Introdução à Python para análise de dados

DESENVOLVE
programa de formação
e inclusão de talentos
em tecnologia

Grupo Boticário

KORU

INTRODUÇÃO À PYTHON PARA ANÁLISE DE DADOS

Python é uma linguagem de programação de propósito geral criada por Guido van Rossum no final dos anos 1980 e lançada em 1991. Seu nome é uma homenagem ao grupo de comédia Monty Python. Uma das características mais apreciadas do Python é sua sintaxe clara e legível, que facilita o aprendizado e a colaboração.

A utilidade do Python se estende por diversas áreas, como desenvolvimento web, automação de tarefas, inteligência artificial e, de maneira crucial para nosso contexto, ciência de dados e análise. Para a análise de dados, o Python se tornou uma das linguagens mais populares entre os cientistas de dados devido à sua facilidade de uso e à sua comunidade ativa.

O ECOSSISTEMA PYTHON PARA ANÁLISE DE DADOS

Pandas

Ferramenta essencial para manipulação e análise de dados tabulares, com estruturas como DataFrames e Series que facilitam o trabalho com dados heterogêneos.

NumPy

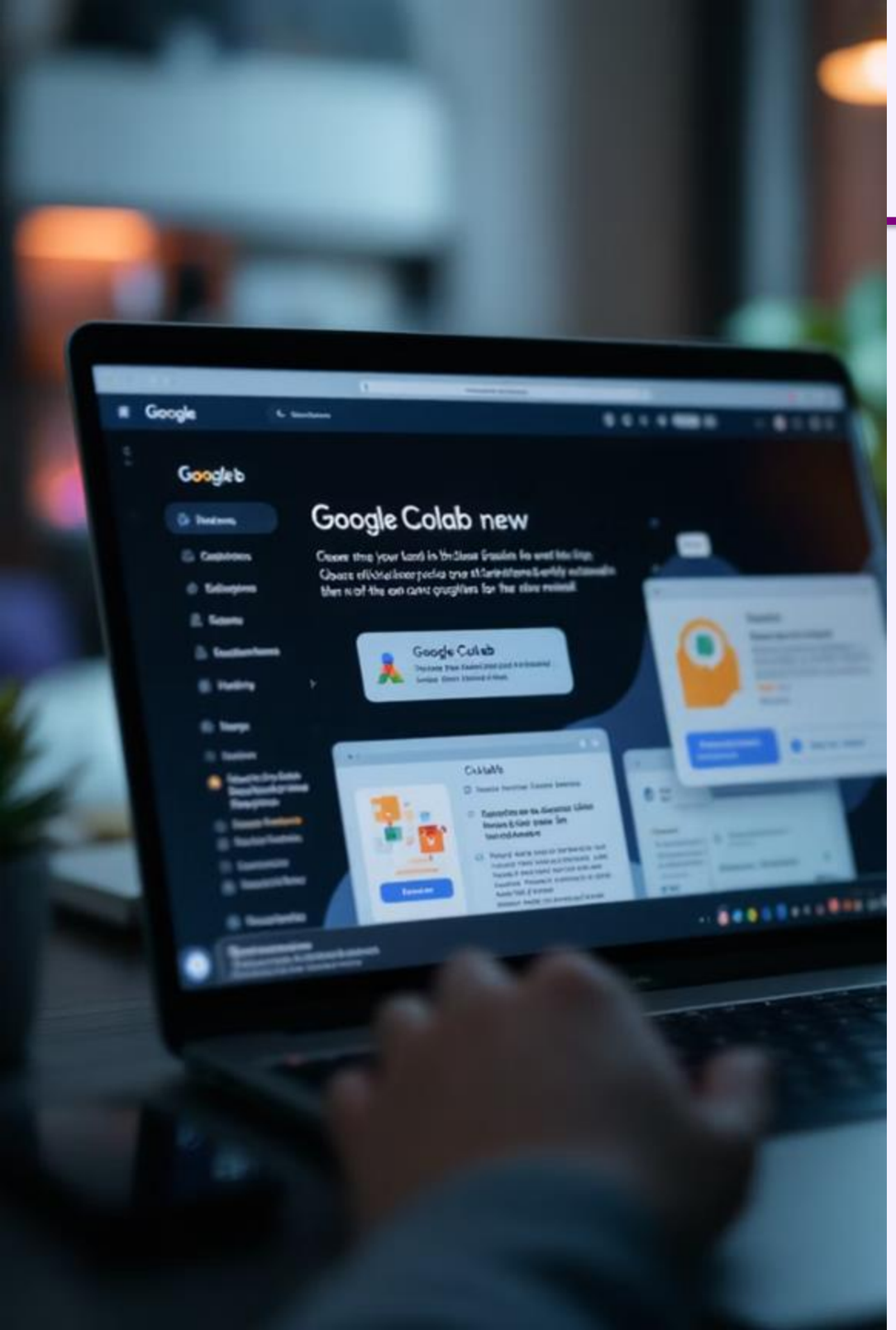
Base para cálculos numéricos e operações eficientes com arrays, suportando computação científica e matemática avançada.

