

# LISTAS

Listas são coleções de objetos que podem ser de qualquer tipo, inclusive outras listas.

As listas em Python são mutáveis, podendo ser alteradas a qualquer momento.

Listas podem ser fatiadas como strings.

Os valores de uma lista podem ser acessados pelo seu índice.

Uma lista pode conter zero ou mais elementos.

O tamanho de uma lista é igual a sua quantidade de elementos.



# CRIAÇÃO DE LISTAS

Criação de uma lista vazia

```
lista = []
```

Uma lista com três elementos strings

```
bancos = ["Banco do Brasil", "Caixa", "Santander"]
```

Cada item de uma lista possui um índice iniciando com zero

```
bancos = ["Banco do Brasil", "Caixa", "Santander"]
```

```
0 = "Banco do Brasil"
```

```
1 = "Caixa"
```

```
2 = "Santander"
```



# ALTERANDO ITENS DA LISTA PELO SEU ÍNDICE

```
bancos= ["Banco do Brasil ", " Caixa", "Santander"]  
print(bancos)  
# Resultado: ["Banco do Brasil ", " Caixa", "Santander"]  
bancos[1] = "Itaú"  
print(bancos)  
# Resultado: ['Banco do Brasil ', 'Itaú', 'Santander']  
bancos[-1] = "C6"  
print(bancos)  
# Resultado: ['Banco do Brasil ', 'Itaú', 'C6']
```

# INCREMENTAR O VALOR DE UMA LISTA

```
print(bancos)
# Resultado: ['Banco do Brasil ', 'Itaú', 'C6']
bancos = bancos + ["Bradesco", "Nubank" ]
print(bancos)
# Resultado: ['Banco do Brasil ', 'Itaú', 'C6', 'Bradesco', 'Nubank']
bancos += ["Safra"]
print(bancos)
# Resultado: ['Banco do Brasil ', 'Itaú', 'C6', 'Bradesco', 'Nubank', 'Safra']
```

# MÉTODOS UTILIZADOS EM LISTAS

1. Para adicionar um item à lista usamos o método **append**
2. Para adicionar um item à lista definindo um índice específico usamos o método **insert**
3. Para remover um elemento da lista usamos o método **remove**
4. Para ordenar os itens de uma lista usamos o método **sort**
5. Para inverter os itens de uma lista usamos o método **reverse**
6. Retornando a quantidade de ocorrências de um elemento em uma lista com **count**
7. Para remover o último elemento de uma lista usamos o método **pop** e retorna o elemento removido
8. A função **del** pode ser usada para remover elementos de uma lista
9. Para limpar uma lista inteira utiliza-se **clear**

# EXEMPLOS

```
lista = [4, 5, 3, 5]
print(lista)
lista.append(2)
print(lista)
lista.insert(2, -3)
print(lista)
lista.remove(4)
print(lista)
lista.sort()
print(lista)
lista.reverse()
print(lista)
```

```
qnt = lista.count(5)
print(qnt)
exc = lista.pop()
print(lista)
print(exc)
del lista[2]
print(lista)
del lista[2:5]
print(lista)
lista.clear()
print(lista)
```

# UMA LISTA DENTRO DE UMA OUTRA LISTA

```
compra = [10.2, 3.35, 16.3, ["tomate", "sabonete", "arroz"]]
print(compra)
produtos = compra[3]
print(produtos)
total = compra[0]+compra[1]+compra[2]
print(total)
letra = ["a", "b", "c"]
num = [2,4,6]
tudo = [letra,num]
print(tudo)
print(f"Letras: {tudo[0]}")
print(f"Numeros: {tudo[1]}")
```

# MÉTODOS UTILIZADOS EM LISTAS

1. Obter o tamanho de uma lista **len**
2. Obter o índice de um determinado elemento da lista **index**

```
letra = ["a", "b", "c"]  
print(f"tamanho da lista: {len(letra)}")  
print(f"endereço da letra b: {letra.index('b')}")
```



