**Кейс 1. Игра «Угадай число»**

В этом руководстве подробно рассмотрен процесс создания игры, в которой пользователь угадывает число, заданное компьютером.

**Предварительные шаги**

На ПК с ОС Windows необходимо установить стандартную среду разработки программ на языке программирования Python. В окне браузера переходим на официальный сайт Python: https://www.python.org/ и во вкладке «Downloads» выбираем для установки последнюю актуальную версию программы.

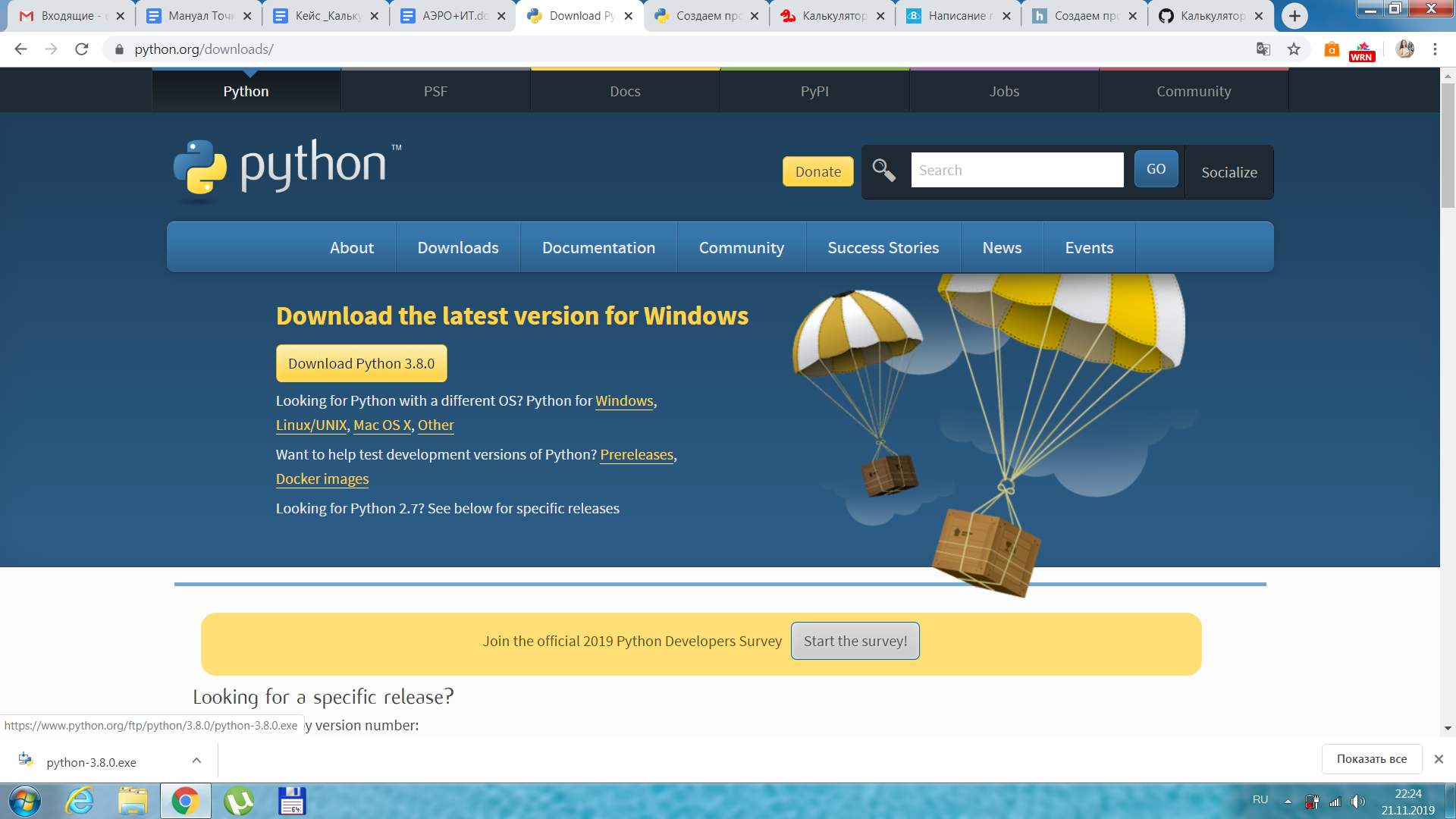


Рис. 1. Официальный сайт для установки среды разработки

Далее необходимо запустить исполняемый файл для установки.



