

# Практическая работа № 3 от 06.03.2020

## Основы Python, часть 2

### ***Цель работы***

Продолжение изучения базового синтаксиса Python.

### ***Задачи работы***

1. Изучить итеративные типы данных Python: списки, кортежи, словари.
2. Изучать основные операции с итеративными типами данных.

### ***Перечень обеспечивающих средств***

1. ПК.
2. Учебно-методическая литература.
3. Задания для самостоятельного выполнения.

### ***Общие теоретические сведения***

#### **Итеративные типы данных Python**

В дополнение к спискам, рассмотренным в предыдущей работе, имеет смысл рассмотреть ещё два важных итеративных типа данных: кортежи и словари.

*Кортеж* (tuple) — заключенная в круглые скобки последовательность значений любых типов, разделенных запятыми.

Примеры кортежей:

```
(1, 'текст', 2, -0.19)
((0, 1), 2, 'три')
```

Основное отличие кортежей и списков: кортежи нельзя изменять и дополнять, списки — можно.

Список можно преобразовать в кортеж с помощью метода tuple() и, наоборот, кортеж в список с помощью метода list().

*Словарь* (dict) — заключенная в фигурные скобки последовательность пар ключ-значение. Ключ и значение разделяются двоеточием, пары между собой — запятыми.

Примеры словарей:

```
{1 : 1, 2 : 'два', 'три' : 3, 'четыре' : 'четыре'}
{'Арсенал' : 'Тула', 'Зенит' : 'Санкт-Петербург', 'Спартак' : 'Москва'}
```

Ключи в словаре не могут повторяться, значения — могут.

Ключами могут выступать только неизменяемые значения, например, числа, строки, кортежи. Значения могут быть любыми, в том числе другими словарями.

## Основные операции со списками, кортежами и словарями

Рассмотрим основные операции на следующих примерах:

```
my_list = [1, 2, 4, 8, 16]
my_tuple = (10, 'десять', 'X')
my_dict = {3 : 'три', 2 : 'два', 1 : 'один'}
```

### Индексация:

Список	Кортеж	Словарь
Можно извлечь элемент по его номеру, начиная с нуля: <code>my_list[1]</code> равно 2. Либо по номеру с конца: <code>my_list[-1]</code> равно 16.	Можно извлечь элемент по его номеру, начиная с нуля: <code>my_tuple[1]</code> равно 'десять'. Либо по номеру с конца: <code>my_tuple[-1]</code> равно 'X'.	Можно извлечь значение по ключу: <code>my_dict[1]</code> равно 'один'.

### Добавление элемента:

Список	Кортеж	Словарь
Можно добавить новый элемент в конец списка: <code>my_list.append(32)</code> Теперь <code>my_list</code> равен <code>[1, 2, 4, 8, 16, 32]</code>  Можно добавить новый элемент в любую позицию списка: <code>my_list.insert(1, 32)</code> После этого <code>my_list</code> будет равен <code>[1, 32, 2, 4, 8, 16]</code>	Нельзя изменить.	Можно добавить новый элемент, указав пару ключ-значение: <code>my_dict[0] = 'ноль'</code> После этого <code>my_dict</code> будет равен <code>{3 : 'три', 2 : 'два', 1 : 'один', 0 : 'ноль'}</code>

### Удаление элемента:

Список	Кортеж	Словарь
Метод <code>pop()</code> возвращает элемент с указанным индексом, удаляя его из списка. Если индекс не указан, то возвращается и удаляется последний элемент. Например, <code>my_list.pop(2)</code>	Нельзя изменить.	Метод <code>pop()</code> возвращает значение с указанным ключом, удаляя пару ключ-значение из словаря. Например, <code>my_dict.pop(3)</code> вернет 'три', а <code>my_dict</code> после этого будет равен <code>{2 : 'два', 1 : 'один'}</code> .

<p>вернет 4, а my_list после этого будет равен [1, 2, 8, 16].</p> <p>Метод remove() удаляет из списка указанный элемент. Например, после выполнения my_list.remove(8) my_list будет равен [1, 2, 4, 16].</p>		
--	--	--

### *Поиск элемента:*

Список	Кортеж	Словарь
<p>Оператор in позволяет проверить, содержится ли элемент в списке. Например, 2 in my_list вернёт True (истина), 5 in my_list вернёт False (ложь).</p>	<p>Оператор in позволяет проверить, содержится ли элемент в кортеже. Например, 10 in my_tuple вернёт True (истина), 'XI' in my_tuple вернёт False (ложь).</p>	<p>Оператор in позволяет проверить, содержится ли ключ в словаре. Например, 3 in my_dict вернёт True (истина), 'три' in my_dict вернёт False (ложь).</p>

### *Сортировка:*

Список	Кортеж	Словарь
<p>Метод sort() сортирует содержимое списка. Пример вызова: my_list.sort().</p> <p>Метод sorted() возвращает новый отсортированный список. Пример вызова: sorted(my_list).</p>	<p>Метод sorted() возвращает новый отсортированный кортеж. Пример вызова: sorted(my_tuple).</p>	<p>Метод sorted() возвращает новый отсортированный список ключей словаря. Пример вызова: sorted(my_dict).</p>

