**Кейс 3. Калькулятор**

В этом руководстве подробно рассмотрен процесс решения задачи разработки собственного программного продукта типа «Калькулятор», со своим собственным интерфейсом.

**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.

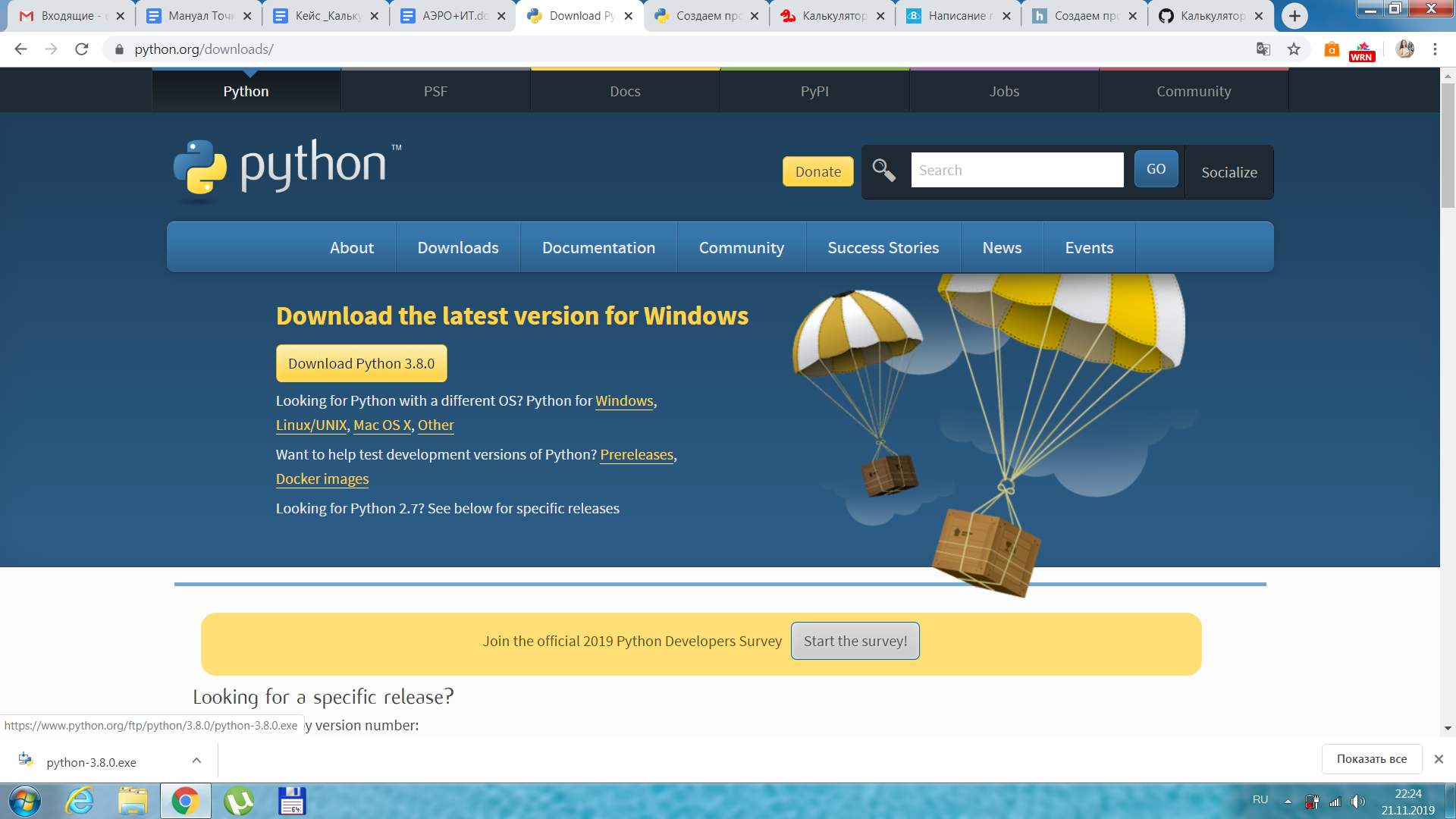


Рис. 1. Официальный сайт для установки среды разработки

Далее необходимо запустить исполняемый файл для установки.