Matplotlib

Biblioteca principal para a criação de visualizações e gráficos, facilitando a análise e comunicação dos dados.

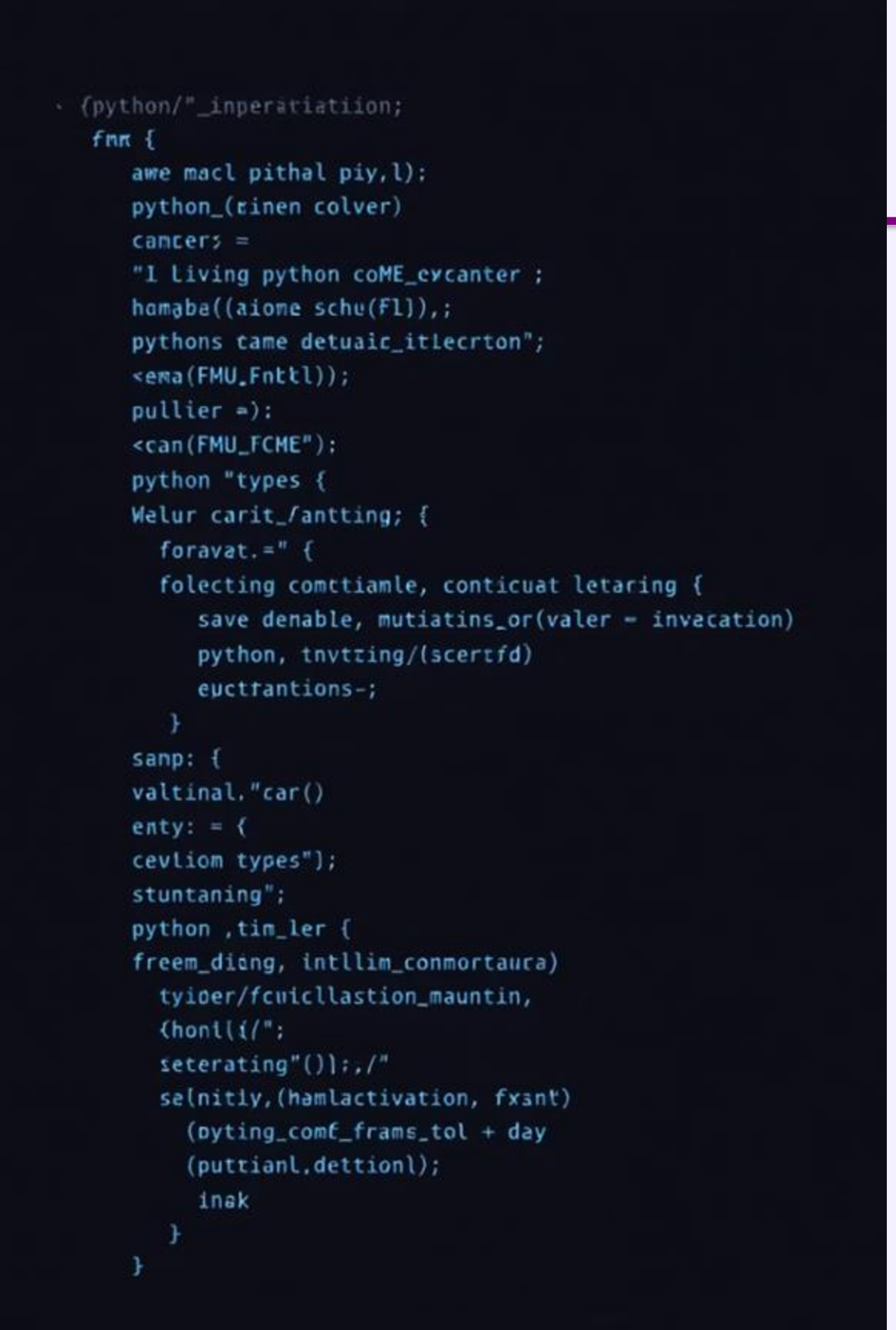
Scikit-learn

Biblioteca para aprendizado de máquina e modelagem preditiva, indispensável para análises avançadas e desenvolvimento de modelos.



