

Tipo de dado **lista**

Sumário

- [Sintaxe de listas](#)
- [Exercício 1 - Impressão de valores da lista](#)
- [Exercício 2 - Soma de valores da lista](#)
- [Exercício 3 - Pertence à lista](#)
- [Exercício 4 - Encontra índice](#)
- [Exercício 5 - Comparação de duas listas](#)
- [Exercício 6 - Lista com valores do usuário](#)

Listas - Motivação

- Até agora trabalhamos com variáveis que armazenam um único valor
- Em muitas situações, precisamos armazenar um conjunto de valores. Uma forma básica de fazermos isso seria definir N variáveis para armazenar N valores de entrada
- Quais os problemas dessa abordagem?

Listas - Motivação

- Quais os problemas dessa abordagem?
 1. Se N for grande, precisaríamos definir dezenas ou até centenas de variáveis
 2. Se N for entrada do usuário, não há como definirmos um número N de novas variáveis durante a execução do código

Listas

- Listas são **variáveis** que permitem o armazenamento de **diversos valores**
- Em C e outras linguagens de programação elas também são chamadas de **arrays** ou **vetores**

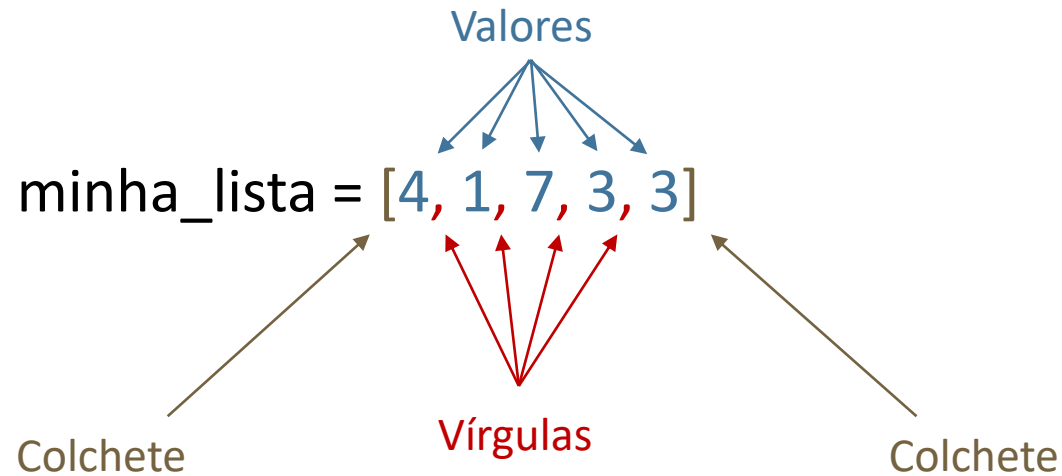
Listas

- Definimos uma lista utilizando a seguinte sintaxe:

```
minha_lista = [4, 1, 7, 3, 3]
```

Listas

- Definimos uma lista utilizando a seguinte sintaxe:

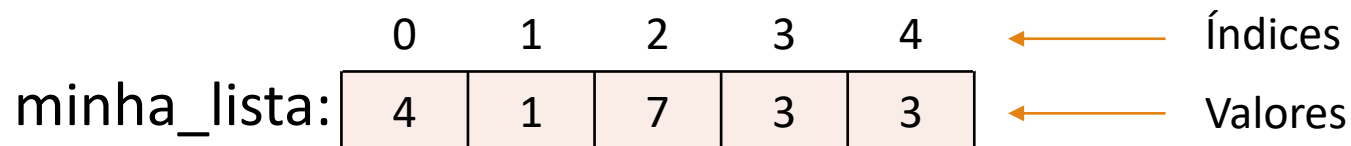


Listas

- Definimos uma lista utilizando a seguinte sintaxe:

```
minha_lista = [4, 1, 7, 3, 3]
```

- Uma lista pode ser entendida como um conjunto de “caixas” contendo valores, e podemos referenciar cada caixa através de um índice:



Listas

minha_lista:

0	1	2	3	4
4	1	7	3	3

- Acessamos cada elemento da lista através do seu respectivo índice
- Por exemplo, o terceiro elemento da lista (número 7) é acessado utilizando o índice 2:

minha_lista[2]

Acessa o número 7, que está na posição de índice 2

Listas

- No código abaixo, o valor impresso na tela será 8, pois **valores[1]** acessa o elemento de índice 1 da lista, que é o valor 2

```
valores = [5, 2, 7, 4]  
num = valores[1]*4  
print(num)
```

Listas

- Importante! Os índices dos elementos de uma lista SEMPRE começam no valor 0

	0	1	2	3	4
minha_lista:	4	1	7	3	3

Listas

Listas podem armazenar qualquer tipo de valor:

- `lista_inteiros = [4, 6, 2, 7, 1, 8, 4, 12, 5, 356]`
- `lista_float = [3.4, 6.4, 8.7, 10.3]`
- `lista_nomes = ["Pedro", "Silvio", "Aline"]`
- `lista_booleanos = [True, False, False, True]`

Até mesmo outras listas:

- `lista_listas = [[1, 5], [4, 1], [7, 10], [12, 4]]`

Função len

A função **len** é utilizada para obtermos o tamanho (número de elementos) de uma lista

No código abaixo, será impresso na tela o valor 5, que é o número de elementos da lista.

```
valores = [5, 2, 7, 4, 3]  
tam = len(valores)  
print(tam)
```

Exercício 1 - Impressão de valores

Faça um programa que imprima todos os valores de uma lista. Cada valor deve ser impresso em uma linha separada. Defina a lista no próprio código (não será entrada do usuário).