Рис. 2. Начало установки Python

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

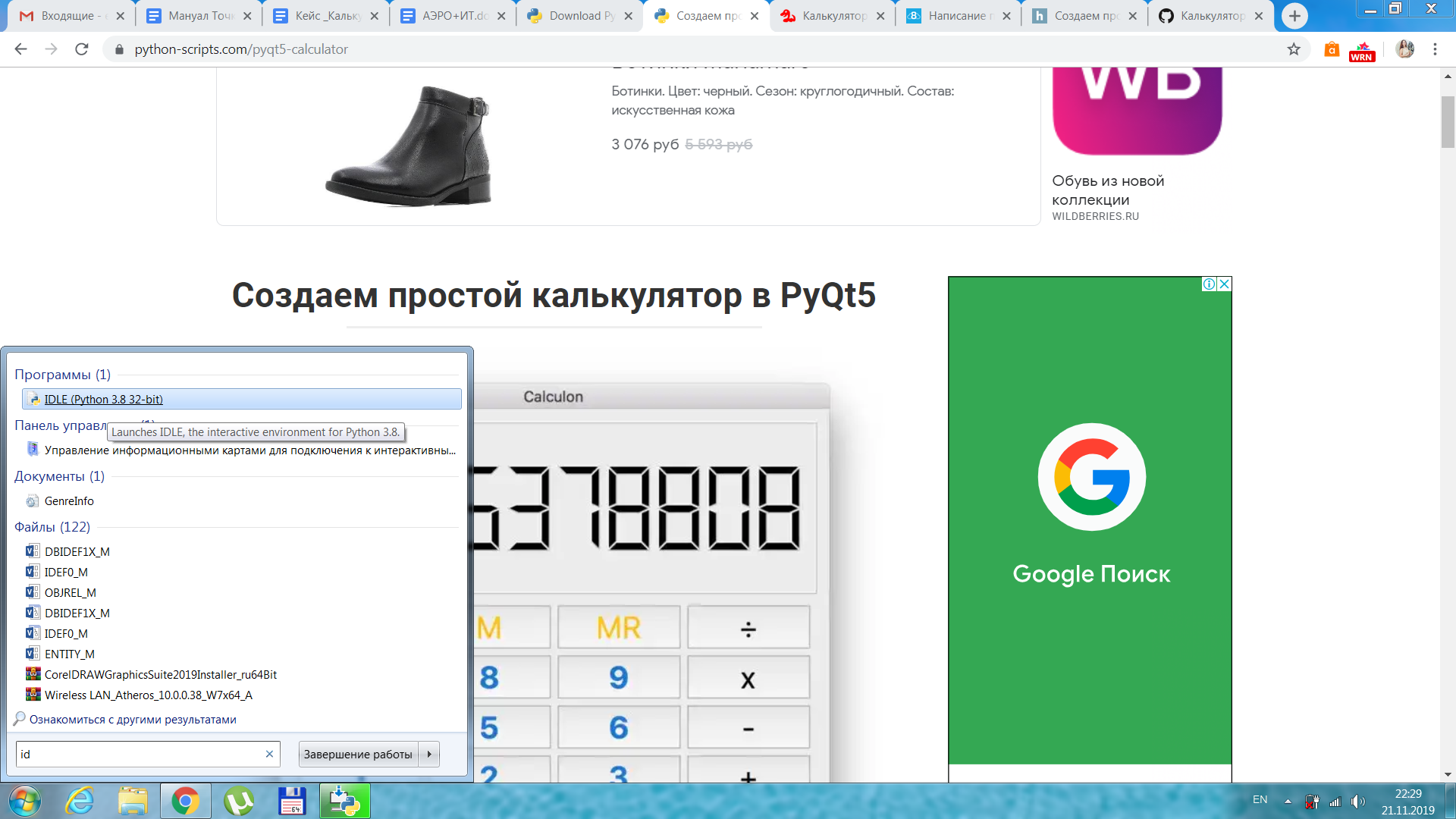


Рис. 3. Где найти IDLE Python

