

## **Listas Tuplas Dicionários etc.. Aninhados**

### **LISTAS ANINHADAS**

#### **1- Soma de Elementos:**

Crie uma função chamada soma elementos que recebe uma lista aninhada e retorna a soma de todos os seus elementos.

#### **2- Maior Elemento:**

Escreva uma função chamada maior elemento que encontra e retorna o maior elemento em uma lista aninhada.

#### **3- Média de Colunas:**

Implemente uma função chamada media colunas que calcula a média de cada coluna em uma matriz e retorna uma lista com os resultados.

#### **4- Multiplicação Escalar:**

Implemente uma função chamada multiplicação escalar que multiplica todos os elementos de uma lista aninhada por um número escalar.

#### **5- Contagem de Ocorrências:**

Escreva uma função chamada contagem ocorrências que conta o número de ocorrências de um elemento em uma lista aninhada.

#### **6- Listas Internas Ordenadas:**

Escreva uma função chamada listas\_internas\_ordenadas que verifica se todas as listas internas de uma lista aninhada estão ordenadas.

#### **7- Elemento Mais Comum por Coluna:**

Implemente uma função chamada elemento\_mais\_comum que encontra o elemento mais comum em cada coluna de uma matriz.

#### **8- Matriz Identidade:**

Escreva uma função chamada matriz\_identidade que verifica se uma matriz é uma matriz identidade.

**9- Adição de Coluna Extra:**

Implemente uma função chamada `adicionar_coluna_extra` que adiciona uma coluna extra a uma matriz, onde cada elemento é a soma dos elementos da coluna correspondente.

**10- Determinante de Matriz 3x3:**

Crie uma função chamada `determinante_matriz_3x3` que calcula o determinante de uma matriz 3x3.

**11- Matriz Mágica:**

Escreva uma função chamada `matriz_magica` que verifica se uma lista aninhada é uma matriz mágica.

**12- Lista composta e análise de dados**

Faça um programa que leia nome e peso de várias pessoas, guardando tudo em uma lista. No final, mostre:

- A) Quantas pessoas foram cadastradas.
- B) Uma listagem com as pessoas mais pesadas.
- C) Uma listagem com as pessoas mais leves.

**13- Listas com pares e ímpares**

Crie um programa onde o usuário possa digitar sete valores numéricos e cadastre-os em uma lista única que mantenha separados os valores pares e ímpares. No final, mostre os valores pares e ímpares em ordem crescente.

**14- Matriz em Python**

Crie um programa que declare uma matriz de dimensão 3x3 e preencha com valores lidos pelo teclado. No final, mostre a matriz na tela, com a formatação correta.

**15- Mais sobre Matriz em Python**

Aprimore o desafio anterior, mostrando no final:

- A) A soma de todos os valores pares digitados.
- B) A soma dos valores da terceira coluna.
- C) O maior valor da segunda linha.

#### **16- Palpites para a Mega Sena**

Faça um programa que ajude um jogador da MEGA SENA a criar palpites. O programa vai perguntar quantos jogos serão gerados e vai sortear 6 números entre 1 e 60 para cada jogo, cadastrando tudo em uma lista composta.

#### **17- Boletim com listas compostas**

Crie um programa que leia nome e duas notas de vários alunos e guarde tudo em uma lista composta. No final, mostre um boletim contendo a média de cada um e permita que o usuário possa mostrar as notas de cada aluno individualmente.

## **DICIONARIOS ANINHADOS**

#### **1- Acesso a Elementos:**

Crie um dicionário aninhado representando informações sobre um livro (título, autor, ano) e acesse cada parte individualmente.

#### **2- Adição de Elementos:**

Adicione um novo capítulo a um dicionário aninhado que representa um livro. Cada capítulo deve ter um título e uma lista de páginas.

#### **3- Atualização de Valores:**

Atualize o ano de publicação de um livro no dicionário aninhado.

#### **4- Remoção de Elementos:**

Remova um capítulo específico do dicionário aninhado do livro.

#### **5- Lista de Todos os Capítulos:**

Extraia e exiba uma lista de todos os títulos dos capítulos do dicionário aninhado do livro.

#### **6- Contagem de Elementos:**

Conte quantos capítulos existem no dicionário aninhado do livro.

#### **7- Ordenação de Capítulos:**

Ordene os títulos dos capítulos em ordem alfabética.

#### **8- Informações de Múltiplos Livros:**

Crie um dicionário que contém informações sobre vários livros, cada um representado por um dicionário aninhado.

#### **9- Busca por Autor:**

Escreva uma função que recebe o nome de um autor e retorna todos os livros desse autor em um dicionário aninhado.

#### **10- Média de Páginas por Capítulo:**

Calcule a média de páginas por capítulo em um livro representado por um dicionário aninhado.

#### **11- Dicionário em Python**

Faça um programa que leia nome e média de um aluno, guardando também a situação em um dicionário. No final, mostre o conteúdo da estrutura na tela.

#### **12- Jogo de Dados em Python**

Crie um programa onde 4 jogadores joguem um dado e tenham resultados aleatórios. Guarde esses resultados em um dicionário em Python. No final, coloque esse dicionário em ordem, sabendo que o vencedor tirou o maior número no dado.

