

## Модуль 1. Арифметические операции в среде Python<sup>1</sup>

### 1. Как начать работу со средой Python

Сначала подготовимся к работе.

#### Задание 1

Попытайтесь на диске найти папку **Рабочая**. Если таковой нет, создайте ее. В папке **Рабочая** создайте папку **Python**, а в ней – папку с *Вашей фамилией*. Далее эту папку будем называть «*Вашей папкой*».

Из папки **Комплект для студентов Python\Examples** в свою папку скопируйте все имеющиеся там файлы.

☞ Последнюю версию среды программирования Python можно скачать с сайта по адресу <http://python.org/download/>.

### 2. Запуск среды

#### Задание 2

Запустите среду программирования Python (Пуск → Все программы → Python 3.3 → IDLE (Python GUI)).

Если среда запустилась, перед Вами появится окно (рис. 1).

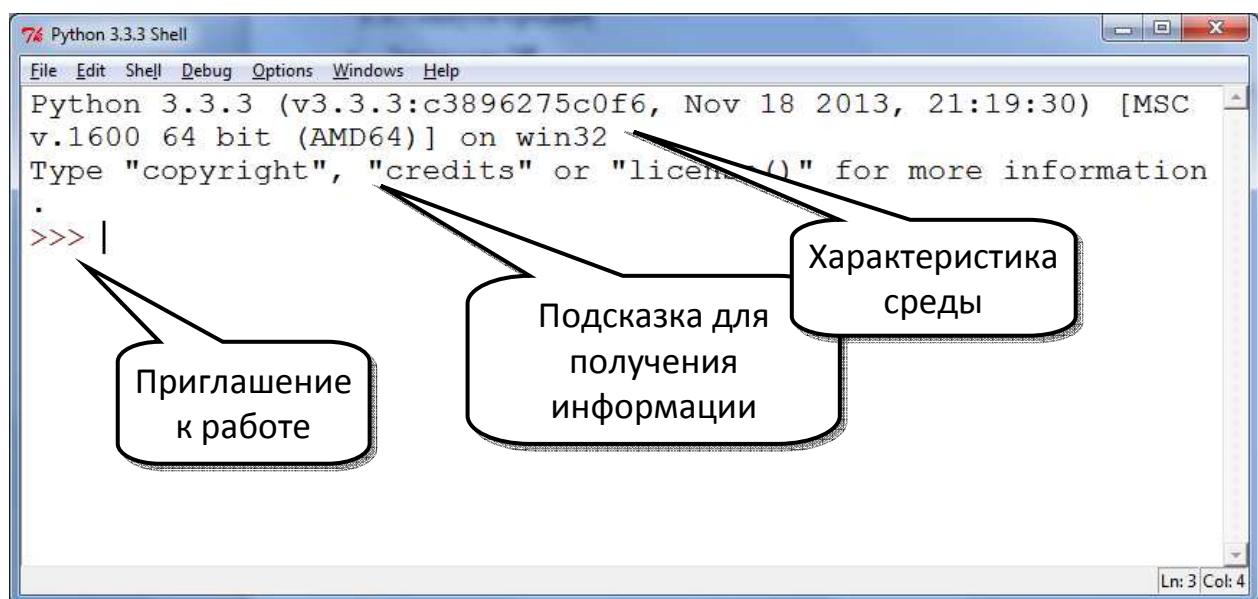


Рис. 1. Структура окна среды Python 3.3.3

### 3. Арифметические операции

В языке Python к простейшим типам данных относят числа и строки.

Числа могут быть целыми (положительными и отрицательными: **123**, **-24**) и вещественными. В вещественных числах вместо десятичной запятой используется десятичная точка: **3.14** **-234.55**

<sup>1</sup> В данном комплексе материалов рассматривается работа со средой Python 3.3.4.