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

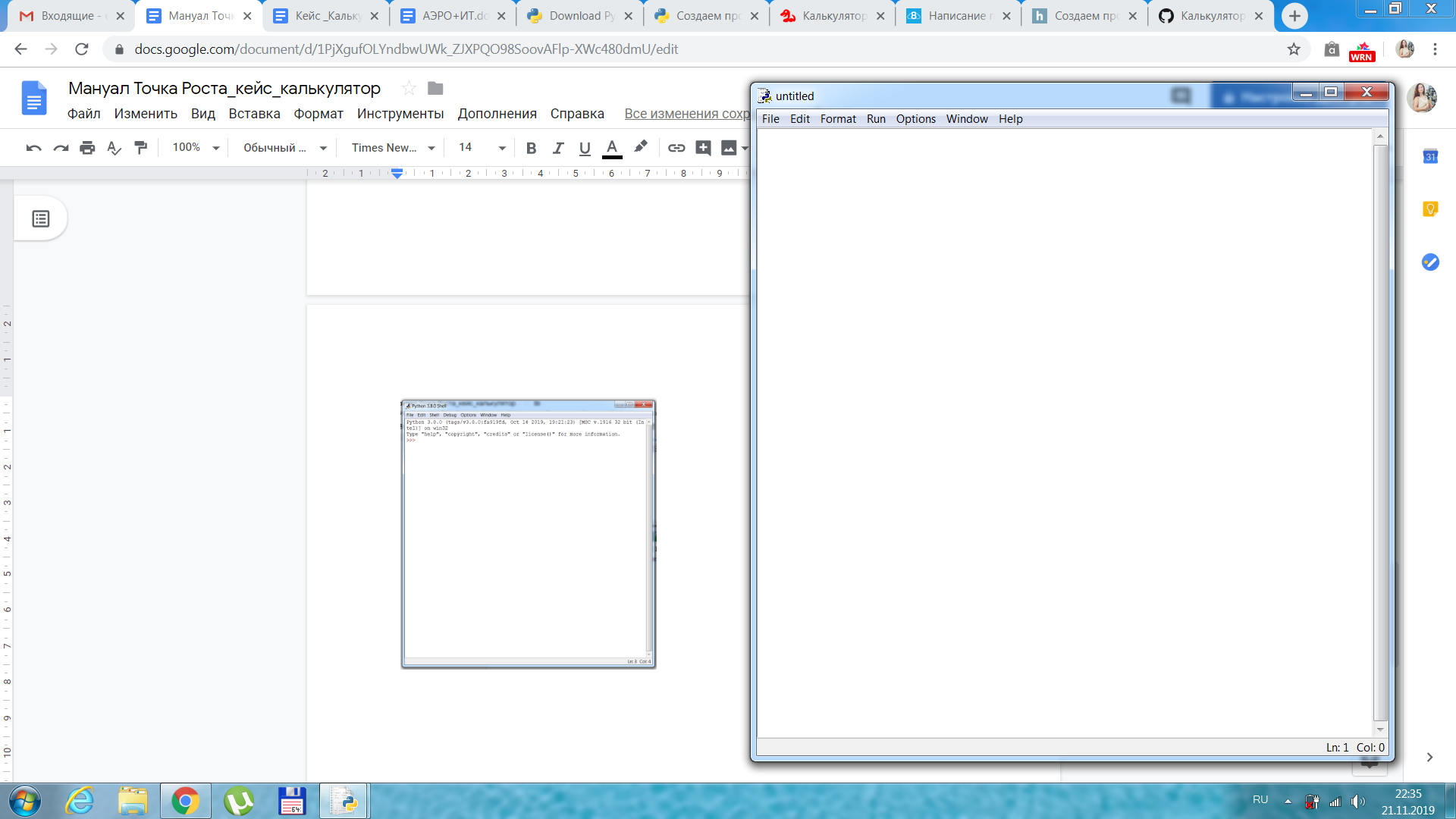
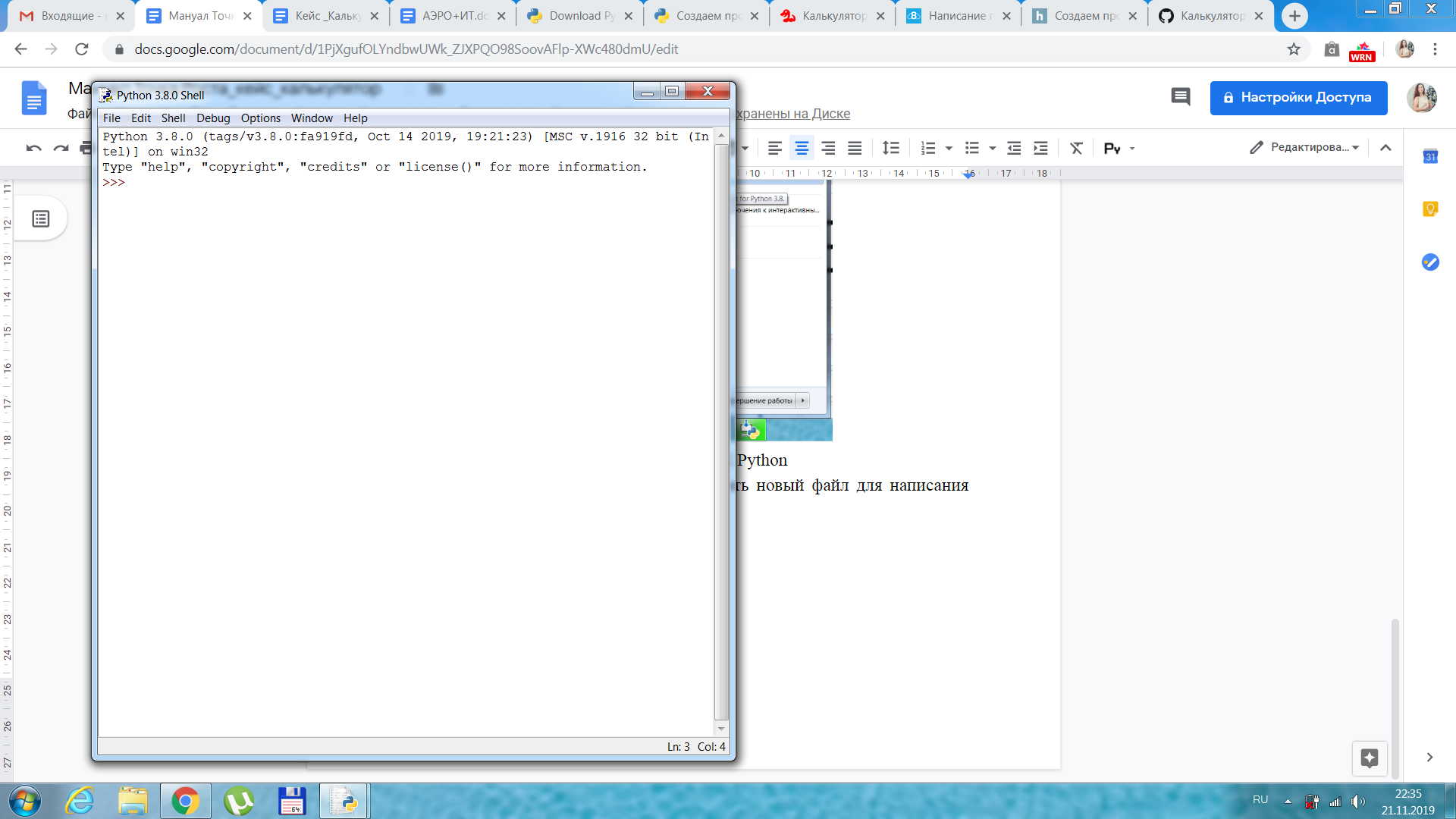


