

Atividade 02 - Bootcamp Avanti: Machine Learning
Aluna: Giulia Lima Duarte

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

```
[15] def filtrar_impares(numeros):  
  
    impares = [numero for numero in numeros if numero % 2 != 0]  
    return impares  
  
print('Questão 1:')  
entrada = input("Digite uma lista de números separados por espaço: ")  
lista_numeros = list(map(int, entrada.split()))  
lista_impares = filtrar_impares(lista_numeros)  
print("Números ímpares:", lista_impares)
```

➡ Questão 1:
Digite uma lista de números separados por espaço: 1 2 3 4 5 6 7 8 9 10
Números ímpares: [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 filtrar_primos(numeros):  
  
    primos = [numero for numero in numeros if eh_primo(numero)]  
    return primos  
  
print('Questão 2:')  
entrada = input("Digite uma lista de números separados por espaço: ")  
lista_numeros = list(map(int, entrada.split()))  
lista_primos = filtrar_primos(lista_numeros)  
print("Números primos:", lista_primos)
```

➡ Questão 2:
Digite uma lista de números separados por espaço: 2 3 4 5 6 7 8 9 10 11 12 13
Números 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 diferenca_simetrica(lista1, lista2):  
  
    set1 = set(lista1)  
    set2 = set(lista2)  
    diferenca = set1.symmetric_difference(set2)  
    return list(diferenca)  
  
print('Questão 3:')  
entrada1 = input("Digite a primeira lista de números separados por espaço: ")  
entrada2 = input("Digite a segunda lista de números separados por espaço: ")  
lista1 = list(map(int, entrada1.split()))  
lista2 = list(map(int, entrada2.split()))  
lista_diferenca_simetrica = diferenca_simetrica(lista1, lista2)  
print("Diferença simétrica:", lista_diferenca_simetrica)
```

Questão 3:
Digite a primeira lista de números separados por espaço: 1 2 3 4 5
Digite a segunda lista de números separados por espaço: 4 5 6 7 8
Diferença simétrica: [1, 2, 3, 6, 7, 8]

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:  
        raise ValueError("A lista deve conter pelo menos dois elementos")  
  
    primeiro_maior = segundo_maior = float('-inf')  
  
    for numero in lista:  
        if numero > primeiro_maior:  
            segundo_maior = primeiro_maior  
            primeiro_maior = numero  
        elif primeiro_maior > numero > segundo_maior:  
            segundo_maior = numero  
  
    if segundo_maior == float('-inf'):  
        raise ValueError("Não há um segundo maior valor distinto na lista")  
  
    return segundo_maior  
  
print('Questão 4:')  
entrada = input("Digite uma lista de números separados por espaço: ")  
lista_numeros = list(map(int, entrada.split()))  
try:  
    resultado = segundo_maior(lista_numeros)  
    print("O segundo maior valor é:", resultado)  
except ValueError as e:  
    print(e)
```

Questão 4:
Digite uma lista de números separados por espaço: 10 20 4 45 99 45
O segundo maior valor é: 45

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_pessoas):  
    return sorted(lista_pessoas, key=lambda pessoa: pessoa[0])  
  
print('Questão 5:')  
n = int(input("Digite o número de pessoas: "))  
lista_pessoas = []  
  
for _ in range(n):  
    nome = input("Digite o nome da pessoa: ")  
    idade = int(input("Digite a idade da pessoa: "))  
    lista_pessoas.append((nome, idade))  
  
lista_ordenada = ordenar_por_nome(lista_pessoas)  
print("Lista ordenada por nome:", lista_ordenada)
```

Questão 5:
Digite o número de pessoas: 5
Digite o nome da pessoa: Carla
Digite a idade da pessoa: 10
Digite o nome da pessoa: Ricardo
Digite a idade da pessoa: 18
Digite o nome da pessoa: Ana
Digite a idade da pessoa: 20
Digite o nome da pessoa: Luiz
Digite a idade da pessoa: 32
Digite o nome da pessoa: Beatriz
Digite a idade da pessoa: 5
Lista ordenada por nome: [('Ana', 20), ('Beatriz ', 5), ('Carla', 10), ('Luiz', 32), ('Ricardo', 18)]

