



← К списку

Методы списков Python

list.append(x)

x: Элемент, который требуется добавить в список.

```
my_list = []  
my_list.append(1)  
my_list # [1]  
my_list.append(3)  
my_list # [1, 3]
```

Для добавления множества элементов используйте **extend()**. Внимание: Данный метод модифицирует исходный объект на месте, возвращая при этом **None**.

list.clear(i)

Удаляет из списка все имеющиеся в нём значения.

```
my_list = [1, 2, 3]  
my_list.clear() # None  
my_list # []
```

Действие метода эквивалентно выражению **del my_list[:]**.

list.copy()

Возвращает копию списка. Внимание: Возвращаемая копия является поверхностной (без рекурсивного копирования вложенных элементов).

```
my_list = [1, 2, 3]
my_list_copy = my_list.copy() # [1, 2, 3]
my_list # []
```

Действие метода эквивалентно выражению `my_list[:]`.

list.count(x)

```
random_list = [4, 1, 5, 4, 10, 4]
random_list.count(4) # 3
```

Метод count считает количество значений x в списке.

list.extend(iterable)

Дополняет список элементами из указанного объекта.

- **it**: Объект, поддерживающий итерирование, элементами которого требуется дополнить список.

```
my_list = []
my_list.extend([1, 2, 3]) # None
my_list # [1, 2, 3]
my_list.extend('add') # None
my_list # [1, 2, 3, 'a', 'd', 'd']
```

Для добавления единственного элемента используйте **append()**.

Внимание: Данный метод модифицирует исходный объект на месте, возвращая при этом **None**.

list.index(x, start, end)

Метод возвращает положение первого индекса, со значением `x`. Также можно указать границы поиска **start** и **end**.

list.insert(i, x)

Вставляет указанный элемент перед указанным индексом

- **i**: Позиция (индекс), перед которой требуется поместить элемент. Нумерация ведётся с нуля. Поддерживается отрицательная индексация.
- **x**: Элемент, который требуется поместить в список.

Метод модифицирует

```
my_list = [1, 3]

my_list.insert(1, 2)
my_list # [1, 2, 3]

my_list.insert(-1, 4)
my_list # [1, 2, 4, 3]
```

Внимание: Данный метод модифицирует исходный объект на месте, возвращая при этом **None**.

list.pop(i)

Возвращает элемент на указанной позиции, удаляя его из списка.

- **i=None**: Позиция искомого элемента в списке (целое число). Если не указана, считается что имеется в виду последний элемент списка. Отрицательные числа поддерживаются.

```
my_list = [1, 2, 3, 4, 5]
last = my_list.pop() # 5
my_list # [1, 2, 3, 4]
second = my_list.pop(-3) # 2
my_list # [1, 3, 4]
first = my_list.pop(0) # 1
```

```
my_list # [3, 4]
```

Чтобы удалить элемент из списка не возвращая его, воспользуйтесь `list.remove()`.

`list.remove(x)`

Удаляет из списка указанный элемент.

- **x**: Элемент, который требуется удалить из списка. Если элемент отсутствует в списке, возбуждается **ValueError**. Удаляется только первый обнаруженный в списке элемент, значение которого совпадает со значением переданного в метод.

```
my_list = [1, 3]

my_list.remove(1)
my_list # [3]

my_list.remove(4) # ValueError
```

Внимание: Данный метод модифицирует исходный объект на месте, возвращая при этом **None**.

`list.reverse()`

Перестраивает элементы списка в обратном порядке.

```
my_list = [1, 'two', 'a', 4]
my_list.reverse() # None
my_list # [4, 'a', 'two', 1]
```

Внимание: Данный метод модифицирует исходный объект на месте, возвращая при этом **None**.

list.sort(key=None, reverse=False)

Сортирует элементы списка на месте.

- **key=None**: Функция, принимающая аргументом элемент, используемая для получения из этого элемента значения для сравнения его с другими.
- **reverse=False**: Флаг, указывающий следует ли производить сортировку в обратном порядке.

```
my_list = [1, 'two', 'a', 4, 'a']  
# Попытка упорядочить/сравнить несравнимые типы вызовет исключение  
my_list.sort() # TypeError: unorderable types: str() <= int()  
  
# Отсортируем «вручную», так чтобы 'a' были в конце.  
my_list.sort(key=lambda val: val == 'a') # None  
  
# Фактически мы отсортировали в соответствии  
# с маской [False, False, False, True, True]  
my_list # ['two', 4, 1, 'a', 'a']
```

Внимание: Данный метод модифицирует исходный объект на месте, возвращая при этом **None**.



О курсе

Самоучитель

Справочник

Программа курса

Примеры задач

Информация

Контакты

Оплата

Договор

