

Метод list.copy() в Python, копия списка

[Справочник по языку Python3](#) / [Операции с изменяемыми последовательностями в Python](#) / Метод list.copy() в Python, копия списка

Создание неглубокой копии списка

Синтаксис:

```
sequence.copy()
```

Параметры:

- sequence - изменяемая последовательность, [list](#) или [bytearray](#),

Результат:

- новый объект, копия последовательности sequence.

Описание:

Операция позволяет [создавать мелкую копию последовательности](#). Метод sequence.copy() создает мелкую копию последовательности sequence.

Операция копирования последовательности sequence эквивалентна выполнению выражения sequence[:].

Метод sequence.copy() был включен для обеспечения согласованности с интерфейсами изменяемых контейнеров, которые не поддерживают срезы, такие как [dict](#) и [set](#). copy() не входит в состав ABC collections.abc.MutableSequence, но большинство классов изменяемых последовательностей предоставляют его.

Эта операция поддерживаются изменяемыми [типами последовательностей](#).

Смотрите так же [модуль copy](#), что бы сделать глубокую копию списка, а так же различия между глубокой и неглубокой копией.

Примеры копирования списка.

- [Общие примеры копирования списка;](#)
- [Копирование списка срезом;](#)

Общие примеры.

```
>>> x = [3, 6, 9, 12, 15, 18]
>>> y = x.copy()
>>> y[2] = 20000
>>> x      # не изменился
[3, 6, 9, 12, 15, 18]
>>> y      # Измененная копия x
[3, 6, 20000, 12, 15, 18]
```

Альтернативная операция создания копии списка.

```
# Альтернативная операция создания копии
>>> z = x[:]
>>> z[-1] = 15000
>>> x      # не изменился
[3, 6, 9, 12, 15, 18]
>>> z      # Измененная копия x
[3, 6, 9, 12, 15, 15000]
```

Содержание раздела:

- [ОБЗОРНАЯ СТРАНИЦА РАЗДЕЛА](#)
- [Изменение/замена элемента списка по индексу.](#)

- [Изменение части списка операцией среза](#)
- [Изменение списка срезом с заданным шагом](#)
- [Удаление части списка операцией среза](#)
- [Удаление части списка по срезу с заданным шагом](#)
- [Метод `list.append\(\)`, добавляет значение в конец списка](#)
- [Метод `list.clear\(\)`, очищает список](#)
- [Метод `list.copy\(\)`, копия списка](#)
- [Метод `list.extend\(\)`, расширяет список другой последовательностью](#)
- [Расширение списка его содержимым](#)
- [Метод `list.insert\(\)`, вставить элемент по индексу](#)
- [Метод `list.pop\(\)`, получение с удалением элемент списка](#)
- [Метод `list.remove\(\)`, удаление элемента списка по значению](#)
- [Метод `list.reverse\(\)`, разворачивает элементы списка](#)
- [Удаление элемента списка по индексу](#)
- [Удаление дубликатов из списка Python с сохранением очередности](#)

ХОЧУ ПОМОЧЬ
ПРОЕКТУ

[DOCS-Python.ru](#)™, 2024 г.

(Внимание! При копировании материала ссылка на источник обязательна)

[@docs_python_ru](#)