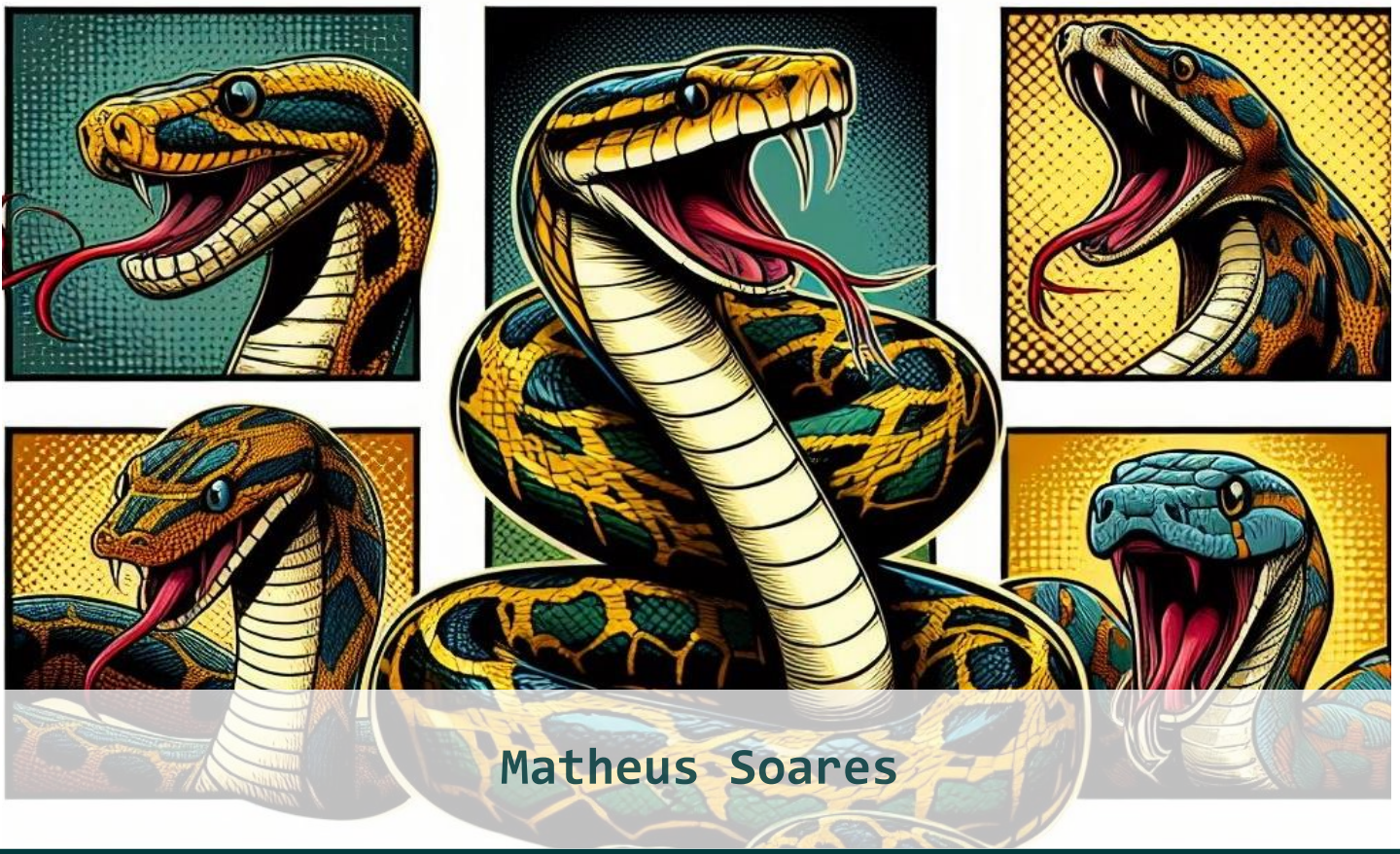
DOMINANDO PYTHON



Matheus Soares

EXPLORANDO O PODER DA SERPENTE DE DADOS

ESTRUTURAS DE DADOS

Um dos aspectos mais importantes do Python é sua rica variedade de estruturas de dados, que permitem organizar e manipular dados de maneira eficiente. Compreender e utilizar essas estruturas de dados é fundamental para escrever código Python eficiente e organizado. Cada estrutura tem suas próprias vantagens e aplicações, e escolher a correta para cada situação pode melhorar significativamente o desempenho e a clareza do seu código.

Introdução

Neste capítulo do eBook, exploraremos as principais estruturas de dados em Python, com explicações claras e exemplos práticos. Uma estrutura de dados é uma maneira de organizar e armazenar dados de uma forma que possa ser acessada e manipulada de maneira eficiente.

