

Lista de Exercícios – Bibliotecas Importantes

1. Biblioteca `math`: Constantes e Funções Matemáticas

Enunciado:

Usando a biblioteca `math`, calcule:

1. A raiz quadrada de 144.
2. O valor de `pi` multiplicado por 2 (use `math.tau`).

```
import math

raiz = math.sqrt(144)
tau_valor = math.tau

print(raiz)
print(tau_valor)
```

Resposta:

```
12.0
6.283185307179586
```

2. Biblioteca `random`: Trabalhando com Aleatoriedade

Enunciado:

1. Sorteie um número entre 1 e 10.
2. Escolha aleatoriamente um nome de uma lista.

```
import random

numero = random.randint(1, 10)
nome = random.choice(["Luana", "Carlos", "Marina", "Pedro"])

print(numero)
print(nome)
```

Resposta:

(Saída pode variar)

7

Marina

3. Biblioteca **threading**: Execução Paralela

Enunciado:

Crie duas funções simples que imprimem mensagens diferentes e execute-as ao mesmo tempo com **threading**.

```
import threading
import time

def tarefa1():
    print("Tarefa 1 iniciada")
    time.sleep(1)
    print("Tarefa 1 finalizada")

def tarefa2():
    print("Tarefa 2 iniciada")
    time.sleep(1)
    print("Tarefa 2 finalizada")

t1 = threading.Thread(target=tarefa1)
t2 = threading.Thread(target=tarefa2)

t1.start()
t2.start()
```

Resposta:

Saída (a ordem pode variar):

Tarefa 1 iniciada
Tarefa 2 iniciada
Tarefa 1 finalizada
Tarefa 2 finalizada

4. Biblioteca **operator**: Ordenando Dicionários

Enunciado:

Ordene o seguinte dicionário pelo valor (pontuação) de forma decrescente.

```
import operator
```

```
pontuacoes = {"Ana": 80, "Bruno": 95, "Carla": 70}
ordenado = sorted(pontuacoes.items(), key=operator.itemgetter(1), reverse=True)
print(ordenado)
```

Resposta:

```
[('Bruno', 95), ('Ana', 80), ('Carla', 70)]
```

5. Biblioteca **NumPy**: Trabalhando com Matrizes

Enunciado:

Crie uma matriz 2x2 com o NumPy e calcule a soma de todos os elementos.

```
import numpy as np

matriz = np.array([[1, 2], [3, 4]])
soma = np.sum(matriz)

print(matriz)
print("Soma:", soma)
```

Resposta:

```
[[1 2]
 [3 4]]
Soma: 10
```

6. Biblioteca **Pandas**: Manipulando Dados em Tabelas

Enunciado:

Crie um DataFrame com nomes e idades e mostre apenas as pessoas com mais de 18 anos.

```
import pandas as pd

dados = pd.DataFrame({
    "Nome": ["Ana", "Bruno", "Carla"],
    "Idade": [17, 22, 19]
})

adultos = dados[dados["Idade"] > 18]
print(adultos)
```

Resposta:

	Nome	Idade
1	Bruno	22
2	Carla	19