

Операции с изменяемыми последовательностями в Python



new.sipuni.ru

РЕКЛАМА

ChatGPT повысит уровень ваших продаж!

Анализ качества продаж • Нейросеть • Автоматизация рутины • Эффективные продажи

[Узнать больше](#)

[Справочник по языку Python3.](#) / Операции с изменяемыми последовательностями в Python

Здесь перечислены [операции с изменяемыми последовательностями](#) с их описанием и примерами использования в коде.

Напоминаем: Изменяемые последовательности поддерживают все [общие операции с последовательностями](#).

[Изменение/замена элемента списка по индексу в Python](#)

Операция `sequence[i] = x` позволяет заменить значение элемента последовательности `sequence` по его индексу `i` новым значением `x`. Эта операция поддерживается изменяемыми типами последовательностей.

[Изменение части списка операцией среза в Python](#)

Операция `sequence[i:j] = iterable` позволяет заменить элементы последовательности `sequence` по указанному срезу `[i:j]` содержимым итерации `iterable`. Эта операция поддерживается изменяемыми типами последовательностей.

[Изменение списка срезом с заданным шагом в Python](#)

Операция `sequence[i:j:k] = iterable` позволяет заменить элементы последовательности `sequence` по срезу от индекса `i` до индекса `j` с шагом `k` содержимым итерации `iterable`. Итерация должна иметь ту же длину, что и срез последовательности.

[Удаление части списка операцией среза в Python](#)

Операция `del sequence[i:j]` позволяет удалить элементы в последовательности `sequence` по определенному срезу `[i:j]`. Эта операция поддерживается изменяемыми типами последовательностей.

[Удаление части списка по срезу с заданным шагом в Python](#)

Операция `del sequence[i:j:k]` позволяет удалить элементы в последовательности `sequence` по срезу от индекса `i` до индекса `j` с шагом `k`. Эта операция поддерживается изменяемыми типами последовательностей.

[Метод `list.append\(\)` в Python, добавляет значение в конец списка](#)

Операция `sequence.append(x)` позволяет добавить в конец последовательности `sequence` новый объект `x`. Эта операция поддерживается изменяемыми типами последовательностей.

[Метод `list.clear\(\)` в Python, очищает список](#)

Операция `sequence.clear()` позволяет удалить все элементы из последовательности `sequence`. Эта операция поддерживается изменяемыми типами последовательностей.

РЕКЛАМА

Стань частью
компании YADRO

УЗНАТЬ

oneweekoffer.yadro.com

Ищем ведущих программистов
в команду YADRO.

→

РЕКЛАМА

Забирай бесплатно 2 урока по ме-
тодам программирования!

→

[Метод `list.copy\(\)` в Python, копия списка](#)

Операция `sequence.copy()` создает мелкую копию последовательности `sequence`. Операция создания мелкой копии эквивалентна выполнению выражения `sequence[:]`. Эта операция поддерживается изменяемыми типами последовательностей.

[Метод `list.extend\(\)` в Python, расширяет список другой последовательностью](#)

Метод `sequence.extend(iterable)` позволяет расширить последовательность `sequence` содержимым итерации `iterable`. Альтернативный способ расширения `sequence += iterable`. Эта операция поддерживается изменяемыми типами последовательностей.

[Расширение списка его содержимым в Python](#)

Метод позволяет расширить последовательность `sequence` ее же содержимым в `n` раз. Нулевые и отрицательные значения `n` очищают последовательность. Эта операция поддерживается изменяемыми типами последовательностей.

[Метод `list.insert\(\)` в Python, вставить элемент по индексу](#)

Метод `sequence.insert(i, x)` вставляет объект `x` в последовательность `sequence` по индексу `i`. Эквивалентна выполнению `sequence[i:i] = [x]`. Эта операция поддерживается изменяемыми типами последовательностей.

[Метод `list.pop\(\)` в Python, получение с удалением элемент списка](#)

Метод `sequence.pop(i)` возвращает значение элемента с индексом `i`, удаляя его из последовательности `sequence`. Необязательный индекс `i` по умолчанию равен `-1`. Эта операция поддерживается изменяемыми типами последовательностей.

[Метод `list.remove\(\)` в Python, удаление элемента списка по значению](#)

Метод `sequence.remove(x)` производит удаление первого элемента, значение которого равно `x` из последовательности `sequence`. Метод `remove()` вызывает `ValueError`, когда объект `x` не найден в последовательности `sequence`.

[Метод `list.reverse\(\)` в Python, разворачивает элементы списка](#)

Метод `sequence.reverse()` переворачивает элементы последовательности `sequence`. Изменяет последовательность не копируя для экономии памяти при реверсировании большой последовательности.

[Удаление элемента списка по индексу в Python](#)

Выражение `del sequence[i]` удаляет элемент последовательности `sequence` по индексу `i`. Выражение вызывает `IndexError`, когда значение индекса `i` находится за пределами длины последовательности.

[Удаление дубликатов из списка Python с сохранением очередности](#)

Удалить дубликаты значений из списка Python можно двумя способами.

ХОЧУ ПОМОЧЬ
ПРОЕКТУ

РЕКЛАМА

Бесплатный курс



Разработка игр на Python

Очно в вашем городе для учащихся

code.top-academy.ru

Бесплатный курс школьникам: Разработка 2D-игр на Python

Разработка игр на Python. Построение сайтов. Обучим любой из профессий Бесплатно!

Блог

Новости

Вакансии

Узнать больше

[DOCS-Python.ru](#)[™], 2023 г.

(**Внимание!** При копировании материала ссылка на источник обязательна)

[@docs_python_ru](#)