Рис. 2. Начало установки Python

После завершения процесса установки в окне «Пуск» появится IDLE – стандартная среда разработки.

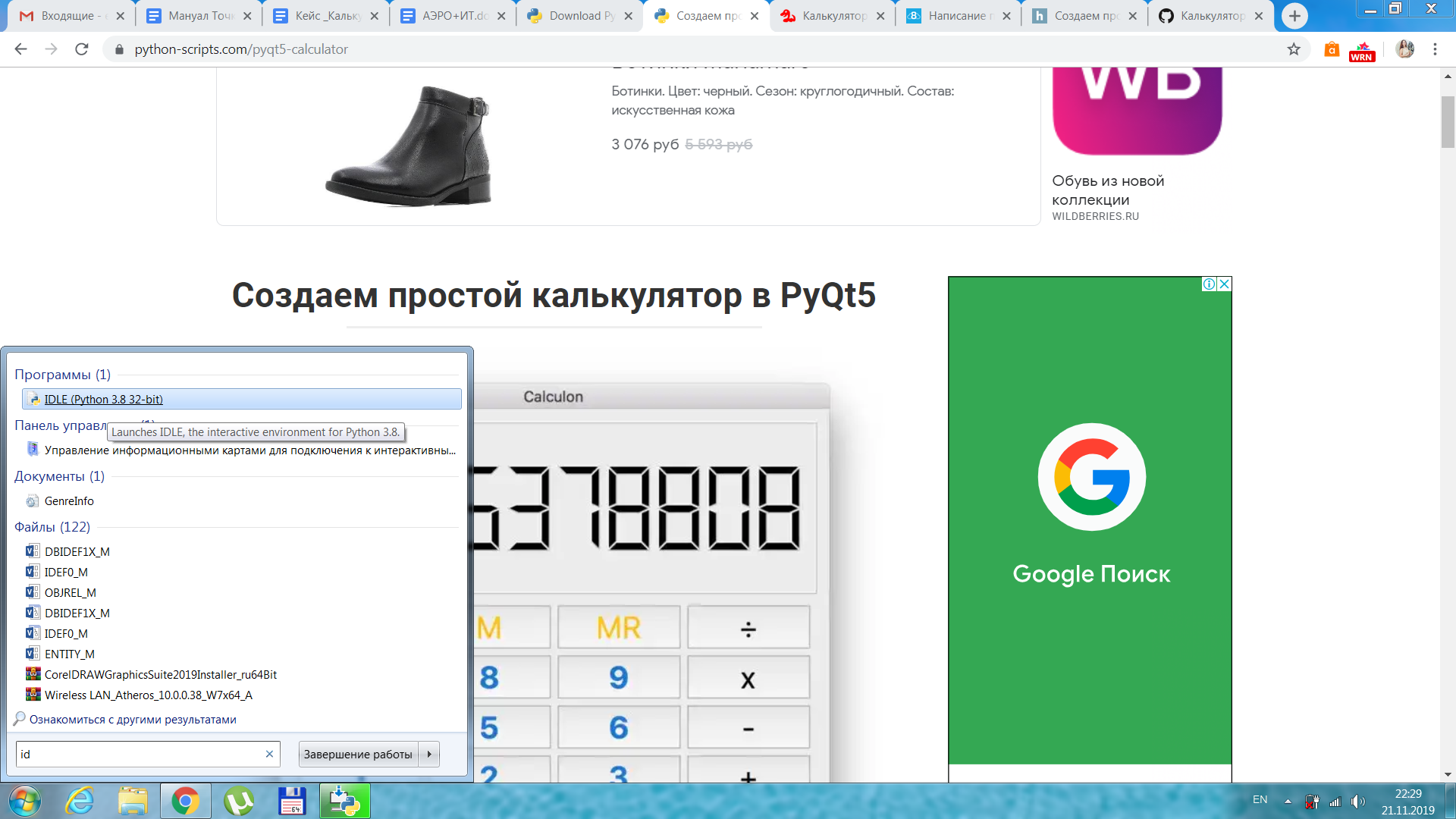


Рис. 3. Где найти IDLE Python

Необходимо запустить программу IDLE Python и создать новый файл для написания текста программы (File/New File).

