

Python - listas de classificação

O método `sort()` da classe `list` reorganiza os itens em ordem crescente ou decrescente com o uso do mecanismo de ordenação lexicográfica. A ordenação está in-loco, no sentido de que o rearranjo ocorre no mesmo objeto de lista e não retorna um novo objeto.

Sintaxe

```
list1.sort(key, reverse)
```

Parâmetros

- **Key** - A função aplicada a cada item da lista. O valor de retorno é usado para realizar a classificação. Opcional
- **inverter** - Valor booleano. Se definido como `True`, a classificação ocorrerá em ordem decrescente. Opcional

Valor de retorno

Este método retorna Nenhum.

Exemplo 1

Agora vamos dar uma olhada em alguns exemplos para entender como podemos classificar listas em Python -

```
list1 = ['physics', 'Biology', 'chemistry', 'maths']
print ("list before sort", list1)
list1.sort()
print ("list after sort : ", list1)

print ("Descending sort")

list2 = [10,16, 9, 24, 5]
print ("list before sort", list2)
list2.sort()
print ("list after sort : ", list2)
```

Ele produzirá a seguinte **saída** -



```
list before sort ['physics', 'Biology', 'chemistry', 'maths']  
list after sort: ['Biology', 'chemistry', 'maths', 'physics']  
Descending sort  
list before sort [10, 16, 9, 24, 5]  
list after sort : [5, 9, 10, 16, 24]
```

Exemplo 2

Neste exemplo, o método `str.lower()` é usado como parâmetro chave no método `sort()`.

```
list1 = ['Physics', 'biology', 'Biomechanics', 'psychology']  
print ("list before sort", list1)  
list1.sort(key=str.lower)  
print ("list after sort : ", list1)
```

Ele produzirá a seguinte **saída** -

```
list before sort ['Physics', 'biology', 'Biomechanics', 'psychology']  
list after sort : ['biology', 'Biomechanics', 'Physics', 'psychology']
```

Exemplo 3

Vamos usar uma função definida pelo usuário como parâmetro chave no método `sort()`. A `myfunction()` usa o operador `%` para retornar o restante, com base no qual a classificação é feita.

```
def myfunction(x):  
    return x%10  
list1 = [17, 23, 46, 51, 90]  
print ("list before sort", list1)  
list1.sort(key=myfunction)  
print ("list after sort : ", list1)
```

Ele produzirá a seguinte **saída** -

```
list before sort [17, 23, 46, 51, 90]  
list after sort: [90, 51, 23, 46, 17]
```