

# Atividade S2 A3

**Nome:** Adalberto Caldeira Brant Filho

**Repositório GitHub:** <https://github.com/adalbertobrant/lipai>

## Código das Videoaulas

### Aula 01 - Tipos de Dados - Listas

As listas podem ser:

- \* ordenadas --> aceitam ordenação
- + mutáveis --> os elementos da lista podem ser editados ou terem seus valores alterados, com inclusão ou exclusão
- indexáveis --> podem ser acessados através de um índice

```
""" Aula 01 - Tipos de Dados - Listas """

# exemplo lista vazia
nomes = []
print(nomes, type(nomes))

nomes = ['jósé', 'maria', 'pedro', 'paulo', 'marcos']
print(nomes, type(nomes))

# permite adicionar tipos diferentes
coisas = ['panela', 2, 1.614, ['outra_lista', 'coisas'], 100]
print(coisas, type(coisas))

# permite acessar elementos através de um índice
print(coisas[0])
print(coisas[0:3])
print(coisas[-1])
print(coisas[:2])
print(coisas[3:6])

# modificação de elementos na lista

lista_nomes = ['Maria', 'Marta', 'Ana', 'Priscila', 'Raquel']
print(lista_nomes[0])
lista_nomes[3:] = ['Pedro', 'João']
print(lista_nomes)

# tamanho de uma lista
print(len(lista_nomes))

# adicionar elementos na lista usando o append

nomes = []
nomes.append('Josefo')
```

```

nomes.append('Marta Gomes')

# insere um elemento em posição específica
# o insert desloca os elementos e adiciona elementos de acordo com o índice
passado

nomes.insert(0, 'Guilherme Tavares')
print(nomes)

# método extend - acrescenta uma lista em outra lista no final da mesma
nomes2 = ['carlos gomes', 'caio silva']
nomes.extend(nomes2)
print(nomes)

# remover elementos da lista - passando o elemento a ser removido
# deve ser usado sempre com tratamento de erros
print(nomes)
nomes.remove('carlos gomes')
print(nomes)

# del - deleta nomes passando o índice, remove da memória
print(nomes)
del nomes[0]
print(nomes)
# se usar del nomes apenas , irá apagar a variável da memória

# limpar o conteúdo da lista deixando ela vazia
print(f'Lista nomes -> {nomes}')
nomes.clear() # limpa elementos da lista (todos)
print(nomes)

# iteração em listas
for nome in lista_nomes:
    print(nome)

for i in range(len(lista_nomes)):
    print(lista_nomes[i])

```

## Aula 02 -Tipos de Dados - Tuplas

Tuplas são conjuntos de elementos que são ordenadas e tem diferenças em relação as listas por serem imutáveis

As tuplas são:

- + ordenadas
- + imutáveis
- + indexáveis

```

""" Aula 02 - Tipos de Dados - Tuplas """

# criação da tupla

nomes = ('Maria', 'Pedro', 'João')
print(nomes, type(nomes))

```

```

# acessar os elementos da tupla da mesma forma que acessa em uma lista
print(nomes[0])
print(nomes[0:3])
print(nomes[-1])

# a leitura de dados na tupla e lista são iguais
# a tupla não aceita modificação de valores

# iteração em tuplas é igual a de listas
for nome in nomes:
    print(nome)

# outra forma de criar tuplas

nomes2 = 'ana', 'amelia', 'marcos'
print(nomes2, type(nomes2))

# unpack - atribuir para variáveis os valores que estão na tupla
nome1 = nomes[0]
nome2 = nomes[1]
nome3 = nomes[2]

# usando o unpack temos:
nome1, nome2, nome3 = nomes
# o unpack deve ser igual a quantidade de elementos da tupla ou o seu fatiamento
nome1, nome2 = nomes[0:2] # unpack fatiado
print(nome1, nome2)

# juntas duas tuplas

todos_nomes = nomes + nomes2
print(todos_nomes)

# para escolher entre listas e tuplas , avaliar os dados se imutáveis preferir
usar tupla

```

## Aula 03 - Tipos de Dados - Set ( Conjuntos )

Set significa conjunto em português e eles tem as seguintes características:

- \* não são ordenados
- \* são mutáveis
- \* não tem um índice, não indexáveis
- \* não aceitam elementos duplicados ( apenas elementos únicos )

```

""" Aula 03 - Tipos de Dados - Set ( Conjuntos ) """
# criar um set
numeros = {1, 2, 3, 4, 5, 6, 6.25, 6.5, 7, 8}
print(numeros, type(numeros))

# acesso dos elementos no set
for numero in numeros:
    print(numero)