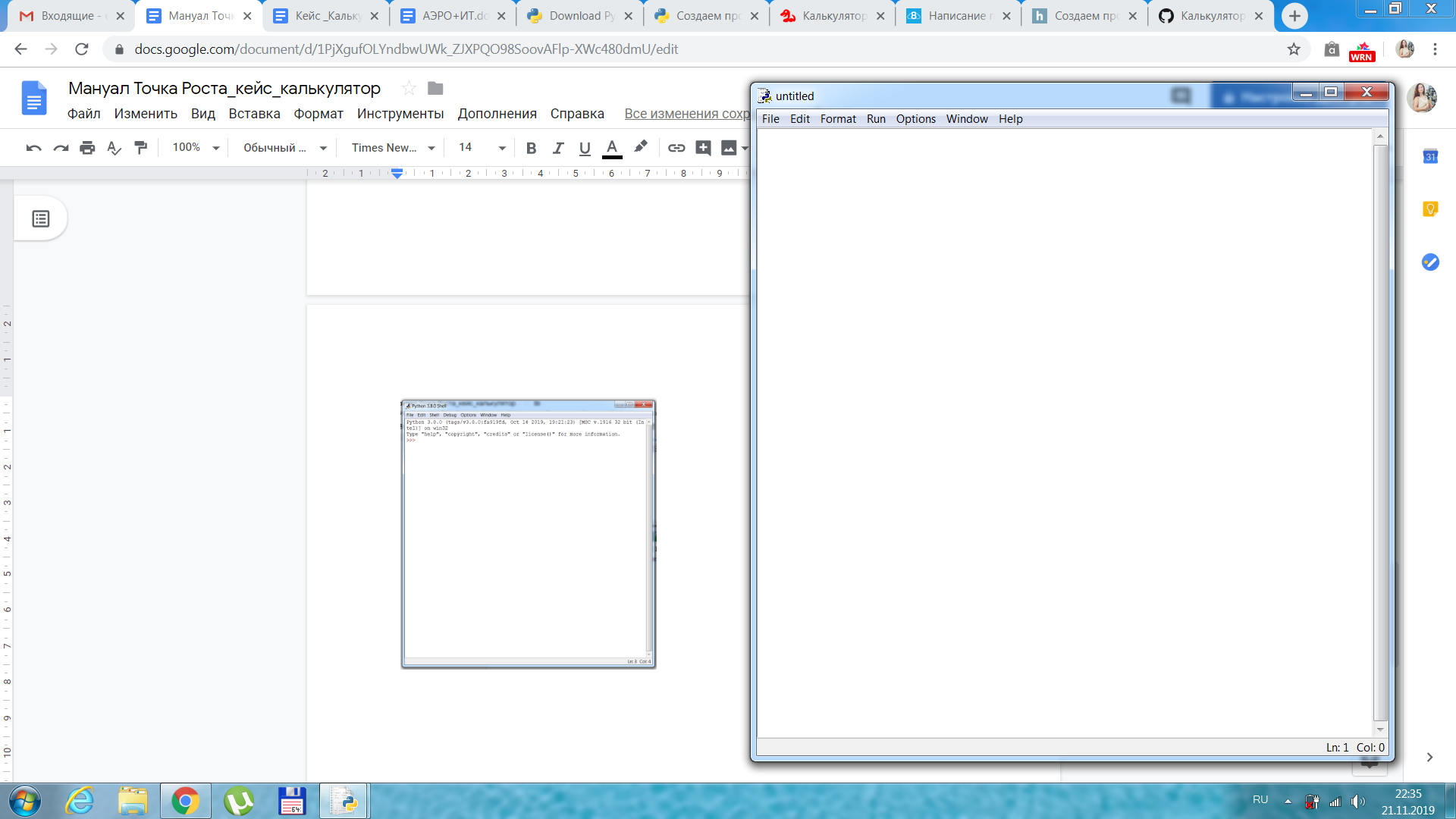
