**Язык программирования Python**

В 1989 году началось создание языка программирования Python Гвидо ван Россумом. Назван он был в честь любимого шоу создателя– «Летающий цирк Монти Пайтона» («Monty Python's Flying Circus»). Существует 3 версии Python. В настоящее время самым актуальным является Python 3.

Python – это интерпретируемый, универсальный и передовой язык программирования, который используется в следующих областях:

* анализ данных;
* написание прикладных программных систем;
* разработка графических интерфейсов;
* написание сетевого программного обеспечения;
* взаимодействие с базами данных.

К особенностям Python относится *динамическая типизация*. Типы объектов во время выполнения программы строго не определены, они могут изменяться. Так же для определения переменных не нужно объявлять их тип.

Для выделения блоков кода используется «:».

Python позволяет писать операции на одной строке с помощью разделителя «;», но делать так не рекомендуется, такой код достаточно сложно читать.

Однострочный комментарий в Python начинается с #, а многострочный выделяется с помощью “…”.

Python поддерживает функциональное программирование.

Существуют два режима запуска Python:

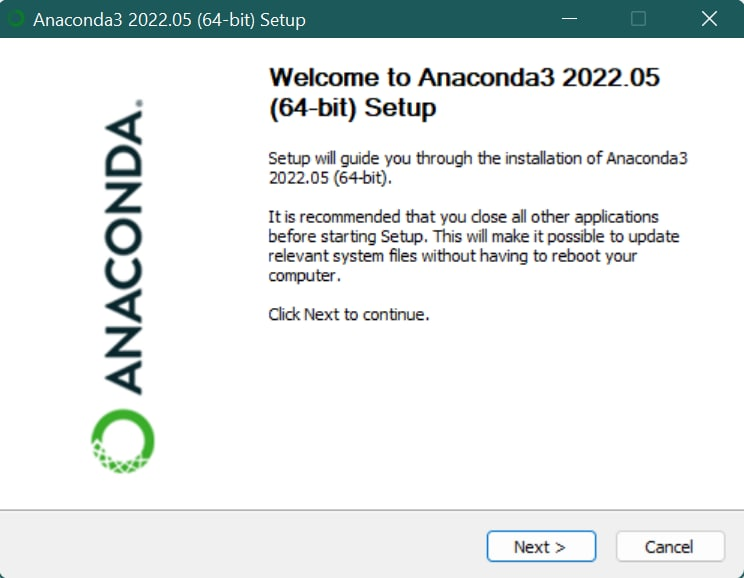
1. Стандартный режим. Необходимо сохранить файл с кодом с расширением .py, затем подать этот файл на вход интерпретатору для запуска кода.
2. Интерактивный режим. Результат работы команд виден сразу.

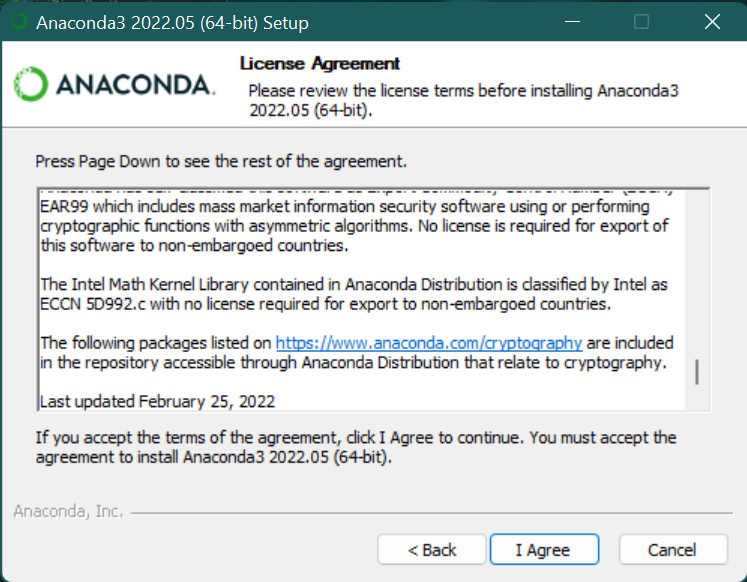
**Установка Python**

Один из самых простых способов установки Python – через дистрибутив Anaconda. Он содержит не только Python, но и необходимые библиотеки.

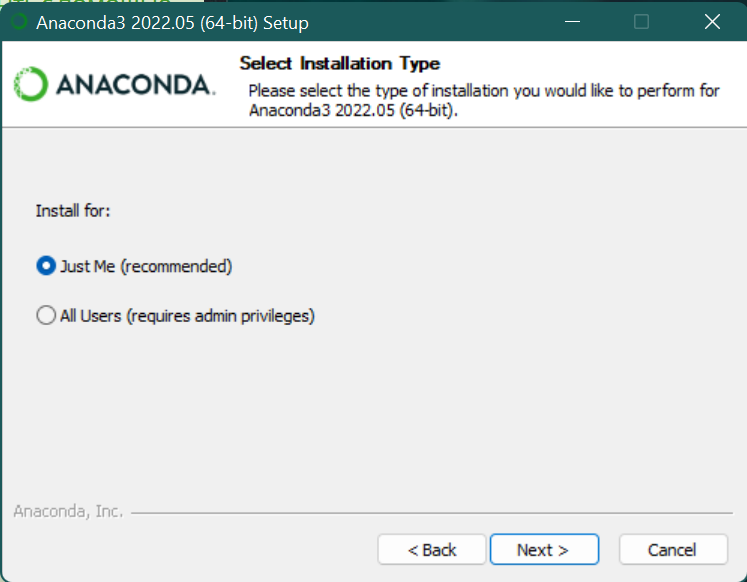
Скачать Anaconda можно тут: <https://www.anaconda.com/>

Далее необходимо установить Anaconda.

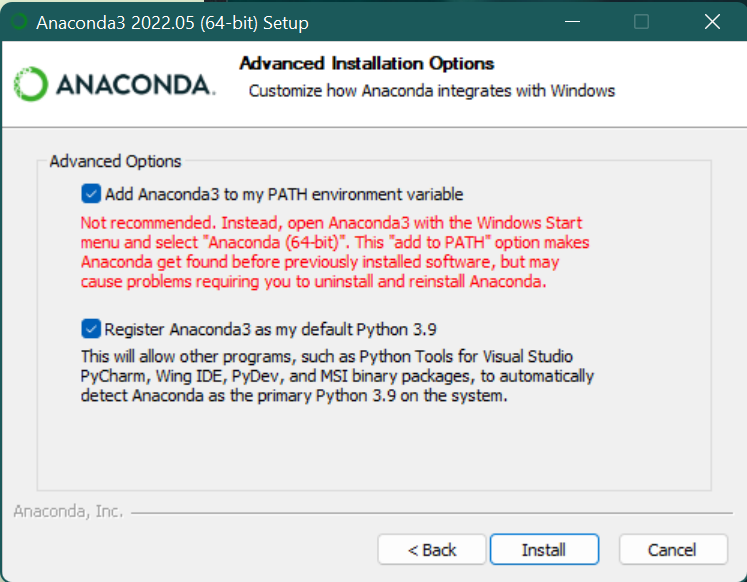


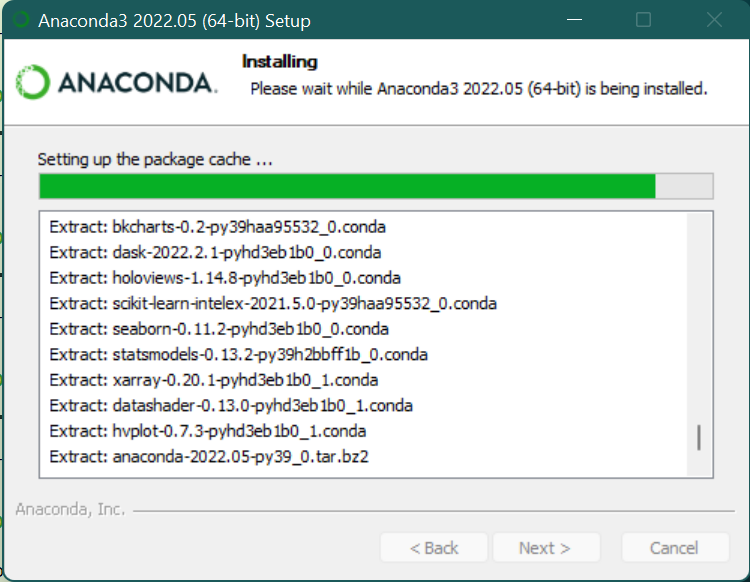


Необходимо ознакомиться с лицензией и нажать I agree.

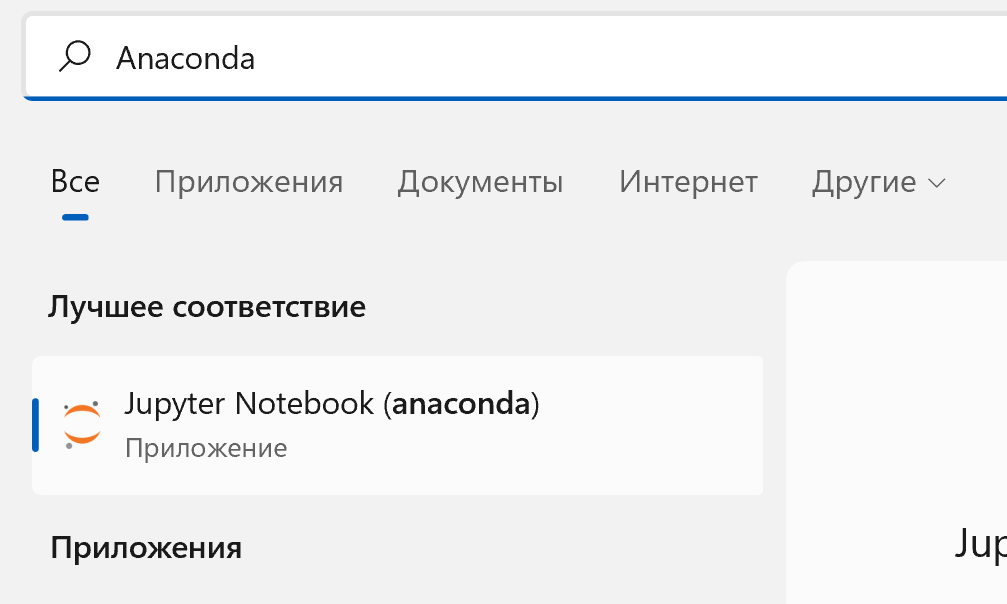


Далее предлагается внести изменения в окружение для того, чтобы Python мог по умолчанию запускаться из Anaconda. Выбрана первая галочка для того, чтобы в пункте PATH был прописан путь до расположения Anaconda. В дальнейшем это позволит писать команды в консоли Windows (cmd) команды для установки/удаления/обновления библиотек Python, например pip install/conda install

