

● Atividade 02 (ATIV-02)

Gisele Oliveira

1. Escreva uma função que receba uma lista de números e retorne outra lista com os números ímpares.

```
def obter_impares(numeros):  
    impares = []  
    for numero in numeros:  
        if numero % 2 != 0:  
            impares.append(numero)  
    return impares
```

```
lista = [1, 2, 3, 4, 5, 6, 7, 8, 9]  
impares = obter_impares(lista)  
print(impares)
```

```
[1, 3, 5, 7, 9]
```

2. Escreva uma função que receba uma lista de números e retorne outra lista com os números primos presentes.

```
def eh_primo(numero):  
    if numero < 2:  
        return False  
    for i in range(2, int(numero ** 0.5) + 1):  
        if numero % i == 0:  
            return False  
    return True
```

```
def obter_primos(lista_numeros):  
    return [numero for numero in lista_numeros if eh_primo(numero)]
```

```
lista = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]  
primos = obter_primos(lista)  
print(primos)  
[2, 3, 5, 7, 11, 13]
```

3. Escreva uma função que receba duas listas e retorne outra lista com os elementos que estão presentes em apenas uma das listas.

```
def elementos_unicos(lista1, lista2):  
    unicos_lista1 = [item for item in lista1 if item not in lista2]  
    unicos_lista2 = [item for item in lista2 if item not in lista1]  
    return unicos_lista1 + unicos_lista2  
  
lista1 = [1, 2, 3, 4, 5, 6, 7, 8]  
lista2 = [3, 4, 5, 6, 7, 8, 9, 10]  
resultado = elementos_unicos(lista1, lista2)  
print(resultado)  
[1, 2, 9, 10]
```

4. Dada uma lista de números inteiros, escreva uma função para encontrar o segundo maior valor na lista.

```
def segundo_maior(lista):  
    if len(lista) < 2:  
        return None  
  
    lista_unica = list(set(lista))  
    lista_unica.sort(reverse=True)  
  
    if len(lista_unica) < 2:  
        return None  
  
    return lista_unica[1]  
  
lista = [1, 2, 4, 65, 77, 65, 99, 88, 85]  
resultado = segundo_maior(lista)  
print(resultado)
```

5. Crie uma função que receba uma lista de tuplas, cada uma contendo o nome e a idade de uma pessoa, e retorne a lista ordenada pelo nome das pessoas em ordem alfabética.

```
def ordenar_por_nome(lista_tuplas):  
    return sorted(lista_tuplas, key=lambda x: x[0])  
  
lista_pessoas = [  
    ("Gisele", 26),  
    ("Eduardo", 29),  
    ("Bruno", 28),  
    ("Amanda", 30)  
]  
  
resultado = ordenar_por_nome(lista_pessoas)  
print(resultado)  
[('Amanda', 30), ('Bruno', 28), ('Eduardo', 29), ('Gisele', 26)]
```

6. Observe os espaços sublinhados e complete o código.

```
import matplotlib.pyplot as plt  
import numpy as np  
  
fig, axs = plt.subplots(ncols=2, nrows=2, figsize=(5.5, 3.5),  
layout="constrained")  
  
for row in range(2):  
    for col in range(2):  
        axs[row, col].annotate(f'axs[{row}, {col}]', (0.5, 0.5),  
transform=axs[row, col].transAxes,  
ha='center', va='center', fontsize=18,  
color='darkgrey')  
fig.suptitle('plt.subplots()')
```

7. Observe os espaços sublinhados e complete o código.

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt

x = np.linspace(-2 * np.pi, 2 * np.pi, 100)
y = np.sin(x)

fig,ax = plt.subplots()
ax.plot(x, y)
```

8. Utilizando pandas, como realizar a leitura de um arquivo CSV em um DataFrame e exibir as primeiras linhas?

```
df = pd.read_csv(''.csv')
print(df.head())
```

9. Utilizando pandas, como selecionar uma coluna específica e filtrar linhas em um “DataFrame” com base em uma condição?

```
coluna_especifica = df['nome_da_coluna']
filtro = df[df['nome_da_coluna'] > valor]
```

10. Utilizando pandas, como lidar com valores ausentes (NaN) em um DataFrame?

```
valores_ausentes = df.isnull() print(valores_ausentes)

df_sem_nan = df.dropna()
df_sem_nan_colunas = df.dropna(axis=1)
```