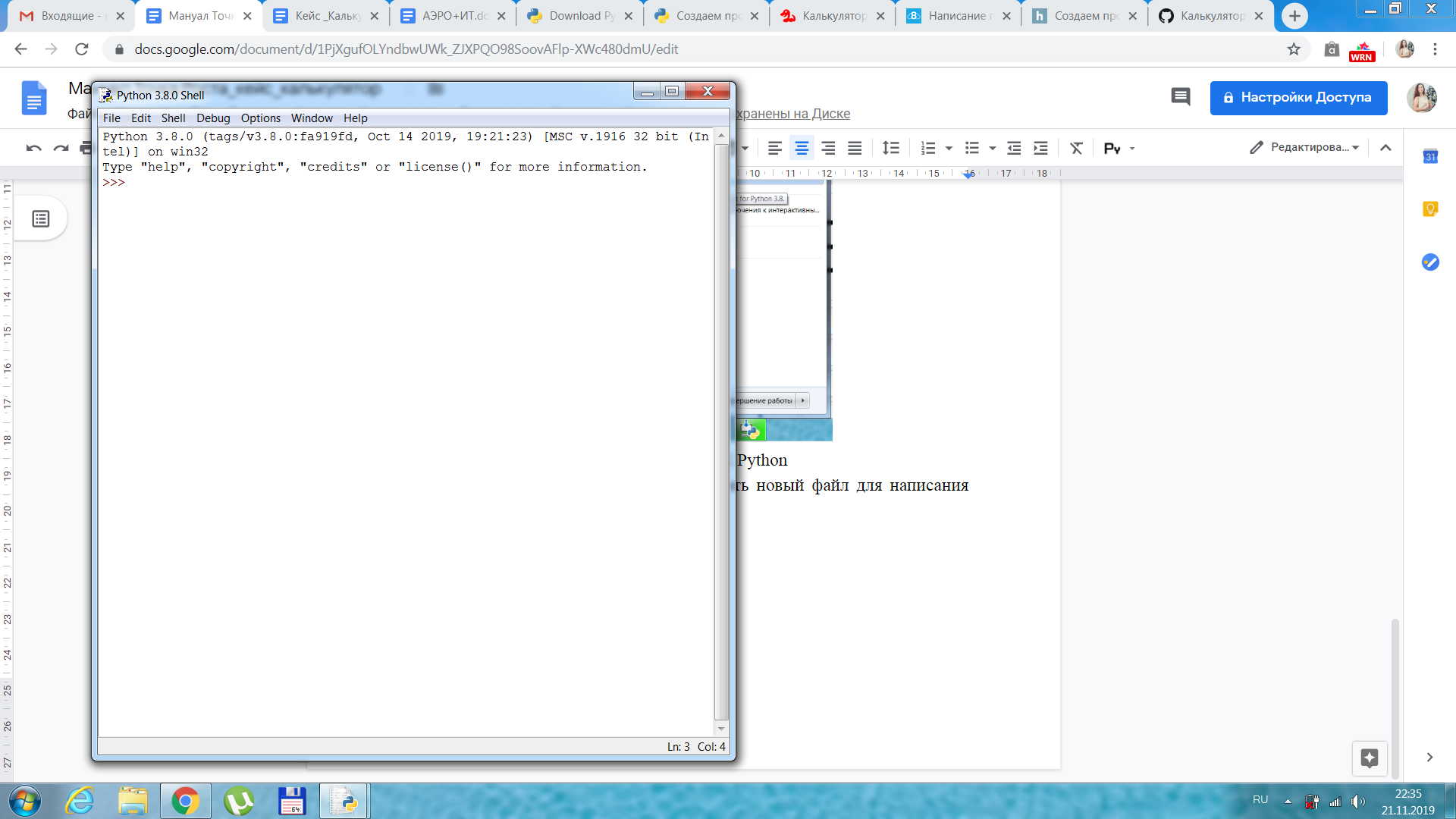