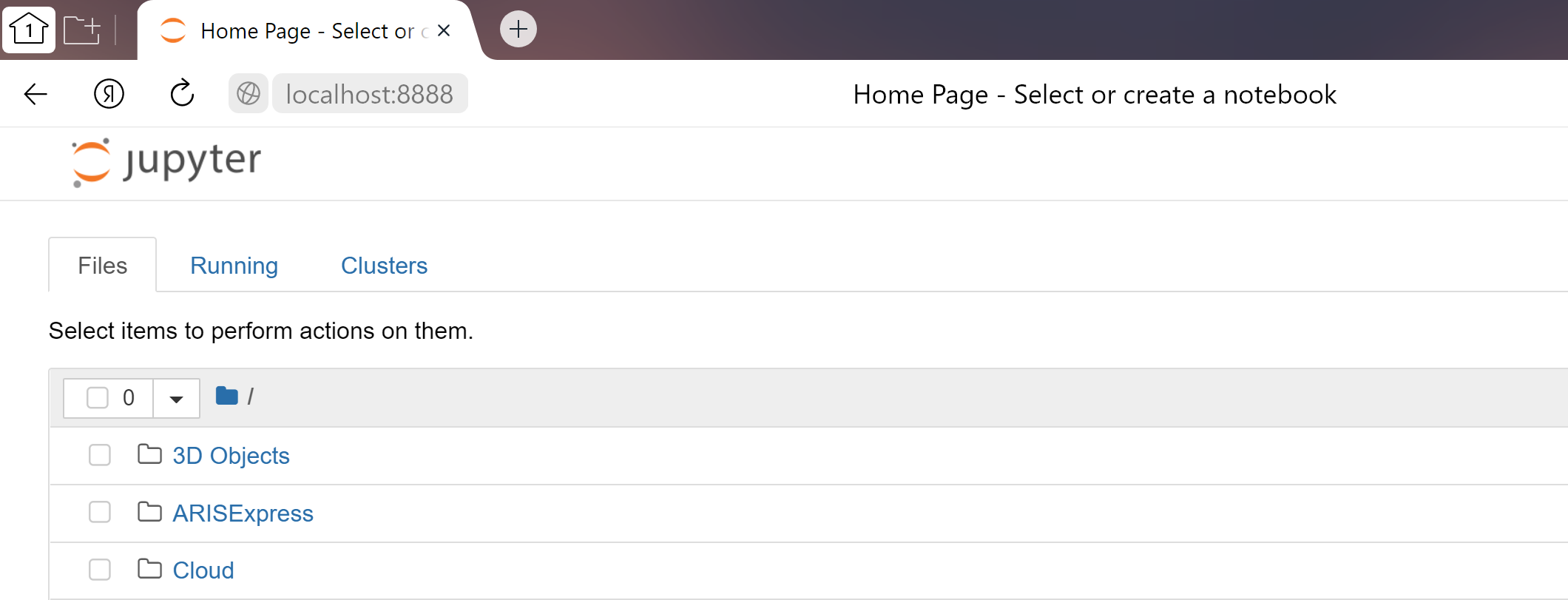




Проверим, работает ли Jupyter Notebook.

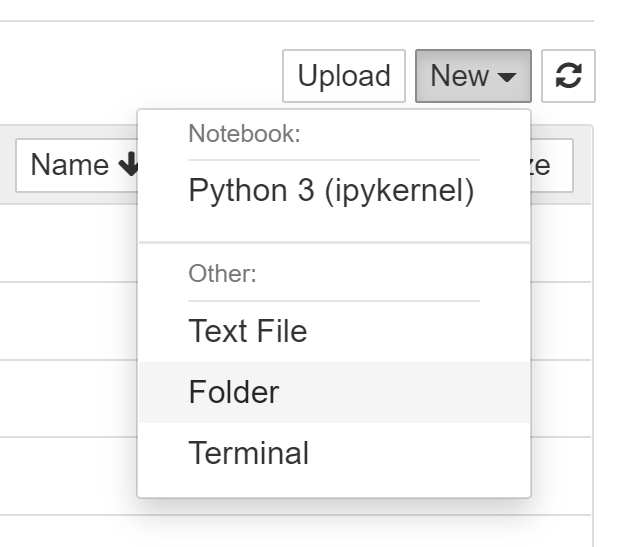


Должна открыться новая вкладка в браузере.



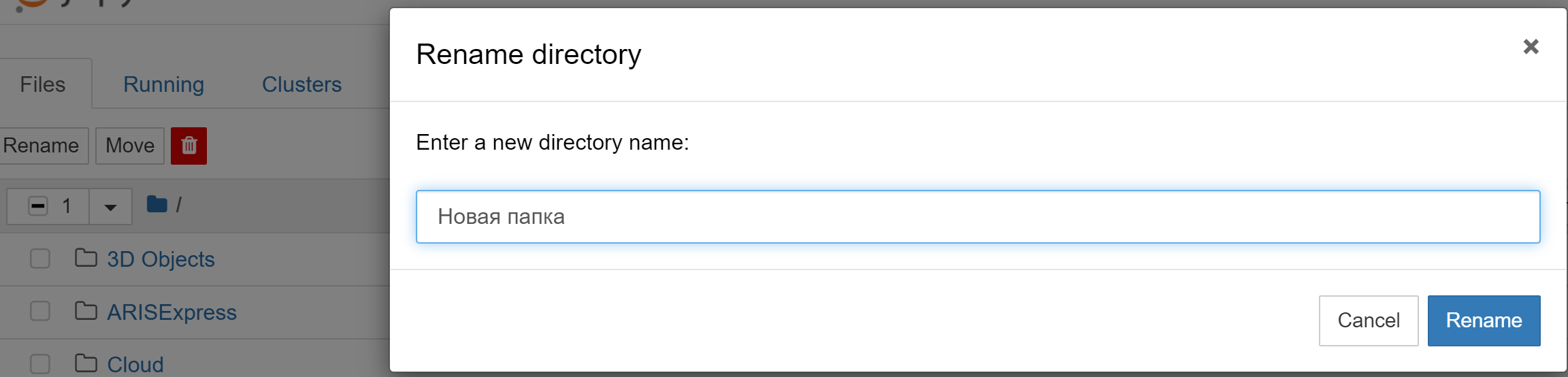
**Работа с Jupyter Notebook**

Создадим новую папку.

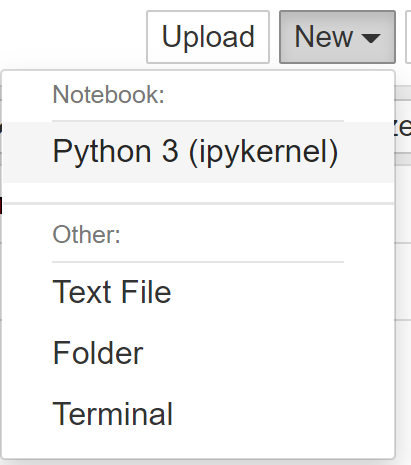


Для того, чтобы переименовать папку необходимо выбрать ее и в левом верхнем углу выбрать «Rename».