Рис. 4. Запуск интерпретатора и текстового редактора кода IDLE Python

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

**Написание программной части для калькулятора**

1. Строка ввода операндов

Для начала нужно написать строку ввода, с помощью которой пользователи смогут вводить данные для вычислений в калькуляторе. Для этого используйте встроенную функцию input(), которая принимает сгенерированный пользователем ввод с клавиатуры. В круглых скобках функции input() можно передать строку – инструкцию пользователю.

Чтобы программа могла выполнять математические вычисления, она не должна принимать никаких данных, кроме чисел. Функцию input() нужно передать внутри функции int(), чтобы преобразовать ввод в целое число.



Рис. 5. Ввод операндов

Протестируйте введенный код Run/Run Module (F5), убедитесь, что, если будут введены символы, пробелы или буквы, программа вернет ошибку.

1. Добавление операторов

Передайте строку внутри функции input() и присвойте переменной значение ввода (к примеру, это будет переменная operation).

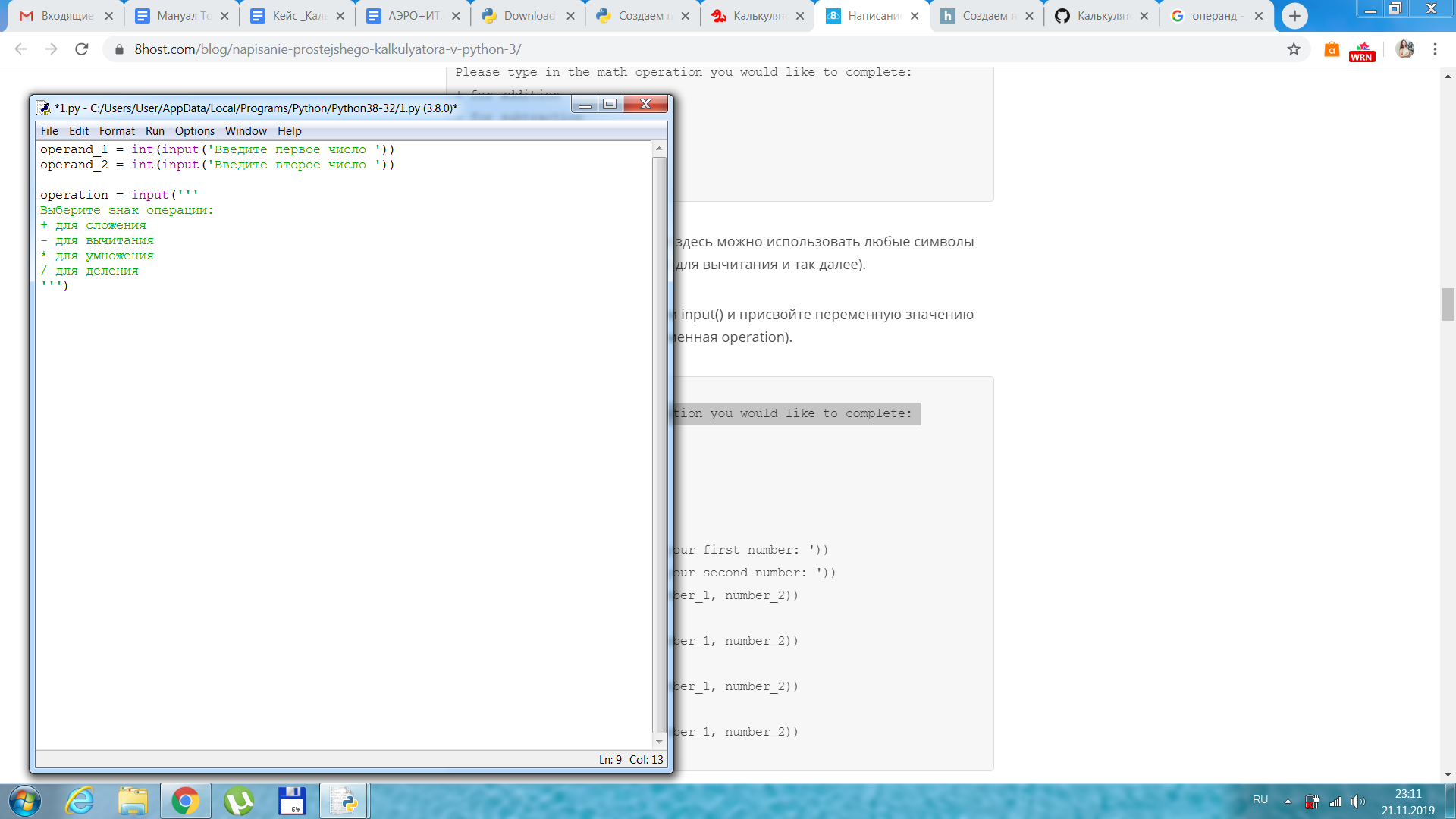


Рис. 6. Выбор операции

1. Условный оператор

Чтобы программа работала, нужно добавить условный оператор if.

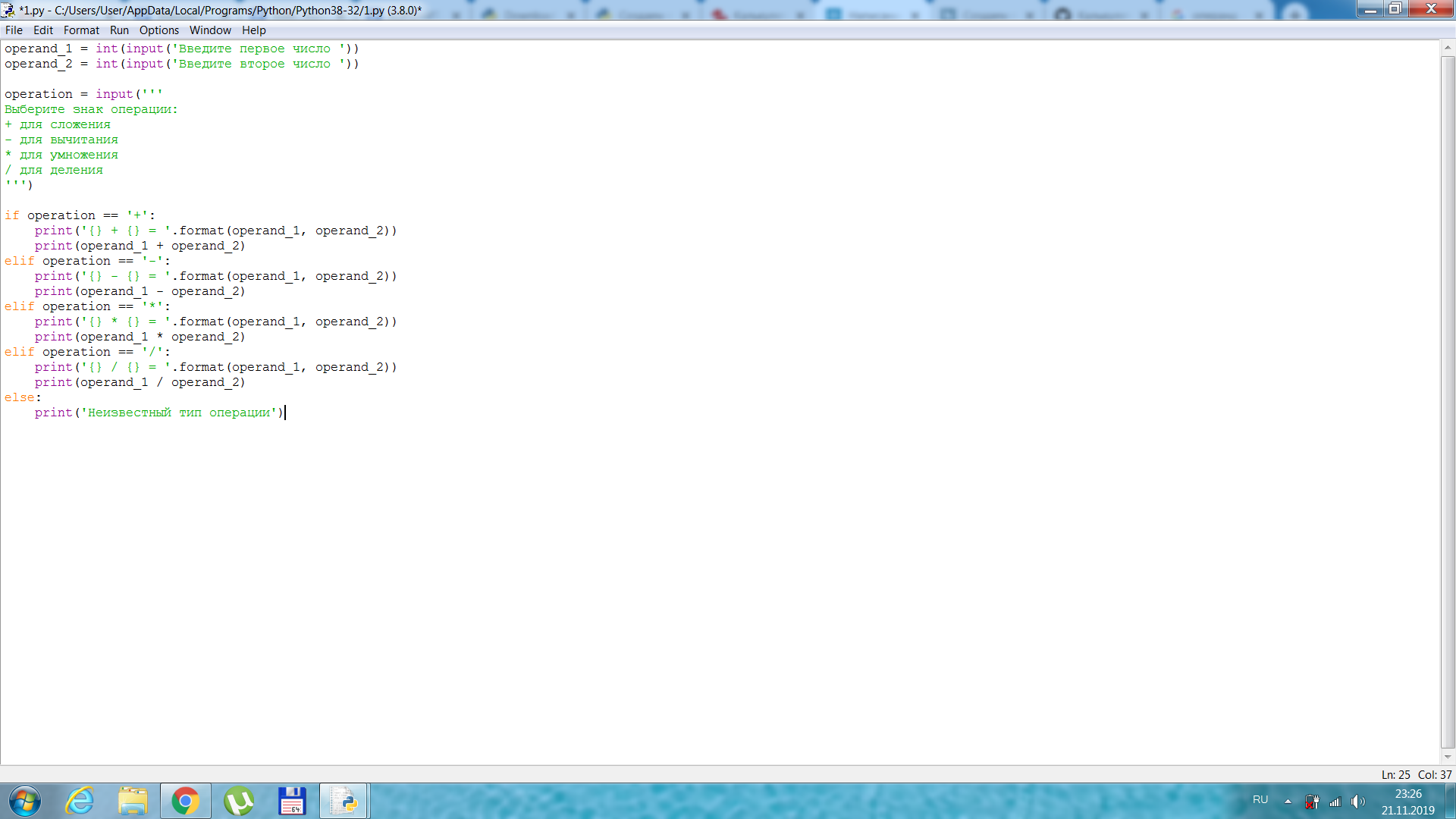


Рис. 7. Добавление условного оператора

Инструкция print('{} + {} = '.format(operand\_1, operand\_2)) отвечает за форматирование ввода выражения, чтобы пользователь убедился в правильности ввода данных.

Инструкция print(operand\_1 + operand\_2) отвечает за вывод результата операции.

