

Listas

Listas em Python são coleções ordenadas e mutáveis, definidas por colchetes []

Exercício: Crie uma lista chamada vendas contendo alguns valores.

Calcule e imprima:

- a) A soma total dos valores.
- b) O maior e o menor valor.
- c) O número total de itens na lista.

```
vendas=[1,2,3,6]
```

Exercício: Inclusão e Inserção de Elementos

Utilize a lista vendas do exercício anterior. Adicione o valor 500 ao final da lista e, em seguida, insira o valor 999 na segunda posição (índice 1). Imprima a lista após as modificações.

Slices

O fatiamento (slice) permite trabalhar com trechos ou subconjuntos da lista. A sintaxe é [start:stop:step].

Exercício: Dada a lista alfabeto = ['A', 'B', 'C', 'D', 'E', 'F', 'G'], utilize o fatiamento para extrair e imprimir uma sublista contendo apenas os elementos a partir do índice 2 até o índice 5 (exclusivo).

```
alfabeto = ['A', 'B', 'C', 'D', 'E', 'F', 'G']
```

Exercício: Fatiamento com Passo (Step) e Inversão

Utilize a lista numeros = [3,6,9,12,15]

- a) Extraia e imprima todos os números ímpares (começando do índice 0 e pulando de 2 em 2).
- b) Imprima a lista completa em ordem inversa.

Split

A função `split()` é usada para quebrar uma *string* em uma lista de substrings, utilizando um separador específico.

Exercício: Você recebeu uma linha de dados de um arquivo CSV que lista nomes de países. Use a função `split()` para transformar a string `registro_paises` em uma lista, separando os nomes pela vírgula.

```
registro_paises = "Brasil,Argentina,Chile,Uruguai,Paraguai"
```

Exercício: Processamento de Registro e Conversão de Tipo

Receba um registro de funcionário onde os campos estão separados por ponto-e-vírgula. Quebre o registro e, em seguida, verifique se o salário (terceiro campo) é maior que 50000. Lembre-se de que os valores vêm como *string* e precisam ser convertidos para `float` para comparação numérica.

```
Ex: registro_funcionario = "ID005;Maria Silva;12500.75;Contabilidade"
```

Dicionários (Dictionaries)

Dicionários funcionam com pares de **chave e valor**, definidos por chaves {}. As chaves devem ser únicas e são usadas para acessar os dados, não a posição numérica.

Exercício: Crie um dicionário chamado `aluno` contendo chaves para nome, idade e curso. Imprima o nome do aluno acessando-o diretamente pela chave.

Exercício: Verificação de Chave antes do Acesso

Crie um dicionário chamado `estoque` com alguns produtos. Peça ao usuário que digite um nome de produto. Antes de tentar imprimir o preço, verifique se a chave digitada existe no dicionário para evitar erros.

```
estoque = {  
    "TV": 1500,  
    "Rádio": 200,  
    "Microondas": 900  
}
```

Módulos (Modules)

Exercício: Módulos são arquivos externos que contêm definições e instruções que podem ser importadas. Podemos importar módulos nativos do Python ou módulos criados por nós mesmos.

Importe o módulo random (módulo nativo do Python).

Gere uma lista de 5 números aleatórios no intervalo de 1 a 100 e imprima-a.

Tuplas (Tuples)

Tuplas são coleções ordenadas definidas por parênteses (). A principal característica é que elas são **imutáveis**, ou seja, não podem ser alteradas (você não pode usar `.append()`, por exemplo).

Exercício: Acessando Elementos e Manipulando uma Lista Dentro da Tupla

Crie uma tupla que contenha uma string, um número inteiro e, no terceiro elemento, uma lista de nomes.

Acesse o primeiro nome dentro dessa lista.

Acrescente o nome Felipe ao final da lista dos nomes.

```
dados_empresa = ("Matriz 01", 2023, ["Alice", "Roberto", "Carla"])
```