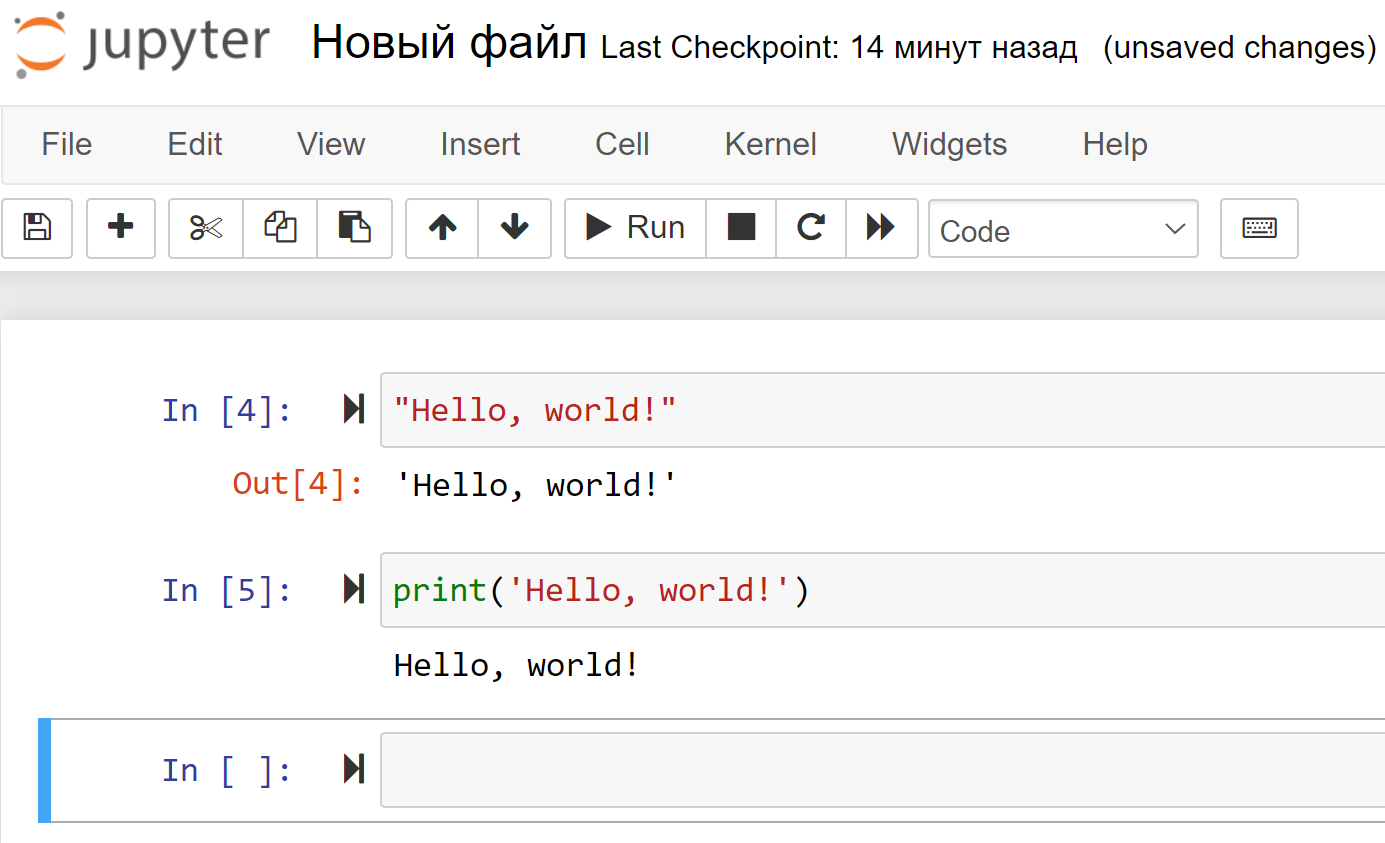




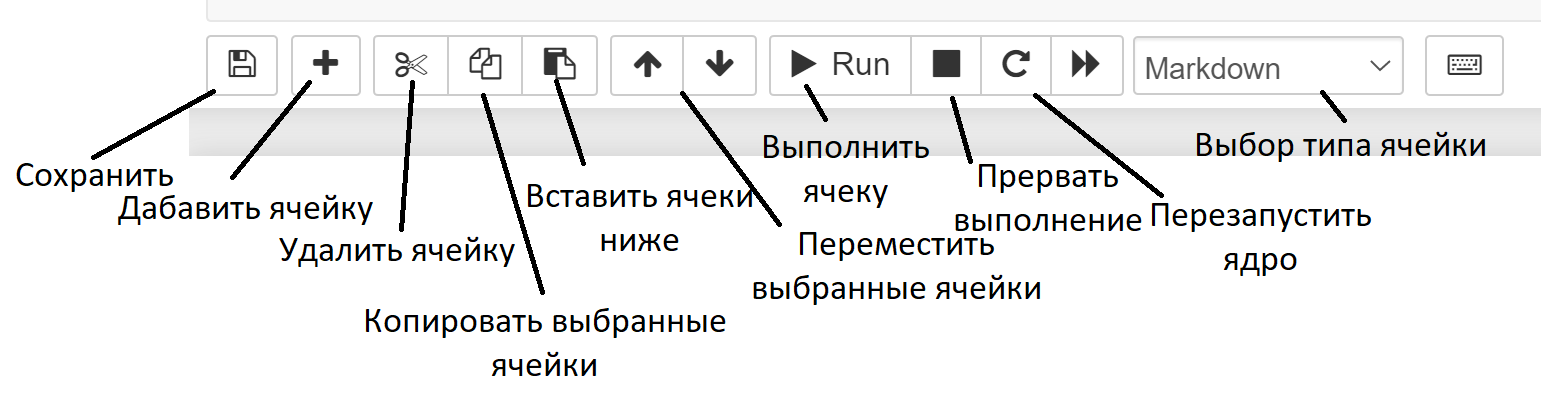
Создадим новый файл IPython. Файл с расширением ipynb.



Выведем предложение «Hello, world!»

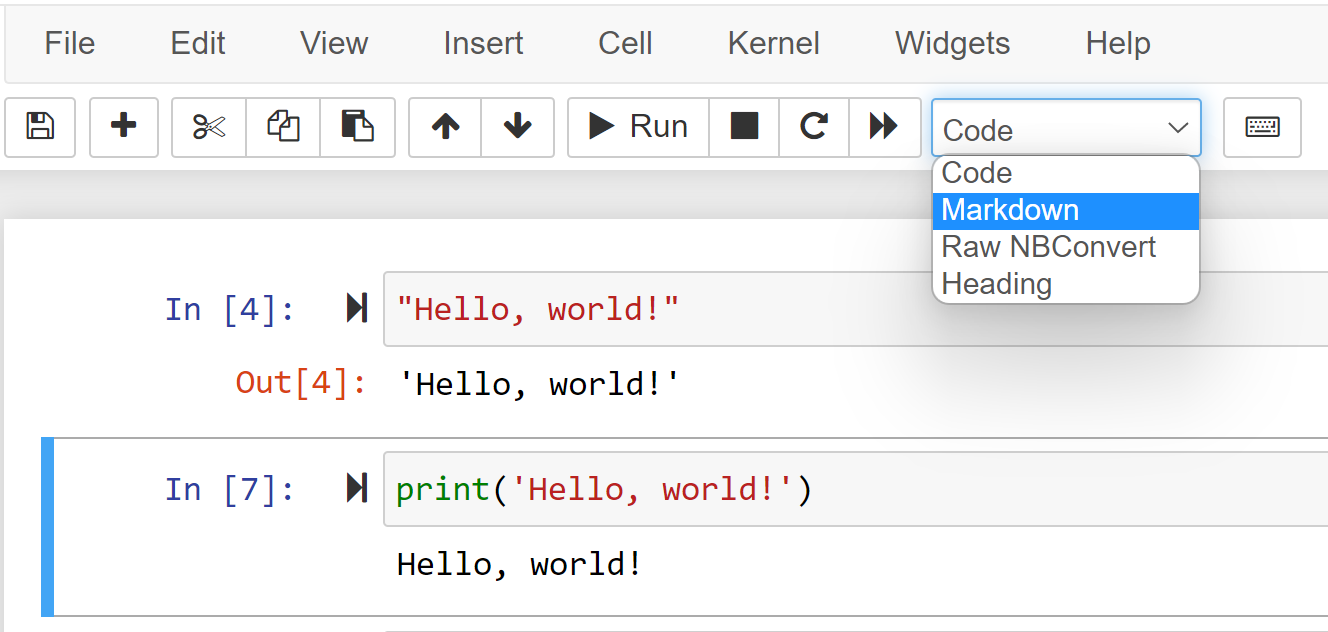


Для запуска кода в ячейке необходимо использовать комбинацию клавиш «Shift+Enter».

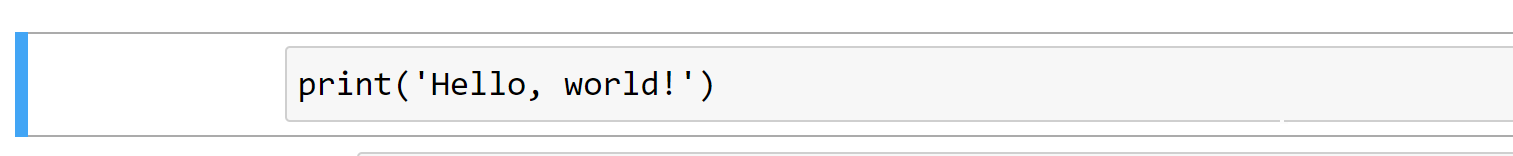


Для того, чтобы выделить несколько ячеек, необходимо с зажатым «Shift» их выбрать.

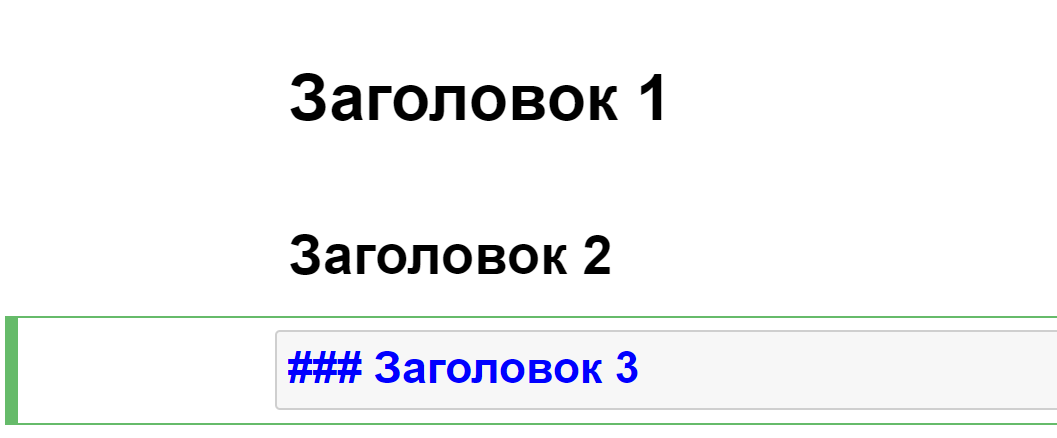
Существует два вида ячеек в Jupyter Notebook: ячейка для выполнения кода (Code) и ячейка для написания текста (Markdown). Тип ячейки всегда можно изменить, например:



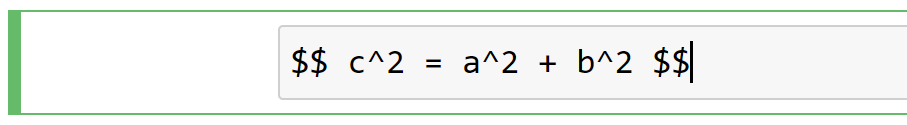
Результат:

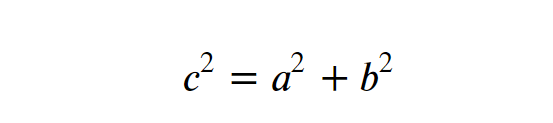


Для того, чтобы сделать заголовок в текстовой ячейке, необходимо использовать #. Чем больше #, тем меньше будет уровень заголовка. Пример:



В ячейках для текста также можно вставлять формулы с помощью знака $$, например:

