

## Funções do Básico ao Avançado

### 1. Calculadora Simples:

Crie uma calculadora simples que recebe dois números e realiza operações de soma, subtração, multiplicação e divisão.

### 2. Funções para sortear e somar

Faça um programa que tenha uma lista chamada números e duas funções chamadas sorteia() e somaPar(). A primeira função vai sortear 5 números e vai colocá-los dentro da lista e a segunda função vai mostrar a soma entre todos os valores pares sorteados pela função anterior.

### 3. Função que descobre o maior

Faça um programa que tenha uma função chamada maior(), que receba vários parâmetros com valores inteiros. Seu programa tem que analisar todos os valores e dizer qual deles é o maior.

### 4. Função que calcula área

Faça um programa que tenha uma função chamada área(), que receba as dimensões de um terreno retangular (largura e comprimento) e mostre a área do terreno.

### 5. Verificador de Números Pares:

Escreva uma função que recebe varios números e verifica se é par. A função deve retornar **True** se o número for par e **False** caso contrário.

### 6. Contador de Palavras:

Crie uma função que recebe uma frase como entrada e conta o número de palavras na frase.

### 7. Tabuada:

Crie um programa que gera a tabuada de um número fornecido pelo usuário.

### 8. Adivinhe o Número:

Desenvolva um jogo onde o computador escolhe um número aleatório e o jogador tem que adivinhar qual é. O programa deve fornecer dicas se o palpite do jogador está alto ou baixo.

### **9. Validador de Senhas Fortes:**

Crie uma função que verifica se uma senha atende aos critérios de uma senha forte, incluindo comprimento mínimo, uso de letras maiúsculas, minúsculas, números e caracteres especiais.

### **10. Analisador de Números Primos:**

Desenvolva um programa que verifica se um número fornecido pelo usuário é primo.

### **11. Funções para votação**

Crie um programa que tenha uma função chamada voto() que vai receber como parâmetro o ano de nascimento de uma pessoa, retornando um valor literal indicando se uma pessoa tem voto NEGADO, OPCIONAL e OBRIGATÓRIO nas eleições.

### **12. Função para Fatorial**

Crie um programa que tenha uma função fatorial() que receba dois parâmetros: o primeiro que indique o número a calcular e outro chamado show, que será um valor lógico (opcional) indicando se será mostrado ou não na tela o processo de cálculo do fatorial.

### **13. Ficha do Jogador**

Faça um programa que tenha uma função chamada ficha(), que receba dois parâmetros opcionais: o nome de um jogador e quantos gols ele marcou. O programa deverá ser capaz de mostrar a ficha do jogador, mesmo que algum dado não tenha sido informado corretamente.

### **14. Analisando e gerando Dicionários**

Faça um programa que tenha uma função notas() que pode receber várias notas de alunos e vai retornar um dicionário com as seguintes informações:

- Quantidade de notas
- A maior nota
- A menor nota
- A média da turma
- A situação (opcional)

### **15. Interactive helping system in Python**

Faça um mini-sistema que utilize o Interactive Help do Python. O usuário vai digitar o comando e o manual vai aparecer. Quando o usuário digitar a palavra 'FIM', o programa se encerrará. Importante: use cores.

**16. Validador de Expressões Matemáticas:**

Escreva uma função que valida se uma expressão matemática fornecida pelo usuário é válida, considerando parênteses, colchetes e chaves.

***Exercícios em Python de nível intermediário, abordando funções com parâmetros, funções sem parâmetros, estruturas de repetição, estruturas condicionais, list comprehension, \*args, \*\*kwargs e outros conceitos avançados:***

**17. Soma e Media de Elementos:**

Crie uma função que recebe uma lista de números como parâmetro e retorna a soma de todos os elementos dessa lista. No Final Escreva uma função que calcula a média de uma lista de números fornecida como parâmetro.

**18. Inversão de String:**

Implemente uma função que recebe uma string como entrada e retorna a string invertida.

**19. Comprimento das Palavras:**

Crie uma função que recebe uma lista de palavras como parâmetro e retorna uma lista com o comprimento de cada palavra.

**20. Verificador de Palíndromos:**

Escreva uma função que verifica se uma palavra é um palíndromo, levando em conta maiúsculas e minúsculas.