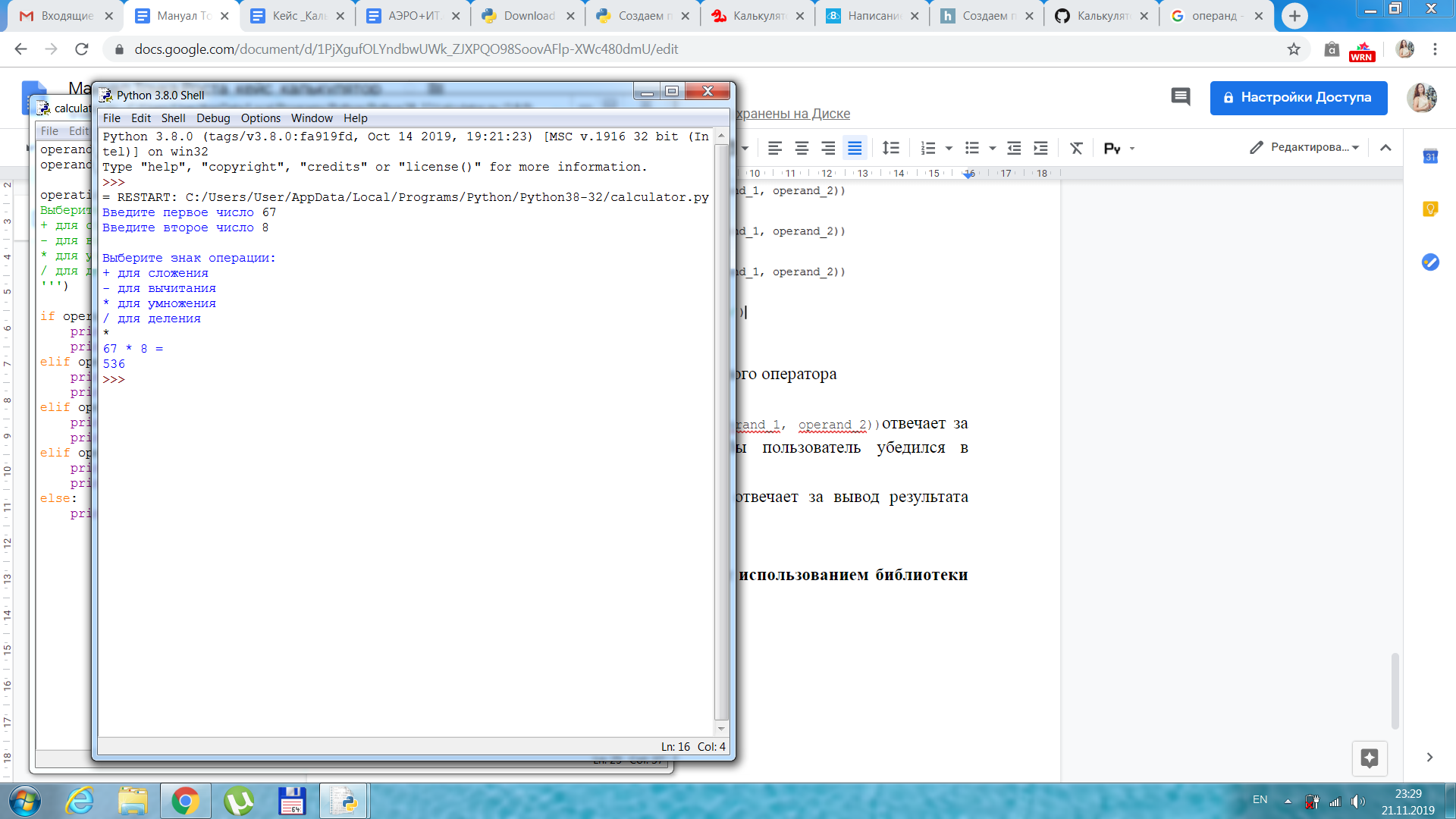


Рис. 8. Результат выполнения программы

1. Определение функции

Определим функцию calculate() с помощью инструкции def – чтобы программа запускалась, нужно добавить функцию в конец файла.