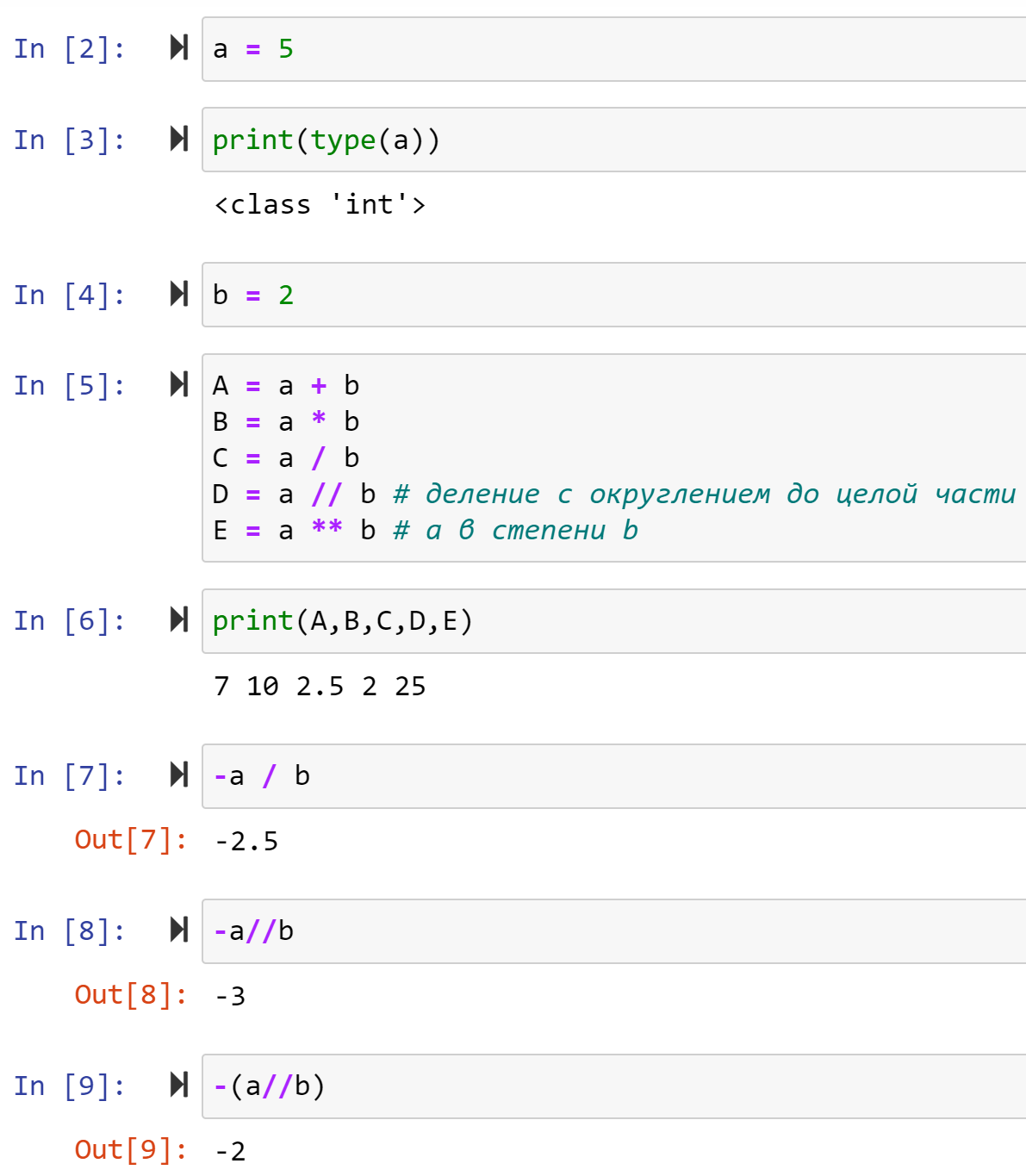




**Типы переменных Python**

В Python 3 есть следующие базовые типы переменных: bool, str, float, int, complex.

*Целочисленные переменные (int) и основные операции с ними:*

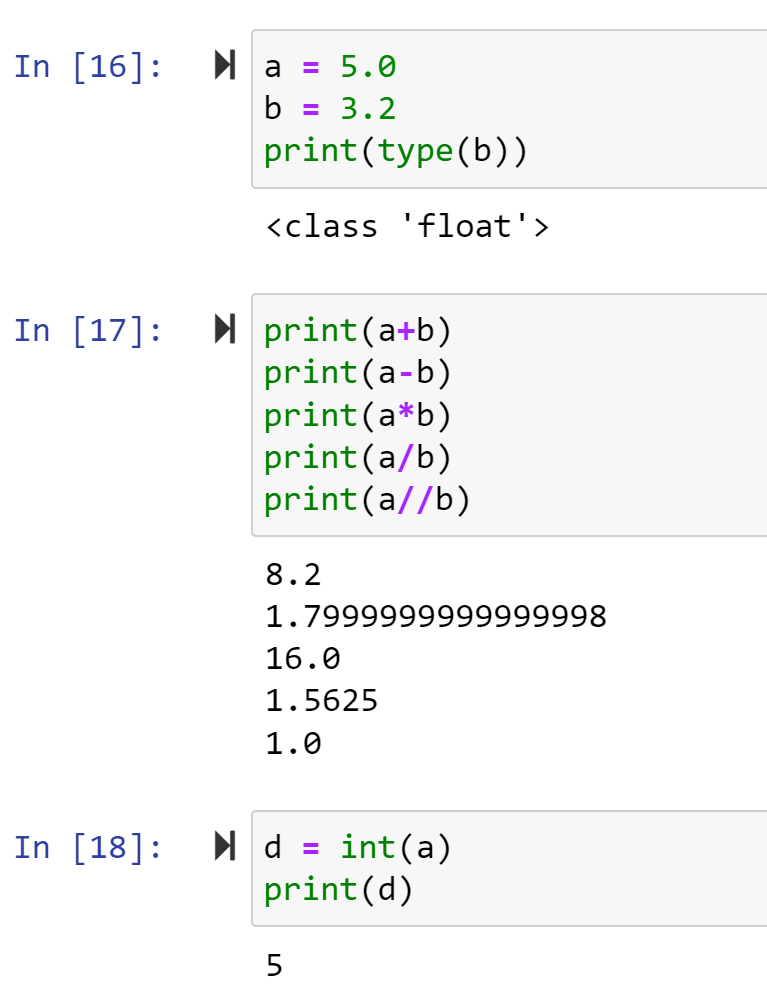


При целочисленном делении отрицательного целого числа -5 на 2 получаем -3, идет округление в меньшую сторону. Чтобы получить такой же результат, как и в случае с положительными числами, необходимо сначала поделить положительные числа, а затем добавить минус (In [9]).

