

PRÁTICA LABORATORIAL 04

Objectivos:

Listas, tuplos, matrizes, dicionários, List Comprehension, estruturas de repetição

1. Criar um tuplo com números de 1 a 5 e imprimir o terceiro elemento.
2. Criar um tuplo com nomes e desempacotar em variáveis.
3. Concatenar dois tuplos e verificar se um elemento existe no resultado.
4. Criar uma matriz 3x3 e imprimir o elemento da segunda linha e terceira coluna.
5. Usar list comprehension para criar uma matriz identidade 3x3. (Pesquisar matriz identidade)
6. Criar uma matriz 2x3 e somar todos os elementos.
7. Criar um dicionário com 3 pares chave-valor:
 - 7.1.1. Imprimir todas as chaves.
 - 7.1.2. Atualizar o valor de uma chave existente
 - 7.1.3. Adicionar uma nova chave.
 - 7.1.4. Remover uma chave e mostrar o dicionário restante.
 - 7.1.5. Verificar se uma chave existe no dicionário.
 - 7.1.6. Iterar sobre o dicionário e imprimir cada par chave-valor.
8. Cria uma nova lista com os quadrados dos números de 0 a 9 usando list comprehension.
9. Usa list comprehension para criar uma lista com os números pares entre 0 e 20.
10. Usa list comprehension para obter os números maiores que 5 de uma lista de inteiros.
11. Cria uma lista com as letras de uma palavra, mas apenas as vogais (usando list comprehension).
12. Usando List Comprehension, descubra os números repetidos entre duas listas à sua escolha;
13. Usando List Comprehension, indique palavras que contenham menos de 4 letras a partir de um texto à sua escolha.
14. Crie uma lista de strings com os nomes de cinco frutas.
15. Utilize list comprehension para criar uma lista que contenha o comprimento de cada string da lista de frutas.
16. Utilize list comprehension para criar uma lista que contenha todas as frutas em maiúsculas.
17. Imprima as duas listas resultantes.
18. Crie uma list que gere uma lista de strings formatadas a partir da seguinte lista de números: (pesquisar por {x:02})
19. numeros = [1, 2, 3, 4, 5], de forma que o resultado seja ["01", "02", "03", "04", "05"].
20. Use list comprehension para criar uma lista contendo apenas os elementos da lista original que têm mais de

5 caracteres e terminam com a letra 'o': palavras = ["comando", "python", "programação", "código", "desenvolvimento"].

21. Use list comprehension para criar uma lista de strings onde cada string é formada pelas três primeiras letras de cada palavra na lista original: palavras = ["Python", "List", "Comprehension", "Exercício", "Prática"].
22. Crie uma list comprehension que gere uma lista de strings onde cada string é a concatenação da palavra original com ela mesma invertida: palavras = ["sol", "lua", "estrela"]
(o resultado deve ser ["sollos", "luaaul", "estrelaalertse"]).
23. Use list comprehension para criar uma lista de strings, onde cada string é "sim" se a palavra for um palíndromo, ou "não" caso contrário: palavras = ["arara", "python", "ovo", "programação", "radar"].

Bom trabalho! 😊