**21. Média com args:**

Implemente uma função que utiliza \*args para calcular a média de uma quantidade variável de números.

**22. Dicionário de Complimentos com kwargs:**

Desenvolva uma função que utiliza \*\*kwargs para criar um dicionário com pares chave-valor, onde as chaves são strings e os valores são seus complimentos.

**23. Lista de Números Pares:**

Crie uma função que recebe uma lista de números como parâmetro e retorna uma lista apenas com os números pares.

**24. Ordenação de Strings por Comprimento:**

Desenvolva uma função que recebe uma lista de strings e retorna uma nova lista com as strings ordenadas por comprimento.

**25. Contagem de Vogais:**

Implemente uma função que recebe uma frase como parâmetro e conta o número de ocorrências de cada vogal.

**26. Elementos Comuns entre Listas:**

Crie uma função que recebe duas listas como parâmetro e retorna uma lista com os elementos comuns entre as duas listas.

**27. Lista de Números Ímpares:**

Escreva uma função que recebe uma lista de números como parâmetro e retorna uma lista apenas com os números ímpares.

**28. Simulação de Caixa Eletrônico:**

Implemente um programa que simula um caixa eletrônico, permitindo saques, depósitos e consulta de saldo

**29. Jogo da Velha:**

Desenvolva um programa que simula um jogo da velha para dois jogadores.

**30. União de Elementos Únicos entre Listas:**

Crie uma função que recebe duas listas como parâmetro e retorna uma lista com os elementos únicos de ambas as listas, sem repetições.

**Exercícios em Python de nível avançado, abrangendo funções com parâmetros, funções sem parâmetros, estruturas de repetição, estruturas condicionais, list comprehension, \*args, \*\*kwargs e conceitos mais avançados:**

**31. Algoritmo de Ordenação Avançado:**

Implemente uma função que utiliza um algoritmo de ordenação eficiente, como QuickSort ou MergeSort, para ordenar uma lista de números.

**32. Operações Assíncronas com asyncio:**

Crie um programa que utiliza asyncio para realizar operações assíncronas, como chamadas de API simultâneas, melhorando a eficiência do código.

**33. Análise de Sentimentos com NLP:**

Implemente uma função que utiliza uma biblioteca de processamento de linguagem natural (NLP) para realizar a análise de sentimentos em um texto.

**34. Web Scraping com BeautifulSoup ou Scrapy:**

Escreva um script que faz o scraping de dados de uma página web utilizando BeautifulSoup ou Scrapy.

**35. Autenticação Segura com JWT:**

Implemente um sistema de autenticação seguro utilizando tokens JWT (JSON Web Tokens) para garantir a segurança da comunicação entre cliente e servidor.

**36. Classificação com Machine Learning:**

Desenvolva uma função que utiliza machine learning para realizar a classificação de dados, utilizando uma biblioteca como Scikit-Learn.

**37. API RESTful com Flask ou Django:**

Crie um programa que utiliza Flask ou Django para construir uma API RESTful, proporcionando interações eficientes entre diferentes partes de uma aplicação.

**38. Agrupamento com K-means:**

Elabore uma função que utiliza o algoritmo K-means para realizar agrupamento de dados, identificando padrões complexos.

**39. Sistema de Recomendação com Filtragem Colaborativa:**

Implemente um sistema de recomendação de itens utilizando algoritmos de filtragem colaborativa, explorando preferências de usuários similares.

**40. Análise de Séries Temporais:**

Desenvolva uma função que realiza a análise de séries temporais, utilizando bibliotecas como Pandas e NumPy para identificar tendências e padrões temporais.

**41. Visualizações de Dados com Matplotlib:**

Crie um programa que utiliza a biblioteca Matplotlib para criar visualizações complexas de dados, facilitando a interpretação de informações.

**42. Classificação com Árvore de Decisão:**

Elabore uma função que utiliza o algoritmo de árvore de decisão para realizar a classificação de dados, explorando diferentes ramos de decisão.

**43. Automação com Selenium:**

Escreva um script que utiliza o Selenium para automatizar interações com páginas web, tornando o processo de teste e extração de dados mais eficiente.

**44. Treinamento de Modelo com TensorFlow ou PyTorch:**

Implemente uma função que realiza o treinamento de um modelo de aprendizado profundo utilizando frameworks como TensorFlow ou PyTorch.