*Комплексные переменные (complex):*

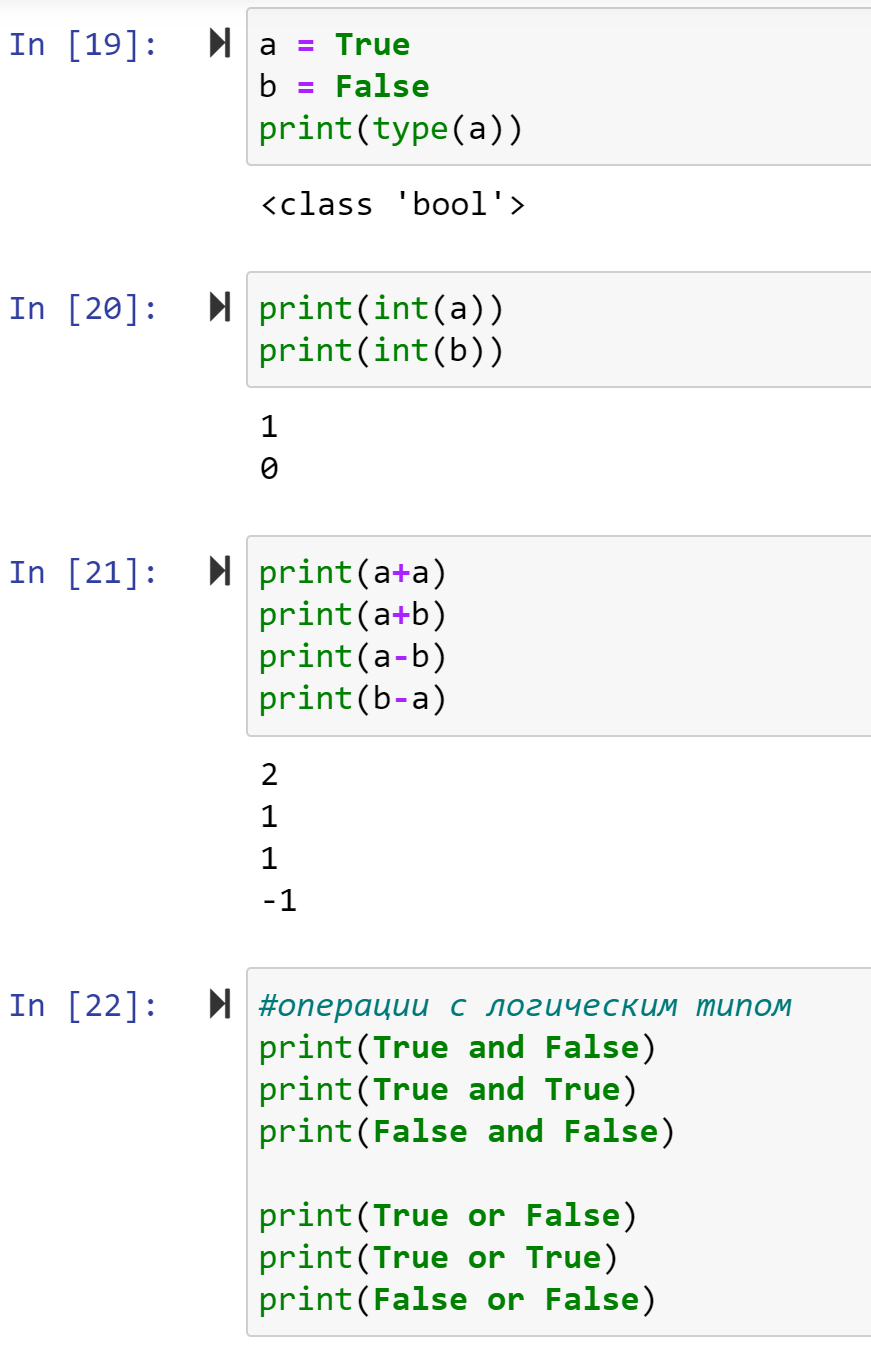


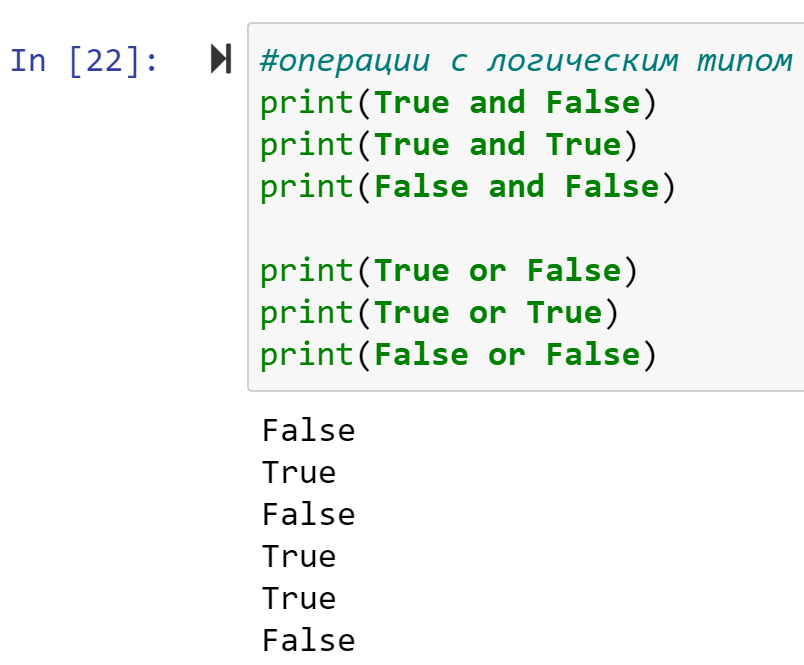
*Вещественные числа (float):*

**

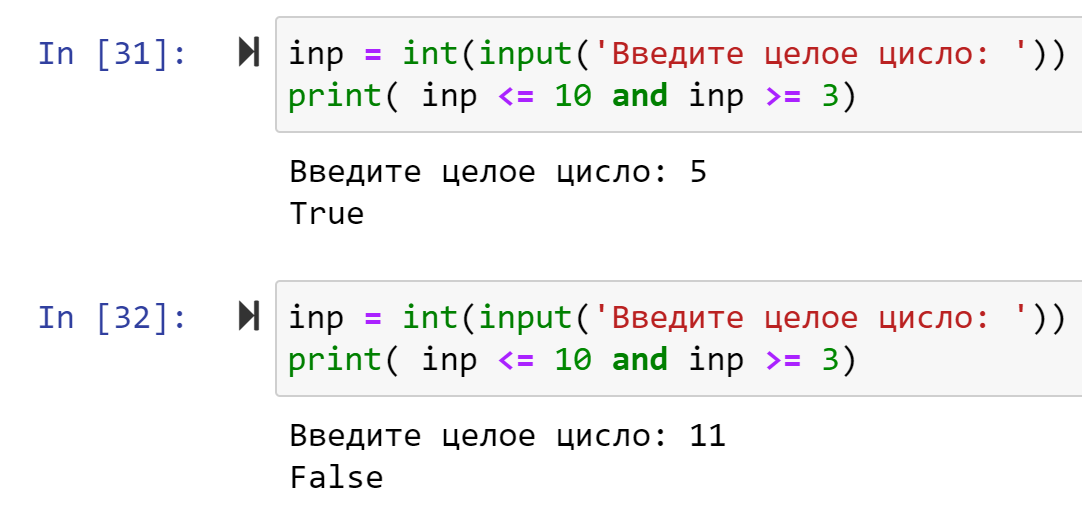
*Логический тип (bool):*

Логический тип (True или False) можно привести к целочисленному типу.





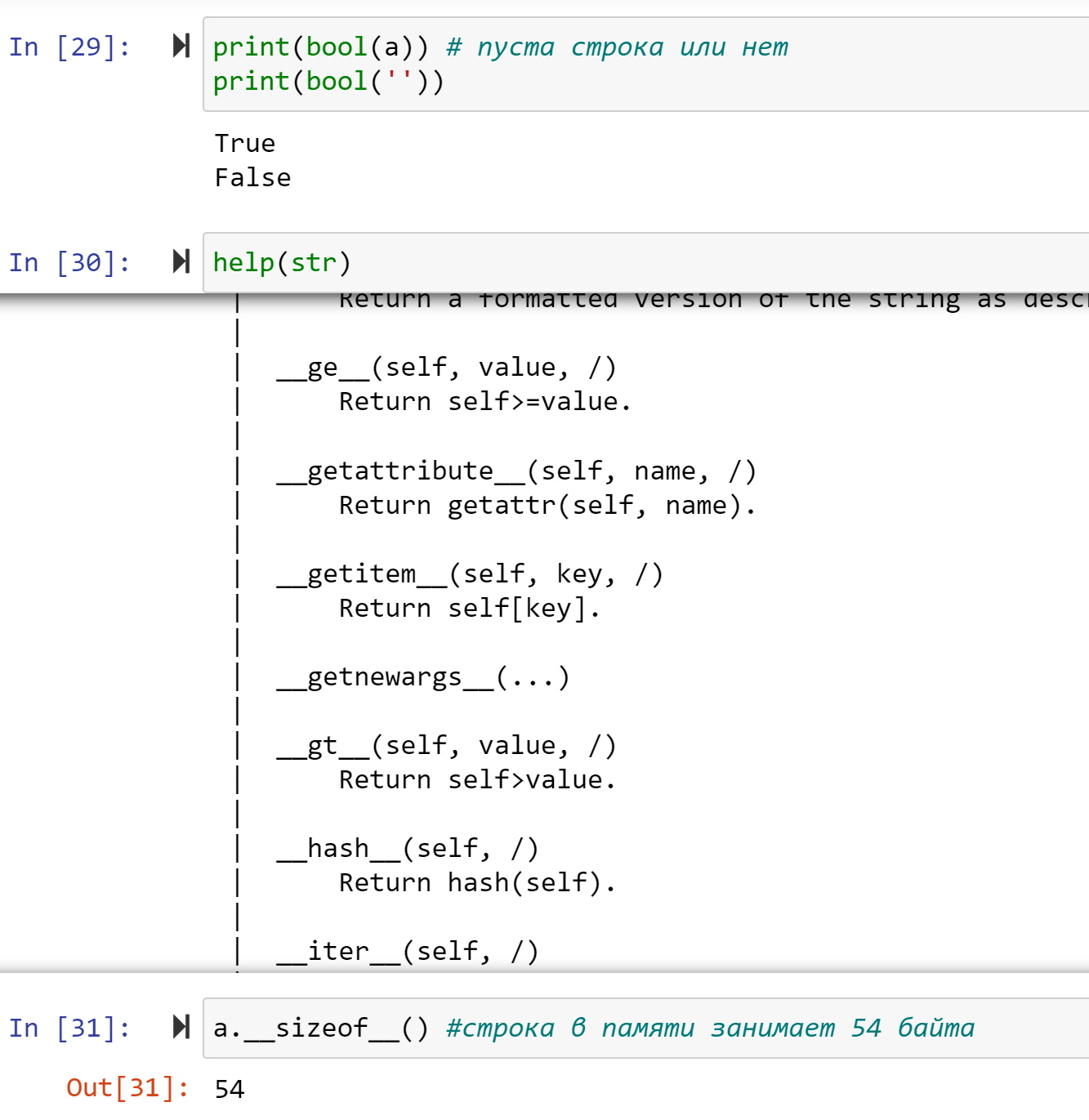
Проверка числа, подающегося на вход:



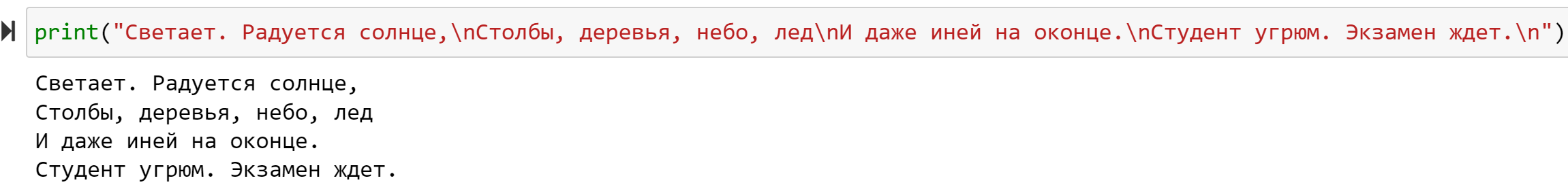
*Строковый тип (str):*

В Python строка – это последовательность символов, фиксация которой начинается с 0. Строки являются неизменяемыми.

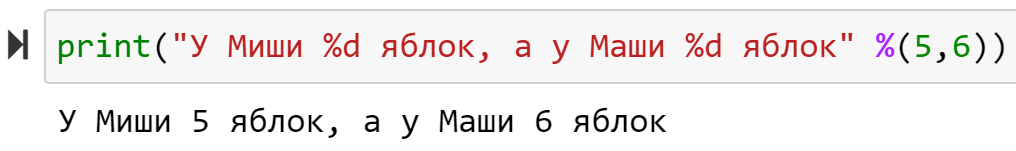




\n – символ перевода на новую строку.