Рис. 4. Запуск интерпретатора и текстового редактора кода IDLE Python

Сохраните файл до начала работ (File/Save As…), выбрав необходимую папку и дав ему имя «chislo», расширение .py установится автоматически.

**Упражнения по поиску элементов в массиве**

Для группировки множества элементов в Python используется список list, который может быть записан как индексированная последовательность значений, разделенных запятыми, заключенная в квадратные скобки. Списки имеют произвольную вложенность, т. е. могут включать в себя любые вложенные списки. Физически список представляет собой массив указателей (адресов) на его элементы.

### Задача: найти заданный элемент в списке

### 

Рис. 5. Решение задачи поиска заданного числа в массиве

### В цикле мы сравниваем каждый элемент списка с эталоном (point). Если нашлось совпадение, то мы выводим соответствующее сообщение и выходим из цикла. Если элементы списка перебраны, но совпадений так и не было, срабатывает else и выполняется команда print («не содержит»). Комментарий к инструкции point = int(input(«какое число требуется найти?»)): переменной point присваиваем значение, которое ввел пользователь в ответ на запрос «какое число…?», при этом помним, что все данные, поступившие от пользователя символьные! Применяем функцию int для преобразования данных в целочисленный тип.

### Задача: найти максимальный (минимальный) элемент в списке

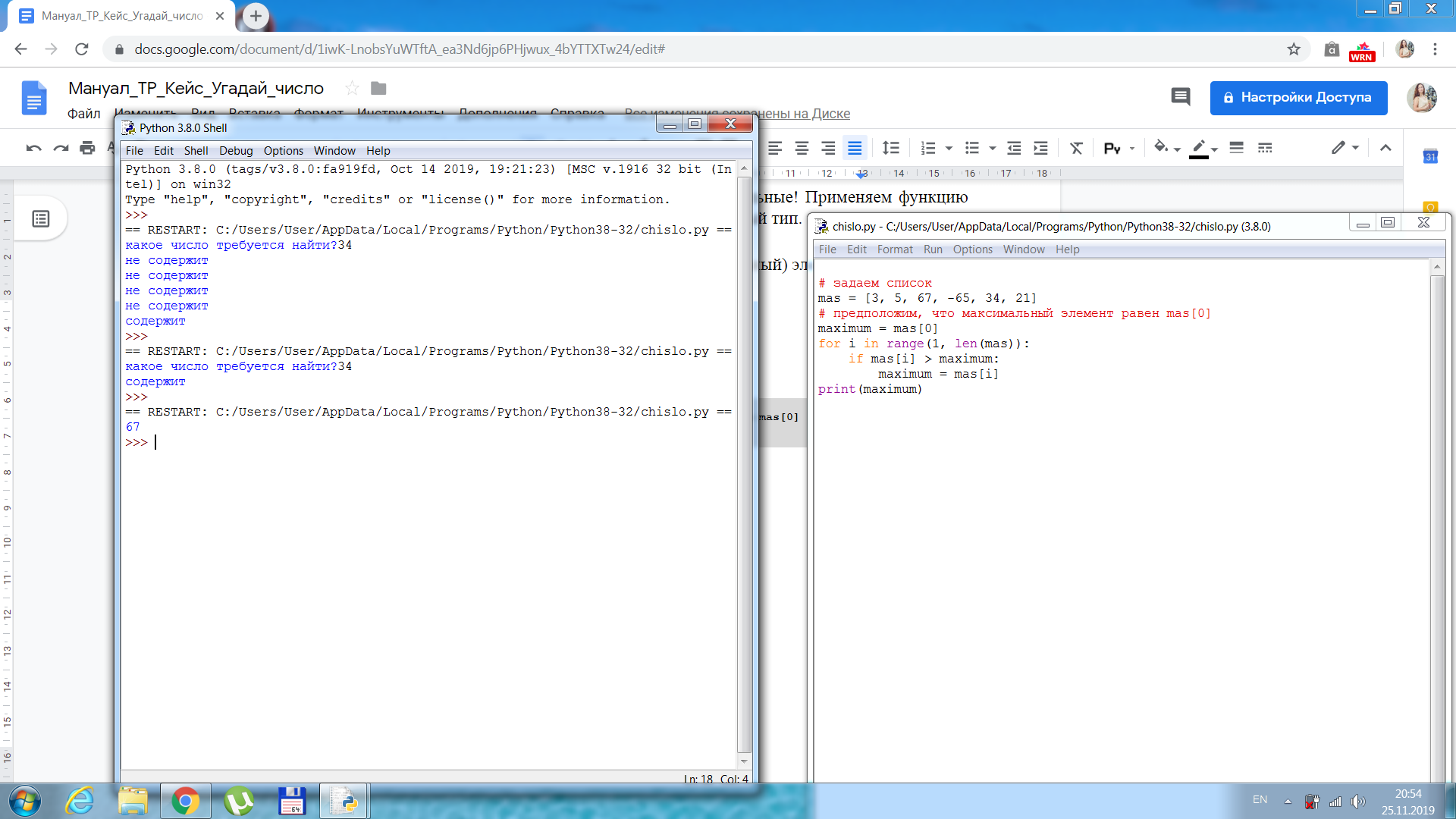


Рис. 6. Решение задачи поиска максимального(минимального) числа в массиве

### Для поиска минимального элемента списка необходимо заменить знак на < (меньше). Что происходит в цикле for? Перебираются все элементы по порядку. Если найдутся элементы списка, большие (меньшие) mas[0] – значение максимума (минимума) обновится. В противном случае значение максимума (минимума) останется равным mas[0].

Потренируйтесь в нахождении элементов с заданными свойствами.

