1. \*\*Números Ímpares:\*\*

```python

def impares(numeros):

"""Retorna uma lista com os números ímpares de uma lista dada.

Args:

numeros: Uma lista de números.

Returns:

Uma lista com os números ímpares da lista de entrada.

"""

impares = []

for numero in numeros:

if numero % 2 != 0:

impares.append(numero)

return impares

# Exemplo de uso

lista = [1, 2, 3, 4, 5, 6, 7, 8, 9]

impares\_lista = impares(lista)

print(f"Números ímpares: {impares\_lista}")

```

2. \*\*Números Primos:\*\*

```python

def primos(numeros):

"""Retorna uma lista com os números primos de uma lista dada.

Args:

numeros: Uma lista de números.

Returns:

Uma lista com os números primos da lista de entrada.

"""

primos = []

for numero in numeros:

if numero <= 1:

continue # 1 não é primo

eh\_primo = True

for i in range(2, int(numero\*\*0.5) + 1):

if numero % i == 0:

eh\_primo = False

break

if eh\_primo:

primos.append(numero)

return primos

# Exemplo de uso

lista = [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

primos\_lista = primos(lista)

print(f"Números primos: {primos\_lista}")

```

3. \*\*Elementos Únicos:\*\*

```python

def unicos(lista1, lista2):

"""Retorna uma lista com os elementos que estão presentes em apenas uma das listas.

Args:

lista1: A primeira lista.

lista2: A segunda lista.

Returns:

Uma lista com os elementos únicos das duas listas.

"""

unicos = []

for elemento in lista1:

if elemento not in lista2:

unicos.append(elemento)

for elemento in lista2:

if elemento not in lista1:

unicos.append(elemento)

return unicos

# Exemplo de uso

lista1 = [1, 2, 3, 4, 5]

lista2 = [3, 4, 5, 6, 7]

unicos\_lista = unicos(lista1, lista2)

print(f"Elementos únicos: {unicos\_lista}")

```

4. \*\*Segundo Maior Valor:\*\*

```python

def segundo\_maior(numeros):

"""Retorna o segundo maior valor em uma lista de números.

Args:

numeros: Uma lista de números.

Returns:

O segundo maior valor na lista.

"""

if len(numeros) < 2:

return None # Se a lista tiver menos de 2 elementos, não há segundo maior

numeros.sort() # Ordena a lista em ordem crescente

return numeros[-2] # Retorna o segundo elemento do final (o segundo maior)

# Exemplo de uso

lista = [1, 5, 3, 2, 4]

segundo\_maior\_valor = segundo\_maior(lista)

print(f"Segundo maior valor: {segundo\_maior\_valor}")

```

\*\*Explicação:\*\*

5. \*\*Tuplas Ordenadas:\*\*

```python

def ordenar\_por\_nome(pessoas):

"""Retorna uma lista de tuplas ordenada pelo nome das pessoas em ordem alfabética.

Args:

pessoas: Uma lista de tuplas, cada uma contendo o nome e a idade de uma pessoa.

Returns:

Uma lista de tuplas ordenada pelo nome das pessoas em ordem alfabética.

"""

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

# Exemplo de uso

pessoas = [('João', 25), ('Maria', 30), ('Pedro', 28), ('Ana', 22)]

pessoas\_ordenadas = ordenar\_por\_nome(pessoas)

print(f"Pessoas ordenadas por nome: {pessoas\_ordenadas}")

```

6. \*\*Completando o Código:\*\*

```python

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()')

plt.show()

```

7. \*\*Completando o Código:\*\*

```python

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)

plt.show()

```

\*\*Pandas:\*\*

8. \*\*Leitura de CSV:\*\*

```python

import pandas as pd

# Ler o arquivo CSV em um DataFrame

df = pd.read\_csv('nome\_do\_arquivo.csv')

# Exibir as primeiras linhas do DataFrame

print(df.head())

```

9. \*\*Selecionar Coluna e Filtrar:\*\*

```python

import pandas as pd

# Ler o arquivo CSV em um DataFrame

df = pd.read\_csv('nome\_do\_arquivo.csv')

# Selecionar a coluna 'idade'

idade = df['idade']

# Filtrar linhas onde a idade é maior que 30

df\_filtrado = df[df['idade'] > 30]

# Exibir o DataFrame filtrado

print(df\_filtrado)

```

10. \*\*Lidando com NaN:\*\*

```python

import pandas as pd

# Ler o arquivo CSV em um DataFrame

df = pd.read\_csv('nome\_do\_arquivo.csv')

# Substituir valores NaN na coluna 'idade' pela média da coluna

df['idade'].fillna(df['idade'].mean(), inplace=True)

# Exibir o DataFrame com valores NaN preenchidos

print(df)

```