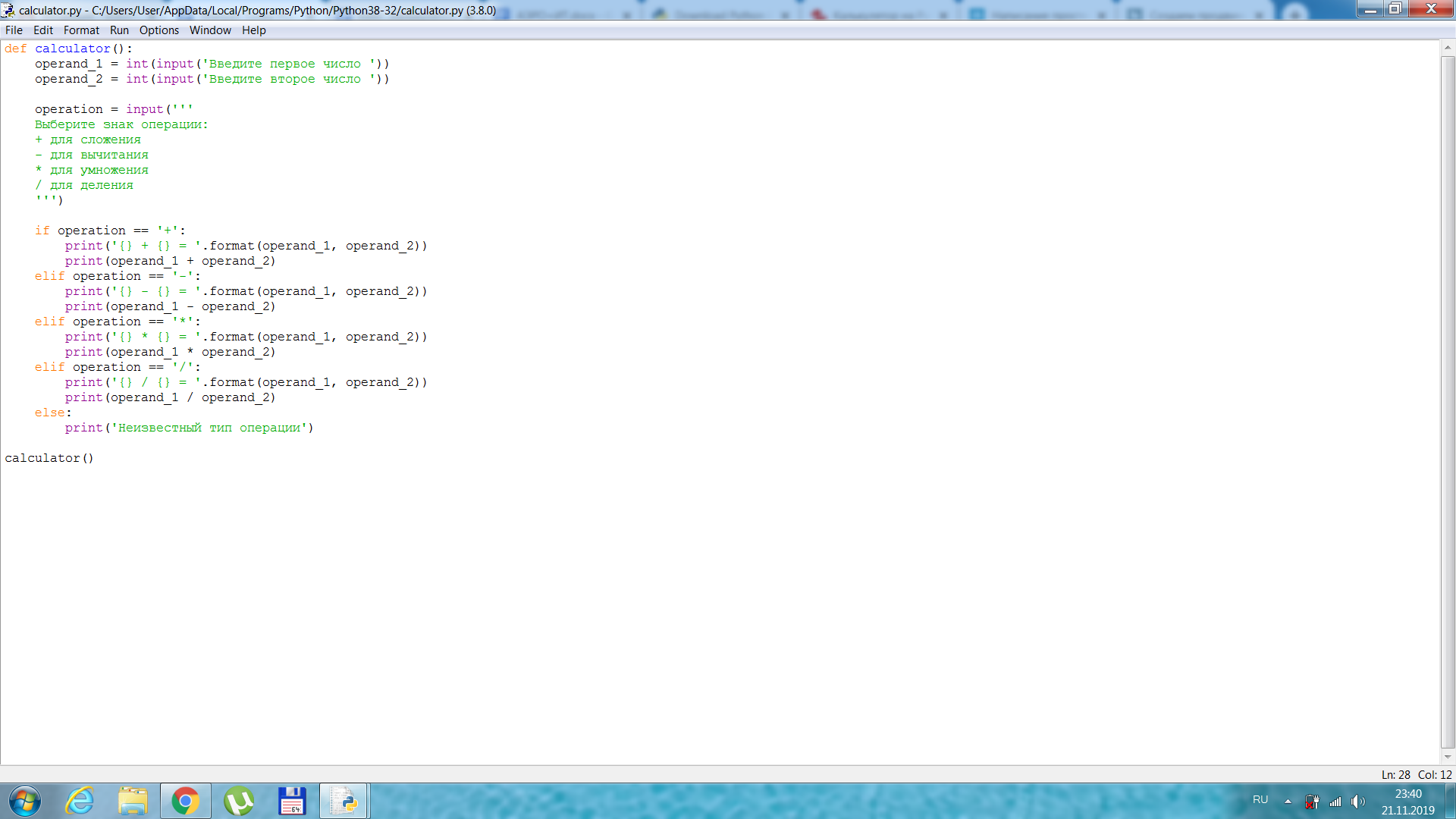


Рис. 9. Итоговый код программы

**Создание внешнего вида калькулятора с использованием библиотеки Tkinter**

Для создания внешнего вида импортируем модули. Библиотека Tkinter необходима для создания «окна» – пользовательского интерфейса программы. Модуль math понадобится для математических операций. Модуль sys обеспечивает доступ к некоторым переменным и функциям, взаимодействующим с интерпретатором Python.

Создаем окно приложения – объект Tk с заголовком Calculator. В [кортеже](https://all-python.ru/osnovy/kortezh.html) buttons будут храниться обозначения для кнопок. В [список](https://all-python.ru/osnovy/spiski.html) stack будут добавляться введенные числа и операции, которые надо совершить. Переменная activeStr предназначена для хранения набираемого числа.

Создаем надпись для вывода набираемых значений и результатов. В цикле создаем кнопки. Расположение кнопок и надписей осуществляется в табличном виде с помощью упаковщика grid. И в завершение запускаем цикл обработки событий mainloop. У надписи выставлена ширина 35, для того чтобы оформление кнопок подстраивалось под надпись. И в результате кнопки при этом значении лучше выглядят.

Для того чтобы кнопки правильно работали, пришлось для каждой из кнопок создать свою функцию с помощью lambda (о функции lambda: https://pythonworld.ru/tipy-dannyx-v-python/vse-o-funkciyax-i-ix-argumentax.html).

