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

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

Exemplo:

```
numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
resultado = filtrar_impares(numeros)
print(resultado)
```

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

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

def filtrar_primos(numeros):
    primos = [numero for numero in numeros if sefor_primo(numero)]
    return primos
```

Exemplo

```
numeros = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
resultado = filtrar_primos(numeros)
print(resultado)
```

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

```
def diferenca_simetrica(lista1, lista2):
    set1 = set(lista1)
    set2 = set(lista2)
diferenca = set1.symmetric_difference(set2)
return list(diferenca)
```

Exemplo

```
lista1 = [1, 2, 3, 4, 5]
lista2 = [4, 5, 6, 7, 8]
resultado = diferenca_simetrica(lista1, lista2)
print(resultado)
```

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

```
def segundo_maior(numeros):
    if len(numeros) < 2:
        raise ValueError("A lista deve conter pelo menos dois elementos.")

    primeiro = segundo = float('-inf')
```

```
    for numero in numeros:
        if numero > primeiro:
            segundo = primeiro
            primeiro = numero
        elif primeiro > numero > segundo:
            segundo = numero

    if segundo == float('-inf'):
        raise ValueError("A lista deve conter pelo menos dois valores distintos.")

    return segundo
```

Exemplo

```
numeros = [10, 20, 4, 45, 99]
resultado = segundo_maior(numeros)
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(pessoas):

return sorted(pessoas, key=lambda pessoa: pessoa[0])
```

Exemplo

```
pessoas = [("Ana", 30), ("Carlos", 22), ("Beatriz", 25), ("Daniel", 27)]

resultado = ordenar_por_nome(pessoas)

print(resultado)
```

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?

```
import pandas as pd
df = pd.read_csv('caminho/para/seu_arquivo.csv')
  Exibição das primeiras linhas do DataFrame
print(df.head())
```

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

```
import pandas as pd
```

Exemplo

```
data = {
        'Nome': ['Ana', 'Carlos', 'Beatriz', 'Daniel'],
        'Idade': [30, 22, 25, 27],
        'Salario': [5000, 4000, 4500, 4200]
}
df = pd.DataFrame(data)
```

coluna_idade = df['Idade']

df_filtrado = df[df['Idade'] > 25]

```
print("Coluna Idade:")
print(coluna_idade)
```

```
print("\nDataFrame filtrado com Idade > 25:")

print(df_filtrado)
```

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

DataFrame?

Isso vai depende da situação e da necessidade. Podemos usar "isna","dropna","fillna", "replace", "interpolate" por exemplo.