# VERIFICANDO A EXISTÊNCIA DE UM ITEM NA LISTA


```
letras = ["a", "b", "c", "d", "e", "f"]
var = input("informe uma letra: ")
if var.lower() in letras:
    print(f"A Letra '{var.lower()}' está na lista")
else:
    print(f"A Letra '{var.lower()}' NÃO está na lista")
```

# ADICIONANDO ELEMENTOS FORNECIDOS PELO USUÁRIO À LISTA

```
nova = []  
while True:  
    num = int(input("Digite um numero inteiro(0 para sair): "))  
    if num==0:  
        break  
    nova.append(num)  
print(nova)
```

## DESAFIO

Escreva um programa que solicite vários números ao usuário, sendo um de cada vez, possibilitando encerrar a entrada de dados informando zero. Armazene os números em uma lista, depois percorra esta lista alimentando outras duas listas, uma com números pares e outra com números ímpares. Ordene e imprima os números em ordem crescente e some os valores e imprima o resultado.



# EXERCÍCIO 1

Escreva um programa que solicite vários números ao usuário, sendo um de cada vez, possibilitando encerrar a entrada de dados informando zero. Adicione os números informados em uma lista e, ao final do programa, imprima a soma de todos os números, a multiplicação de todos os números, o maior e o menor número informado.

## Exemplo de execução:

Informe um número (zero para sair): 10

Informe um número (zero para sair): 5

Informe um número (zero para sair): 20

Informe um número (zero para sair): 0



## EXERCÍCIO 2

Escreva um programa que solicite várias palavras ao usuário, sendo uma de cada vez, possibilitando encerrar a entrada de dados informando zero. Adicione as palavras informadas em uma lista. Após o usuário informar "0" (zero), solicite a palavra que ele deseja contar. Desta forma, o programa deverá contar as ocorrências daquela palavra. Ao final imprima o resultado.

### Exemplo de execução:

Informe uma palavra (zero para sair): Olá  
Informe uma palavra (zero para sair): Casa  
Informe uma palavra (zero para sair):  
Bicicleta  
Informe uma palavra (zero para sair): Olá

Informe uma palavra (zero para sair): Olá  
Informe uma palavra (zero para sair): Hello  
Informe uma palavra (zero para sair): 0  
Informe a palavra que deseja contar: Olá  
Temos 3 ocorrências de Olá.

## EXERCÍCIO 3

Escreva um programa que solicite vários números ao usuário, sendo um de cada vez, possibilitando encerrar a entrada de dados informando zero. Adicione os números em uma lista, crie uma outra lista contendo os números sem repetição e uma outra contendo os números que se repetem na primeira lista.

Exemplo de execução:

Informe um número (zero para sair): 10  
Informe um número (zero para sair): 5  
Informe um número (zero para sair): 4  
Informe um número (zero para sair): 10  
Informe um número (zero para sair): 88  
Informe um número (zero para sair): 4

Informe um número (zero para sair): 10  
Informe um número (zero para sair): 4  
Informe um número (zero para sair): 8  
Informe um número (zero para sair): 0  
Números informados: [10, 5, 4, 10, 88, 4, 10, 4, 8]  
Números se repetição: [10, 5, 4, 88, 8]  
Somente números que se repetiram: [10, 4]

## EXERCÍCIO 4

Escreva um programa que solicite vários números ao usuário, sendo um de cada vez, possibilitando encerrar a entrada de dados informando zero. Adicione os números em uma lista, crie uma outra lista contendo os números sem repetição e uma outra contendo os números que se repetem na primeira lista.

Exemplo de execução:

Informe um número (zero para sair): 10  
Informe um número (zero para sair): 5  
Informe um número (zero para sair): 4  
Informe um número (zero para sair): 10  
Informe um número (zero para sair): 88  
Informe um número (zero para sair): 4  
Informe um número (zero para sair): 10

Informe um número (zero para sair): 4  
Informe um número (zero para sair): 8  
Informe um número (zero para sair): 0  
Números informados: [10, 5, 4, 10, 88, 4, 10, 4, 8]  
Números se repetição: [10, 5, 4, 88, 8]  
Somente números que se repetiram: [10, 4]