6. Observe os espaços sublinhados e complete o código.

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

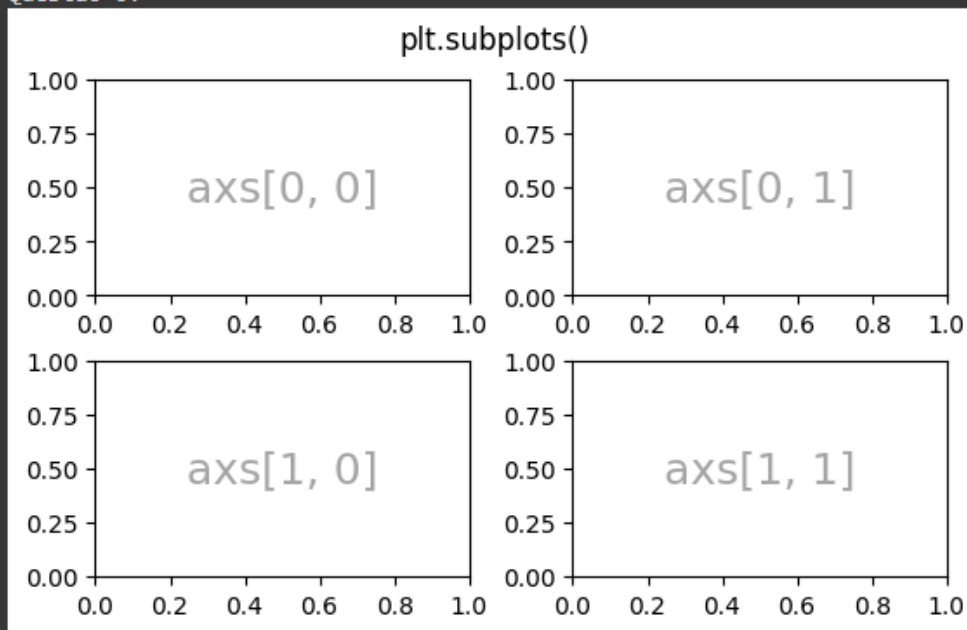
```
[19] import matplotlib.pyplot as plt
import numpy as np

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()')
print('Questão 6:')
plt.show()
```



Questão 6:



7. Observe os espaços sublinhados e complete o código.

```
import numpy as np
import _____ as mpl
import _____ as plt

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

____, ____ = plt.subplots()
ax.____(____, ____)
```

```

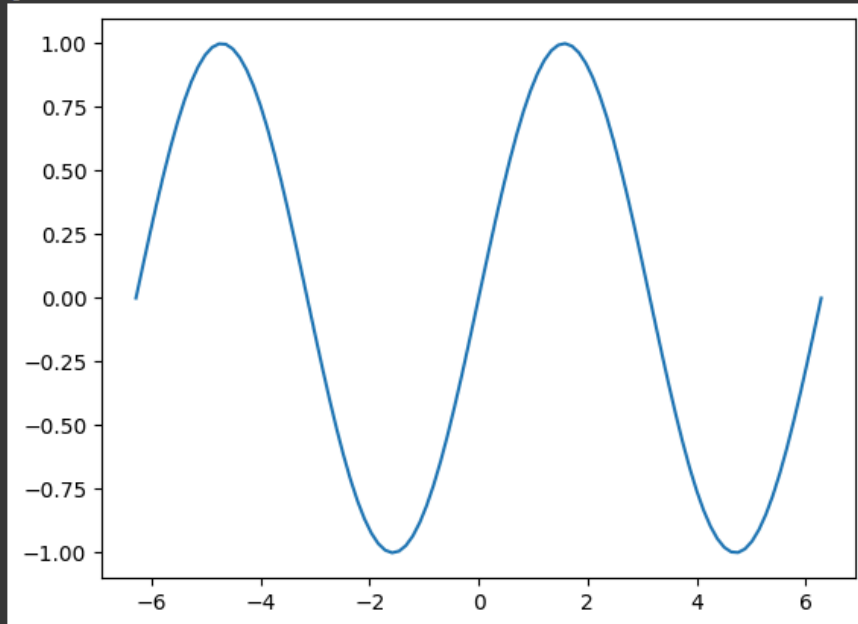
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt

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

fig, ax = plt.subplots()
ax.plot(x, y)
print('Questão 7:')
plt.show()

```

↔ Questão 7:



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('/content/RickAndMortyScripts.csv')

print('Questão 8:')
print(df.head())

```

↔ Questão 8:

	index	season	no.	episode	no.	episode name	name \
0	0	1	1	Pilot	Rick		
1	1	1	1	Pilot	Morty		
2	2	1	1	Pilot	Rick		
3	3	1	1	Pilot	Morty		
4	4	1	1	Pilot	Rick		

	line
0	Morty! You gotta come on. Jus'... you gotta co...
1	What, Rick? What's going on?
2	I got a surprise for you, Morty.
3	It's the middle of the night. What are you tal...
4	Come on, I got a surprise for you. Come on, h...

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

```
import pandas as pd

file_path = '/content/RickAndMortyScripts.csv'
df = pd.read_csv(file_path)

print('Questão 9:')
coluna_name = df['name']
print("Coluna 'name':")
print(coluna_name.head())

filtro_rick = df[df['name'] == 'Rick']
print("\nLinhas onde o nome é 'Rick':")
print(filtro_rick.head())
```

```
Questão 9:
Coluna 'name':
0    Rick
1    Morty
2    Rick
3    Morty
4    Rick
Name: name, dtype: object

