

A SOCIEDADE DO python

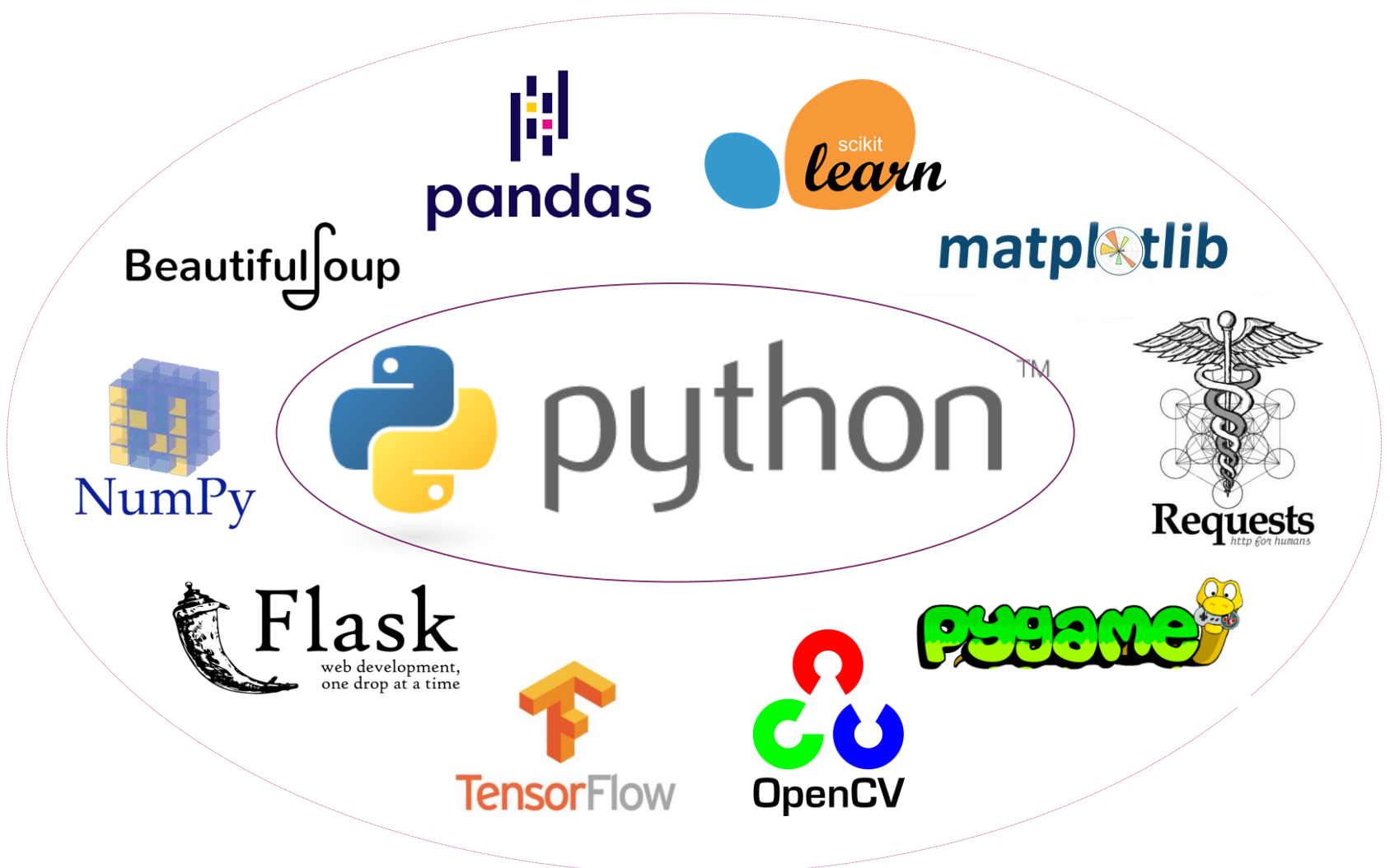
uma jornada na programação



Amanda Letícia P. Medeiros

Bibliotecas Python que você precisa conhecer

Python é uma linguagem de programação poderosa com um rico ecossistema de bibliotecas que a tornam versátil e eficiente para uma ampla gama de aplicações. Vamos explorar algumas das bibliotecas Python mais essenciais com explicações simples e exemplos de código em contextos reais.



