

Códigos

MICROATIVIDADE 1

```
• listas.py

#Microatividade 1: Descrever a manipulação
#da estrutura de dados lista em Python

lista_mesclada = [1, 2, 3, "Olá, Python", True, 12.6]
print(type(lista_mesclada))
print(lista_mesclada)

lista_mesclada.append("Lista alinhada")
print(lista_mesclada)

lista_mesclada.insert(4, 5)
print(lista_mesclada)

print(len(lista_mesclada))

lista_mesclada.remove(2)
print(lista_mesclada)

print(lista_mesclada[1])

nova_lista_mesclada = lista_mesclada[:4]
print("Conteúdo da nova lista mesclada é:", nova_lista_mesclada)
```

MICORATIVIDADE 2

```
• tuplas.py

#Microatividade 2: Descrever a manipulação
#da estrutura de dados tupla em Python

primeira_tupla = 1 ,2 , 3, 4, "Olá, Tupla"
print(type(primeira_tupla))
print("O conteúdo da tupla é:", primeira_tupla)

indice_4 = primeira_tupla.index(4)
print("O indice do 4º elemento é:", indice_4)

contem_3 = 3 in primeira_tupla
print("A tupla contém o elemento 3?", contem_3)

contem_33 = 33 in primeira_tupla
print("A tupla contém o elemento 33?", contem_33)
```

MICORATIVIDADE 3

```
• sets.py

#Microatividade 3: Descrever a manipulação
#da estrutura de dados set em Python

set_inicial = {11, 12, 13, 14}
print(type(set_inicial))
print("O conteúdo inicial do set é: ", set_inicial)

set_inicial.add(15)
print("O conteúdo do set com o elemento 15 adicionado:", set_inicial)

set_inicial.update([1,2,3,4,5])
print("O conteúdo do set com os novos elementos adicionados é:", set_inicial)

set_inicial.discard(13)
print("Conteúdo do set_inicial após remover 13:", set_inicial)

novo_set = {20, 21, 23, 1, 2}
print("Conteúdo do novo_set:", novo_set)

print("União entre set_inicial e novo_set:", set_inicial.union(novo_set))

print("Interseção entre set_inicial e novo_set:", set_inicial.intersection(novo_set))

print("Diferença entre set_inicial e novo_set:", set_inicial.difference(novo_set))

print("Diferença simétrica entre set_inicial e novo_set:", set_inicial.symmetric_difference(novo_set))
```

MICORATIVIDADE 4

```
• dicionarios.py

#Microatividade 4: Descrever a criação da
#estrutura de dados dicionário em Python

meu_dicionario = {
    1: "Python",
    2: "Java",
    3: "PHP"
}

print("Conteúdo do meu_dicionario:", meu_dicionario)

print("Tipo de dados de meu_dicionario:", type(meu_dicionario))

print("Valor da chave 'linguagem' para o código 1:", meu_dicionario.get(1))
print("Valor da chave 'linguagem' para o código 2:", meu_dicionario.get(2))
print("Valor da chave 'linguagem' para o código 3:", meu_dicionario.get(3))

print("Tamanho de meu_dicionario:", len(meu_dicionario))

dicionario_frutas = {
    1: {"nome": "limão", "tipo": "ácida"},
    2: {"nome": "laranja", "tipo": "ácida"},
    3: {"nome": "manga", "tipo": "semiácida"},
    4: {"nome": "maçã", "tipo": "semiácida"},
    5: {"nome": "banana", "tipo": "doce"},
    6: {"nome": "mamão", "tipo": "doce"}
}

print("Valores da chave 1 em dicionario_frutas:", dicionario_frutas[1])

print("Valores da chave 2 em dicionario_frutas:", dicionario_frutas[2])

for chave, valor in dicionario_frutas.items():
    print(f"Valores da chave {chave} em dicionario_frutas: nome = {valor['nome']}, tipo = {valor['tipo']}")
```

MICORATIVIDADE 5

```
• dicionarios2.py

#Microatividade 5: Descrever a atualização
#de dados em um dicionário

dicionario_pessoas = {
    1: {'nome': 'Maria', 'idade': 26, 'nacionalidade': 'brasileira'}
}

novos_elementos = {
    2: {'nome': 'João', 'idade': 30, 'nacionalidade': 'portuguesa'},
    3: {'nome': 'Ana', 'idade': 22, 'nacionalidade': 'espanhola'}
}

dicionario_pessoas.update(novos_elementos)
print("Dicionário de pessoas atualizado:", dicionario_pessoas)

novo_dicionario_pessoas = dicionario_pessoas.copy()
print("Novo dicionário de pessoas:", novo_dicionario_pessoas)

elemento_removido = dicionario_pessoas.pop(2)
print("Conteúdo do primeiro dicionário após remover o elemento:", dicionario_pessoas)

print("Elemento removido:", elemento_removido)
elemento_removido = dicionario_pessoas.popitem()

print("Conteúdo do primeiro dicionário após remover o último elemento:", dicionario_pessoas)

dicionario_pessoas.clear()
novo_dicionario_pessoas.clear()
```

Missão Prática | Estruturando os Dados

main.py

```
# Missão Prática | Estruturando os Dados
# funções da missão prática

def calcular_media(notas):
    if len(notas) != 4:
        raise ValueError("A lista de notas deve conter exatamente 4 elementos.")
    return sum(notas) / len(notas)

def verificar_reprovacao(media):
    return media < 6

def alunos_reprovados(dados_alunos):
    reprovados = []
    for matricula, nome, notas in dados_alunos:
        media = calcular_media(notas)
        if verificar_reprovacao(media):
            reprovados.append(f"Aluno Reprovado: {nome} - Matrícula: {matricula} - Média Final: {media:.2f}")
    return reprovados

operacoes.py

#Missão Prática | Estruturando os Dados
#Operações da missão prática

from main import calcular_media, verificar_reprovacao, alunos_reprovados

dados_alunos = [
    (26, "Maria", [8, 7, 5, 9]),
    (101, "Ana", [9, 9, 8, 9]),
    (13, "João", [6, 5, 5, 5]),
    (37, "Ágatha", [8, 6, 7.5, 9]),
    (72,"Joaquin", [6, 5.5, 5, 7]),
    (5, "Félix", [10, 8, 8, 8])
]

alunos_reprovados = alunos_reprovados(dados_alunos)

for aluno in alunos_reprovados:
    print(aluno)
```