Задача: вывести номера элементов списка, которые больше своих соседей

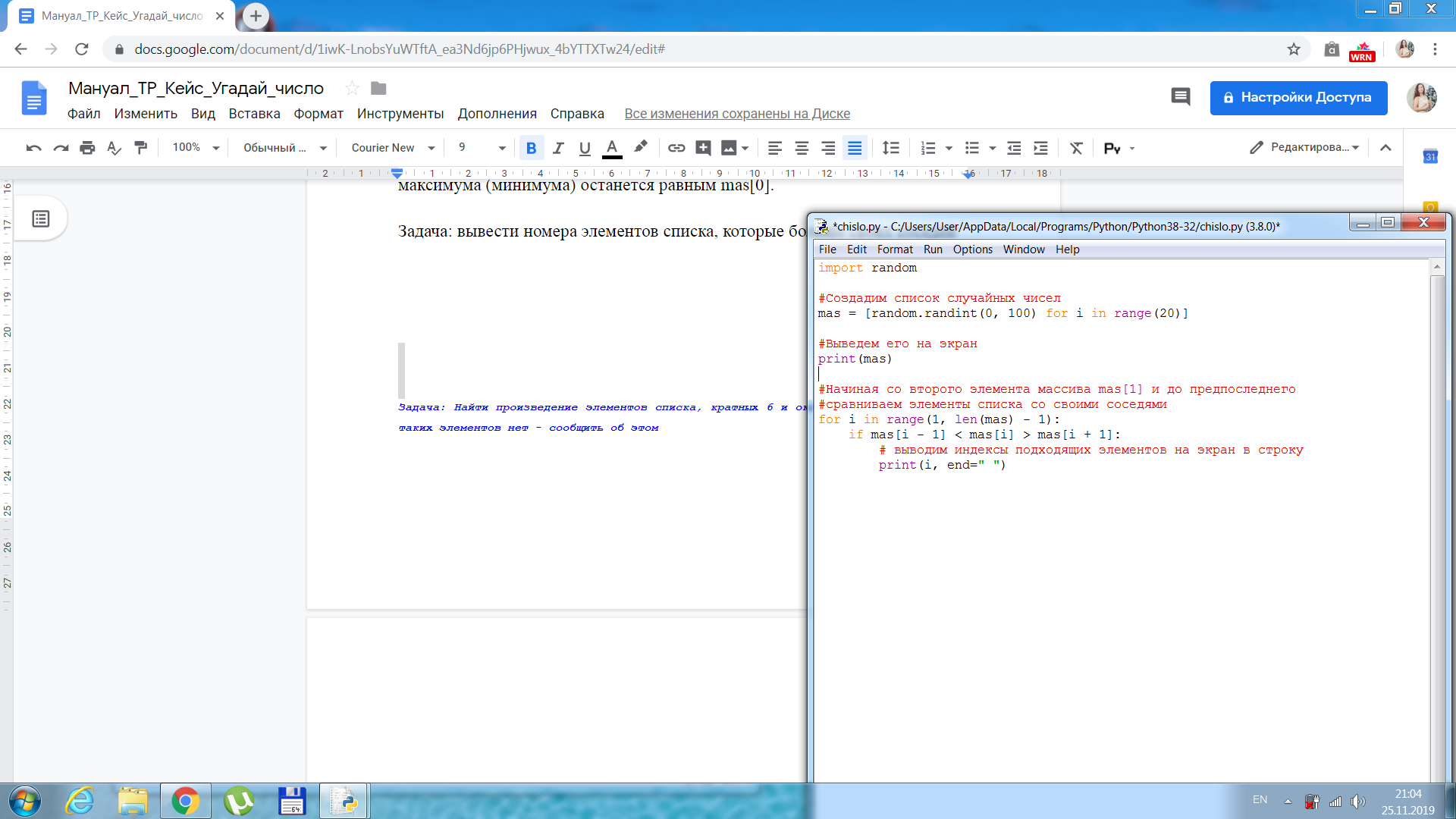


Рис. 7. Решение задачи поиска элементов списка, которые больше своих соседей

Модуль random, подключенный с помощью команды import, позволяет воспользоваться встроенной функцией генерации случайных чисел. В данном случае генерируется список из 20 элементов, числа выбираются в диапазоне от 0 до 100.

# **Двоичный поиск**

Описание алгоритма.

1. Находится средний элемент последовательности. Для этого первый и последний индексы связываются с переменными, а индекс среднего элемента вычисляется.

2. Значение среднего элемента сравнивается с искомым значением. В зависимости от того, больше оно или меньше значения среднего элемента, дальнейший поиск будет происходить только в левой или только в правой половинах массива. Если значение среднего элемента оказывается равным искомому, поиск завершается.

3. Иначе одна из границ исследуемой последовательности сдвигается. Если искомое значение больше значения среднего элемента, то нижняя граница сдвигается за средний элемент на один элемент справа. Если искомое значение меньше значения среднего элемента, то верхняя граница сдвигается на элемент перед средним.

4. Снова находится средний элемент теперь уже в выбранной половине. Описанный выше алгоритм повторяется для данного среза.

