

Atividade 01 - Estruturas de dados compostas

1. Dada uma lista de números inteiros, implemente uma compreensão de lista que crie uma nova lista contendo apenas os números pares elevados ao quadrado.

```
1. lista = [1, 2, 4, 5, 7, 8, 10]
2. lista_pares_quadrado = [num**2 for num in lista if num % 2 == 0]
3. print(lista_pares_quadrado)
```

```
[4, 16, 64, 100]
```

2. Dadas duas listas de strings prefixos e sufixos, implemente uma compreensão de lista que crie uma nova lista contendo todas as concatenações possíveis entre um prefixo e um sufixo.

```
1. prefixos = ["in", "anti", "ante", "pos"]
2. sufixos = ["felizmente", "comum", "moderno", "paro"]
3.
4. combinacoes = [prefixo + sufixo for prefixo in prefixos for sufixo in sufixos]
5.
6. print(combinacoes)
```

```
['infelizmente', 'incomum', 'inmoderno', 'inparo', 'antifelizmente', 'anticomum',
'antimoderno', 'antiparo', 'antefelizmente', 'antecomum', 'antemoderno', 'anteparo',
'posfelizmente', 'poscomum', 'posmoderno', 'posparo']
```

3. Dadas duas listas de números inteiros a e b, implemente um programa em Python que:
 - Converta ambas as listas em conjuntos (sets).
 - Encontre a união desses dois conjuntos.
 - Exiba o conjunto resultante.

```
1. a = [5, 2, 3, 4]
2. b = [2, 5, 3, 8]
3.
4. con_a = set(a)
5. con_b = set(b)
6.
7. con_c = con_a.union(con_b)
8.
9. print(con_c)
```

```
{2, 3, 4, 5, 8}
```

4. Dada uma string, implemente um programa em Python que conte a frequência de cada caractere na string usando um dicionário. Exemplo: texto = 'banana'

Saída: {'b': 1, 'a': 3, 'n': 2}

```
1. string = "paralelepipedo"
2. dic = {}
3. for letra in string:
4.     dic[letra] = string.count(letra)
5. print(dic)
```

```
{ 'p': 3, 'a': 2, 'r': 1, 'l': 2, 'e': 3, 'i': 1, 'd': 1, 'o': 1 }
```