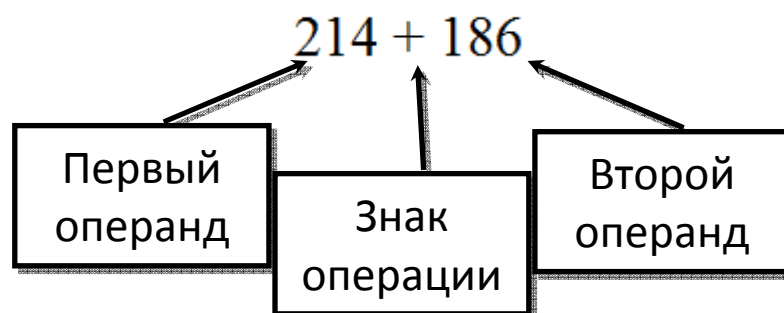
С числами возможны следующие операции:

Таблица 1

Арифметические операции	
Знак операции	Название операции
+	Сумма двух операндов
–	Разность двух операндов
*	Произведение двух операндов
/	Деление двух операндов
//	Целая часть от деления двух операндов
%	Остаток от деления
**	Возведение в степень

Арифметические операции используются в арифметических выражениях. Операции сложения и вычитания могут быть унарными (т.е. там может быть один операнд) и бинарными. Например:



Унарные операции : +23      -17      +34.678  
 Бинарные операции : 56+24      74–24      118 % 33



Все остальные операции являются бинарными.

#### 4. Арифметические операции: практикум

Среда программирования Python представляет интерпретатор.

-  **Интерпретатор** – программа (разновидность транслятора) или аппаратное средство, выполняющее *интерпретацию*<sup>2</sup>.
-  **Интерпретация** – пооператорный (покомандный, построчный) анализ, обработка и тут же выполнение исходной программы или запроса (в отличие от компиляции, при которой программа транслируется без её выполнения)<sup>2</sup>.

<sup>2</sup> Википедия: Интерпретатор : <http://ru.wikipedia.org/wiki/интерпретатор>

Интерпретатор можно использовать как простейший калькулятор. Достаточно набрать выражение, нажать **ENTER** и в окне появится результат этого выражения. Например:

```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:1
9:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more
information.
>>> 5+7
12
>>> 5.5*3
16.5
>>> 77/7
11.0
>>>
```

### Задание 3

- 1) Наберите в строке приглашения (`>>>`) выражение `4/2` и нажмите **ENTER**; выясните, какое число получено: вещественное или целое<sup>3</sup>.
- 2) Наберите в строке приглашения выражения `5 % 2`; выясните, какое число получено: вещественное или целое.
- 3) Поупражняйтесь в вычислениях с помощью интерпретатора Python.

### Задание 4

Проведите эксперимент и найдите ответы на следующие вопросы:

Операнд 1	Операция	Операнд 2	Результат
Целое	+	Целое	(целое или вещественное?)
Целое	+	Вещественное	(целое или вещественное?)
Вещественное	+	Целое	(целое или вещественное?)
Вещественное	+	Вещественное	(целое или вещественное?)

Обобщите полученные результаты:

- а) Если хотя бы один операнд в сумме является вещественным, то тип результата –
- б) Если все операнды в сумме являются целыми, то тип результата –

<sup>3</sup> Вещественное от целого отличается наличием десятичной точки.

в) Если все операнды в сумме являются вещественными, то тип результата –

### Задание 5

Догадайтесь, какое из обобщений в предыдущем задании лишнее.

### Задание 6

Проведите эксперимент для всех остальных операций из таблицы 1. Обобщите полученные вами результаты и заполните таблицу из файла Python\_Examples\Таблица\_3.doc.