#### **13- Cadastro de Trabalhador em Python**

Crie um programa que leia nome, ano de nascimento e carteira de trabalho e cadastre-o (com idade) em um dicionário. Se por acaso a CTPS for diferente de ZERO, o dicionário receberá também o ano de contratação e o salário. Calcule e acrescente, além da idade, com quantos anos a pessoa vai se aposentar.

#### **14- Cadastro de Jogador de Futebol**

Crie um programa que gerencie o aproveitamento de um jogador de futebol. O programa vai ler o nome do jogador e quantas partidas ele jogou. Depois vai ler a quantidade de gols feitos em cada partida. No final, tudo isso será guardado em um dicionário, incluindo o total de gols feitos durante o campeonato.

### 15- Unindo dicionários e listas

Crie um programa que leia nome, sexo e idade de várias pessoas, guardando os dados de cada pessoa em um dicionário e todos os dicionários em uma lista. No final, mostre:

- A) Quantas pessoas foram cadastradas
- B) A média de idade
- C) Uma lista com as mulheres
- D) Uma lista de pessoas com idade acima da média

### 16- Cadastro de Jogadores de Futebol

Crie um programa que gerencie o aproveitamento de vários jogadores de futebol. O programa vai ler o nome do jogador e quantas partidas ele jogou. Depois vai ler a quantidade de gols feitos em cada partida. No final, tudo isso será guardado em um dicionário, incluindo o total de gols feitos durante o campeonato. Mostre a sequência que de cada gol que o jogador fez do maior para o menor

## TUPLAS ANINHADAS

### 1- Acesso a Elementos:

Crie uma tupla aninhada representando informações sobre um ponto no espaço tridimensional e acesse cada coordenada individualmente.

### 2- Comprimento de Tuplas Internas:

Escreva uma função que aceita uma lista de tuplas e retorna uma lista com o comprimento de cada tupla interna.

### 3- Maior Elemento em Cada Tupla:

Escreva uma função que encontre o maior elemento em cada tupla de uma lista de tuplas.

### 4- Remoção de Elementos:

Remova um elemento específico de uma tupla aninhada.

### 5- Soma Element-wise:

Escreva uma função que some duas tuplas aninhadas element-wise.

**6- Ordenação de Tuplas:**

Ordene uma lista de tuplas com base no primeiro elemento de cada tupla.

**7- Contagem de Elementos:**

Dada uma lista de tuplas, conte quantas vezes um elemento específico aparece em todas as tuplas

**8- Multiplicação de Elementos:**

Escreva uma função que multiplica cada elemento de uma tupla aninhada por um número escalar.

**9- Verificação de Existência:**

Verifique se um elemento específico existe em uma tupla aninhada.

**10- Contagem de Tuplas com Comprimento Específico:**

Dada uma lista de tuplas, conte quantas tuplas têm um comprimento específico.

**11- Concatenação de Tuplas:**

Concatene duas tuplas aninhadas.

**12- Inversão de Tuplas Internas:**

Inverta a ordem dos elementos em cada tupla de uma lista de tuplas.

## CONJUNTOS

**1- Criação de Conjuntos Aninhados:**

Crie um conjunto aninhado contendo vários conjuntos de números inteiros.

**2- União de Conjuntos:**

Escreva uma função que recebe dois conjuntos aninhados e retorna a união de todos os conjuntos.

**3- Diferença de Conjuntos:**

Crie uma função que encontre a diferença entre dois conjuntos aninhados.

**4- Verificação de Subconjunto:**

Escreva uma função que verifica se todos os conjuntos em um conjunto aninhado são subconjuntos de um conjunto maior.

**5- Criação de Conjuntos Dinâmicos:**

Escreva uma função que cria conjuntos aninhados com base em parâmetros fornecidos.

**6- Contagem de Elementos Únicos:**

Dado um conjunto aninhado, conte quantos elementos únicos estão presentes em todos os conjuntos.

**7- Remoção de Conjuntos Vazios:**

Remova todos os conjuntos vazios de um conjunto aninhado.

**8- Combinação de Conjuntos:**

Crie um conjunto que contenha todas as combinações possíveis de pares de elementos dos conjuntos em um conjunto aninhado.

**9- Verificação de Disjunção:**

Escreva uma função que verifica se todos os conjuntos em um conjunto aninhado são disjuntos entre si.

**10- Ordenação de Conjuntos:**

Ordene os conjuntos em um conjunto aninhado com base no número de elementos que contêm.

**11- Média de Elementos:**

Calcule a média de todos os elementos em conjuntos aninhados de números.

#### **12- Conjuntos com Múltiplos Elementos Comuns:**

Encontre todos os conjuntos que compartilham pelo menos dois elementos em um conjunto aninhado.

#### **13- Verificação de Inclusão:**

Escreva uma função que verifica se um conjunto é incluído em todos os conjuntos de um conjunto aninhado.

#### **14- Adição de Conjuntos Dinâmicos:**

Escreva uma função que adiciona conjuntos a um conjunto aninhado com base em uma lista de conjuntos fornecida.