```

```

# adicionar itens no set
numeros.add(100)
print(numeros)

# se adicionar um item que já existe no set
# apesar de rodar o programa ele não é adicionado
numeros.add(100)
print(numeros)
print('-----')
# adicionar elementos de um set em outro
print(numeros)
numeros2 = {42, 3, 1, 11, 17}
print(numeros2)
numeros.update(numeros2)
print(numeros, type(numeros))
print('-----')

# remove elementos de um set
numeros.remove(100)
print(numeros)

# copiando um set para outro
outro_set = numeros.copy()
print(outro_set)

```

## Aula 04 - Tipos de Dados - Dicionários ( dic )

Dicionários em python sempre tem uma chave e um valor associado.

Em um dicionário não existe um índice , sendo considerado uma coleção do tipo chave, valor.

A chave é única.

O dicionário é mutável.

```

""" Aula 04 - Tipos de Dados - Dicionário (dic) """
# criar um dicionário
carro = {
    'marca': 'ford',
    'modelo': 'mustang',
    'ano': 1964
}
print(carro, type(carro))

# acessar os valores de um dicionário
print(carro['marca'])
print(carro.get('marca'))

# pegar todas as chaves
print(carro.keys(), type(carro.keys())) # retorna um conjunto de chaves

# pegar todos os valores
print(carro.values(), type(carro.values()))

# pegar todos os pares (chave:valor)
print(carro.items(), type(carro.items())) # retorna conjunto de tuplas

```

```

# verifica se a chave existe
print("marca" in carro)
print("cor " in carro)

# adicionar chave/valor de forma dinâmica
carro['cor'] = 'azul'
print(carro)
print('cor' in carro)

# remover um item do dicionario através de sua chave
carro.pop('marca')
print(carro)

# o pop no dicionário retorna o valor removido
valor_removido = carro.pop('cor')
print(f'A cor {valor_removido} foi removida do dicionário{carro}')

# iterar com o dicionário
for key in carro:
    print(key, carro[key], type(key))

for key in carro.keys():
    print(key)

for value in carro.values():
    print(value)

for key, value in carro.items():
    print(key, value)

# lista de dicionários

aluno1 = {
    'nome': 'João',
    'email': 'joao@mail.com',
    'notas': [10, 7, 8]
}

aluno2 = {
    'nome': 'Maria',
    'email': 'maria@mail.com',
    'notas': [10, 5, 6]
}

alunos = [aluno1, aluno2]

for aluno in alunos:
    print(aluno['nome'], aluno['email'])
    for nota in aluno['notas']:
        print(nota)

```

## Exercícios de fixação

## Ex01

```
"""
ex01.py
o Solicite ao usuário 3 números.
o Armazene os valores em uma lista.
o Ao final, apresente o menor e o maior elemento.

"""

lista = []
CONTADOR = 0
while CONTADOR < 3:
    lista.append(int(input('Entre um valor para a lista -> ')))
    CONTADOR += 1

menor = maior = lista[0]

for x in lista:
    if x < menor:
        menor = x
    if x > maior:
        maior = x

print(f'O menor valor da lista {lista} é {menor}')

print(f'O maior valor da lista {lista} é {maior}')
```

## Ex02

```
"""
ex02.py
o Solicite ao usuário o mês do ano em formato numérico: 1, 2, 3, ..., 12.
o Apresente o nome do mês correspondente (ex.: entrada 3 → saída
Março).
o Implementar usando uma Tupla (tuple).
"""

mes_ano = int(input("Entre o mês do ano -> "))

meses = ("Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho",
         "Julho", "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro")

if 1 <= mes_ano <= 12:
    print(meses[mes_ano - 1])
```

## Ex03

```
"""
ex03.py
o Solicite ao usuário o mês do ano em formato numérico: 1, 2, 3, ..., 12.
o Apresente o nome do mês correspondente (ex.: entrada 3 → saída
Março).
o Implementar usando um Dicionário (dict).
"""

mes_ano = int(input("Entre o mês do ano -> "))
```

```
meses = {
    1: "Janeiro",
    2: "Fevereiro",
    3: "Março",
    4: "Abril",
    5: "Maio",
    6: "Junho",
    7: "Julho",
    8: "Agosto",
    9: "Setembro",
    10: "Outubro",
    11: "Novembro",
    12: "Dezembro"
}

if 1 <= mes_ano <= 12:
    if mes_ano in meses:
        print(meses[mes_ano])
    else:
        print("Entre um mês de 1 a 12 apenas")
```