# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**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.

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**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”**

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**Listas.**

Você pode alterar os 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']
```

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**Listas.**

Você pode incrementar o valor em uma lista como nos exemplos abaixo.

```
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']
```

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**Listas.**

**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 **sorted**.

**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**.

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**Listas.**

**Métodos utilizados em listas.**

**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**.

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**Listas.**

Métodos utilizados em listas.

```
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)
```

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**Listas.**

É possível ter uma lista dentro de 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]}")
```

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**Listas.**

**Outros 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')}")
```

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**Listas.**

Verificar a existência de um item dentro da 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")
```

# LÓGICA DE PROGRAMAÇÃO

LINGUAGEM DE PROGRAMAÇÃO

**Listas.**

Adicionar elementos fornecidos pelo usuário em uma lista.

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

# VAMOS PRATICAR.

EXEMPLOS E ATIVIDADES DE LISTAS