AMBIENTE DE TRABALHO PYTHON

O Google Colab é um ambiente baseado em nuvem que permite escrever e executar código Python diretamente do navegador, sem necessidade de instalação local. É a ferramenta que utilizaremos para todas as práticas e análises, facilitando o acesso e o compartilhamento de notebooks.



SINTAXE BÁSICA E TIPOS DE DADOS

Tipos de Dados Básicos

- Inteiro (int): Para números inteiros, essenciais para contagens e indexação
- Decimal (float): Para números reais com parte decimal
- Texto (str): Sequências de caracteres para dados não numéricos
- Booleano (bool): Representa valores lógicos True e False

Operadores

- Aritméticos: +, -, *, /
- Comparação: >, <, ==, !=, >=, <=
- Lógicos: and, or, not

Indentação

Em Python, a indentação (espaços no início da linha) define blocos de código, contribuindo para a legibilidade. Dois-pontos (:) marcam o início de um bloco indentado.

ESTRUTURAS CONDICIONAIS E DE REPETIÇÃO

Instrução if-else

Permite que o programa tome decisões baseadas em condições. O bloco if é executado quando a condição é verdadeira, e o bloco else quando é falsa.

Expressões Booleanas

Fundamentais para o controle de fluxo, são avaliadas como True ou False e determinam quais blocos de código serão executados.

