

ATIVIDADE FUNÇÕES COM ARRAYS

1. Adicionando elementos à lista

Crie um programa que declare uma lista vazia chamada `numeros`.

Em seguida, peça ao usuário que digite **cinco números inteiros** e utilize o método `append()` para adicioná-los à lista.

Após todas as inserções, exiba a lista completa.

2. Inserindo elementos em posições específicas

Crie uma lista inicial chamada `nomes` contendo os valores `["Ana", "Carlos", "Beatriz"]`.

Utilize o método `insert()` para inserir o nome `"João"` na **segunda posição** (índice 1) da lista.

Depois, exiba a lista atualizada.

3. Removendo elementos pelo valor

Crie uma lista chamada `frutas` com os elementos `["maçã", "banana", "laranja", "uva"]`.

Peça ao usuário que digite o nome de uma fruta e use o método `remove()` para eliminá-la da lista.

Se o elemento não existir, exiba uma mensagem informando que a fruta não foi encontrada. Mostre a lista após a remoção.

4. Removendo elementos pelo índice

Crie uma lista chamada `valores` contendo `[10, 20, 30, 40, 50]`.

Peça ao usuário que informe uma posição (índice) para remover e utilize o método `pop()` para eliminar o item daquela posição.

Exiba o valor removido e a lista atualizada.

Caso o índice informado não exista, exiba uma mensagem de erro adequada.

5. Contando ocorrências de um valor

Crie uma lista de números contendo valores repetidos, por exemplo: `[2, 4, 2, 8, 2, 5, 4]`.

Peça ao usuário que informe um número e utilize o método `count()` para descobrir **quantas vezes** ele aparece na lista.

Mostre o resultado com uma mensagem explicativa.

6. Unindo duas listas

Crie duas listas chamadas `lista_a` e `lista_b` com alguns valores inteiros.

Use o método `extend()` para **juntar os elementos da segunda lista na primeira**.

Exiba o resultado final.

7. Ordenando e invertendo listas

Crie uma lista chamada `idades` com os valores `[15, 22, 30, 18, 25]`.

Utilize o método `sort()` para ordenar os números em **ordem crescente** e mostre o resultado. Em seguida, utilize o método `reverse()` para inverter a ordem dos elementos e exiba novamente.

8. Localizando a posição de um elemento

Crie uma lista chamada `cidades` contendo `["São Paulo", "Rio de Janeiro", "Recife", "Curitiba"]`.

Peça ao usuário que digite o nome de uma cidade e use o método `index()` para descobrir **em que posição ela está** na lista. Exiba a posição encontrada.

Caso o nome informado não exista, exiba uma mensagem informando que o item não foi encontrado.

9. Copiando listas

Crie uma lista chamada `original` com alguns números inteiros.

Em seguida, crie uma cópia da lista usando o método `copy()`.

Adicione um novo elemento apenas à lista copiada e exiba as duas listas (original e cópia).

10. Jogo simples de adivinhação com listas

Crie uma lista chamada `numeros` com os valores `[1, 2, 3, 4, 5]`.

Utilize o módulo `random` e a função `choice()` para escolher um número aleatoriamente da lista. Peça ao usuário que tente adivinhar qual número foi sorteado. Se o palpite for igual ao número escolhido, exiba uma mensagem de acerto; caso contrário, informe o número correto.