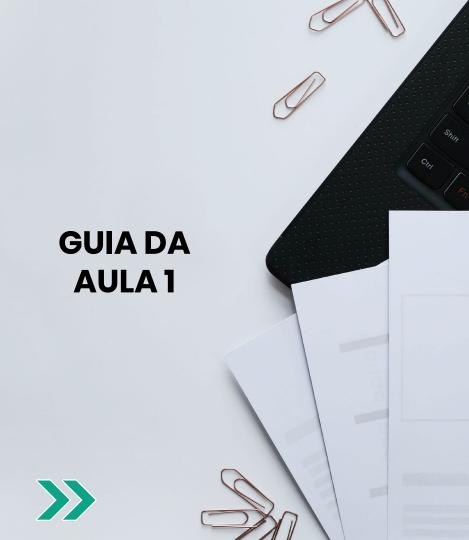


# Python para análise de dados





# **ESTRUTURA DE DADOS**







# **Estruture listas**

- Motivação
- Definição
- Operações
- Métodos
- Conversão
- Revisitando a motivação



Acompanhe aqui os temas que serão tratados na videoaula





#### 1. Motivação

O aplicativo do seu banco registra toda a sua movimentação financeira. Ao final do dia, o app consolida o saldo final para que você possa controlar sua vida financeira.

```
In [ ]:
         dia 11 saldo inicial = 1000
In [ ]:
         dia 11 transacao 1 = 243
         dia 11 transacao 2 = -798.58
         dia 11 transacao 3 = 427.12
         dia 11 transacao 4 = -10.91
In [ ]:
         dia 11 saldo final = dia 11 saldo inicial + \
                               dia_11_transacao_1 + \
                               dia 11_transacao_2 + \
                               dia 11 transacao 3 + \
                               dia 11 transacao 4
```





## 1. Motivação

```
In [ ]: print(dia_11_saldo_final)
```

Será que existe uma forma melhor de armazenar as transações diárias?





## 2. Definição

Armazenam sequências mutáveis e ordenadas de valores. São do tipo list:

```
In []:
    usuario_web = [
        'André Perez',
        'andre.perez',
        'andre123',
        'andre.perez@gmail.com'
]

    print(usuario_web)
    print(type(usuario_web))
```





## 2. Definição

```
In [ ]:
         idade = 20
         saldo em conta = 723.15
         usuario loggedin = True
         usuario web = [
              'André Perez',
             idade,
              'andre.perez',
             'andre123',
             'andre.perez@gmail.com',
             saldo em conta,
             usuario loggedin
         print(usuario web)
         print(type(usuario_web))
```





## 3. Operações

As operações da estrutura do tipo *list* são: + (concatenação).

#### **Exemplo**: Fabricantes de *hardware mobile*

```
In []:
    fabricantes_mobile_china = ['xiaomi', 'huawei']
    fabricantes_mobile_eua = ['apple', 'motorola']
    fabricantes_mobile = fabricantes_mobile_china + fabricantes_mobile_eua

    print(fabricantes_mobile_china)
    print(fabricantes_mobile_eua)
    print(fabricantes_mobile)
```





#### 3. Operações

Outra operação muito utilizada é a de fatiamento (slicing), semelhante ao de strings:

#### Fatiamento fixo:

```
In []: print(f'0: {fabricantes_mobile[0]}')
    print(f'-1: {fabricantes_mobile[-
    1]}')
```

#### Fatiamento por intervalo:

```
In []: fabricantes_mobile_china = fabricantes_mobile[0:2]
    fabricantes_mobile_eua = fabricantes_mobile[2:len(fabricantes_mobile)]

print('china: ' + str(fabricantes_mobile_china))
print('eua: ' + str(fabricantes_mobile_eua))
```





#### 3. Operações

Podemos adicionar elementos a uma posição específica da lista:

```
In []: print(fabricantes_mobile)

In []: fabricantes_mobile[2] = 'nokia'
    print(fabricantes_mobile)
```





#### 4. Métodos

São métodos nativos do Python que nos ajudam a trabalhar no dia a dia com listas.

```
In []: juros = [0.05, 0.07, 0.02, 0.04, 0.08]
print(juros)

In []: # inserir um elemento sem substituir: list.insert(index, val)
juros.insert(0, 0.10)
print(juros)

In []: # inserir um elemento no fim da lista: list.append(val)
juros.append(0.09)
print(juros)
```





#### 4. Métodos

```
In []: # remover um elemento pelo valor: list.remove(val)
    juros.remove(0.1)
    print(juros)

In []: # remover um elemento pelo indice: list.pop(val)
    terceiro_juros = juros.pop(2)
    print(terceiro_juros)
In []: print(juros)
```





#### 5. Conversão

Podemos converter alguns tipos de variáveis em listas, como strings.

```
In []: email = 'andre.perez@gmail.com'
    caracteres_email = list(email)

    print(email)
    print(caracteres_email)
```





#### 6. Revisitando a motivação

```
In [ ]:
         dia 11 saldo inicial = 1000
In [ ]:
         dia 11 transacoes = []
         dia 11 transacoes.append(243)
         dia 11 transacoes.append(-798.58)
         dia 11 transacoes.append(427.12)
         dia 11 transacoes.append(-10.91)
         print(dia 11 transacoes)
In [ ]:
         dia 11 saldo final = dia 11 saldo inicial + \
                               dia 11 transacoes[0] + \
                               dia 11 transacoes[1] + \
                               dia 11 transacoes[2] + \
                               dia 11 transacoes[3]
         print(dia 11 saldo final)
```