## **Sets**

1 - União de Sets Crie dois sets e realize a união entre eles, utilizando o método `union` e o operador `|`. Exercício 2: Interseção de Sets Elabore dois sets e encontre a interseção entre eles, utilizando o método `intersection` e o operador `&`.

2 - Diferença de Sets Crie dois sets e determine a diferença entre eles, utilizando o método `difference` e o operador `-`.

3 - Atualização de Sets Elabore dois sets e atualize o primeiro com os elementos do segundo, utilizando o método `update` e o operador `|=`.

4 - Remoção de Elementos em Comum Crie dois sets e remova os elementos comuns entre eles, utilizando o método `difference_update` e o operador `-=`.

5 - Verificação de Igualdade Desenvolva dois sets e verifique se são iguais, utilizando o método `==`.

6 - Adição de Elementos Crie um set e adicione novos elementos, utilizando o método `add`.

7 - Remoção de Elementos Elabore um set e remova um elemento específico, utilizando o método `remove` ou `discard`.



8 - Limpeza de Set Desenvolva um set e remova todos os elementos, utilizando o método `clear`.

9 - Verificação de Diferença Simétrica Crie dois sets e determine a diferença simétrica entre eles, utilizando o método `symmetric_difference` e o operador `^`.

10 - Conjunto Imutável Elabore um set e converta-o em um conjunto imutável (frozen set), utilizando a função `frozenset`.

11 - Concatenação de Sets Crie dois sets e concatene-os, convertendo o resultado para um set, utilizando a função `set()`.

12 - Remoção Aleatória de Elemento Desenvolva um set e remova um elemento aleatório, utilizando o método `pop`.

13 - Adição de Sets Disjuntos Elabore dois sets e adicione os elementos do segundo ao primeiro, excluindo elementos comuns, utilizando o método `isdisjoint`.

14 - Exclusão de Elemento pelo Valor Crie um set e exclua um elemento pelo seu valor, utilizando o método `discard` ou `remove`.

15 - Cópia Profunda de Sets Desenvolva um set e realize uma cópia profunda, utilizando o método `copy()`.

16 - Verificação de Subconjunto Próprio Elabore dois sets e verifique se um é subconjunto próprio do outro, utilizando o método `issubset` e `!=`.

17 - Atualização de Sets com Interseção Crie dois sets e atualize o primeiro incluindo apenas os elementos comuns aos dois, utilizando o método `intersection_update`.

# Collections

- 1: Contador de Elementos Utilize a classe Counter do módulo collections para contar a ocorrência de cada elemento em uma lista.
- 2: DefaultDict para Contagem Utilize um defaultdict para contar a ocorrência de cada elemento em uma lista, atribuindo um valor padrão de 0.
- 3: Agrupamento por Chave Crie uma função que utilize defaultdict para agrupar elementos de uma lista com base em uma chave específica.
- 4: Concatenação de Listas em Dict Utilize um defaultdict para criar um dicionário onde as chaves são letras e os valores são listas de palavras que começam com essa letra.
- 5: Encontrar Elemento com Maior Ocorrência Desenvolva uma função que utilize Counter para encontrar o elemento mais comum em uma lista.
- 6: Adição de Elementos em DefaultDict Utilize um defaultdict para adicionar elementos a grupos específicos, onde os grupos são determinados por uma chave.
- 7: Maior Elemento em DefaultDict Crie uma função que utilize defaultdict para encontrar o elemento com o maior valor em grupos específicos.
- 8: DefaultDict de Conjuntos Utilize um defaultdict para criar um dicionário onde as chaves são números e os valores são conjuntos de palavras associadas a esses números.
- 9: DefaultDict com Lista Aninhada Utilize um defaultdict para criar um dicionário onde as chaves são números e os valores são listas de palavras associadas a esses números.
- 10: Adição de Elementos em DefaultDict de Conjuntos Utilize um defaultdict onde os valores são conjuntos para adicionar elementos a conjuntos específicos.
- 11: DefaultDict com Lista de Tuplas Desenvolva um defaultdict onde os valores são listas de tuplas, associando cada tupla a uma chave.

12: Acesso a Elemento em DefaultDict Aninhado Desenvolva uma função que utilize defaultdict aninhados para acessar um elemento específico.

13: DefaultDict com Lista de Conjuntos Crie um defaultdict onde os valores são listas de conjuntos, associando cada conjunto a uma chave específica.

14: Concatenação de DefaultDicts Desenvolva uma função que utilize defaultdict para concatenar dois dicionários, onde os valores são listas.

15: DefaultDict de DefaultDict de Listas Crie um defaultdict onde os valores são defaultdict de listas, associando cada lista a uma chave específica.

16: DefaultDict de Listas de Dicionários Utilize um defaultdict para criar um dicionário onde as chaves são números e os valores são listas de dicionários associadas a esses números.

17: DefaultDict de DefaultDict de Conjuntos Crie uma estrutura utilizando defaultdict aninhados, onde os valores são conjuntos associados a duas chaves.

18: Counter de Caracteres em Lista de Strings Utilize Counter para contar a ocorrência de caracteres em uma lista de strings.