



Урок 23. Python: Создание списка или массива

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 23 из 33 · 45 минут · Модуль «Массивы и списки»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

В журнале мест нужен список нулей и одна отметка в середине. Почему присваивание другому имени не создаёт независимую копию журнала?

Результат урока: Создать одномерный список, безопасно обратиться к элементу и отличить копирование от второго имени того же списка.

01. Проблемная ситуация

0–5 минут. Нарисуйте четыре ячейки с индексами 0–3. Поставьте 1 в ячейку с индексом 4//2. Что должно происходить, если ячеек вообще нет?

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Размер, индекс и изменяемый объект

Список Python — упорядоченная изменяемая последовательность. В этом уроке он содержит только целые числа. `[0]*n` создаёт `n` нулей; допустимые неотрицательные индексы идут от 0 до `n-1`. Для `n=0` список пуст, и запись `a[0]=1` невозможна.

Целочисленное `n//2` выбирает один средний индекс: при чётном размере — правый из двух центральных. Условие `if n>0` защищает запись. `a[i]=x` меняет существующий элемент, а `append(x)` добавляет новый и увеличивает длину.

После `b=a` оба имени указывают на один список. После `b=a.copy()` внешние списки различны; для нашего одномерного списка целых этого достаточно для независимого изменения элементов. Копия поверхностная: правило не распространяется автоматически на вложенные списки.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

При `n=4` создаём `[0,0,0,0]`, индекс `4//2=2`, получаем `[0,0,1,0]`. При `n=1` индекс 0, результат `[1]`. При `n=0` возвращаем `[]`, не выполняя запись.

```
def make_slots(n):
    if n < 0 or n > 8:
        return None
    a = [0] * n
    if n > 0:
        a[n // 2] = 1
    return a
```

```
for n in [0, 1, 2, 4, 5, 8, 9, -1]:
    print(n, make_slots(n))
```

n = 4	четыре независимые позиции
Индексы	0 1 2 3
До записи	0 0 0 0
После a[2]=1	0 0 1 0

Одномерный список из четырёх целых: изменение одного элемента не меняет длину.



03. Практика 1. Проверяем материал

Середина списка и пустой случай

14–25 минут. Составьте восемь прогнозов. Для каждого допустимого n запишите длину и индекс единицы; для пустого списка такого индекса нет.

n	<code>make_slots(n)</code>
0; 1; 2	<code>[]; [1]; [0,1]</code>
4; 5	<code>[0,0,1,0]; [0,0,1,0,0]</code>
8	<code>[0,0,0,0,1,0,0,0]</code>
9; -1	<code>None; None</code>

2 минуты — рисунок, 4 — запуск, 3 — временно удалите защиту $n > 0$ и проверьте $n = 0$, 2 — восстановите код. Критерии: границы 0 и 8, индекс середины, отличие `[]` от `None`, объяснение `IndexError`.

04. Практика 2. Перенос способа

Новая версия без изменения исходника

25–33 минуты. Реализуйте `changed(a,i,x)`: вернуть копию списка с заменой на позиции i ; исходный a не менять; при $i < 0$ или $i \geq \text{len}(a)$ вернуть `None`. За 2 минуты проверьте мысленно $a = [2,4,6]$, $i = 1$, $x = 9$; за 4 напишите функцию, за 2 проверьте ещё $i = 0, 2, 3, -1$ и пустой a . Критерии: копия, исходник, обе границы, длина.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты предскажите результат $a = [1,2]$; $b = a$; $b[0] = 8$. За 3 сравните с $b = a.\text{copy}()$. За 2 решите, какой вариант подходит для хранения исходной и исправленной версий журнала.

06. Индивидуальная проверка

40–45 минут. Найдите `make_slots(3)`, результат `changed([1,2],0,7)`, исходный список после вызова и объясните отказ для `changed([],0,7)`. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

[] — допустимый результат для нулевого размера, None — отказ для -1 и 9. Без защиты запись в пустой список вызывает IndexError. Не следует превращать корректный пустой результат в ошибку; восстановленный код дан выше.

Практика 2: разбор и критерии

Основной результат [2,9,6], исходник [2,4,6]. Для замен на 0 и 2 получаются [9,4,6] и [2,4,9]; для 3, -1 и пустого списка — None. Здесь отрицательные индексы специально запрещены контрактом, хотя Python обычно их поддерживает.

```
def changed(a, i, x):
    if i < 0 or i >= len(a):
        return None
    b = a.copy()
    b[i] = x
    return b

a = [2, 4, 6]
for i in [1, 0, 2, 3, -1]:
    print(i, changed(a, i, 9), a)
print(changed([], 0, 9))
```

Обсуждение: возможный вывод

В первом случае a тоже станет [8,2]; во втором сохранится [1,2]. Для независимых версий нужна копия. Разные имена не доказывают наличие разных объектов. Здесь элементы — целые, а не вложенные коллекции.

Индивидуальная проверка: ответ и критерии

[0,1,0]; [7,2]; исходный [1,2]. Пустой список не содержит позиции 0, поэтому результат None. По одному баллу за пункт.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Python 3.11+ и IDLE или редактор; запуск python lesson.py (либо python3 lesson.py). Внешние библиотеки не нужны. Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 8 класс. Предпосылки: целые числа, функции, if, индексы строк. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание. Матрица программы предназначена учителю.

Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ



07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Массивы и списки	Создать одномерный список, безопасно обратиться к элементу и отличить копирование от второго имени того же списка.	Исходник, трассировка и таблица проверок
Создание, ввод и обработка массивов; перебор элементов	Новая версия без изменения исходника	Самостоятельный алгоритм, ключ и индивидуальная проверка

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Python: структуры данных

<https://docs.python.org/3/tutorial/datastructures.html>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 18.10.2026.