### *Слияние:*

Список	Кортеж	Словарь
<p>Метод extend() добавляет к первому списку элементы второго списка. Например, для l1 = [0,1] и l2 = [2, 3] после выполнения l1.extend(l2) l1 равен [0,1,2,3].</p> <p>Оператор + создает новый список, состоящий из элементов первого и второго списков. Например, для l1 = [0,1] и</p>	<p>Оператор + создает новый кортеж, состоящий из элементов первого и второго кортежей. Например, для t1 = (0, 1) и t2 = (2, 3) t1+t2 равно (0,1,2,3). Сами t1 и t2 при этом не изменяются.</p>	<p>Метод update() добавляет переданные пары ключ-значение к словарю. Например, для d1 = {0:0, 1:1} и d2 = {2:2, 3:3} после выполнения d1.update(d2) d1 будет равен {0:0, 1:1, 2:2, 3:3}.</p>

l2 = [2, 3] l1+l2 равно [0,1,2,3]. Сами l1 и l2 при этом не изменяются.		
---	--	--

### Получение длины:

Список	Кортеж	Словарь
Метод len возвращает количество элементов в списке. Например, len(my_list) вернет 5.	Метод len возвращает количество элементов в кортеже. Например, len(my_tuple) вернет 3.	Метод len возвращает количество элементов в словаре. Например, len(my_dict) вернет 3.

### Копирование:

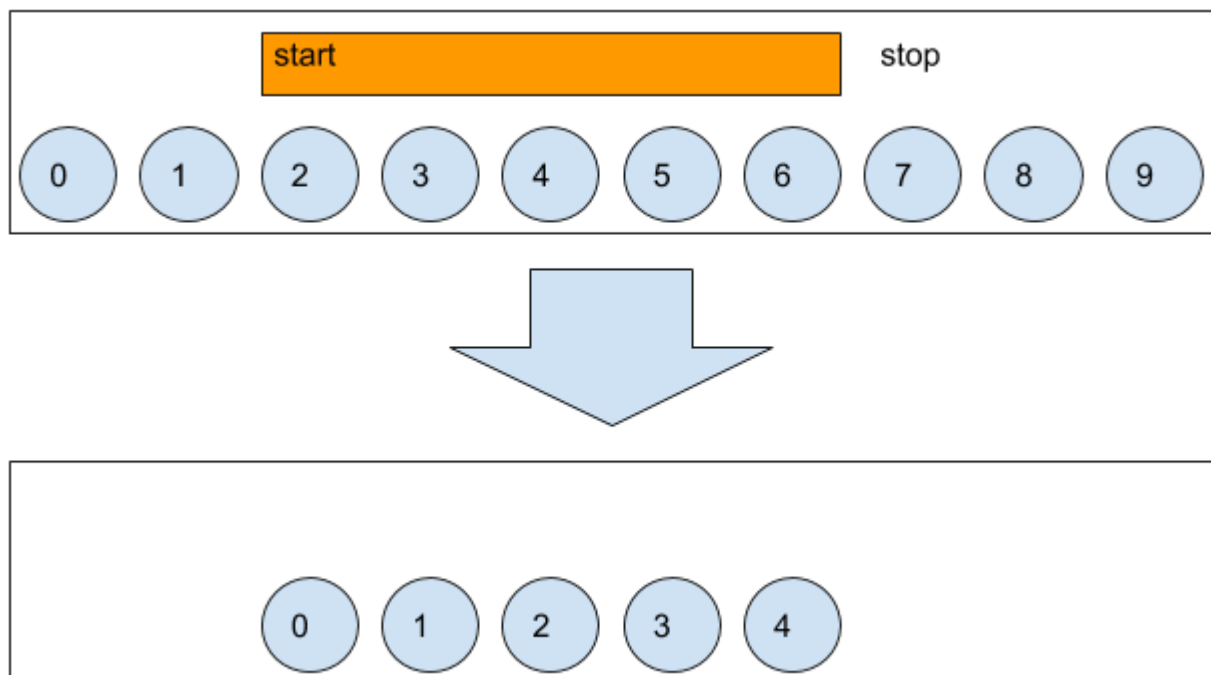
Список	Кортеж	Словарь
Метод copy() создает копию списка. Пример вызова: my_list.copy().	Не имеет особого смысла, т. к. кортеж неизменяем. Два кортежа с одними и теми же значениями — по сути один и тот же кортеж.	Метод copy() создает копию словаря. Пример вызова: my_dict.copy().

### Слайсинг:

Слайсинг — это более мощный аналог индексирования. Слайсинг применим к спискам, кортежам и строкам.

Нотация слайсинга аналогична нотации функции range():

start : stop : step – начало : остановка : шаг.



