

Lógica de Programação

Manipulando Lista com Python

Prof.: Caio Malheiros

caio.duarte@sp.senai.br

Plano de aula

- 1.O que são listas?
- 2.Criando uma lista
- 3.Acessando elementos
- 4.Modificando elementos
- 5.Adicionando elementos
- 6.Removendo elementos
- 7.Operadores com listas

O que são Listas?

Uma lista é uma coleção ordenada de elementos.

Em Python, listas podem conter diferentes tipos de dados.



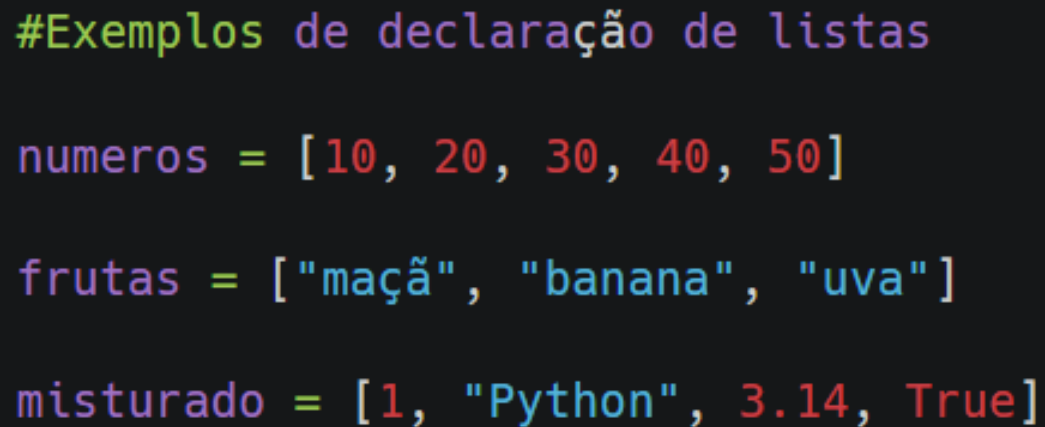
```
#Declarando uma lista em Python
```

```
minha_lista = [1, 2, 3, 4, 5]
```

Exemplos de Listas

Listas podem ser criadas usando colchetes [].

Exemplo:



```
#Exemplos de declaração de listas

numeros = [10, 20, 30, 40, 50]

frutas = ["maçã", "banana", "uva"]

misturado = [1, "Python", 3.14, True]
```

Acessando Elementos

Usamos índices para acessar os elementos da lista.

O primeiro elemento tem índice 0.

Exemplo



```
frutas = ["maçã", "banana", "uva"]
```


```
print(frutas[0]) # Saída: maçã
```

```
print(frutas[-1]) # Saída: uva
```

Modificando Elementos

Podemos alterar valores dentro de uma lista.

Exemplo:



```
numeros = [10, 20, 30]

numeros[1] = 50

print(numeros) # Saída: [10, 50, 30]
```

Adicionando Elementos

append() adiciona um elemento ao final da lista.

insert() adiciona um elemento em uma posição específica.

Exemplo:



```
lista = [1, 2, 3]
```

```
lista.append(4) # [1, 2, 3, 4]
```


```
lista.insert(1, 10) # [1, 10, 2, 3, 4]
```

Removendo Elementos

remove() remove o primeiro elemento encontrado.

pop() remove um elemento pelo índice e retorna o valor removido.

Exemplo:



```
lista = [1, 2, 3, 4]


lista.remove(2)  # [1, 3, 4]

elemento = lista.pop(1)
# Remove 3, lista fica [1, 4]
```


Percorrendo Listas

Podemos usar **for** para percorrer elementos de uma lista.

Exemplo:



```
numeros = [10, 20, 30]

for numero in numeros:
    print(numero)

#Qual será a saída?
```

Operadores com Listas

+ concatena listas.

* repete elementos da lista.

- Exemplo

```
lista1 = [1, 2, 3]
lista2 = [4, 5]

print(lista1 + lista2)
# [1, 2, 3, 4, 5]

print(lista1 * 2)
# [1, 2, 3, 1, 2, 3]
```

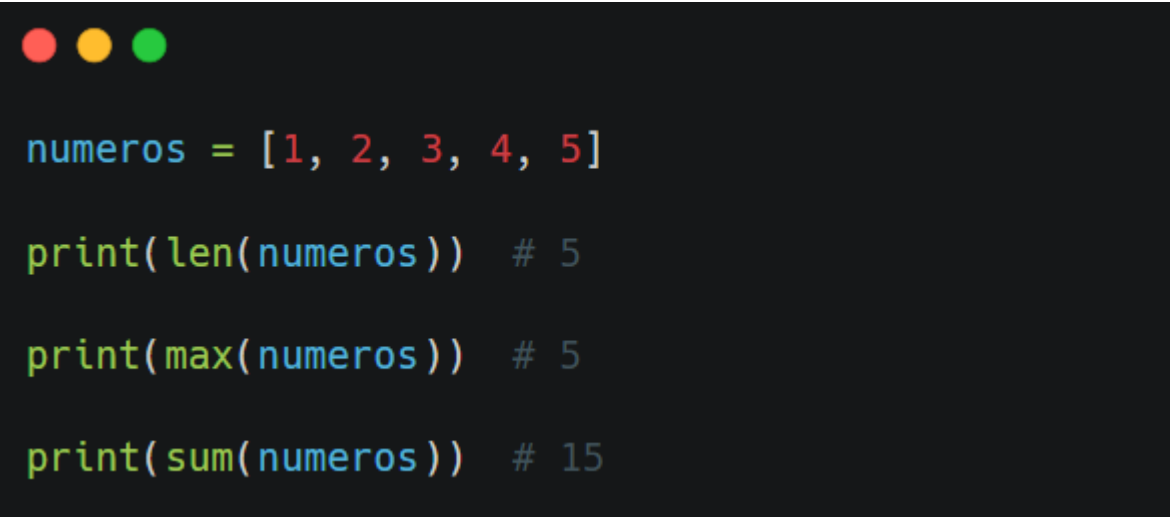
Funções úteis com Listas

len(lista): retorna o tamanho da lista.

max(lista): retorna o maior elemento.

min(lista): retorna o menor elemento.

sum(lista): retorna a soma dos elementos.

A screenshot of a terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The terminal displays a Python script that defines a list of numbers and uses built-in functions to calculate its length, maximum value, and sum.

```
numeros = [1, 2, 3, 4, 5]
print(len(numeros)) # 5
print(max(numeros)) # 5
print(sum(numeros)) # 15
```

Conclusão

1. Listas são uma estrutura de dados fundamental em Python.
2. Permitem armazenar, acessar e modificar coleções de dados de forma flexível.
3. Muitas operações e métodos facilitam seu uso no dia a dia.

Dúvidas?
Ótimo dia para todos!