**Копирование списков**

Научимся копировать списки в языке Python. Пусть у нас есть список a = [2, 3, 4]. Оказывается, что операция b = a не создаёт новый список. Такая операция делает имя b ссылкой на тот же список, на который ссылается имя a. Таким образом, a и b — это два разных имени, ссылающиеся на один список. Поэтому если мы модифицируем список a, например, добавив число 5 в конец списка: a.append(5), то модифицируется и список b.

Нам же нужно выполнить такую операцию, чтобы имя b ссылалось на список, который является копией списка a, а не на сам список a. Для этого можно использовать присваивание с использованием среза: b = a[:]. Также можно использовать генератор: b = [elem **for** elem **in** a].

Если есть необходимость создать копию двумерного списка, то возникают дополнительные сложности. Чтобы их избежать, рекомендуется выполнять копирование с использованием специальных функций из модуля copy. Подключить модуль copy можно, написав **from**copy **import**\* в начале программы.

**Создание двумерных списков**

Пусть теперь нам нужно создать двумерный список из 3 строк и 4 столбцов, заполненный нулями. Кажется, что такой двумерный список можно создать следующим образом: a = [[0] \* 4] \* 3. Но тут возникает проблема. При таком способе a[0], a[1] и a[2] являются ссылками на один и тот же список [0] \* 4. Поэтому после операции a[0][0] = 1, окажется, что элементы a[1][0] и a[2][0] тоже стали равны числу 1.

Чтобы правильно создать двумерный список, заполненный нулями и состоящий из n строк и m столбцов, необходимо, чтобы каждая строка списка создавалась заново. Есть несколько способов сделать это.

Например, можно создать список из n нулей, а затем каждый элемент этого списка заменить на список из m нулей. Во время замены список из m нулей будет создаваться заново, поэтому каждая из n строк окажется ссылкой на свой независимый список из m нулей:

a = [0] \* n

**for** i **in** range(n):

a[i] = [0] \* m

Можно создать изначально пустой список, а потом n раз добавить в конец этого списка новый элемент, который является списком из m нулей:

a = []

**for** i **in** range(n):

a.append([0] \* m)

Также для создания двумерного списка можно использовать генератор, который создает список из nn элементов, каждый из которых будет списком, состоящим из mm нулей (подробнее о генераторах списков будет рассказано в следующем модуле):

a = [[0] \* m **for** i **in** range(n)]

В каждом из этих трёх способов очередная строка создается независимо от остальных: заново конструируется список [0] \* m, а не копируются ссылки на один и тот же список.

[Вопросы-ответы по видеолекции](https://edu.sirius.online/#/qa/324_3380_122523)