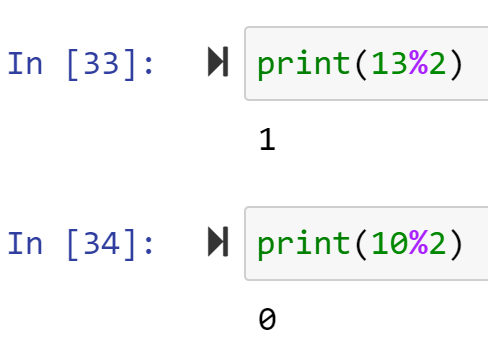


Python поддерживает форматированный вывод строк с помощью оператора форматирования %. Например:

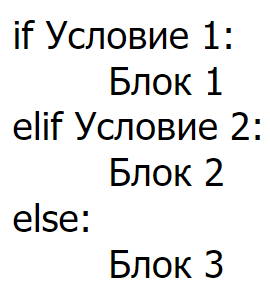


*Условный оператор (if, else, elif):*

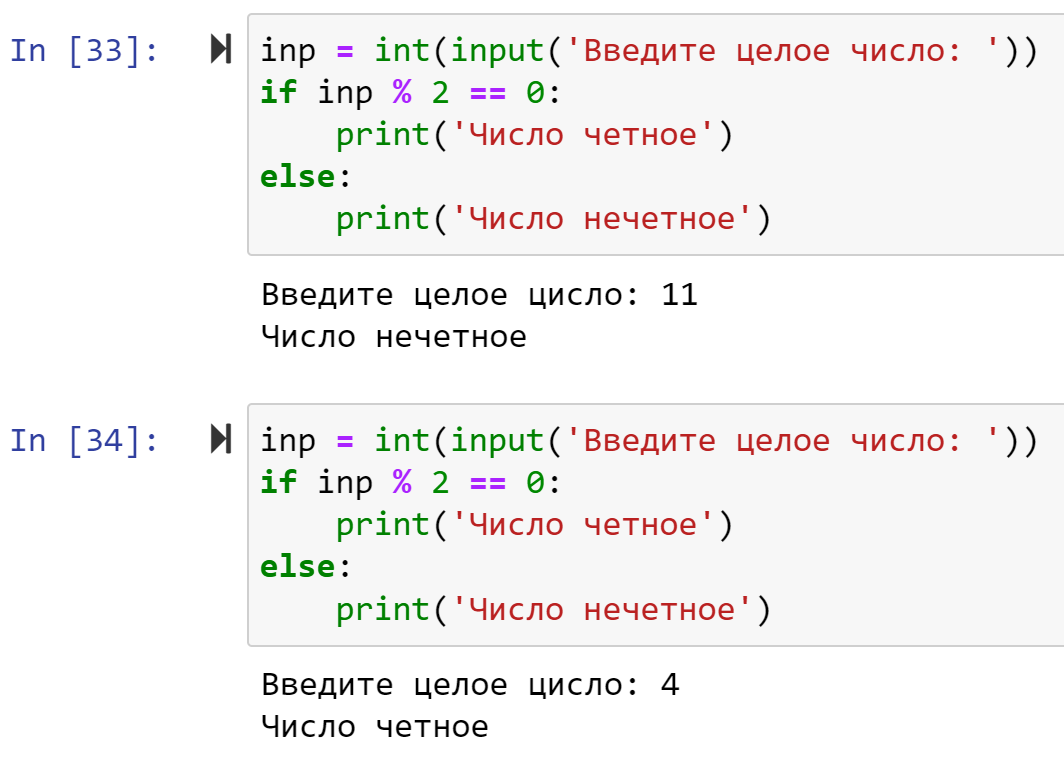
Классическая задача – проверка числа на четность. Для этого используется знак %, который позволяет посчитать остаток от деления, например:



Пример реализации условия в Python. В начале пишется ключевое слово if, затем блок кода, который должен выполниться при выполненном условии. Затем, если условие не выполняется, но есть еще одно условие, то пишется ключевое слово elif, после чего снова идет блок кода. Если же ни одно из условий не выполняется, то пишется ключевое слово else и соответствующий ему блок кода. В Python блоки кода выделяются отступом.



Пример:

