

# Списки

**№ урока:** 7 **Курс:** Python Starter

**Средства обучения:** Python 3.4; интегрированная среда разработки (PyCharm 4 (рекомендуется) или Microsoft Visual Studio 2013 + Python Tools for Visual Studio)

## Обзор, цель и назначение урока

В уроке рассматриваются списки – структура данных, которая позволяет хранить несколько значений и в Python часто используется вместо массивов в других языках программирования.

## Изучив материал данного занятия, учащийся сможет:

- Использовать списки для хранения и обработки множества значений
- Получать отдельные элементы списков и строк, а также их сечения
- Работать со списками: добавлять и удалять данные, изменять существующие значения, проверять элементы на вхождение в список
- Понимать разницу между изменяемыми и неизменяемыми структурами данных

## Содержание урока

1. Понятие списка
2. Индексы, срезы списков
3. Проверка элементов на вхождение в список
4. Работа со списками (получение значений элементов, добавление и удаление значений)

## Резюме

- Массив – набор фиксированного количества однотипных элементов, расположенных в памяти непосредственно друг за другом, доступ к которым осуществляется по индексу (номеру).
- Индекс – целое число, которое используется для доступа к элементу массива.
- Размерность массива – максимальное количество его элементов, или, иначе, количество индексов, необходимое для однозначного доступа к элементу массива.
- Массив может иметь несколько размерностей и образовывать двумерные (матрицы) или многомерные массивы. В большинстве языков это реализуется при помощи массивов, состоящих из массивов.
- Список – это тип данных, представляющий собой упорядоченную последовательность определённых значений, которые могут повторяться.
- Количество элементов списка может быть произвольным.
- В Python вместо массивов, как правило, используются списки. Однако в случаях, когда производительности списков недостаточно, можно воспользоваться специальными средствами: классом `array` из стандартной библиотеки языка, в котором типы данных элементов списка фиксированы, или сторонними библиотеками (такими, как `NumPy`), которые предлагают эффективную реализацию классических массивов.
- Тип данных, который реализует списки, – `list`.
- Список в Python является изменяемой (*mutable*) структурой данных. То есть, его элементы можно изменять после создания списка.
- Список в Python является гетерогенной структурой данных. Это значит, что в нём можно хранить значения разных типов.
- Для создания списка в Python следует записать его элементы через запятую в квадратных скобках.
- Для создания пустого списка следует просто записать пустые квадратные скобки.

- Пример:

```
my_list = [1, 2, 3]
empty_list = []
```

- В Python списки можно индексировать, как массивы.
- Отсчёт индексов идёт с нуля.
- Для получения элемента по индексу следует после имени списка указать индекс в квадратных скобках. Пример:

```
my_list[2]
```

- Можно также индексировать списки с конца. В таком случае задаётся отрицательный индекс, начиная с -1.
- Можно также получить группу элементов по их индексам. Эта операция называется срезом списка (list slicing).
- Для получения среза списка в квадратных скобках указывается индекс первого элемента, затем через двоеточие записывается индекс, на единицу больший индекса последнего требуемого элемента:

```
my_list[start:end]
```

- Можно также добавить ещё одно двоеточие и шаг среза:
- ```
my_list[start:end:step]
```
- Можно не указывать начальный и/или конечный индексы. Значением по умолчанию для начального индекса является 0, для конечного – длина списка. Если не указан ни начальный, ни конечный индексы, ни шаг, то `my_list[:]` вернёт копию списка.
  - Для проверки, является ли указанное значение элементом списка, используется операция `in`.
  - Для получения количества элементов в списке используется функция `len`.
  - Функцию `len`, индексацию, получение срезов и проверку элемента на вхождение можно также применять ко строкам.
  - Для добавления нового элемента в список используется метод `append`:
- ```
my_list.append(value)
```
- Для удаления элемента списка по индексу используется оператор `del`:
- ```
del my_list[index]
```
- Для изменения элемента списка используется оператор присваивания:
- ```
my_list[index] = value
```
- Для обхода элементов списка используется цикл `for`:
- ```
for element in my_list:
    # выполняем операции с текущим элементом element
    pass
```

## Закрепление материала

- Что такое массив?
- Что такое индекс?
- Что такое список?
- Являются ли классические массивы частью языка Python? Что используется вместо них?
- Как создать список в Python?
- Какой индекс имеет первый элемент списка?
- Какой индекс имеет последний элемент списка?
- Что такое сечение списка?
- Как узнать длину списка?
- Как добавить новый элемент в список?
- Как удалить элемент из списка?
- Как обработать каждый элемент списка?

## Дополнительное задание

Задание

Создайте список, введите количество его элементов и сами значения, выведите эти значения на экран в обратном порядке.

### Самостоятельная деятельность учащегося

#### Задание 1

Создайте список и введите его значения. Найдите наибольший и наименьший элемент списка, а также сумму и среднее арифметическое его значений.

#### Задание 2

Перепишите решение последней задачи из шестого урока так, чтобы она не использовала рекурсию и не вычисляла все промежуточные количества вариантов путей множество раз (что крайне неэффективно), а сохраняла их в списке.

#### Задание 3

Простым называется число, которое делится нацело только на единицу и само себя. Число 1 не считается простым. Напишите программу, которая находит все простые числа в заданном промежутке, выводит их на экран, а затем по требованию пользователя выводит их сумму либо произведение.

### Рекомендуемые ресурсы

Документация по Python

<https://docs.python.org/3/library/stdtypes.html#list>

Статьи в Википедии о ключевых понятиях, рассмотренных на этом уроке

[https://ru.wikipedia.org/wiki/Массив\\_\(программирование\)](https://ru.wikipedia.org/wiki/Массив_(программирование))

<https://ru.wikipedia.org/wiki/Список>