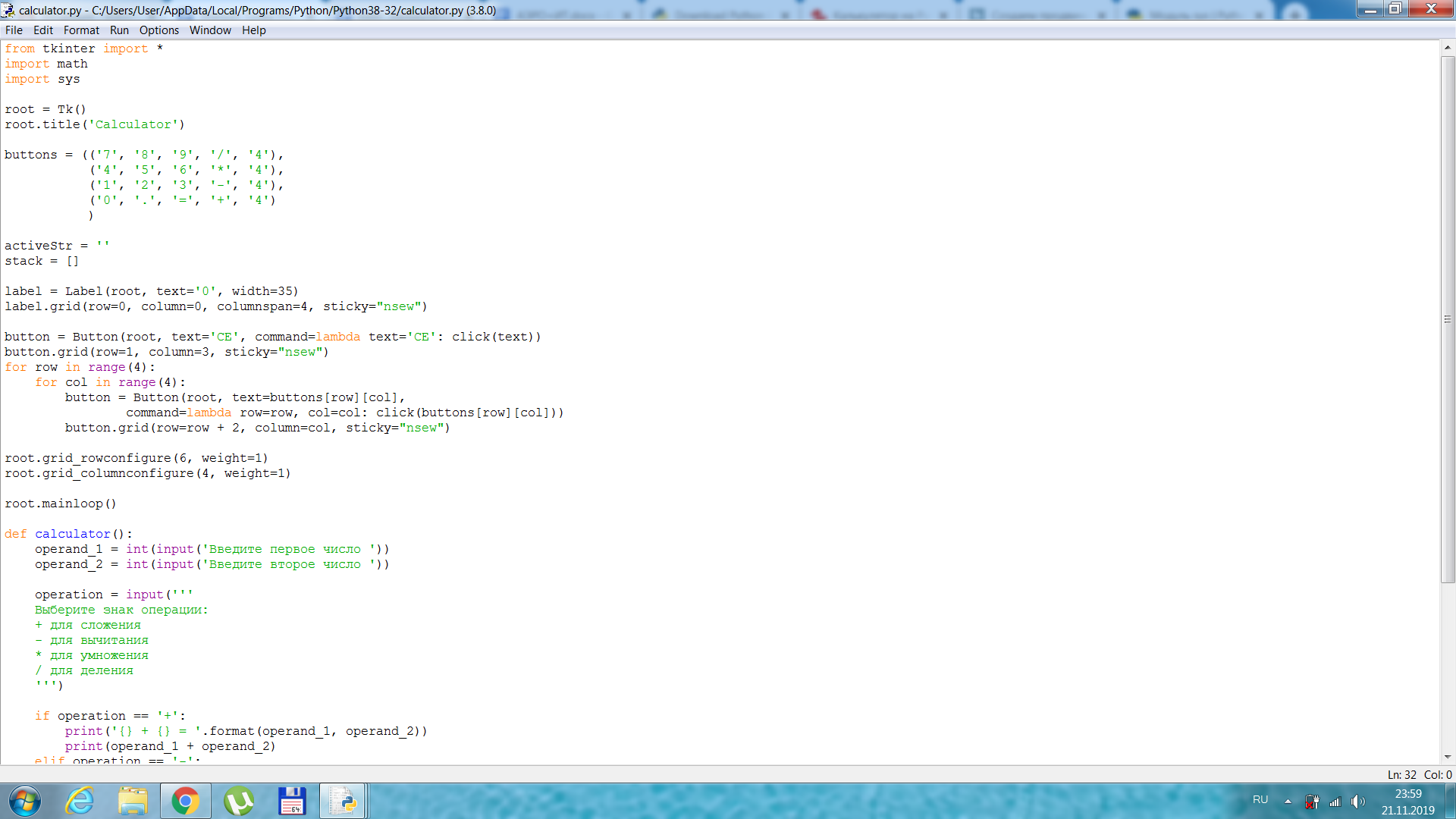


Рис. 10. Импорт библиотек и программирование внешнего вида

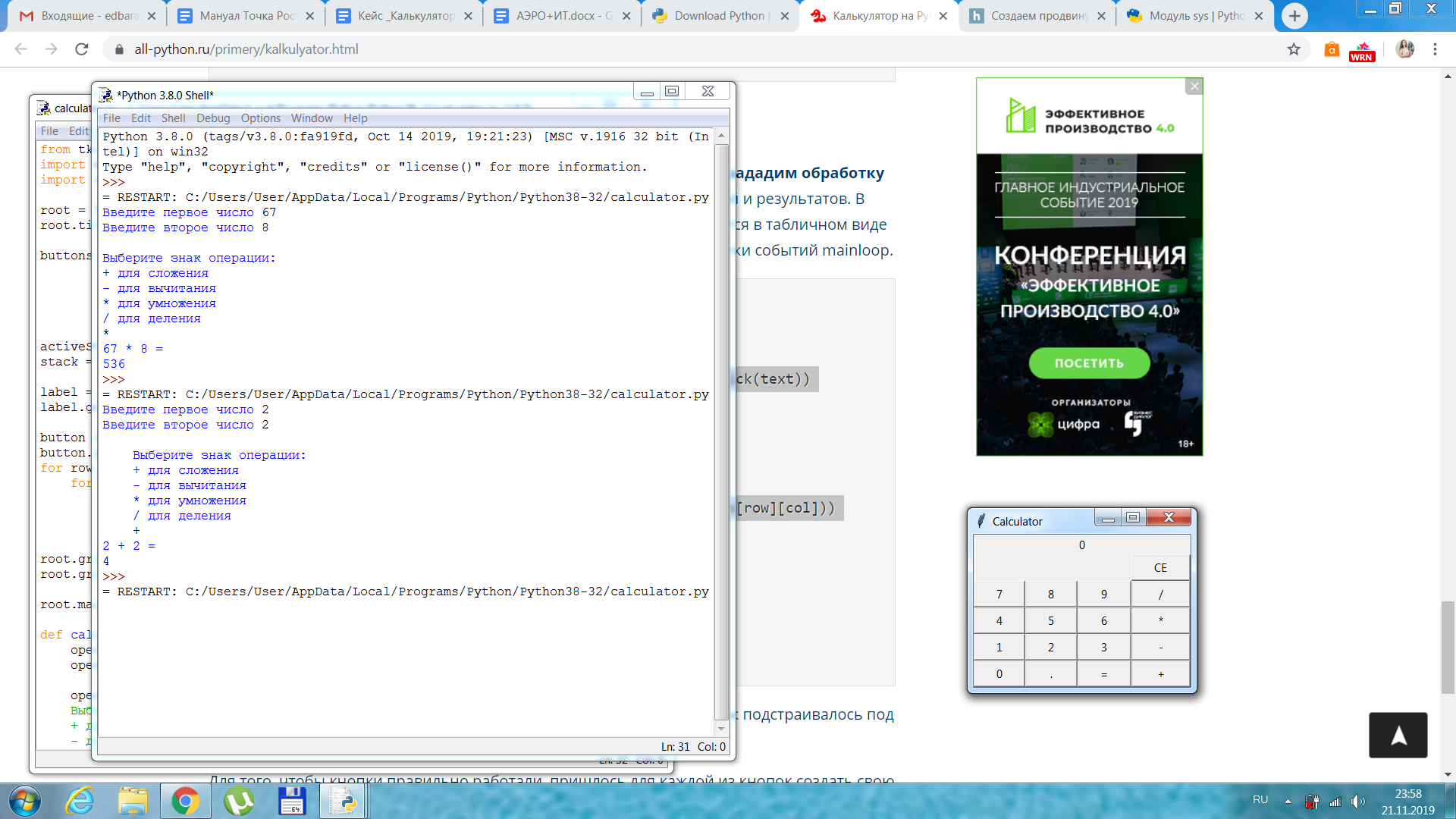


Рис. 11. Результат выполнения

Чтобы запрограммировать вычисление результата, определим функцию calculate(), которая получает из списка stack операнды и операцию. Результат отображается в надписи label. Получать из списка строки будем с помощью метода pop.