Двоичный поиск – очень мощный метод. Если, например, длина массива равна 1023, после первого сравнения область сужается до 511 элементов, а после второй – до 255. Легко подсчитать, что для поиска в массиве из 1023 элементов достаточно 10 сравнений.

Алгоритм поиска перебором должен просмотреть весь список в поисках данного элемента, но если исходный массив уже отсортирован, то элемент в нем можно найти гораздо быстрее, если воспользоваться идеей двоичного (бинарного) поиска.

**Решение кейса**

Как быстро найти загаданное число?

Знание о том, больше или меньше загаданное число нашей попытки, очень сильно облегчает задачу. Мы можем взять число из середины последовательности. Если не угадаем, у нас тогда останется половина вариантов, это верно. Но мы избавимся и от целой другой половины! В случае 10 это не так важно, но если нам загадали число от 1 до 100, мы даже неправильным предположением убираем 50 вариантов!

С оставшейся половиной можно проделать то же самое.

К примеру, загадали число от 1 до 100 (это число 43):

– Это число 50?

– Меньше.

– Это число 25?

– Больше.

– Это число 37?

– Больше.

– Это число 43?

– Верно!

Мы справились всего за 4 шага! Ведь в нашем алгоритме мы каждый раз делим оставшийся интервал на 2.