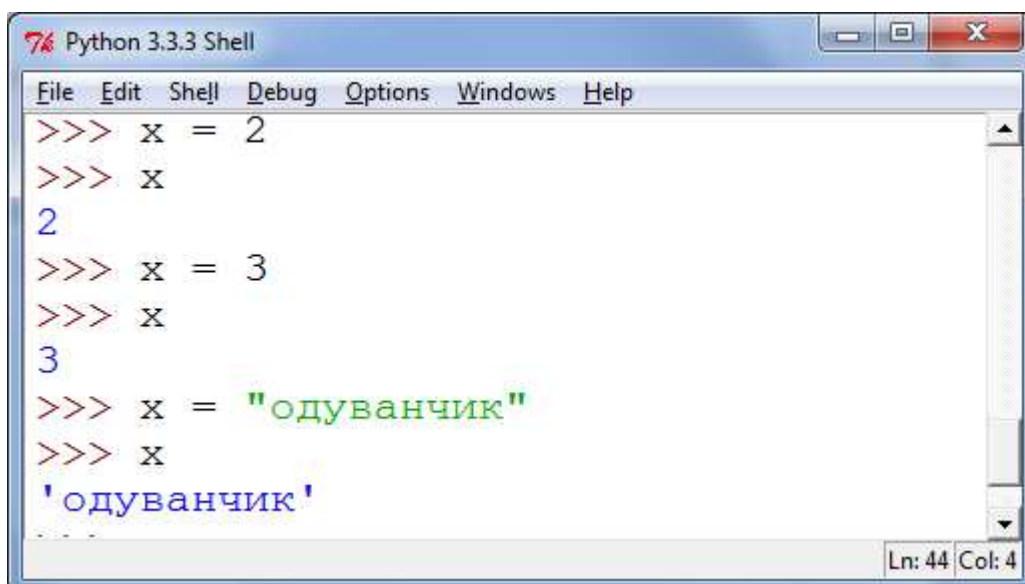
## 5. Операция присваивания

Работать с числами, конечно, здорово, но в программировании принято работать с переменными. Переменная – это блок памяти, который имеет имя и, как правило, определенный тип. Под «типом» понимают тип данных, которые может храниться в переменной.

Имя блока памяти (его часто называют «идентификатор») представляет собой последовательность символов, которая может включать буквы (как английские, так и русские), цифры и знак подчеркивания (и только эти!). Например:

```
mothers  
Mike_1999  
Лошадь_Пржевальского  
Шансон_1_2_2
```

Операция присваивания позволяет задать каждой переменной какое-то значение. Например:

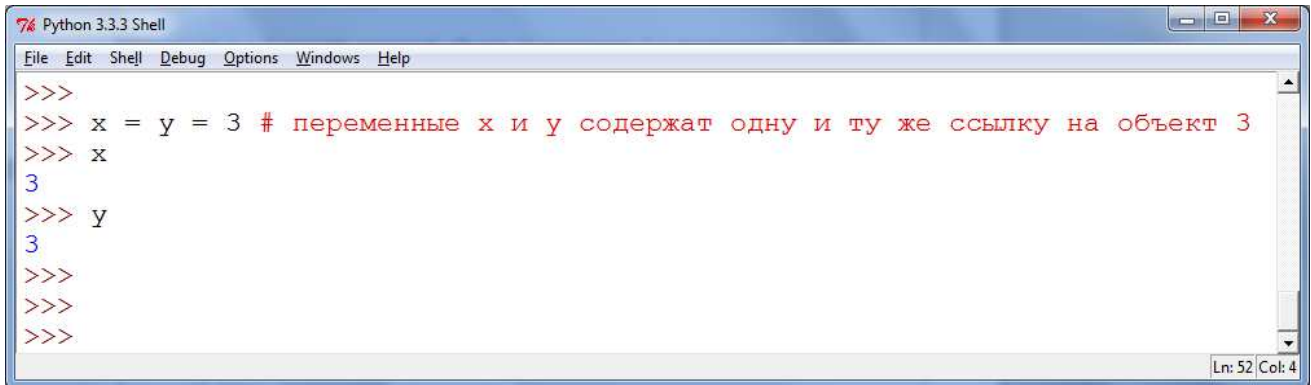


```
Python 3.3.3 Shell  
File Edit Shell Debug Options Windows Help  
>>> x = 2  
>>> x  
2  
>>> x = 3  
>>> x  
3  
>>> x = "одуванчик"  
>>> x  
'одуванчик'
```

Как видно из примера, если в строке приглашения задать просто имя переменной, то будет выведено ее значение.

Значение в кавычках (или апострофах) – это строковые данные.

