
Listas 4

Docente: Simone Neves

Exemplo de programa em Python:

Entrada com ``input()`` (quando aplicável)

Processamento == COM LISTA

Saída com ``print()``

Uma lista é uma coleção de valores que podem ser alterados.

Pode conter diferentes tipos de dados.

É semelhante a um vetor em outras linguagens.

Exemplo:

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

```
# Criando uma lista com números
```

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

```
# Exibindo a lista
```

```
print("Minha lista:", numeros)
```



Saída esperada:

```
Minha lista: [10, 20, 30, 40]
```

```
.
```

Criando uma Lista

Use colchetes `[]` para criar uma lista.

Exemplo: `frutas = ['maçã', 'banana', 'laranja']`

Também é possível criar listas vazias: `lista_vazia = []`

Criando uma lista com entrada do usuário

```
frutas = []
```

```
for i in range(3):
```

```
    fruta = input(f"Digite o nome da {i+1}ª fruta: ")
```

```
    frutas.append(fruta)
```

```
print("Lista de frutas:", frutas)
```

Saída

Digite o nome da 1ª fruta: maçã

Digite o nome da 2ª fruta: banana

Digite o nome da 3ª fruta: uva

Lista de frutas: ['maçã', 'banana', 'uva']

Acessando Elementos

Índices começam em 0.

``frutas[0]`` retorna o primeiro item.

``frutas[-1]`` retorna o último item.

código

```
frutas = ['maçã', 'banana', 'laranja']
```

```
print("Primeira fruta:", frutas[0])
```

```
print("Última fruta:", frutas[-1])
```



Saída esperada:

Primeira fruta: maçã

Última fruta: laranja

Modificando uma Lista

Você pode alterar o valor de um item usando o índice.

Exemplo: `frutas[1] = 'uva'`



Código de exemplo:



Saída esperada:

```
python
frutas = ['maçã', 'banana', 'laranja']
frutas[1] = 'uva'
print("Lista modificada:", frutas)
.
```

Lista modificada: ['maçã', 'uva', 'laranja']

Adicionando Elementos

 Explicação:

`append()` adiciona no final.


`insert()` adiciona em uma posição específica.

 **Saída esperada:**


Lista atualizada: ['maçã', 'abacaxi', 'banana', 'laranja']

 Código de exemplo:

```
python
frutas = ['maçã', 'banana']
frutas.append('laranja')
frutas.insert(1, 'abacaxi')
print("Lista atualizada:", frutas)
```

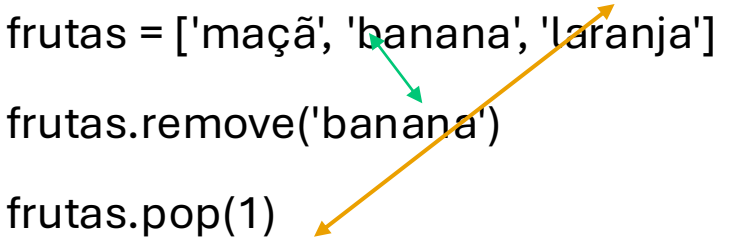



Removendo Elementos

-  Explicação:
- `remove()` elimina um valor específico.
- `pop()` remove pelo índice.

 Código de exemplo:

```
python
frutas = ['maçã', 'banana', 'laranja']
frutas.remove('banana')
frutas.pop(1)
print("Lista após remoção:", frutas)
```



 Saída esperada:

Lista após remoção: ['maçã']

Copiando Listas

 Explicação:

cópia a referência.

Use `copy()` ou fatiamento (`[:]`) para cópia real.

•  Saída esperada:

python

```
lista1 = ['a', 'b', 'c']
```

```
lista2 = lista1.copy()
```

```
lista2.append('d')
```

```
print("Lista 1:", lista1)
```

```
print("Lista 2:", lista2)
```

• Lista 1: ['a', 'b', 'c']

• Lista 2: ['a', 'b', 'c', 'd']



.

Percorrendo Listas

Use `for` para percorrer todos os elementos.

Ou use `range(len(lista))` com índices.

exemplo:

```
nomes = ['Ana', 'Bruno', 'Carlos']
```

```
print("Com for direto:")
```

```
for nome in nomes:
```

```
    print(nome)
```

```
print("Com índices:")
```

```
for i in range(len(nomes)):
```

```
    print(nomes[i])
```

Saída

Com for direto:

Ana

Bruno

Carlos

Com índices:

Ana

Bruno

Carlos

Métodos Úteis

Explicação:

``len()`` retorna o tamanho

``sort()`` ordena

``reverse()`` inverte

``in`` verifica se existe

```
print("9 está na lista?", 9 in  
      numeros)
```



Saída esperada:

```
numeros = [5, 2, 9, 1]
```

```
print("Tamanho:", len(numeros))
```

```
numeros.sort()
```

```
print("Ordenado:", numeros)
```

```
numeros.reverse()
```

```
print("Invertido:", numeros)
```

Tamanho: 4

Ordenado: [1, 2, 5, 9]

Invertido: [9, 5, 2, 1]

9 está na lista? True

-
- listas são versáteis e fundamentais em Python.
 - Podem ser criadas, modificadas, copiadas e percorridas facilmente.
 - Pratique bastante para se familiarizar com suas funcionalidades.