Start (начало):

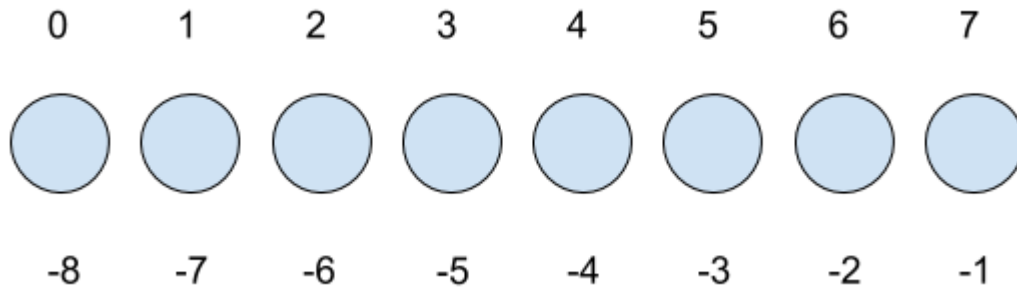
Если не указать, то считается равным 0.

Если указать отрицательное число, то элемент берется с конца списка или кортежа.

Stop (остановка):

Если не указать, то считается равным длине списка или кортежа.

Если указать отрицательное число, то элемент берется с конца списка или кортежа.



Step (шаг):

Число, на которое увеличивается индекс при проходе по списку или кортежу.

Если не указать, то считается равным 1.

Если указать отрицательное число, то отсчёт идет от конца к началу.

Слайсы можно использовать в большинстве методов или операторов, требующих указания индекса.

## **Задание**

### **Часть 1**

- Сделайте форк репозитория

<https://github.com/mosalov/EmptyJupyterNotebookForBinder>

- Откройте сайт Binder: <https://mybinder.org/>.
- В поле «GitHub repository name or URL» укажите ссылку на свой репозиторий. Нажмите кнопку «launch», дождитесь открытия репозитория.
- Откройте (кликните) файл «empty\_notebook.ipynb».
- Напишите код, который создаст и выведет список квадратов натуральных чисел от 1 до 1000.
- Сохраните файл Jupyter notebook с названием «Задание 3\_1.ipynb» и загрузите его в созданный репозиторий.

### **Часть 2**

- Вернитесь к файлу «empty\_notebook.ipynb», открытому в Binder.
- Создайте кортеж из натуральных чисел от 1 до 20.
- С помощью слайсинга создайте кортеж, содержащий числа с 5 по 15 включительно.
- С помощью слайсинга создайте кортеж, содержащий последние 7 элементов первоначального кортежа.
- С помощью слайсинга создайте кортеж (7, 9, 11, 13).
- Сохраните файл Jupyter notebook с названием «Задание 3\_2.ipynb» и загрузите его в созданный репозиторий.

### **Часть 3**

- Вернитесь к файлу «empty\_notebook.ipynb», открытому в Binder.
- Создайте список, состоящий из букв вашего имени (с помощью метода `list('строка')` или любым другим удобным способом).
- Создайте копию списка (второй список).
- Отсортируйте второй список.
- Создайте третий список, который состоит из элементов первого и второго списков.
- Из третьего списка удалите первые 3 элемента.
- Выведите получившиеся первый, второй и третий списки.
- Сохраните файл Jupyter notebook с названием «Задание 3\_3.ipynb» и загрузите его в созданный репозиторий.

### **Часть 4**

- Вернитесь к файлу «empty\_notebook.ipynb», открытому в Binder.
- Создайте пустой словарь.

- Добавьте в словарь элементы «1 : 10», «'abc' : 'def'», «0 : [0, 1, 2]», «(0, 1) : 'a'».
- Прибавьте 5 к значению 10.
- Добавьте к строке 'def' ее копию (должно получиться 'defdef').
- В список [0, 1, 2] добавьте элемент 3, а затем удалите элемент 0.
- Удалите элемент со значением 'a'.
- Выведите получившийся словарь.
- Сохраните файл Jupyter notebook с названием «Задание 3\_4.ipynb» и загрузите его в созданный репозиторий.

### ***Контрольные вопросы***

1. Какой тип данных: список, кортеж или словарь — вы считаете правильным использовать для указанных ниже данных и почему:
  - а) список студентов в группе,
  - б) список оценок студента,
  - в) данные студента: ФИО, адрес, дата рождения.
2. Напишите функцию, которая принимает на вход список натуральных чисел и возвращает словарь, в котором ключами являются элементы входного списка, а значениями — кортежи простых чисел, меньших или равных ключу.

### ***Требования к отчету***

Требуется представить отчет в виде письма на адрес [mosalov.op@ut-mo.ru](mailto:mosalov.op@ut-mo.ru) с указанием ФИО и ссылки на репозиторий с сохраненными файлами Jupyter notebook.

### ***Литература***

1. <https://pythonworld.ru/samouchitel-python>
2. <https://www.coursera.org/learn/diving-in-python>
3. <https://stepik.org/course/67/promo>
4. <http://pythontutor.ru/>