

## Python XI: Встроенные Сортировки

В Python существует два метода сортировки: `sort()` и `sorted()`. Они оба выполняют сортировку элементов списка, но различаются в том, как они работают и как они изменяют исходный список.

Метод `sort()` является методом списка и изменяет порядок элементов в списке на месте. Он не создает новый список, а изменяет текущий список. Метод `sorted()` является функцией, которая возвращает новый список, содержащий отсортированные элементы из исходного списка.

Пример использования метода `sort()`:

```
numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
numbers.sort()
print(numbers) # [1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]
```

Пример использования функции `sorted()`:

```
numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
sorted_numbers = sorted(numbers)
print(sorted_numbers) # [1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]
```

Кроме того, функция `sorted()` может принимать дополнительный параметр `reverse`, который указывает направление сортировки (по умолчанию `reverse=False`, т.е. сортировка по возрастанию):

```
numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
sorted_numbers_desc = sorted(numbers, reverse=True)
print(sorted_numbers_desc) # [9, 6, 5, 5, 5, 4, 3, 3, 2, 1, 1]
```

Метод `sort()` также может принимать параметр `reverse`, но он не возвращает отсортированный список, а изменяет исходный список:

```
numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
numbers.sort(reverse=True)
print(numbers) # [9, 6, 5, 5, 5, 4, 3, 3, 2, 1, 1]
```

Существует возможность использовать функцию `sorted()` и метод `sort()` с параметром `key`, который позволяет указать функцию, по которой будет производиться сортировка. Это позволяет сортировать объекты не только по значениям, но и по другим свойствам. То есть так же можно использовать лямбда функции и свои функции. Важно запомнить что передавать функцию надо без `()`, так как если указать круглые скобки, вы ее вызовете.

Пример сортировки списка строк по длине:

```
fruits = ['apple', 'banana', 'kiwi', 'orange', 'pear']
sorted_fruits = sorted(fruits, key=len)
print(sorted_fruits) # ['kiwi', 'pear', 'apple', 'banana',
'orange']
```

Важно запомнить, что функция `sort()` выполняет сортировку "на месте" (in-place), то есть изменяет исходный список, а функция `sorted()` создает новый отсортированный список, не изменяя исходный.

```
# Использование sort()
numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
numbers.sort()
print(numbers) # [1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]

# Использование sorted()
numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]
sorted_numbers = sorted(numbers)
print(sorted_numbers) # [1, 1, 2, 3, 3, 4, 5, 5, 5, 6, 9]
print(numbers) # [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5] - исходный
список не изменился
```