Existem várias estruturas de dados em Python, cada uma com suas próprias características e usos específicos. Algumas das estruturas de dados padrões em Python incluem,

1. **Listas (list)**: Uma lista é uma coleção ordenada de elementos que podem ser de diferentes tipos de dados. Os elementos em uma lista são acessados por meio de índices (posições), e a lista pode ser modificada adicionando, removendo ou alterando elementos.
2. **Tuplas (tuple)**: Uma tupla é semelhante a uma lista, mas é imutável, o que significa que seus elementos não podem ser alterados após a criação da tupla. As tuplas são geralmente usadas para representar coleções de dados heterogêneos e são acessadas por meio de índices.
3. **Dicionários (dict)**: Um dicionário é uma coleção não ordenada de pares chave-valor, onde cada chave é única e mapeada para um valor correspondente. Os valores em um dicionário são acessados por meio de suas chaves, e os dicionários são úteis para armazenar e recuperar dados de forma eficiente.
4. **Conjuntos (set)**: Um conjunto é uma coleção não ordenada de elementos únicos e imutáveis. Os conjuntos são usados principalmente para realizar operações de conjunto, como união, interseção e diferença, e são úteis para remover duplicatas de outras coleções.

Listas: Armazenando Coleções de Itens

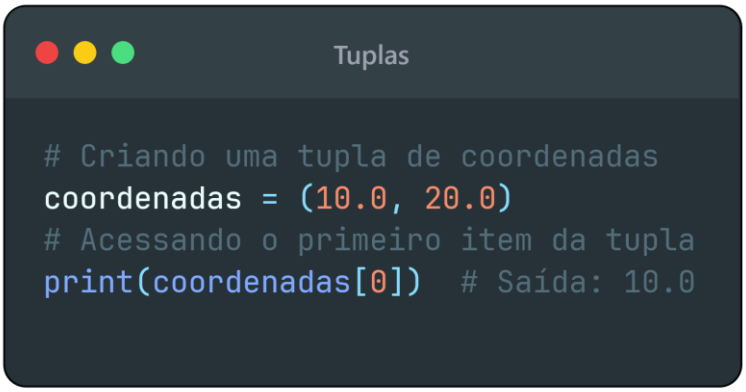
As listas são uma das estruturas de dados mais usadas em Python. Elas permitem armazenar uma coleção ordenada de itens que podem ser modificados.

Um conjunto de parâmetros consiste em uma lista com nenhum ou mais elementos. Porém é possível uma lista com um único parâmetro. As listas armazenam tipos diferentes de dados.
(Fonte: Alura)

Listas em Python podem armazenar de maneira sequencial qualquer tipo de objeto. Podemos criar listas utilizando o construtor list, a função range ou colocando valores separados por vírgula dentro de colchetes. Listas são objetos mutáveis, portanto podemos alterar seus valores após a criação.
(Fonte: DIO)

Tuplas: Coleções Imutáveis

As tuplas são similares às listas, mas são imutáveis, ou seja, uma vez criadas, seus valores não podem ser alterados.



Tuplas são estruturas de dados muito parecidas com as listas, a principal diferença é que tuplas são imutáveis enquanto listas são mutáveis. Podemos criar tuplas através da classe tuple, ou colocando valores separados por vírgula de parênteses.
(Fonte: DIO)

Conjuntos: Coleções de Itens Únicos

Conjuntos são coleções não ordenadas de itens únicos. Eles são úteis para eliminar duplicatas e realizar operações matemáticas como união e interseção.

Conjuntos

```python
# Criando um conjunto de números
numeros = {1, 2, 3, 4, 5}
# Adicionando um item ao conjunto
numeros.add(6)
# Removendo duplicatas de uma lista usando um conjunto
lista_com_duplicatas = [1, 2, 2, 3, 4, 4, 5]
lista_sem_duplicatas = list(set(lista_com_duplicatas))
print(lista_sem_duplicatas)  # Saída: [1, 2, 3, 4, 5]
```