**45. Processamento de Linguagem Natural Avançado:**

Desenvolva um sistema de processamento de linguagem natural que inclua análise de entidades, sentimentos e compreensão contextual.

**46. API Assíncrona com FastAPI:**

Crie uma aplicação que utiliza o framework FastAPI para construir uma API assíncrona, aproveitando a eficiência do código assíncrono.

**47. Análise de Redes Sociais:**

Elabore uma função que realiza a análise de redes sociais, identificando comunidades, influenciadores e padrões de conexão.

**48. Otimização Genética:**

Escreva um programa que utiliza o algoritmo de otimização genética para resolver um problema complexo, como otimização de parâmetros.

**49. Recomendação de Filmes com Aprendizado Profundo:**

Implemente um sistema de recomendação de filmes baseado em aprendizado profundo, explorando arquiteturas de redes neurais.

**50. Análise de Importância de Páginas com PageRank:**

Desenvolva uma função que utiliza o algoritmo de PageRank para analisar a importância de páginas da web em uma rede.

**51. Processamento Avançado de Linguagem Natural com NLTK:**

Crie um script que utiliza a biblioteca Natural Language Toolkit (NLTK) para realizar processamento avançado de linguagem natural.

**52. Análise Estatística Avançada com Statsmodels:**

Elabore uma função que utiliza a biblioteca Statsmodels para realizar análise estatística avançada, explorando modelos estatísticos complexos.

**53. Processamento de Áudio com Librosa:**

Escreva um programa que utiliza a biblioteca Librosa para realizar análise e processamento de áudio, como extração de características musicais.

**54. Reconhecimento de Objetos com TensorFlow e OpenCV:**

Desenvolva um script que utiliza TensorFlow e OpenCV para realizar o reconhecimento de objetos em imagens ou vídeos.

**55. Implementação de um Chatbot:**

Crie um sistema de chatbot utilizando uma biblioteca como Rasa ou ChatterBot, capaz de responder a perguntas e manter diálogos contextuais.

**56. Análise de Grafos com NetworkX:**

Elabore uma função que utiliza a biblioteca NetworkX para realizar análise de grafos, identificando centralidade, caminhos mínimos, entre outros.

**57. Sistema de Recomendação Baseado em Conteúdo:**

Implemente um sistema de recomendação baseado em conteúdo que sugira itens similares com base nas características dos itens e nas preferências do usuário.

**58. Integração de API de Machine Learning:**

Crie um programa que faz a integração com uma API de machine learning, enviando dados para previsões e processando os resultados.

**59. Processamento de Linguagem Natural com SpaCy:**

Desenvolva um script que utiliza a biblioteca SpaCy para realizar tarefas avançadas de processamento de linguagem natural, como análise sintática e named entity recognition (NER).

**60. Implementação de um Framework Web:**

Crie um pequeno framework web, inspirado em frameworks populares como Flask ou Django, para entender os conceitos fundamentais por trás dessas ferramentas.

**61. Animação Gráfica com Pygame:**

Escreva um programa que utiliza a biblioteca Pygame para criar animações gráficas interativas.

**62. Aprendizado Não Supervisionado com K-means:**

Desenvolva uma função que utiliza o algoritmo K-means para realizar aprendizado não supervisionado em dados complexos.



**63. Reconhecimento Facial com OpenCV e Dlib:**

Implemente um sistema de reconhecimento facial utilizando as bibliotecas OpenCV e Dlib.

**64. Sistema de Recomendação de Música com Spotipy:**

Crie um programa que utiliza a biblioteca Spotipy para construir um sistema de recomendação de músicas baseado no histórico de reprodução do usuário.

**65. Otimização de Código com Numba:**

Refatore uma função existente para aproveitar a otimização JIT (Just-In-Time) da biblioteca Numba.

**66. Análise de Sentimento Multilíngue com Transformers:**

Desenvolva uma função que utiliza modelos de transformers, como BERT ou GPT, para realizar análise de sentimento em textos multilíngues.

**67. Integração de Banco de Dados NoSQL:**

Crie um programa que faz a integração com um banco de dados NoSQL, como MongoDB ou Cassandra, para armazenar e recuperar dados.