

# Lists em Python

## Índice

[Access List Items \(Acessar Itens da Lista\)](#)

[Change List Items \(Alterar Itens da Lista\)](#)

[Add List Items \(Adicionar Itens à Lista\)](#)

[Remove List Items \(Remover Itens da Lista\)](#)

[Loop Lists \(Iterar Listas\)](#)

[List Comprehension \(Compreensão de Listas\)](#)

[Sort Lists \(Ordenar Listas\)](#)

[Copy Lists \(Copiar Listas\)](#)

[Join Lists \(Unir Listas\)](#)

[List Methods \(Métodos de Listas\)](#)

## Explicações Adicionais

- **Access List Items** Acessa itens individuais da lista usando índices.
- **Change List Items** Modifica itens existentes em uma lista.
- **Add List Items** Adiciona novos itens à lista.
- **Remove List Items** Remove itens específicos de uma lista.
- **Loop Lists** Percorre todos os itens de uma lista usando loops.
- **List Comprehension** Cria novas listas aplicando uma expressão a cada item de uma lista existente.
- **Sort Lists** Ordena os itens da lista em ordem crescente ou decrescente.
- **Copy Lists** Cria uma cópia da lista.
- **Join Lists** Une duas ou mais listas em uma única lista.
- **List Methods** Utiliza métodos incorporados para manipular e processar listas.

## Access List Items (Acessar Itens da Lista)

Você pode acessar itens individuais de uma lista usando índices. O índice começa em 0.

```
# Código
my_list = ['apple', 'banana', 'cherry']
item = my_list[1]    # Acessa 'banana'

print(item)

# Saída
# banana
```

## Change List Items (Alterar Itens da Lista)

É possível alterar itens existentes em uma lista acessando o índice do item e atribuindo um novo valor.

```
# Código
my_list = ['apple', 'banana', 'cherry']
my_list[1] = 'blueberry' # Altera 'banana' para 'blueberry'

print(my_list)

# Saída
# ['apple', 'blueberry', 'cherry']
```

## Add List Items (Adicionar Itens à Lista)

Você pode adicionar itens a uma lista usando os métodos `append()` ou `extend()`.

```
# Código
my_list = ['apple', 'banana']
my_list.append('cherry') # Adiciona 'cherry' ao final da lista
my_list.extend(['date', 'elderberry']) # Adiciona vários itens ao final da lista

print(my_list)

# Saída
# ['apple', 'banana', 'cherry', 'date', 'elderberry']
```

## Remove List Items (Remover Itens da Lista)

Para remover itens da lista, você pode usar os métodos `remove()` ou `pop()`.

```
# Código
my_list = ['apple', 'banana', 'cherry']
my_list.remove('banana') # Remove 'banana'
popped_item = my_list.pop() # Remove o último item e retorna 'cherry'

print(my_list)
print(popped_item)

# Saída
# ['apple']
# cherry
```

## Loop Lists (Iterar Listas)

Você pode iterar sobre todos os itens de uma lista usando um loop `for`.

```
# Código
my_list = ['apple', 'banana', 'cherry']

for item in my_list:
    print(item)

# Saída
# apple
# banana
# cherry
```

## List Comprehension (Compreensão de Listas)

A compreensão de listas permite criar novas listas aplicando uma expressão a cada item de uma lista existente.

```
# Código
numbers = [1, 2, 3, 4, 5]
squared_numbers = [x ** 2 for x in numbers]

print(squared_numbers)

# Saída
# [1, 4, 9, 16, 25]
```

## Sort Lists (Ordenar Listas)

Você pode ordenar uma lista usando o método `sort()` para ordenar in-place ou `sorted()` para criar uma nova lista ordenada.

```
# Código
my_list = [3, 1, 4, 1, 5, 9]
my_list.sort()    # Ordena a lista em ordem crescente
sorted_list = sorted(my_list, reverse=True)    # Cria uma nova lista ordenada em ordem decrescente

print(my_list)
print(sorted_list)

# Saída
# [1, 1, 3, 4, 5, 9]
# [9, 5, 4, 3, 1, 1]
```

## Copy Lists (Copiar Listas)

Para copiar uma lista, você pode usar a notação de fatiamento ou o método `copy()`.

```
# Código
original_list = ['apple', 'banana', 'cherry']
copied_list = original_list.copy()    # Cria uma cópia rasa da lista
# Alternativamente: copied_list = original_list[:]

print(copied_list)

# Saída
# ['apple', 'banana', 'cherry']
```

## Join Lists (Unir Listas)

Você pode unir duas ou mais listas usando o operador `+`.

```
# Código
list1 = ['apple', 'banana']
list2 = ['cherry', 'date']
joined_list = list1 + list2

print(joined_list)

# Saída
# ['apple', 'banana', 'cherry', 'date']
```

## List Methods (Métodos de Listas)

Listas em Python possuem diversos métodos úteis, como `append()`, `remove()`, `pop()`, entre outros.

```
# Código
my_list = ['apple', 'banana', 'cherry']
my_list.append('date')    # Adiciona 'date'
print(my_list.count('banana'))    # Conta quantas vezes 'banana' aparece

# Saída
# ['apple', 'banana', 'cherry', 'date']
# 1
```