Инструкция `x = 2` создаёт числовой объект со значением 2 и присваивает ссылку на этот объект переменной `x`. Далее инструкция `x = 3` создаёт новый числовой объект со значением 3 и присваивает ссылку на него переменной `x`. А объект со значением 2 удаляется из памяти автоматическим сборщиком мусора т.к. была потеряна последняя ссылка на этот объект. Благодаря этому справедливо следующее:



```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
>>>
>>> x = y = 3 # переменные x и y содержат одну и ту же ссылку на объект 3
>>> x
3
>>> y
3
>>>
>>>
>>>
```

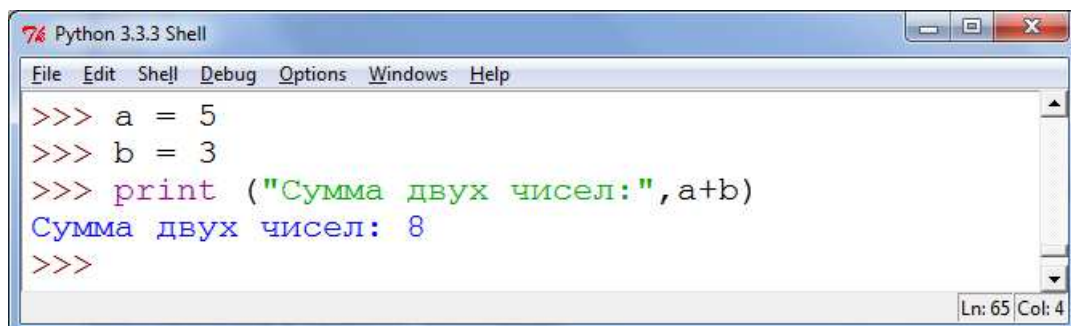
### Задание 7

Сначала устно, а затем с помощью интерпретатора Python выясните, что будет результатом следующей последовательности инструкций:

- a) `>>> x = 777`  
`>>> x = x + 223`  
`>>> x = x % 255`  
`>>> x`
- a) `>>> yx = 465`  
`>>> эх = yx + 535`  
`>>> эх = эх % 97`  
`>>> эх`

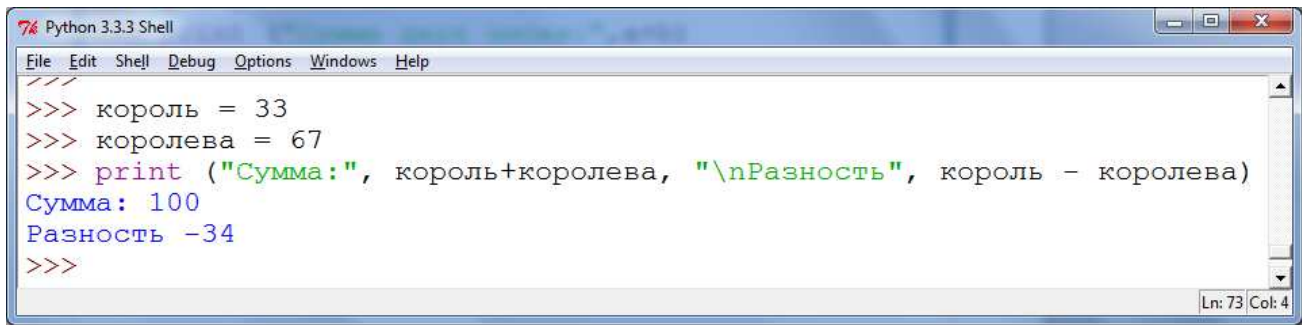
## 5. Процедура вывода `print()`

Процедура вывода обеспечивает вывод информации из списка вывода, находящегося в скобках. Например:



```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
>>> a = 5
>>> b = 3
>>> print ("Сумма двух чисел:", a+b)
Сумма двух чисел: 8
>>>
```

Если при выводе данных требуется разорвать строку, часто используется специальный управляющий набор символов `"\n"`. Например:



```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
>>> король = 33
>>> королева = 67
>>> print ("Сумма:", король+королева, "\nРазность", король - королева)
Сумма: 100
Разность -34
>>>
```

### Задание 8

С помощью интерпретатора Python выясните, что будет результатом следующей последовательности инструкций:

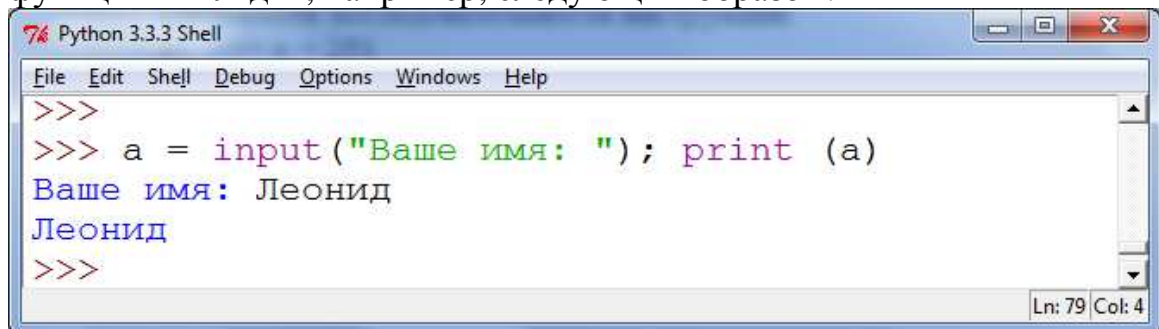
- а) `>>> x = 251`  
`>>> y = 356`  
`>>> z = y % 25`  
`>>> print ("Результат: ", z, x+y, "\n", x**z)`
- а) `>>> ха = "фик"`  
`>>> хи = "ус"`  
`>>> print (ха + хи)`

## 5. Функция ввода input()

Использовать инструкцию присваивания в программе не всегда возможно, поскольку программы, как правило, должны отвечать требованию универсальности. Если это невозможно реализовать, то это будет уже не программа, а легкое недоразумение.

Функция `input` предназначена для ввода строки с клавиатуры. Поэтому приняты три варианта использования.

1. Если требуется ввести с клавиатуры строку в какую-то переменную, то функция выглядит, например, следующим образом:

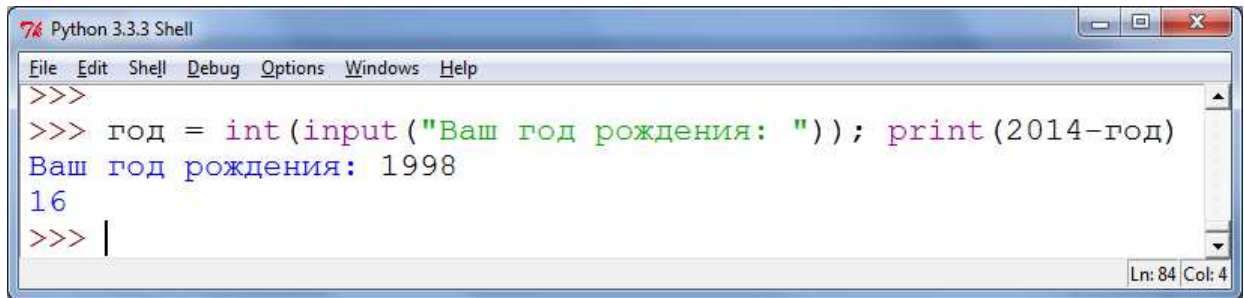


```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
>>>
>>> a = input("Ваше имя: "); print (a)
Ваше имя: Леонид
Леонид
>>>
```

☞ Обратите внимание на одну особенность: в Python принято писать по технологии «Одна инструкция – одна строка». Однако, учитывая, что в Python почти все можно, разрешается написать несколько инструкций в одной строке, если разделить их точкой с запятой. Злоупотреблять этим не следует!



