

Nível 1: Fundamentos

Neste nível, você vai praticar a manipulação básica de listas, loops e condições simples.

Exercício 1: Contando Ocorrências

Crie um programa que conte quantas vezes um número específico aparece em uma lista.

- **Entrada:** Uma lista de números e um número para ser procurado.
- **Saída:** Um número inteiro que representa a quantidade de vezes que o número procurado aparece na lista.

Exemplo:

```
numeros = [1, 5, 2, 8, 5, 3, 5]
```

```
numero_procurado = 5
```

Resultado Esperado:

3

Nível 2: Combinando Lógica

Aqui, você vai usar mais de uma lista e combinar diferentes lógicas de controle, como loops e verificações de pertencimento.

Exercício 2: Encontrando Elementos Comuns

Você tem duas listas e precisa encontrar os elementos que aparecem em ambas. O programa deve gerar uma terceira lista contendo apenas os elementos em comum, sem repetições.

- **Entrada:** Duas listas.
- **Saída:** Uma nova lista com os elementos que as duas listas têm em comum.

Exemplo:

```
lista1 = ["vermelho", "azul", "verde", "amarelo"]
```

```
lista2 = ["verde", "roxo", "azul", "preto"]
```

Resultado Esperado:

```
['azul', 'verde']
```

Nível 3: Lógica Avançada

Neste nível, os desafios exigem mais de uma lista e combinam diferentes lógicas de controle, como loops e verificações de pertencimento.

Exercício 3: Filtrando Números Primos

Sua tarefa é criar um programa que percorra uma lista de números e crie uma **nova lista** contendo apenas os números que forem primos.

- **Entrada:** Uma lista de números inteiros.
- **Saída:** Uma nova lista com os números primos encontrados.

Exemplo:

numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Resultado Esperado:

[2, 3, 5, 7]

Nível 4: Desafio Final

Este é o exercício mais completo, combinando entrada de dados do usuário, loops, validação e manipulação de listas.

Exercício 4: Coletor de Dados Robusto

Crie um programa que colete números de usuário e os armazene em uma lista. O programa deve continuar pedindo números até que o usuário digite -1 para parar. Ele deve validar a entrada para garantir que o que foi digitado é realmente um número antes de prosseguir. Apenas os números entre 0 e 100 devem ser considerados válidos e adicionados à lista. Ao final, imprima a soma dos números válidos e a lista dos números coletados.

- **Entrada:** Vários valores digitados pelo usuário, um de cada vez.
- **Saída:** A soma dos números válidos e a lista dos números coletados.

Exemplo:

Entradas: 10, 50, abc, -5, 101, 20, -1

Resultado Esperado:

Soma dos números válidos: 80

Números coletados: [10, 50, 20]