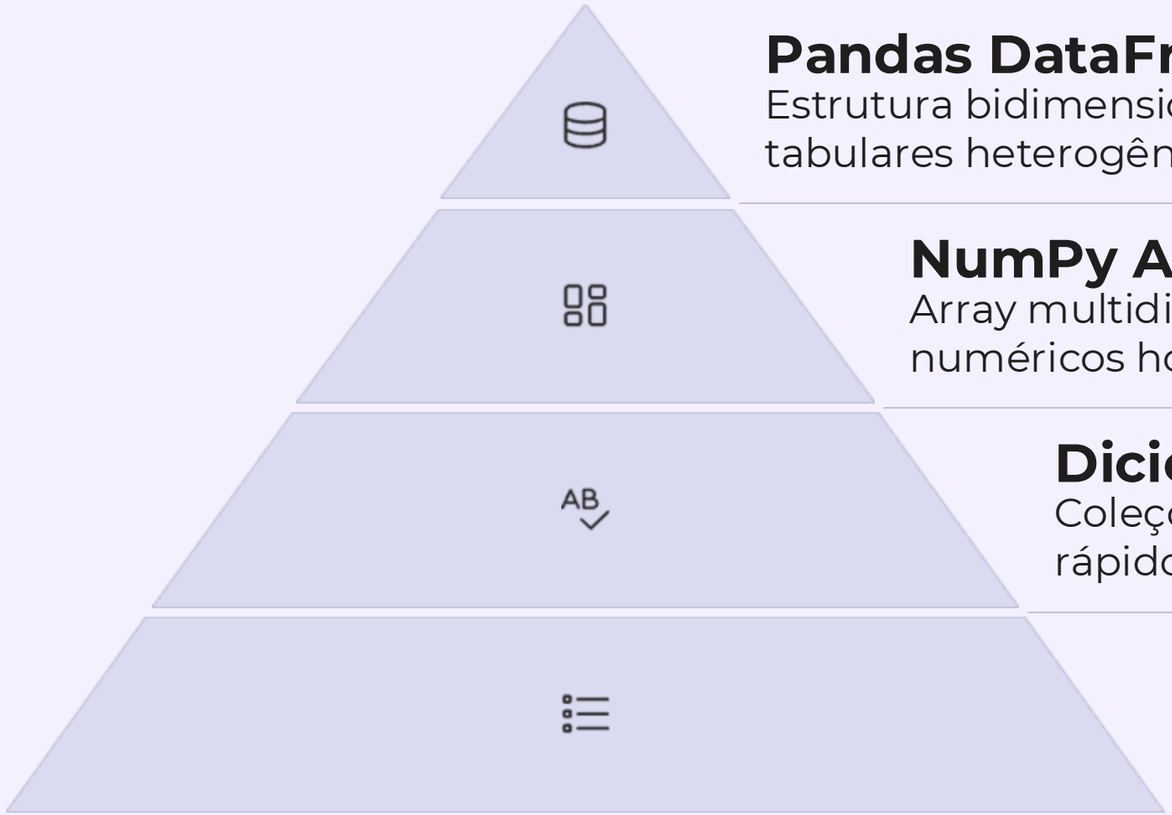
Loop for

Utilizado para iterar sobre objetos iteráveis como strings, listas e tuplas. Executa um bloco de código para cada item da sequência.

Loop while

Executa um bloco de código enquanto uma condição específica for verdadeira. A condição é verificada antes de cada iteração.

ESTRUTURAS DE DADOS EM PYTHON



Pandas DataFrame
Estrutura bidimensional para dados tabulares heterogêneos

NumPy Array
Array multidimensional para dados numéricos homogêneos

Dicionários
Coleções de pares chave-valor para acesso rápido por chave

Listas e Tuplas
Sequências ordenadas de elementos (mutáveis e imutáveis)

A escolha entre estas estruturas depende da natureza dos dados e das operações necessárias. Listas são ideais para coleções ordenadas que podem ser modificadas, dicionários para associar valores a chaves únicas, tuplas para dados imutáveis, e estruturas avançadas como NumPy arrays e Pandas DataFrames para análise de dados eficiente.

PANDAS: A FERRAMENTA ESSENCIAL PARA DADOS

Importação e Criação

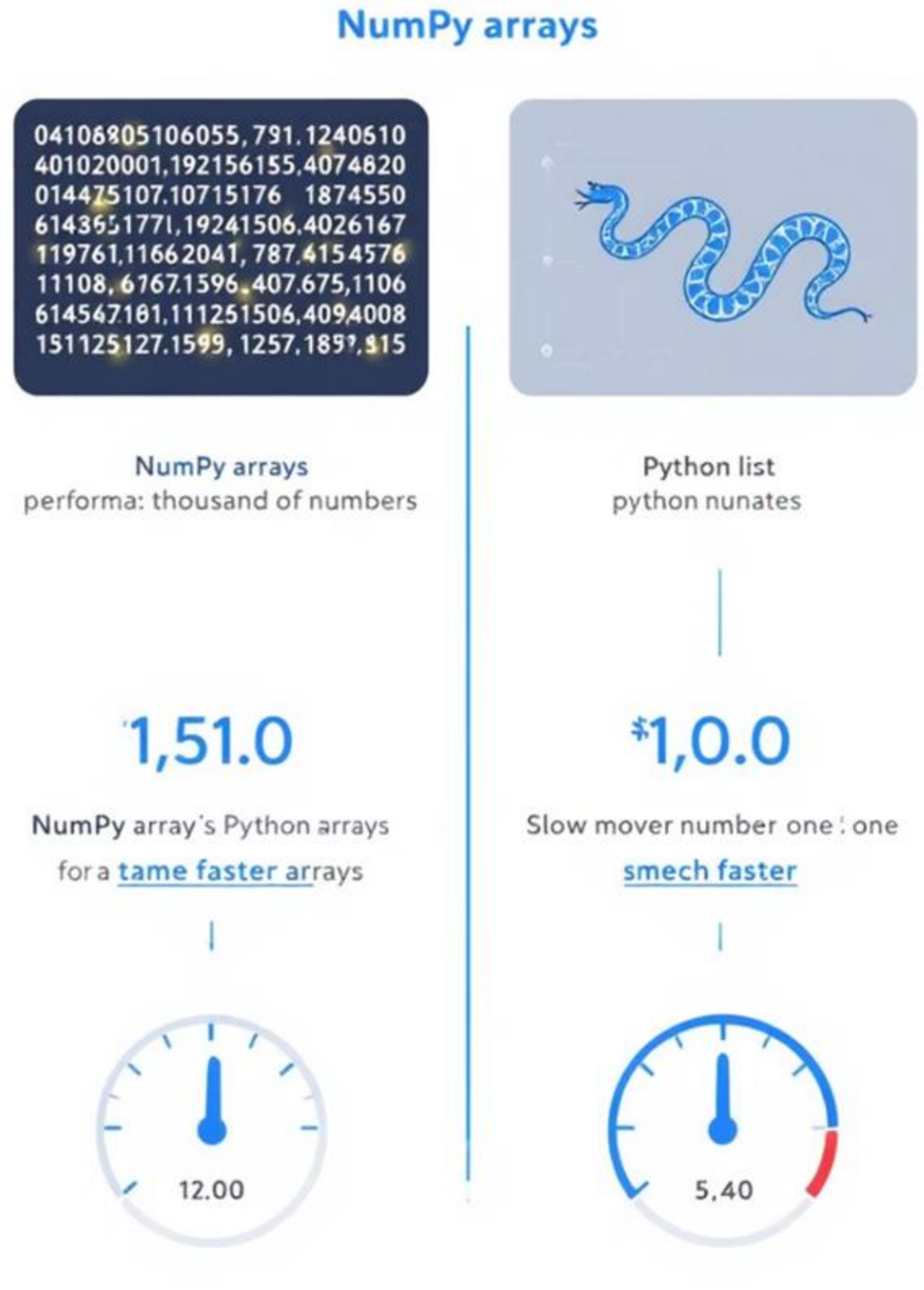
Importe o Pandas com "import pandas as pd" e crie estruturas como Series (unidimensional) e DataFrame (bidimensional) para organizar seus dados.