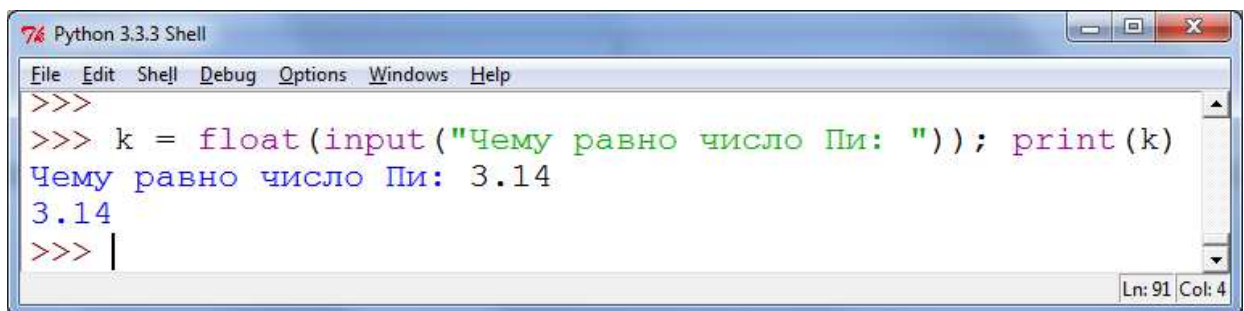
2. Если требуется ввести с клавиатуры целое число, то функция будет выглядеть немного иначе:



```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
>>>
>>> год = int(input("Ваш год рождения: ")); print(2014-год)
Ваш год рождения: 1998
16
>>> |
```

☞ В случае, если строка представляет собой набор цифр – целое число, функция `int()` преобразует текстовое выражение в целое число. Если вводимый текст не будет целым числом, будет выдана ошибка.

3. И, наконец, если требуется ввести вещественное число, будет использована функция `float()`, которая обеспечивает преобразование [правильно] записанного вещественного числа из текстовой в числовую форму:

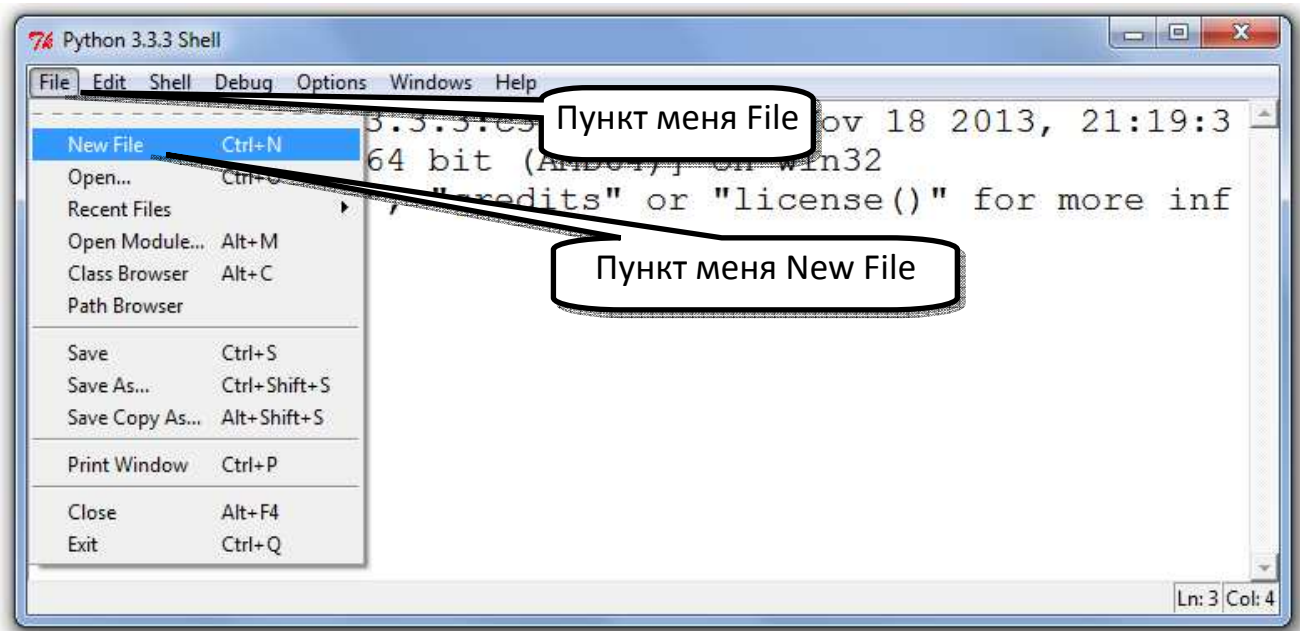


```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
>>>
>>> k = float(input("Чему равно число Пи: ")); print(k)
Чему равно число Пи: 3.14
3.14
>>> |
```