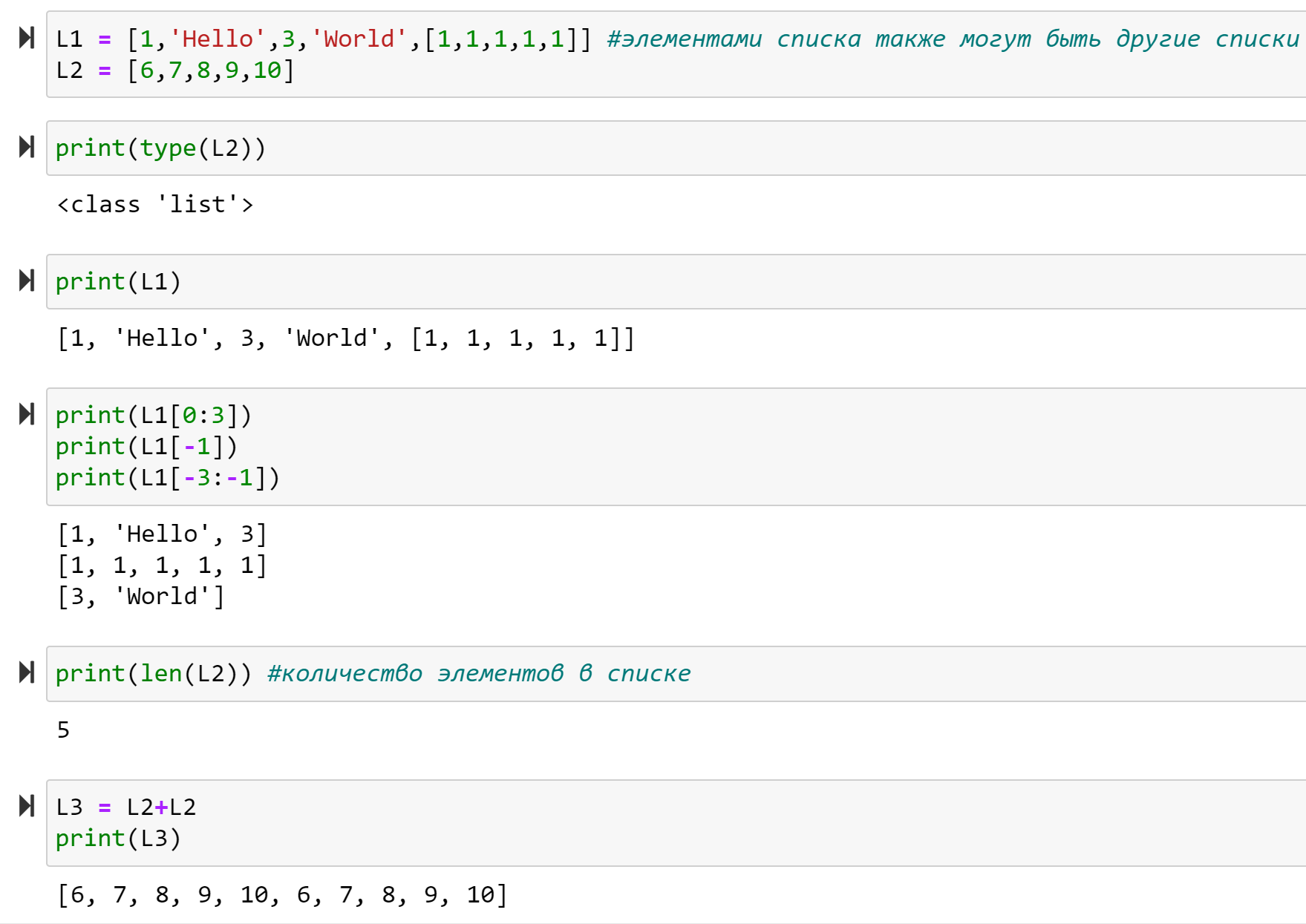


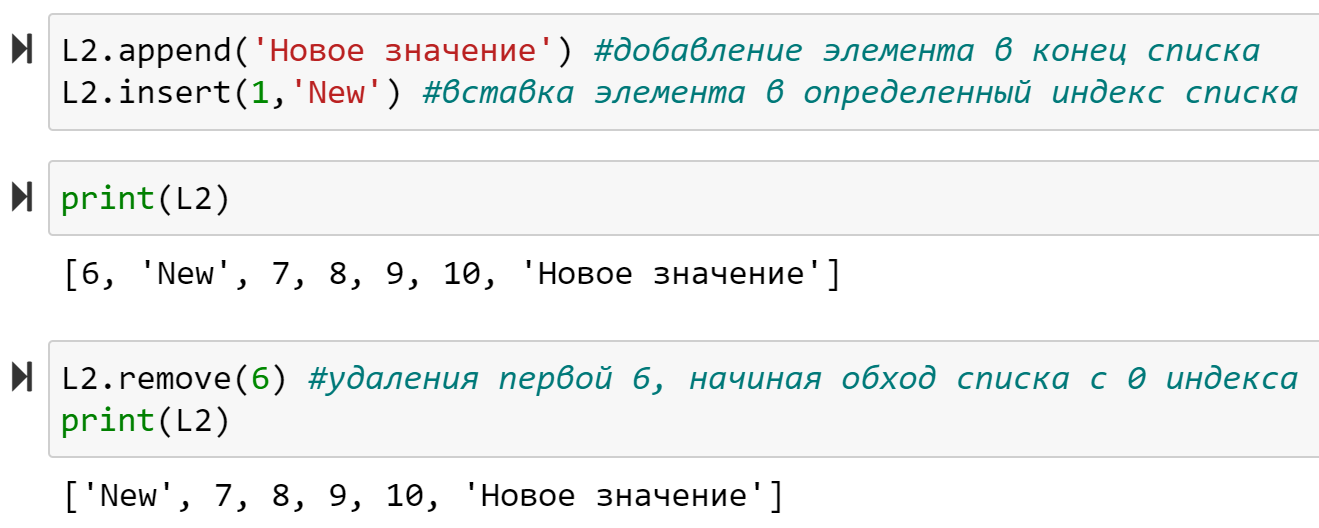
**Структуры данных Python**

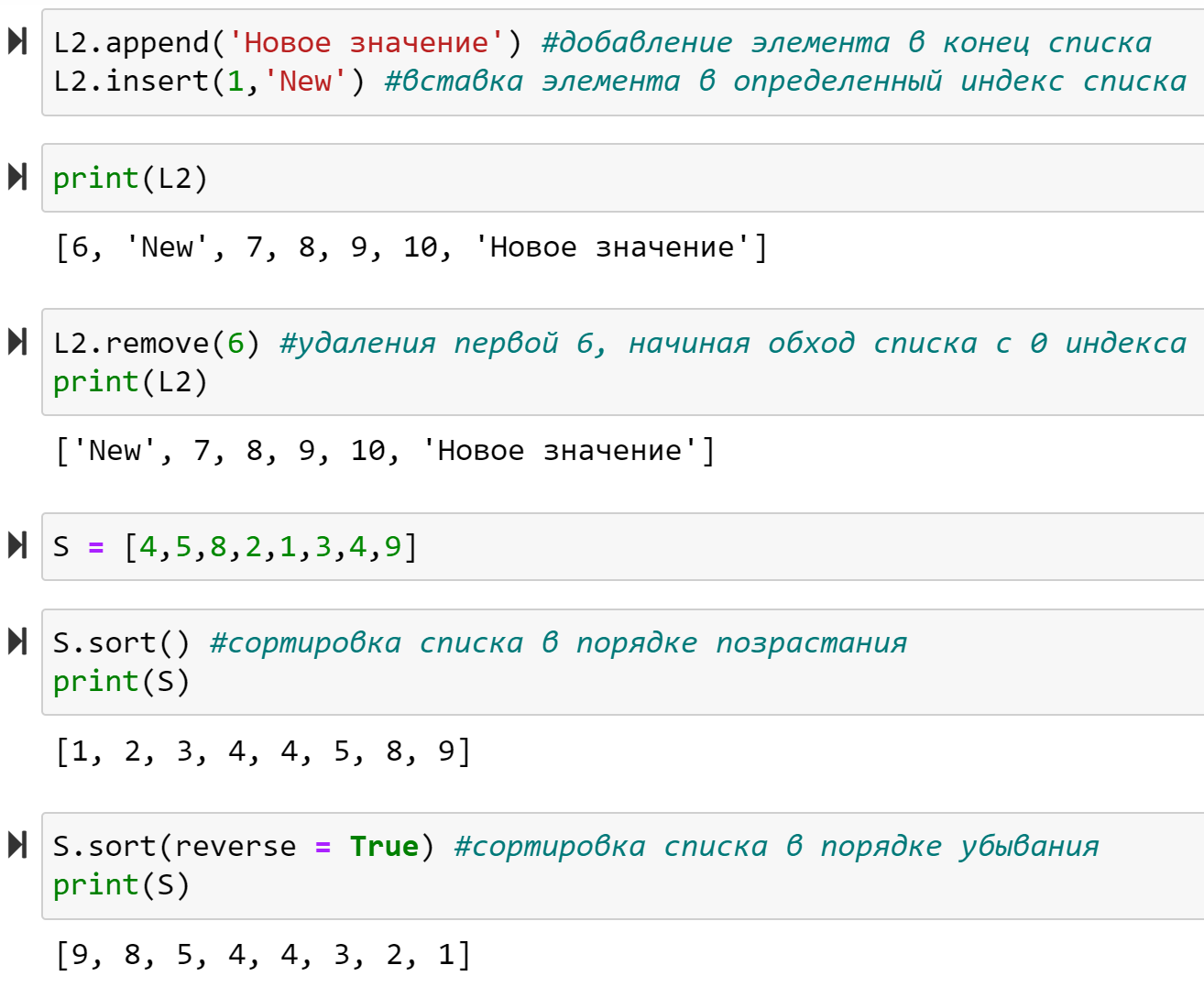
В Python все структуры данных являются объектами. У них есть методы и поля. Структуры данных Python: список (list), кортеж (tuple), множество (set), словарь (dict).

*Список (list).*

Список – это изменяемая структура данных. Может хранить в себе элементы разных типов. Обращение к элементу списка происходит с помощью индекса, начиная с 0. Чтобы обратиться к элементу с конца списка, то используется -1, -2 и т.д. Объявление списка происходит с помощью [] или list(). Пример:





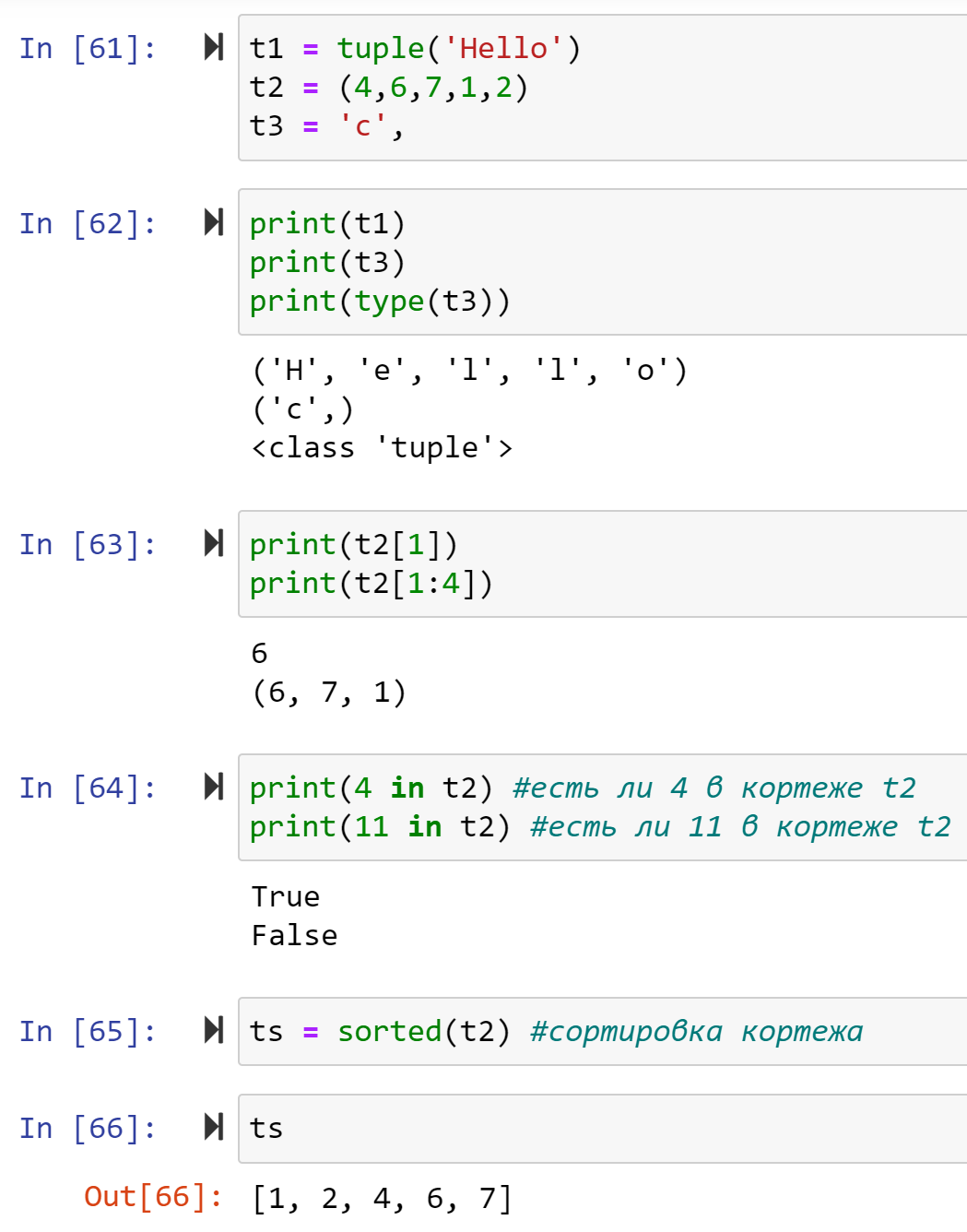


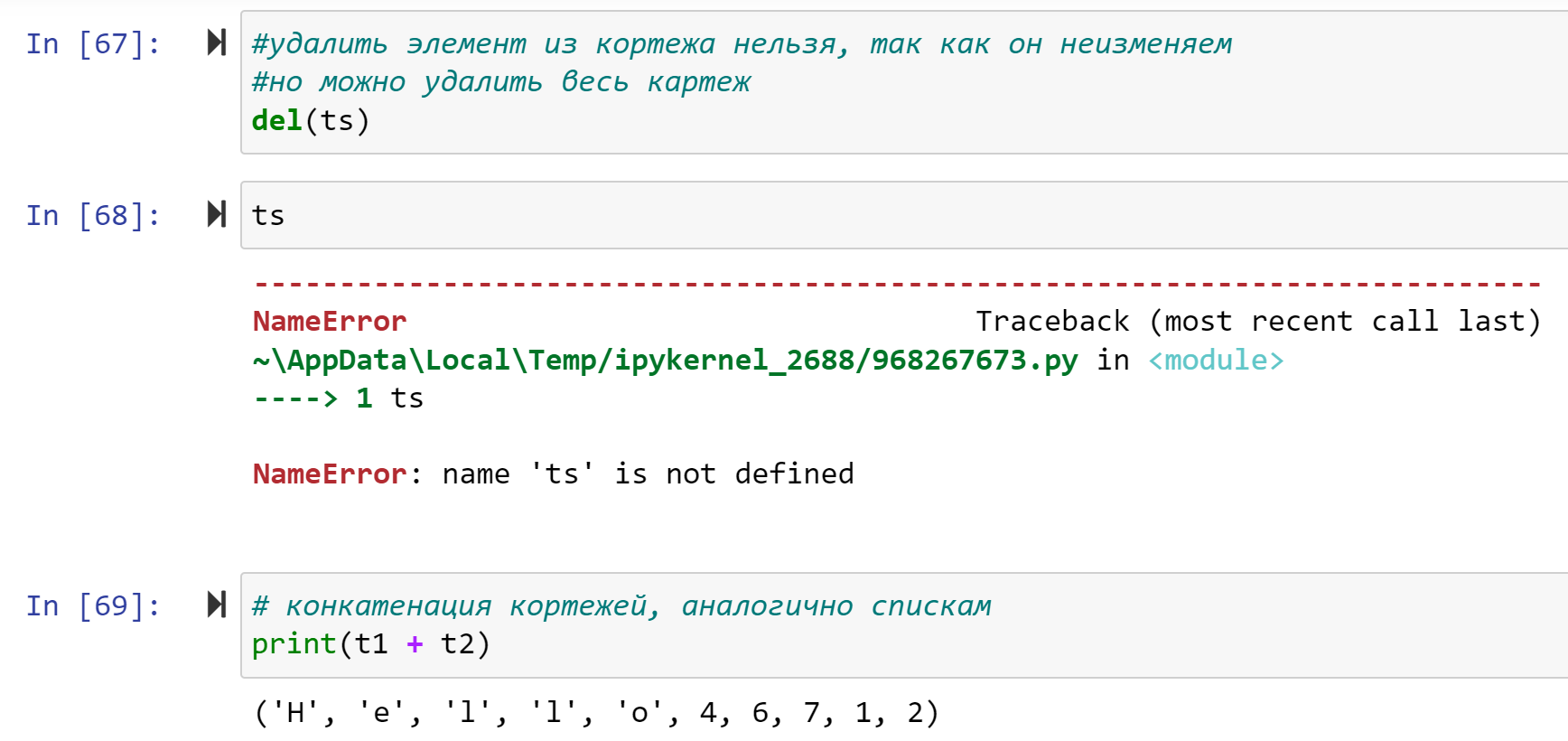


*Кортеж (tuple).*

Кортеж – неизменяемая структура данных. Аналогично списку, может хранить элементы разных типов. Кортеж объявляется через () или tuple().