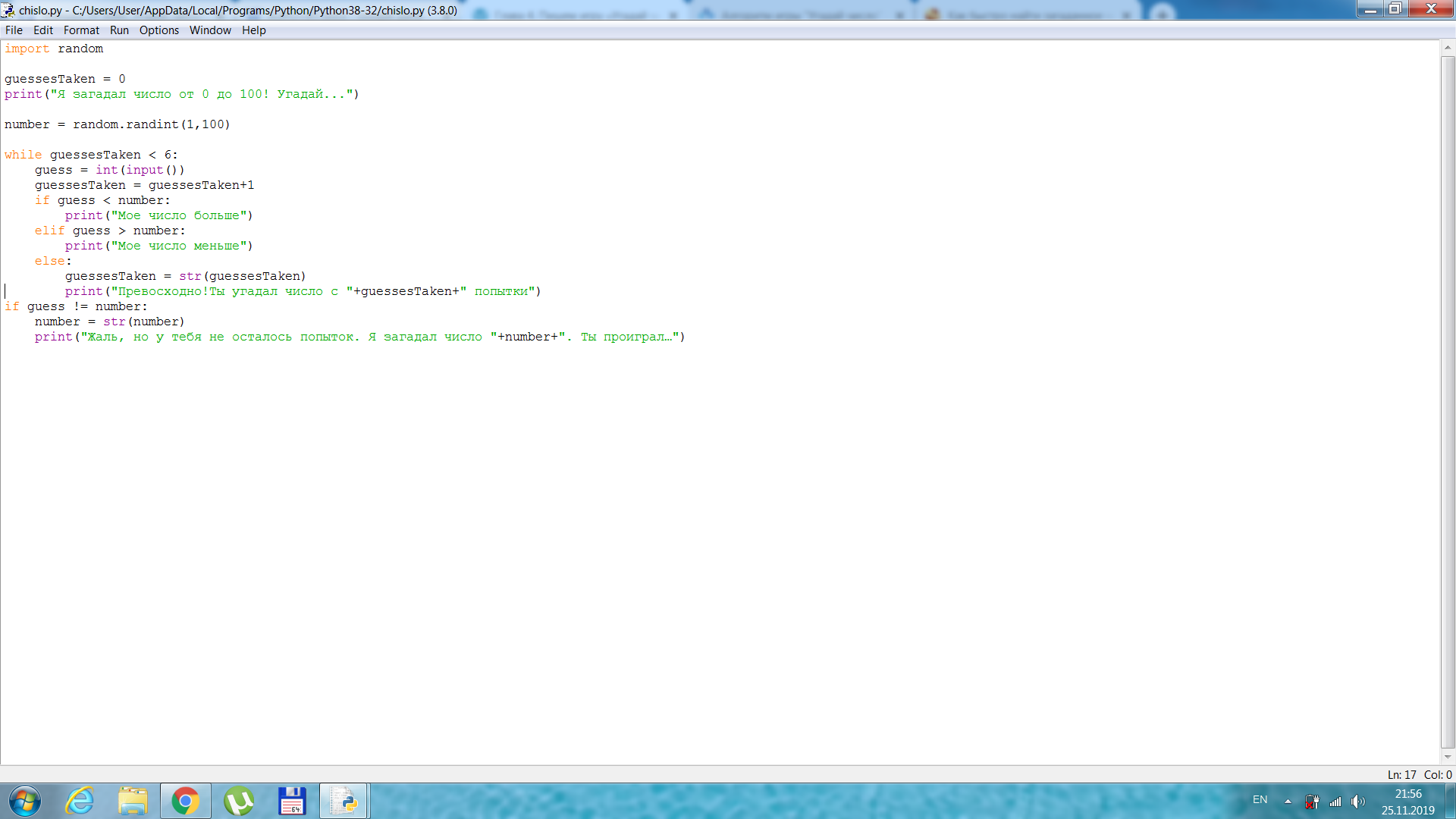


Рис. 8. Программирование игры «Угадай число»

Комментарии.

1. Импорт модуля random для использования функции генерации случайных чисел.
2. Вводим переменную для количества попыток угадать число, инициируем пока 0, но далее в цикле while принимаем количество попыток 6 (можно менять).
3. В переменную number записываем случайное число.
4. Тело цикла while работает таким образом, что введенное число от пользователя сравнивается с загаданным, сообщается больше оно или меньше, увеличивается счетчик попыток; если угадано, то идет сообщение об этом.
5. Цикл прекращается тогда, когда закончились попытки или угадано число; если израсходованы попытки, то последняя команда – это сообщение о проигрыше.