Tratamento de Exceções

Escreva um programa que solicite ao usuário dois números e tente realizar a divisão do primeiro pelo segundo. Utilize try-except para lidar com a exceção de divisão por zero e imprimir uma mensagem de erro apropriada.

Crie um programa que solicite ao usuário que digite um número inteiro. Use try-except para lidar com a exceção ValueError caso o usuário digite algo que não seja um número inteiro, imprimindo uma mensagem de erro.

Modifique o exercício anterior para que, caso ocorra um ValueError, o programa peça novamente ao usuário para digitar um número inteiro até que uma entrada válida seja fornecida.

Estrutura de Repetição for

Imprima todos os números pares de 0 a 20 utilizando um loop for e a função range().

Dada a lista de nomes ["Carlos", "Ana", "Beto", "Maria"], imprima "Olá, [nome]!" para cada nome na lista.

Calcule a soma de todos os números em uma lista dada: [10, 20, 30, 40, 50]. Utilize um loop for.

Imprima as letras da palavra "PYTHON" uma por linha utilizando um loop for.

Utilize um loop for com a cláusula break para encontrar o primeiro número divisível por 7 em uma lista de números: [5, 14, 21, 30, 35]. Ao encontrar, imprima o número e saia do loop.

Utilize um loop for com a cláusula continue para imprimir todos os números ímpares de 1 a 10.

Estrutura de Repetição while

Escreva um programa que peça ao usuário para digitar um número positivo. Enquanto o número digitado não for positivo, continue pedindo a entrada.

Simule um contador regressivo de 10 até 0 utilizando um loop while. Imprima cada número e, ao chegar em 0, imprima "Fogo!".

Crie um jogo simples de adivinhação. Gere um número aleatório entre 1 e 100. Peça ao usuário para adivinhar o número. Enquanto a adivinhação estiver incorreta, informe se o número a ser adivinhado é maior ou menor. Use um loop while.

Utilize um loop while com a cláusula break para permitir que o usuário digite senhas até acertar a senha correta (defina uma senha correta no seu código).

Utilize um loop while com a cláusula continue para imprimir apenas os números pares digitados pelo usuário (o loop deve continuar até que o usuário digite um número negativo para sair).

Escreva um programa que calcule a média de números digitados pelo usuário. O programa deve continuar pedindo números até que o usuário digite 0. Utilize um loop while.

Listas

Crie uma lista de suas 5 comidas favoritas.

Imprima o primeiro item da lista.

Imprima o último item da lista.

Adicione mais uma comida ao final da lista usando append().

Insira uma comida no início da lista usando insert().

Remova uma comida específica da lista usando remove().

Remova a última comida da lista usando pop().

Imprima o tamanho atual da lista usando len().

Verifique se uma determinada comida está na sua lista usando o operador in.

Crie uma segunda lista de 3 outras comidas favoritas e combine as duas listas usando o operador +.

Ordene a lista combinada alfabeticamente usando sort().

Crie uma lista de números e encontre o maior e o menor número da lista sem usar as funções max() e min() diretamente (utilize um loop).

Escreva um programa que receba uma frase do usuário e crie uma lista com cada palavra da frase.

Crie duas listas de números com o mesmo tamanho. Crie uma terceira lista que contenha a soma dos elementos correspondentes das duas primeiras listas.Remova todos os elementos duplicados de uma lista.

Tuplas

Crie uma tupla com os nomes dos dias da semana em português.

Imprima o terceiro dia da semana.

Imprima o número de elementos na tupla usando len().

Verifique se a string "Sábado" está na tupla.

Tente adicionar um novo dia à tupla.

Crie uma tupla com números. Tente modificar o primeiro elemento da tupla.

Converta a tupla dos dias da semana em uma lista e adicione o dia "Fim de Semana" ao final da lista. Converta a lista de volta para uma tupla.

Desempacote a tupla (10, 20, 30) em três variáveis separadas: a, b e c. Imprima os valores das variáveis.

Conjuntos

Crie dois conjuntos de números: conjunto_a = {1, 2, 3, 4, 5} e conjunto_b = {4, 5, 6, 7, 8}.

Encontre a união dos dois conjuntos usando union().

Encontre a interseção dos dois conjuntos usando intersection().

Encontre a diferença entre conjunto_a e conjunto_b usando difference().

Encontre a diferença simétrica entre os dois conjuntos usando symmetric_difference().

Crie um conjunto de letras. Adicione uma nova letra usando add(). Tente adicionar uma letra que já existe.

Remova um elemento específico de um conjunto usando remove(). Tente remover um elemento que não existe. Use discard() para remover um elemento e observe a diferença.

Crie uma lista com elementos duplicados e converta-a em um conjunto para remover as duplicatas. Converta o conjunto de volta para uma lista.

Verifique se um conjunto é subconjunto de outro conjunto usando o método issubset().