



CURSO: <BÁSICO EM MACHINE LEARNING>

- **Atividade 02 (ATIV-02)**

- Tipo: Somativa;
- Tema: Algoritmos básicos de programação em linguagem python e visualização e análise de dados.
- Conteúdo: Módulo 1 e 2.
- Participantes: Individual.
- Avaliação do aluno.
 - Objetivo: Avaliar desempenho do aluno sobre conhecimentos básicos de programação em linguagem python e visualização e análise de dados.
 - Nota: 0 a 3 supercrítico, 4 a 6 crítico, 5 a 7 razoável e 8 a 10 bom;
 - Critérios avaliados: Respostas com coerência, coesão e com exemplos.
- Informações adicionais: A atividade é composta por 5 questões dissertativas sobre python e 5 questões sobre visualização e análise de dados.
- **AO CONCLUIR A ATIVIDADE: ENVIAR APENAS O LINK DO REPOSITÓRIO GITHUB (ESPECIFICAR A BRANCH) PÚBLICO.**

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

`def numeros_impares(numeros):`

`lista_impares=[]`

`for i in numeros:`

`if i %2==1:`

`lista_impares.append(i)`

`return lista_impares`



2. `numeros_impares(numeros)` Escreva uma função que receba uma lista de números e retorne outra lista com os números primos presentes.

```
def is_primo(n):
    if n <= 1:
        return False
    else:
        for i in range(2, n):
            if n % i == 0:
                return False
        return True
def numeros_primos(numeros):
    lista_primos = []
    for i in numeros:
        if is_primo(i):
            lista_primos.append(i)
    return lista_primos
```

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

```
def valores_unicos(lista1, lista2):
    lista_unicos = []
    for i in lista1:
        if i not in lista2:
            lista_unicos.append(i)
    for i in lista2:
        if i not in lista1:
            lista_unicos.append(i)
    return lista_unicos
```



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:  
        return None  
  
    maior = max(numeros)  
    numeros.remove(maior)  
  
    segundo_maior = max(numeros)  
  
    return segundo_maior
```

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):  
    pessoas_ordenadas = sorted(pessoas)  
  
    for nome, idade in pessoas_ordenadas:  
        print(f"{nome} ({idade} anos)")
```

Para ver todos os códigos, clique [aqui](#)

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?

Para ler um arquivo CSV em um DataFrame do Pandas, você pode utilizar a função `read_csv()`. Essa função recebe como argumento o caminho para o arquivo CSV. Após a leitura, você pode usar o método `head()` para exibir as primeiras linhas do DataFrame.

```
import pandas as pd

df = pd.read_csv('meu_arquivo.csv')

print(df.head())
```



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

Para selecionar uma coluna específica e preciso colocar o nome da coluna entre colchetes []. Por exemplo, se você tem um DataFrame chamado df e deseja selecionar a coluna “idade”, escreve-se `df["idade"]`.

Para filtrar linhas com base em uma condição, e preciso colocar a condição desejada dentro dos colchetes [] após o nome do DataFrame. Por exemplo, para filtrar apenas as linhas onde a coluna “idade” é maior que 30, escreve-se `df[df["idade"] > 30]`.

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

Usa-se funções como `isnull()`, `notnull()`, `dropna()`, e `fillna()`. As `isnull()` e `notnull()` permitem detectar valores ausentes ou não-ausentes. Já as `dropna()` e `fillna()` são para lidar esses dados faltantes. Usando a `dropna()` estamos eliminando de forma eficaz qualquer linha ou coluna (conforme nossa especificação) que contenha pelo menos um valor ausente. Usando `fillna()` estamos substituindo os valores ausentes por um valor específico para preenchimento na coluna. Nem sempre é apropriado remover dados, porque isso pode levar à perda de informações valiosas, devemos ponderar qual [e a melhor função para situação.