01

NUMPY



NumPy: Computação Numérica

NumPy é o pacote fundamental para computação científica com Python. Ele oferece suporte para arrays, matrizes e muitas funções matemáticas.

```
● ● ● Operações básicas com arrays

import numpy as np

# Cria um array 1D
arr = np.array([1, 2, 3, 4, 5])

# Realiza operações básicas
print(arr + 5) # [ 6  7  8  9 10]
print(np.mean(arr)) # 3.0
```





PANDAS



Pandas: Manipulação de Dados

Pandas é essencial para análise e manipulação de dados, fornecendo estruturas de dados como DataFrames para lidar com dados estruturados.

```
● ● ● Operações com DataFrame

import pandas as pd

# Cria um DataFrame
data = {
    'Nome': ['Alice', 'Bob', 'Charlie'],
    'Idade': [24, 27, 22]
}
df = pd.DataFrame(data)

# Exibe informações básicas
print(df)
print(df.describe())
```



03

MATPLOTLIB



Matplotlib: Visualização de Dados

Matplotlib é uma biblioteca de plotagem para criar visualizações estáticas, interativas e animadas em Python.

```
● ● ● Plotando um gráfico simples

import matplotlib.pyplot as plt

# Dados
x = [1, 2, 3, 4, 5]
y = [10, 15, 7, 10, 5]

# Cria um gráfico de linha
plt.plot(x, y)
plt.xlabel('Eixo X')
plt.ylabel('Eixo Y')
plt.title('Gráfico de Linha Simples')
plt.show()
```





SCIKIT-LEARN



Scikit-Learn: Aprendizado de Máquina

Scikit-Learn é uma biblioteca poderosa para aprendizado de máquina, oferecendo ferramentas simples e eficientes para mineração e análise de dados.

```
Regressão linear

from sklearn.linear_model import LinearRegression
import numpy as np

# Dados de exemplo
X = np.array([[1], [2], [3], [4], [5]])
y = np.array([1, 3, 2, 5, 4])

# Cria um modelo de regressão linear
modelo = LinearRegression()
modelo.fit(X, y)

# Faz previsões
previsoes = modelo.predict(X)
print(previsoes)
```



05

REQUESTS



Requests: Requisições HTTP

Requests é uma biblioteca simples e elegante para fazer requisições HTTP em Python.

```
Fazendo uma requisição GET

import requests

# Faz uma requisição GET
resposta = requests.get('https://api.github.com')

# Verifica o código de status da resposta
print(resposta.status_code)

# Exibe o conteúdo da resposta
print(resposta.json())
```



06

BEAUTIFUL SOUP



Beautiful Soup: Web Scraping

Beautiful Soup é uma biblioteca para parsing de documentos HTML e XML, facilitando a extração de informações de páginas da web.

```
Scraping de uma Página Web

from bs4 import BeautifulSoup
import requests

# Busca a página web
resposta = requests.get('https://www.example.com')

# Faz o parsing do conteúdo HTML
sopa = BeautifulSoup(resposta.content, 'html.parser')

# Encontra e imprime o título da página
titulo = sopa.find('title')
print(titulo.text)
```





TENSORFLOW



TensorFlow: Aprendizado Profundo

TensorFlow é uma biblioteca open-source para computação numérica e aprendizado de máquina, particularmente adequada para treinar e implantar modelos de aprendizado profundo.

Operação Básica com TensorFlow

```
import tensorflow as tf

# Define uma constante
ola = tf.constant('Olá, TensorFlow!')

# Cria uma sessão para executar a constante
with tf.Session() as sess:
    print(sess.run(ola))
```



08

FLASK



Flask:

Desenvolvimento Web

Flask é um framework web leve para construir aplicações web em Python.

● ● ● Criando uma Aplicação Web Simples

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def home():
    return "Olá, Flask!"

if __name__ == '__main__':
    app.run(debug=True)
```



09

PYGAME



Pygame: Desenvolvimento de Jogos

Pygame é uma biblioteca para desenvolvimento de jogos. Ela fornece funcionalidades para criar gráficos, sons e interação do usuário, permitindo a criação de jogos simples e complexos.

```
● ● ● Criando uma Janela de Jogo Simples

import pygame

# Inicializa o Pygame
pygame.init()
# Define as dimensões da janela
screen = pygame.display.set_mode((640, 480))
# Define a cor da janela
white = (255, 255, 255)
screen.fill(white)
# Atualiza a janela
pygame.display.flip()
# Loop principal do jogo
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

pygame.quit()
```



10

OPENCV



OpenCV: Processamento de Imagens

OpenCV é uma biblioteca open-source para visão computacional e processamento de imagens. Ela oferece ferramentas para capturar, processar e analisar imagens e vídeos.

Carregando e Exibindo uma Imagem

```
import cv2

# Carrega a imagem
image = cv2.imread('example.jpg')

# Exibe a imagem em uma janela
cv2.imshow('Imagem', image)

# Aguarda a tecla 'q' ser pressionada para fechar a janela
while True:
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cv2.destroyAllWindows()
```



AGRADECIMENTOS



Obrigada por ler!

Essas bibliotecas representam apenas uma pequena fração das poderosas ferramentas disponíveis no ecossistema Python. Ao incorporar essas bibliotecas em seus projetos, você pode melhorar significativamente suas capacidades e eficiência na programação. Continue explorando, aprendendo e criando soluções incríveis com Python. Muito sucesso em suas futuras aventuras de programação!

Esse Ebook foi gerado por IA, e diagramado por humano. O passo a passo se encontra no meu GitHub:



Boas codificações e até a próxima!