## 6. Как готовить программы в среде Python

Ну, собственно, ничего сложного здесь нет. Следует:

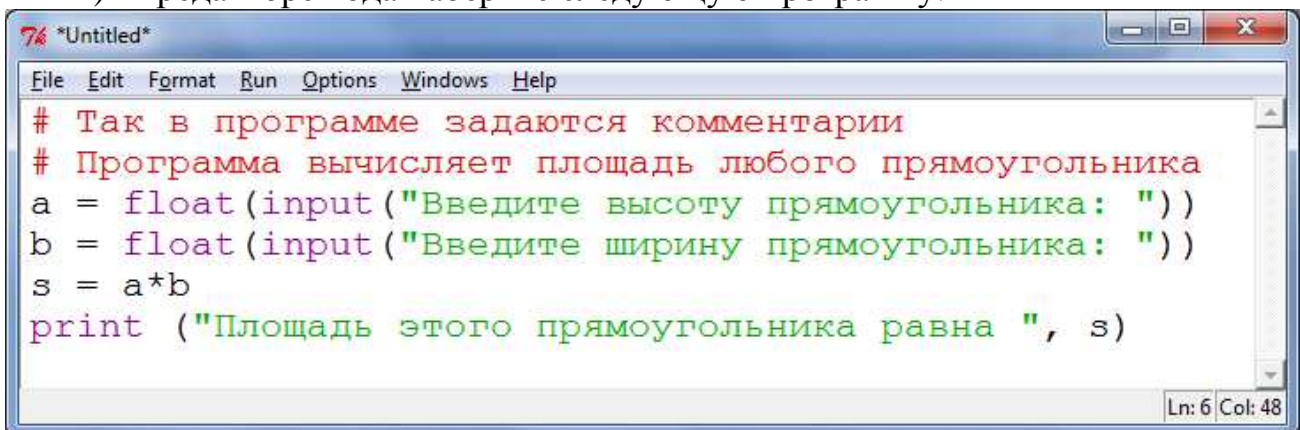
- 1) в строке меню среды Python щелкнуть по пункту **File** и в появившемся подменю выбрать пункт **New File** (что, как Вы хорошо знаете, означает Новый файл);



- 2) после щелчка на пункт **New File** на экране появится окно текстового редактора (очень напоминает Блокнот), в котором и можно набирать текст будущей программы;
- 3) как уже отмечалось, в Python при написании программы принято писать одну инструкцию в одной строке!

### Задание 9

- 1) В редакторе кода наберите следующую программу:



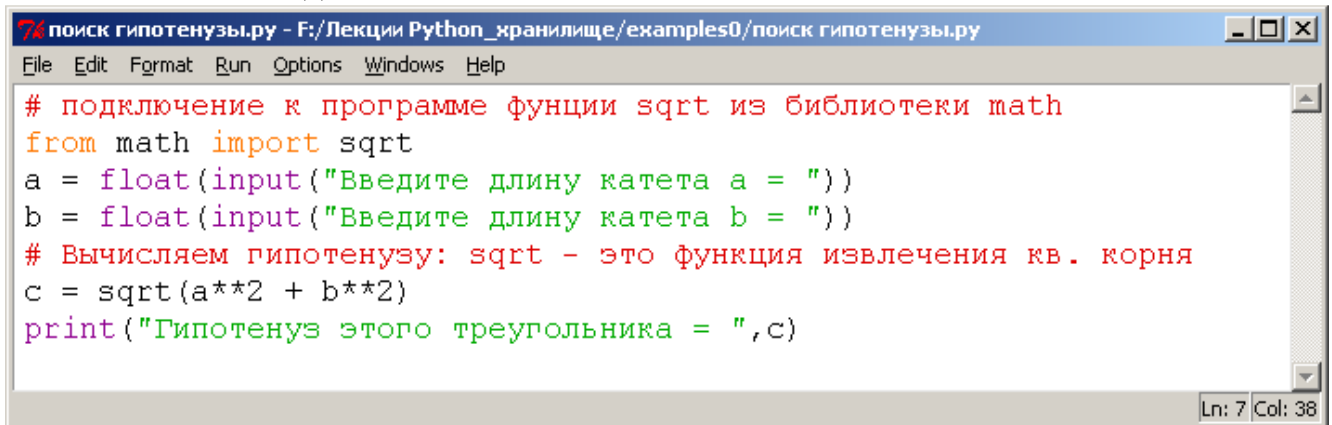
- 2) После подготовки текста программы (на самом деле ее можно готовить в любом текстовом редакторе типа Блокнот) ее нужно сохранить. Для этого щелкните снова по пункту меню **File**, но в этот раз выберите пункт **Save As...** Появится стандартное окно **Сохранить как...** Сохраните созданный Вами текст в Вашей папке (если еще не сделали Задание 1, сделайте его).
- 3) И, наконец, чтобы увидеть как работает программа, запустите ее на выполнение. Для этого достаточно нажать на функциональную клавишу **F5** или воспользоваться пунктом меню **Run → Run Module**.
- 4) Попробуйте запустить подготовленную программу несколько раз и вычислить площади разных прямоугольников с совершенно