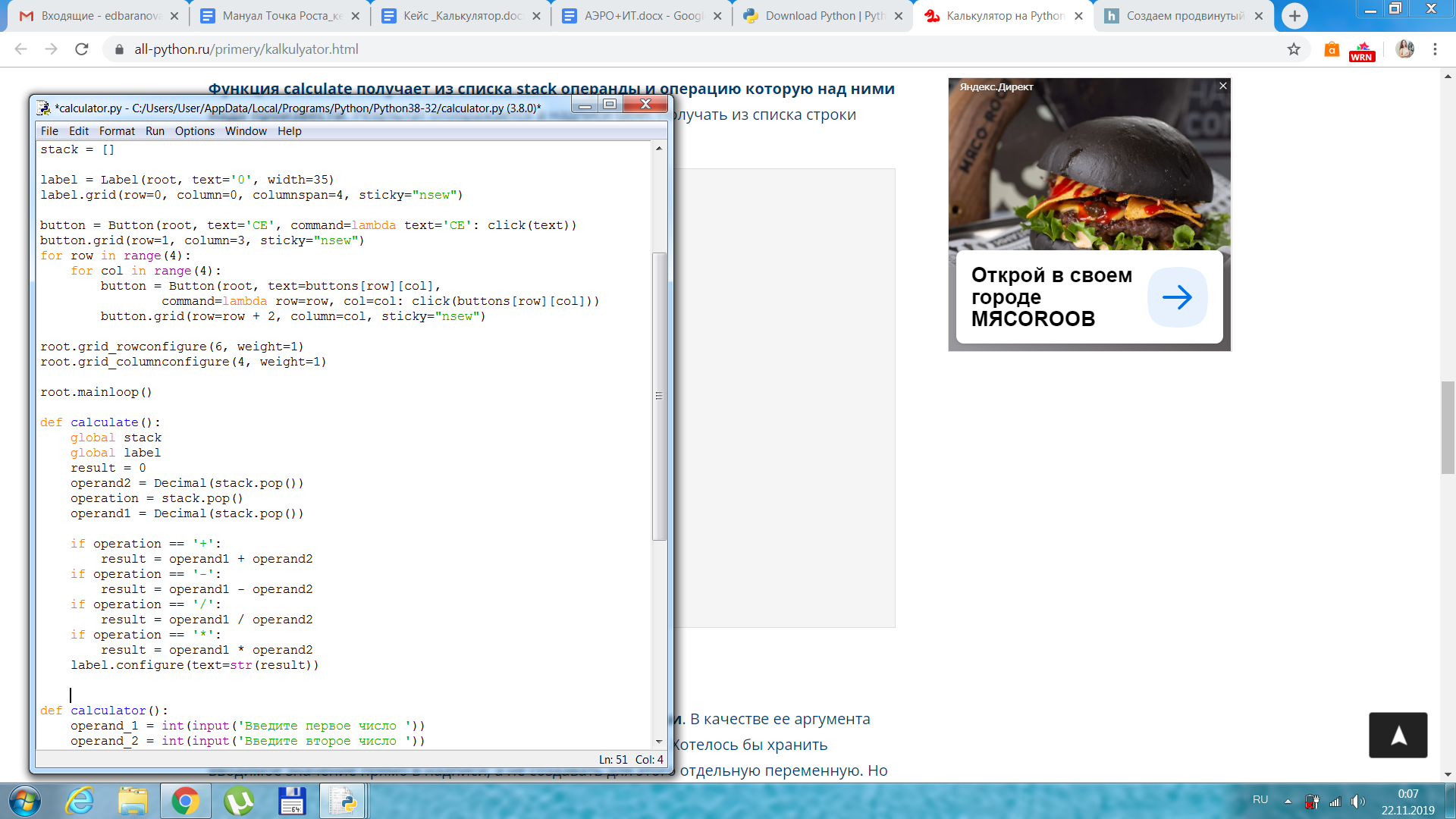


Рис. 12. Вычисление результата

Обработку нажатия кнопок созданного калькулятора определим в функции click. В качестве ее аргумента передается текст, отображаемый на кнопке, которую нажали.

Основу функции составляет перебор вариантов ввода. При вводе цифр (в том числе дробных) заполняется label, иначе – ожидаем ввода операции и вызываем функцию calculate().

В списке с операторами и командами для калькулятора не обязательно их будет три. Но при обработке с помощью метода pop будут рассматриваться три последних введенных значения. А после проведения расчета список очищается. Далее в него добавляется полученный результат на случай, если пользователь нажмет на калькуляторе клавишу операции сразу, а не будет вводить новое число.

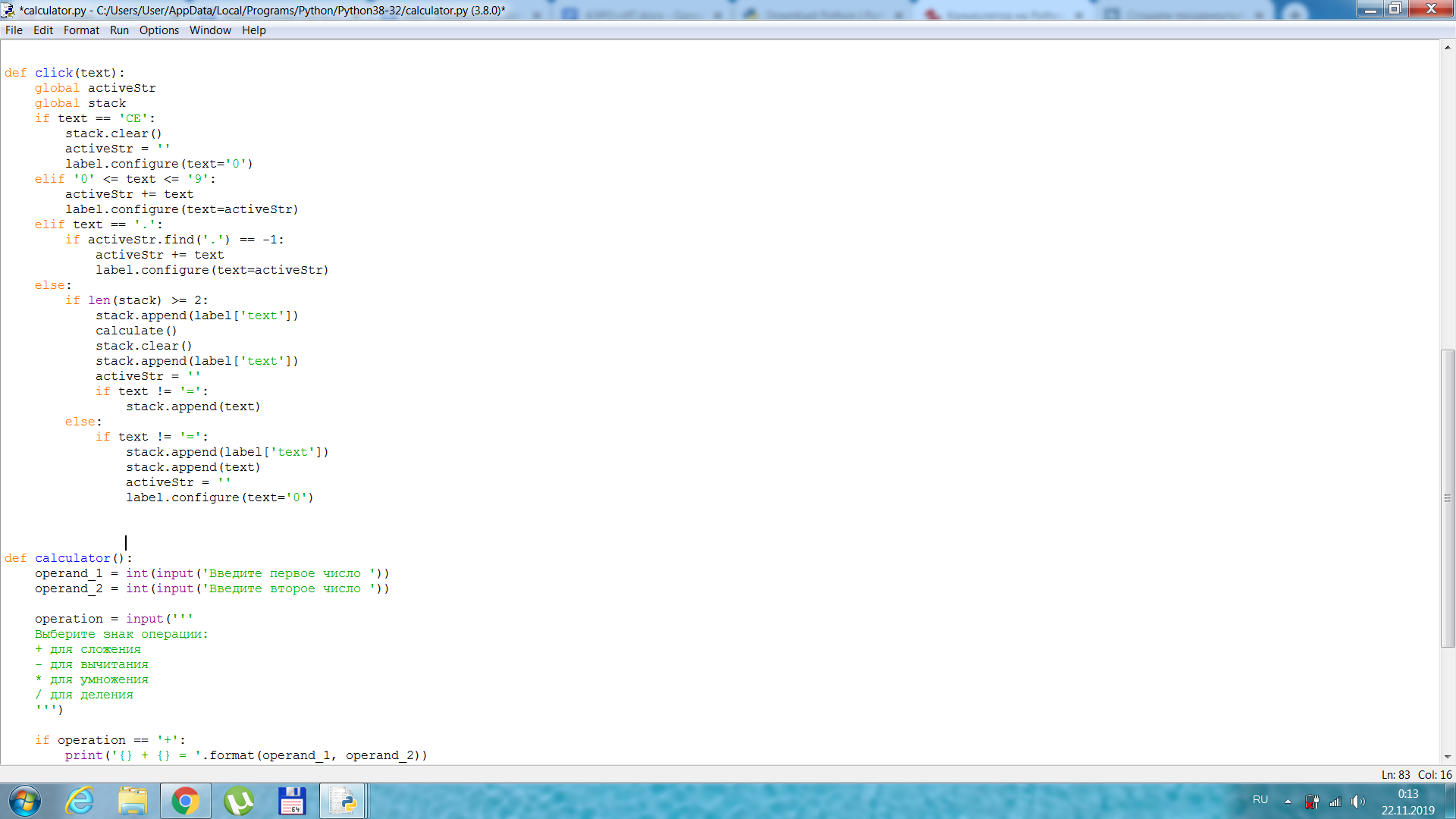


Рис. 13. Функция обработки нажатия кнопки

По окончании программирования протестируйте работу программы.