Linhas onde o nome é 'Rick':
   index  season no.  episode no.  episode name  name \
0      0         1         1      Pilot Rick
2      2         1         1      Pilot Rick
4      4         1         1      Pilot Rick
6      6         1         1      Pilot Rick
7      7         1         1      Pilot Rick

line
0      Morty! You gotta come on. Jus'... you gotta come with me.
2      I got a surprise for you, Morty.
4      Come on, I got a surprise for you. Come on, hurry up.
6      We gotta go, gotta get outta here, come on. Got a surprise for you Morty.
7  What do you think of this... flying vehicle, Morty? I built it outta stuff I found in the garage.
```

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

```
import pandas as pd

df = pd.read_csv('https://raw.githubusercontent.com/pandas-dev/pandas/main/doc/data/titanic.csv')

print('Questão 10:')

print("DataFrame Original:\n", df.head())

print("\nValores Ausentes por Coluna:\n", df.isna().sum())

df_dropna = df.dropna()
print("\nApós remover linhas com qualquer valor NaN:\n", df_dropna.head())

df_fillna = df.fillna(0)
print("\nApós preencher valores NaN com 0:\n", df_fillna.head())

df_fillna_mean = df.fillna(df.mean(numeric_only=True))
print("\nApós preencher valores NaN com a média das colunas numéricas:\n", df_fillna_mean.head())

df_interpolated = df.interpolate()
print("\nApós interpolar valores ausentes:\n", df_interpolated.head())
```



Questão 10:

DataFrame Original:

	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	
2	Heikkinen, Miss Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S



Valores Ausentes por Coluna:

```
PassengerId    0
Survived        0
Pclass         0
Name           0
Sex            0
Age          177
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin         687
Embarked       2
dtype: int64
```

Após remover linhas com qualquer valor NaN:

```
PassengerId  Survived  Pclass  \
1             2         1         1
3             4         1         1
6             7         0         1
10            11         1         3
11            12         1         1
```

```
                Name      Sex  Age  SibSp  \
1  Cumings, Mrs. John Bradley (Florence Briggs Thayer)  female  38.0      1
3      Futrelle, Mrs. Jacques Heath (Lily May Peel)    female  35.0      1
6                      McCarthy, Mr. Timothy J        male  54.0      0
10                      Sandstrom, Miss Marguerite Rut  female   4.0      1
11                      Bonnell, Miss Elizabeth    female  58.0      0
```

```
    Parch  Ticket    Fare Cabin Embarked
1       0  PC 17599  71.2833   C85        C
3       0  113803  53.1000  C123        S
6       0   17463  51.8625  E46        S
10      1  PP 9549  16.7000   G6        S
11      0  113783  26.5500  C103        S
```

Após preencher valores NaN com 0:

```
PassengerId  Survived  Pclass  \
0             1         0         3
1             2         1         1
2             3         1         3
3             4         1         1
4             5         0         3
```




```
      Name      Sex  Age  SibSp  \
0      Braund, Mr. Owen Harris    male  22.0    1
1  Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38.0    1
2      Heikkinen, Miss Laina    female  26.0    0
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)    female  35.0    1
4      Allen, Mr. William Henry    male  35.0    0
```

```
      Parch      Ticket     Fare Cabin Embarked
0         0         A/5 21171   7.2500    0        S
1         0          PC 17599  71.2833   C85        C
2         0  STON/O2. 3101282   7.9250    0        S
3         0       113803  53.1000  C123        S
4         0       373450   8.0500    0        S
```

Após preencher valores NaN com a média das colunas numéricas:

```
      PassengerId  Survived  Pclass  \
0                 1         0       3
1                 2         1       1
2                 3         1       3
3                 4         1       1
4                 5         0       3
```

```
      Name      Sex  Age  SibSp  \
0      Braund, Mr. Owen Harris    male  22.0    1
1  Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38.0    1
2      Heikkinen, Miss Laina    female  26.0    0
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)    female  35.0    1
4      Allen, Mr. William Henry    male  35.0    0
```

```
      Parch      Ticket     Fare Cabin Embarked
0         0         A/5 21171   7.2500   NaN        S
1         0          PC 17599  71.2833   C85        C
2         0  STON/O2. 3101282   7.9250   NaN        S
3         0       113803  53.1000  C123        S
4         0       373450   8.0500   NaN        S
```

Após interpolar valores ausentes:

```
      PassengerId  Survived  Pclass  \
0                 1         0       3
1                 2         1       1
2                 3         1       3
3                 4         1       1
4                 5         0       3
```

```
      Name      Sex  Age  SibSp  \
0      Braund, Mr. Owen Harris    male  22.0    1
1  Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38.0    1
2      Heikkinen, Miss Laina    female  26.0    0
3  Futrelle, Mrs. Jacques Heath (Lily May Peel)    female  35.0    1
4      Allen, Mr. William Henry    male  35.0    0
```

```
      Parch      Ticket     Fare Cabin Embarked
0         0         A/5 21171   7.2500   NaN        S
1         0          PC 17599  71.2833   C85        C
2         0  STON/O2. 3101282   7.9250   NaN        S
3         0       113803  53.1000  C123        S
4         0       373450   8.0500   NaN        S
```

Datasets: [Rick&Morty Scripts | Kaggle](#), [Titanic CSV](#).

Código:  ATIV-02.ipynb