Exemplo:

Uma lista definida como

```
lista_valores = [3, 6, 1, 7, 3, 8]
```

é mostrada na tela da seguinte forma:

```
3  
6  
1  
7  
3  
8
```

Solução

```
valores = [3, 6, 1, 7, 3, 8]
```

```
n = len(valores)
for i in range(0, n):
    print(valores[i])
```

Solução 2

Python permite iterarmos diretamente os elementos de uma lista sem precisarmos utilizar os índices:

```
valores = [3, 6, 1, 7, 3, 8]

n = len(valores)
for valor in valores:
    print(valor)
```

Esse código é considerado mais adequado do que o do slide anterior. Ele também é mais eficiente, ou seja, roda mais rápido.

Mas como essa funcionalidade não existe em C, iremos utilizar a função `range()`, como no slide anterior, para ter códigos mais similares com a versão em C.

Exercício 2 - Soma

Faça um programa que imprima na tela a soma dos valores de uma lista.

Defina a lista no código (não será entrada do usuário).

Solução

```
valores = [3, 6, 1, 7, 3, 8]
```

```
n = len(valores)
```

```
soma = 0
```

```
for i in range(0, n):  
    soma += valores[i]
```

```
print(soma)
```

Exercício 3 - Pertence ou não

Faça um programa que verifique se um número digitado pelo usuário pertence a uma lista. Se o número pertencer à lista, o programa imprime “Número está contido na lista”. Caso contrário, o programa imprime “Número não encontrado”

Defina a lista no código (não será entrada do usuário).

Solução

```
valores = [3, 6, 1, 7, 3, 8]
```

```
valor = int(input("Digite um valor: "))
```

```
n = len(valores)
```

```
# Definimos uma variável chamada encontrou que começa com o  
# valor False. Se em algum momento o número for encontrado,  
# mudamos a variável para True
```

```
encontrou = False
```

```
for i in range(0, n):
```

```
    if valor==valores[i]:
```

```
        encontrou = True
```

```
        # Termina o laço porque o valor já foi encontrado
```

```
        break
```

```
if encontrou:
```

```
    print("O número pertence à lista")
```

```
else:
```

```
    print("O número não pertence à lista")
```

Exercício 4 - Pertence com índice

Faça um programa que verifica se um número digitado pelo usuário pertence a uma lista. **O programa imprime a posição (índice) da primeira ocorrência do número na lista.** Caso contrário, o programa imprime “Número não encontrado”

Defina a lista no código (não será entrada do usuário).

Solução

```
valores = [3, 6, 1, 7, 3, 8]

valor = int(input("Digite um valor: "))

n = len(valores)
# Variável indice armazenará o índice do valor encontrado.
# Iniciar ela como -1 possibilita usar ela para verificar
# se o valor foi encontrado
indice = -1
for i in range(0, n):
    if valor==valores[i]:
        indice = i
        # Termina o laço porque o valor já foi encontrado
        break

if indice!=-1:
    print(f"O número ocorre na posição {indice}")
else:
    print("O número não pertence à lista")
```

Exercício 5 - Comparação

Faça um programa que compare duas listas de mesmo tamanho e identifique se todos os valores delas são iguais e estão na mesma ordem.

Exemplo:

Entrada:

lista1: [4, 2, 6, 5, 7]

lista2: [4, 2, 3, 5, 7]

Saída:

As listas não são iguais

Solução

```
valores1 = [3, 6, 1, 7, 3, 8]
valores2 = [3, 6, 1, 4, 3, 8]
```

```
# Assumimos que as listas possuem o mesmo tamanho. Portanto,
# precisamos obter apenas o tamanho de uma das listas
n = len(valores1)
# Variável iguais começa em True, assumindo que as listas são
# iguais. Se algum elemento for diferente, ela será modificada
# para False
iguais = True
for i in range(0, n):
    if valores1[i] != valores2[i]:
        iguais = False
        break

if iguais:
    print(f"As listas são iguais")
else:
    print("As listas não são iguais")
```


Inserindo novos elementos em listas

- Por enquanto consideramos listas possuindo valores predeterminados
- Como inserir elementos durante a execução do programa?

- Primeiro, definimos uma lista vazia:

```
lista = []
```

- Então, adicionamos elementos utilizando o método **append**:

```
lista.append(4)
```

```
lista.append(7)
```

```
lista.append(2)
```

Exercício 6 - Inserção

Faça um programa que receba valores digitados pelo usuário e os insira em uma lista. Após receber todos os valores, o programa imprime a lista na tela

Solução

```
valores = []

while True:
    valor = input("Digite um valor ou 'e' para encerrar: ")
    if valor=="e":
        break
    else:
        valor = float(valor)
        valores.append(valor)

print(f"A lista digitada foi:\n{valores}")
```