сумасшедшими размерами сторон. Посмотрите, как на это реагирует Python.

И в заключение составьте программы для задач, приведенных в следующем пункте этого модуля. Не забудьте все программы сохранять в своей папке с понятными именами (например, программу по определению объема параллелепипеда можно назвать S-параллелепипеда\_Ивановой.py. Здесь Иванова – фамилия разработчика программы).

Небольшая подсказка:



```
# поиск гипотенузы.py - F:/Лекции Python_хранилище/examples0/поиск гипотенузы.py
File Edit Format Run Options Windows Help

# подключение к программе функции sqrt из библиотеки math
from math import sqrt
a = float(input("Введите длину катета a = "))
b = float(input("Введите длину катета b = "))
# Вычисляем гипотенузу: sqrt - это функция извлечения кв. корня
c = sqrt(a**2 + b**2)
print("Гипотенуз этого треугольника = ",c)

Ln: 7 Col: 38
```

## 7. Задачи для самостоятельной работы

1. Известны значения двух переменных, содержащие числа. Напишите программу, меняющую значения этих переменных между собой без использования третьей, буферной переменной.
2. Вычислите объем и площадь поверхности параллелипипеда. Определите, какие данные должны быть введены, какие – выданы на экран.
3. Составьте программу определения корней квадратного уравнения, имеющего решения.
4. Составьте программу определения идеального веса человека, если он считается по формуле: идеальный вес (кг) = рост (см) – 100.
5. Средняя наценка на товар составляет 18%. Пусть известна розничная цена товара. Вычислите оптовую цену.
6. Даны координаты двух точек. Вычислите расстояние между ними.
7. Вы положили деньги в банк под 15% годовых. Составьте программу определения общей суммы вклада через заданное число месяцев.
8. Длина некоторого отрезка составляет  $p$  метров. Напишите программу перевода ее в русскую неметрическую систему. Указание: 1 верста = 500 саженей, 1 сажень = 3 аршина, 1 аршин = 16 вершков, 1 вершок = 44,45 мм.
9. Составьте программу, которая вычисляет площадь равностороннего треугольника.
10. Напишите программу вычисления стоимости покупки, состоящей из нескольких карандашей, линеек и тетрадей. Их количество и цену задать вводом. Ответ вывести в виде:

Сумма к оплате: ... руб ... коп.

11. Напишите программу, переводящую время, указанное в минутах, во время в часах и минутах.

- ☞ *Пригласите преподавателя и продемонстрируйте ему все выполненные Вами задания. После этого сохраните все программы и завершите работу в среде Python. Будьте готовы ответить на вопросы.*
- ☞ *Если вы обучаетесь в условиях использования дистанционных образовательных технологий, просто вышлите все файлы с подготовленными Вами программами преподавателю (если это предусмотрено технологической картой).*