Кортежи работают быстрее списков за счет своей неизменяемости. Они хранятся в памяти особым образом, поэтому операции с элементами кортежа выполняются быстрее. Кортежи используются, когда есть требования к безопасности.

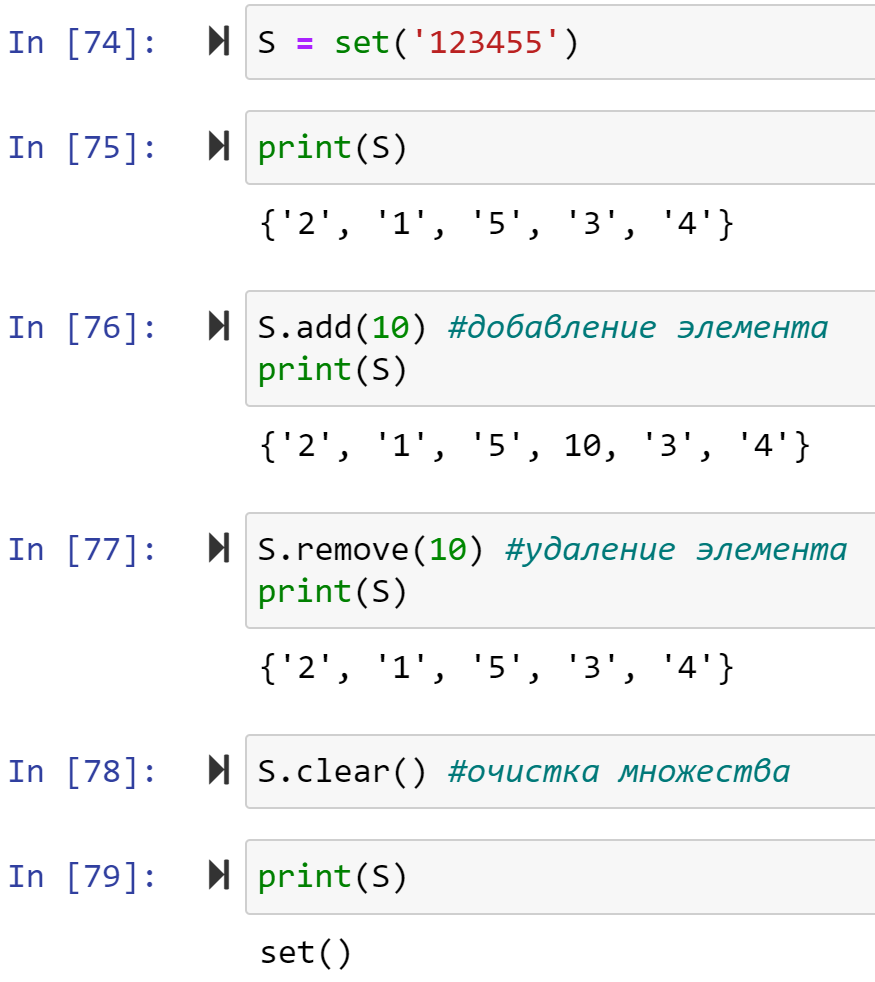






*Множество (set).*

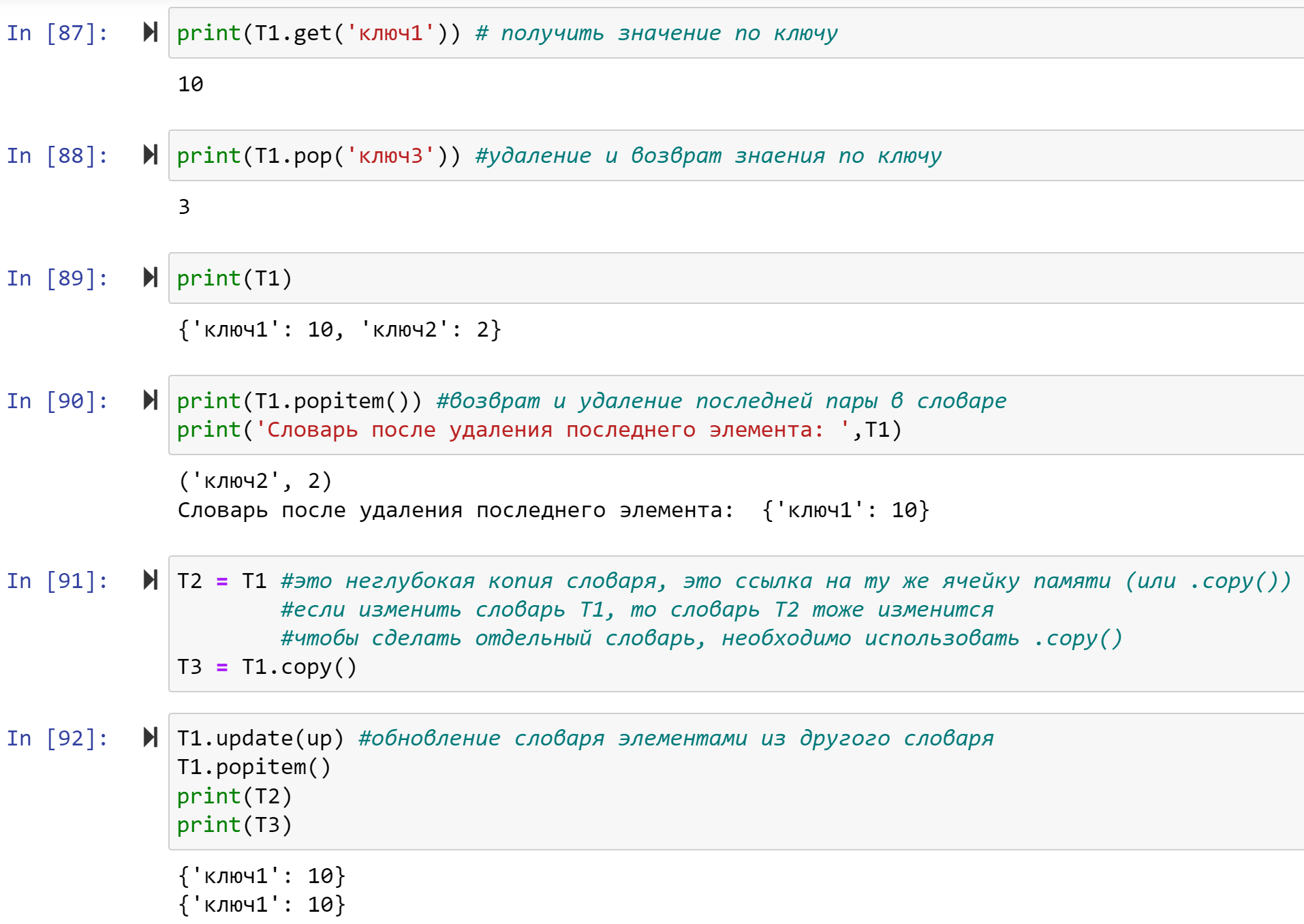
Множество – изменяемая структура данных, которая имеет требование к уникальности элементов. Множество удобно использовать для удаления повторяющихся элементов.

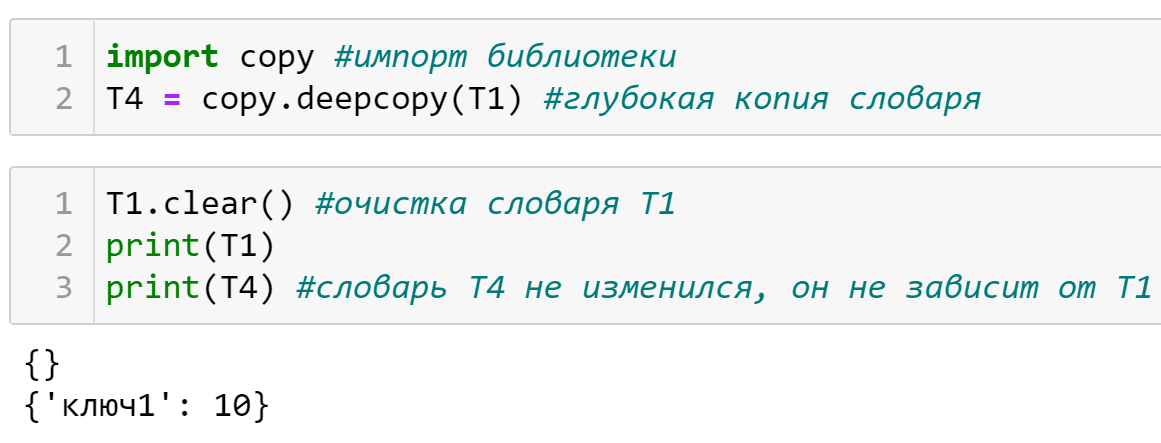


*Словарь (dict).*

Словарь – ассоциативный массив, который хранит пары ключ: значение. Объявляется через {} при указании {ключ1:значение1, ключ2:значение2}. Ключ – уникальный и неизменяемый, поэтому tuple может использоваться в качестве ключа.



****

****

**Практическая работа**

1. Установить Python.
2. Написать программу, которая вычисляет площадь фигуры, параметры которой подаются на вход. Фигуры, которые подаются на вход: треугольник, прямоугольник, круг. Результатом работы является словарь, где ключ – это название фигуры, а значение – это площадь.
3. Написать программу, которая на вход получает: 1) два числа; 2) операцию, которую к ним нужно применить. Должны быть реализованы следующие операции: +, -, /, //, \*\*. Результатом работы программы является одно число.
4. Написать программу, вычисляющую площадь треугольника по переданным длинам трёх его сторон по формуле Герона:

где

На вход программе подаются целые числа, выводом программы должно являться вещественное число, соответствующее площади треугольника.

1. Оформить отчет о проделанной работе. В отчете должен быть приведен код и результат его выполнения.