

## 1. web\_scraping\_extracao.py

*(Aula 1 — Web Scraping básico com BeautifulSoup)*

python

```
import requests
from bs4 import BeautifulSoup

url = "https://sites.google.com/view/vestibular123"

response = requests.get(url)
content = response.content

soup = BeautifulSoup(content, "html.parser")
meta_tags = soup.find_all('meta', {'property': 'og:description'})

texto_salvar = meta_tags[0]['content']

print(texto_salvar)
```

## 2. escrever\_arquivo.py

*(Aula 1 — Salvando dados em arquivo texto)*

python

```
with open('exemplo.txt', 'w') as file:
    file.write('Olá estudantes de ciência de dados!')
```

## 3. ler\_arquivo.py

*(Aula 1 — Lendo dados de um arquivo texto)*

python

```
with open('exemplo.txt', 'r') as file:
    content = file.read()
```

*(Aula 2 — Importação simples de biblioteca)*

```
python

import math

# Calculando a raiz quadrada de 25
raiz_quadrada = math.sqrt(25)

print("A raiz quadrada de 25 é:", raiz_quadrada)
```

## **5. importar\_bibliotecas\_ds.py**

*(Aula 2 — Importação simples de bibliotecas para Ciência de Dados)*

```
python

# Importando a biblioteca pandas para manipulação de dados tabulares
import pandas

# Importando a biblioteca numpy para manipulação de arrays e
operações matemáticas
import numpy

# Importando a biblioteca matplotlib para visualização de dados
import matplotlib

# Importando a biblioteca seaborn para visualização estatística
import seaborn

# Importando a biblioteca scipy para funções matemáticas e estatísticas avançadas
import scipy

# Importando a biblioteca sklearn para aprendizado de máquina
import sklearn
```

## **6. pandas\_dataframe.py**

*(Aula 2 — Criando um DataFrame com pandas)*

python

```
import pandas as pd

# Criando um DataFrame simples
data = {'Nome': ['Alice', 'Bob', 'Charlie'],
        'Idade': [25, 30, 35]}
df = pd.DataFrame(data)

print(df)
```

## 7. numpy\_array.py

*(Aula 2 — Criando array com NumPy)*

python

```
import numpy as np

# Criando um array NumPy
arr = np.array([1, 2, 3, 4, 5])
print(arr)
```

## 8. grafico\_dispersao.py

*(Aula 2 — Usando seaborn e matplotlib)*

python

```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

# Dados
x = np.random.normal(size=100)
y = np.random.normal(size=100)

# Plotando gráfico de dispersão
plt.figure(figsize=(2, 2))
sns.scatterplot(x=x, y=y)
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Gráfico de Dispersão')
plt.show()
```

## 9. grafico\_linha.py

*(Aula 2 — Gráfico de linha com matplotlib)*

python

```
import matplotlib.pyplot as plt

# Dados
x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

# Plotando gráfico de linha
plt.figure(figsize=(2, 2))
plt.plot(x, y)
plt.xlabel('Eixo X')
plt.ylabel('Eixo Y')
plt.title('Gráfico de Linha')
plt.show()
```

## 10. sklearn\_regressao.py

*(Aula 2 — Exemplo com regressão linear)*

python

```
import sklearn.linear_model

# Criando um objeto de regressão linear
regression_model = sklearn.linear_model.LinearRegression()

print("Objeto de regressão linear criado:", regression_model)
```

## 11. scipy\_fatorial.py

*(Aula 2 — Cálculo de fatorial)*

python

```
import scipy

# Calculando o fatorial de 5
fatorial = scipy.special.factorial(5)
print("Fatorial de 5 é:", fatorial)
```

## 12. machine\_learning\_iris.py

*(Aula 2 — Árvore de decisão com Iris dataset)*

python

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier

# Carregando conjunto de dados
iris = datasets.load_iris()

# Separando treino/teste
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target,
                                                    test_size=0.2,
random_state=42)

# Criando modelo
model = DecisionTreeClassifier(max_depth=2)

# Treinando modelo
model.fit(X_train, y_train)

# Predizendo
y_pred = model.predict(X_test)

# Acurácia
accuracy = accuracy_score(y_test, y_pred)
print(f"A acurácia do modelo foi de {accuracy:.2f}.")
```

### 13. importar\_biblioteca\_padrao.py

(Aula 3 — Uso de módulos da biblioteca padrão)

python

```
import os
print(os.listdir('.'))

import datetime
data_atual = datetime.date.today()
print(data_atual)

import json
dados = '{"nome": "Alice", "idade": 30}'
dados_dict = json.loads(dados)
print(dados_dict['nome'])
```

## 14. urllib\_exemplo.py

*(Aula 3 — Consumindo dados de uma URL)*

python

```
from urllib import request

resposta = request.urlopen('https://www.example.com')
print(resposta.read().decode('utf-8'))
```

## 15. regex\_email.py

*(Aula 3 — Encontrando e-mails com regex)*

python

```
import re

padrao = r'\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}\b'
texto = 'Entre em contato pelo e-mail: usuario@example.com'
emails = re.findall(padrao, texto, re.IGNORECASE)
print(emails)
```

## 16. salvar\_reviews.py

*(Aula 4 — Script para extrair comentários e salvar em arquivo)*

python

```
# Exemplo estruturado – substituir pela lógica de extração real
reviews = [
    "Ótimo produto, estou muito satisfeito com a qualidade.",
    "O produto não atendeu às minhas expectativas, não recomendo.",
    "Produto excelente, entrega rápida, recomendo!"
]

with open('reviews.txt', 'w', encoding='utf-8') as f:
    for r in reviews:
        f.write(r + '\n')
```

✅ Agora, se você quiser, já posso gerar todos esses arquivos `.py` organizados em uma pasta para você exportar direto para o repositório.

Quer que eu já prepare esse pacote ZIP com todos eles?