Dicionários: Pares de Chave-Valor

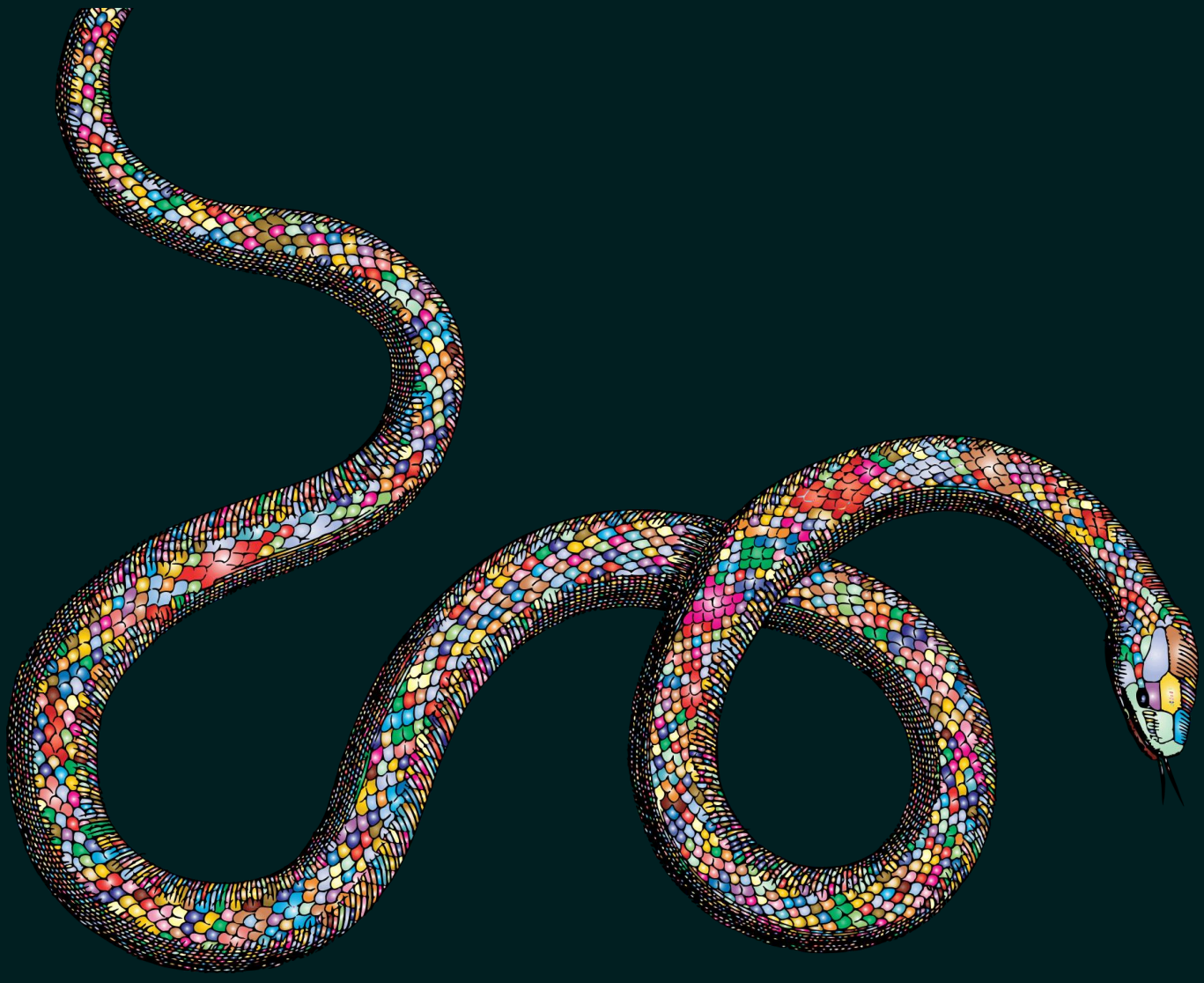
Dicionários são coleções de pares chave-valor. Eles permitem associar valores específicos a chaves únicas, facilitando a busca e manipulação de dados.

Dicionário

```
# Criando um dicionário de informações de contato
contato = {"nome": "João", "telefone": "1234-5678"}
# Acessando o telefone de João
print(contato["telefone"])  # Saída: 1234-5678
# Adicionando um novo par chave-valor
contato["email"] = "joao@example.com"
```

Os dicionários são coleções um pouco diferentes que representam um tipo de mapeamento, e mapeamento são coleções de associações entre dados e valores, em que o primeiro elemento é conhecido como chave e o segundo valor.

- Listas são estruturas de dados que representam um tipo básico de sequência.
- Mapeamentos são coleções sequenciais ordenadas, que utilizam índices para acessar os valores



AGRADECIMENTOS

OBRIGADO POR LER ATÉ AQUI

Esse Ebook foi possui contribuições do autor e de IA.
O passo a passo se encontra no meu GitHub

Esse conteúdo foi criado por meio dos conhecimentos adquiridos por cursos, bootcamp e trilhas de formação. O uso de IA serviu para estruturar o conteúdo e simplificar conteúdo.

https://github.com/matheussooares/python-language