Carregamento de Dados

Utilize funções como pd.read_csv() para carregar dados de arquivos em DataFrames, facilitando a análise de dados tabulares de diversas fontes.

Exploração e Manipulação

Explore seus dados com métodos como .head(), .info() e .describe(), e manipule-os com operações de filtragem, agrupamento e transformação.



NUMPY: O PODER DA COMPUTAÇÃO NUMÉRICA

Característica	Descrição	Benefício
Arrays Multidimensionais	Estruturas homogêneas para dados numéricos	Operações vetorizadas rápidas
Funções Universais (ufuncs)	Operações elemento a elemento em arrays	Processamento eficiente sem loops explícitos
Armazenamento Otimizado	Dados em blocos contíguos de memória	Menor consumo de memória que sequências Python
Integração com C/C++/Fortran	API C fácil de usar	Facilita a integração com código legado

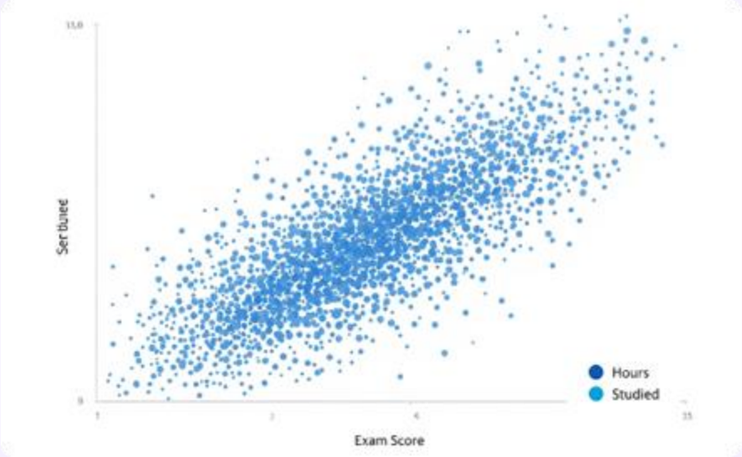
MATPLOTLIB: VISUALIZANDO SEUS DADOS

Gráficos de Linha

Ideais para mostrar tendências ao longo do tempo, conectando pontos de dados sequenciais. Muito utilizados para séries temporais e evolução de métricas.

Gráficos de Barras

Perfeitos para comparar quantidades entre diferentes categorias. Podem ser verticais (colunas) ou horizontais, dependendo da necessidade de visualização.



Gráficos de Dispersão

Fundamentais para visualizar relações entre duas variáveis quantitativas, identificando padrões, correlações e valores atípicos nos dados.

A visualização de dados é crucial na análise, permitindo identificar padrões que não seriam evidentes em tabelas numéricas. O Matplotlib oferece controle detalhado sobre todos os aspectos dos gráficos, desde cores e estilos até anotações e elementos personalizados.


NOS VEMOS NA
PRÓXIMA AULA!


Momento de
avaliar a aula
de hoje!