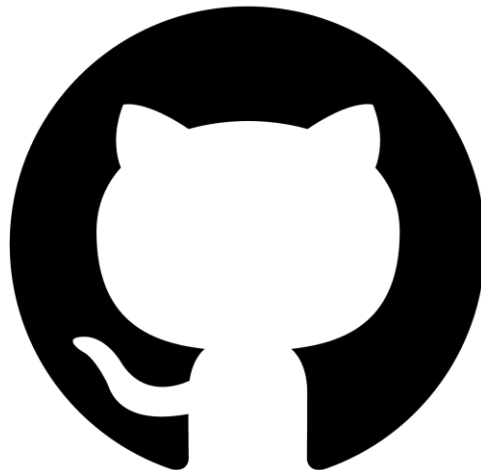


AGRADECIMENTOS

OBRIGADO POR LER ATÉ AQUI!

Esse Ebook foi elaborado por IA , e diagramado por humano. Esse conteúdo foi gerado para fins didáticos e fixação de conteúdo.

Se quiser saber um pouco mais sobre mim, deixarei aqui minhas redes.



<https://github.com/jonissonA>

