

Ввод данных

За ввод данных отвечает функция `input()` .

В переводе с английского: input - ввод

Принцип ее работы заключается в том, что она останавливает выполнение программы и ожидает введения данных от пользователя. Рассмотрим простой пример, где мы предлагаем пользователю ввести его имя, а затем приветствуем его.

Задание 1 name.py

1. Создай каталог **Lesson4**
2. Запусти IDLE и выбери в меню **File ► New File**.
3. Сохрани программу под именем **name.py** в каталоге **Lesson4**
4. В раскрывшемся окне редактора набери программу в которой тебе нужно написать следующий код:

```
# объявим переменную, которая будет хранить имя пользователя:
name = input ('Как тебя зовут? \n')
# выводим на экран приветственную фразу:
print ('Привет, ', name)
```

Здесь есть переменная, **name**. В ней сохраняется последовательность символов, называемая строкой.

Кроме того, здесь используются две функции:

print() Отвечает за вывод чисел и текста на экран.

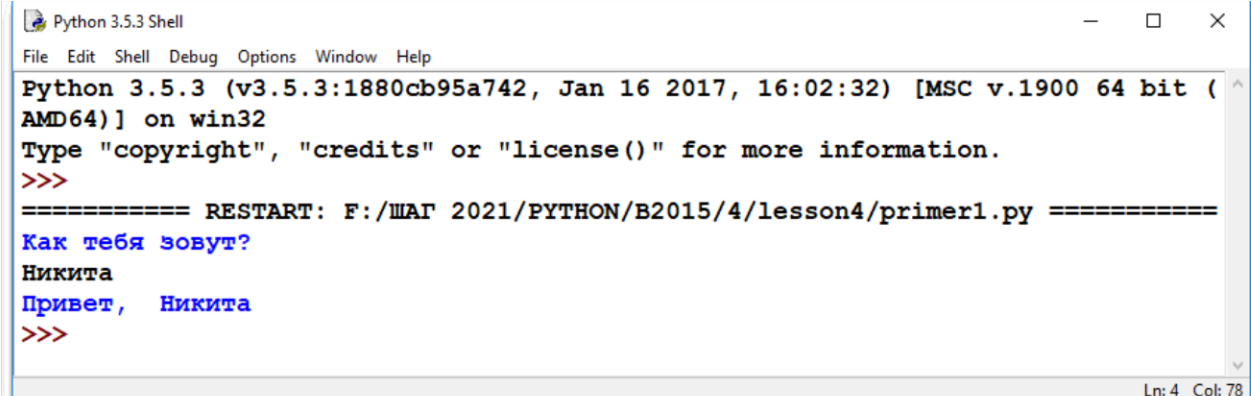
input() Отвечает за ввод чисел и текста с помощью клавиатуры.

5. Сохрани (**File ► Save**) и запусти программу (**Run ► Run Module** или **F5**).

После сохранения и запуска файла мы увидим, что окно оболочки Python запустило программу, выведя на экран вопрос **Как тебя зовут?**. Так как мы поставили после вопроса литерал `\n`, то курсор переходит на новую строку.

Введем свое имя на следующей строке и нажмём клавишу **Enter**. Программа напечатает слово **Привет** , после которого будет следовать введенное имя.

Так как мы не просили программу сделать что-либо еще, то программа будет завершена — и мы опять увидим приглашение `>>>`, как показано на скриншоте.



```
Python 3.5.3 Shell
File Edit Shell Debug Options Window Help
Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 16:02:32) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:/ШАГ 2021/PYTHON/B2015/4/lesson4/primer1.py =====
Как тебя зовут?
Никита
Привет, Никита
>>>
```

Задание 2 class.py

1. Открой файл **name.py**, если ты его закрыл.
2. Сохрани этот файл под новым именем **class.py** в папке **Lesson4**. Для этого нажми **File ► Save As ► class.py**
3. Файл **name.py** автоматически закроется, а вместо него будет открыт **class.py** с такой же программой.
4. Работать будем в файле **class.py**
5. Добавь следующий фрагмент кода (я выделила жирным):

```
# объявим переменную, которая будет хранить имя пользова-
теля:
name = input ('Как тебя зовут? \n')
# выводим на экран приветственную фразу:
print ('Привет, ', name)

# объявим переменную, которая будет хранить номер класса:
my_class = input ('В каком классе ты учишься? \n')
# выводим на экран фразу об учебе в классе:
print (name, 'учится в', my_class, 'классе')
```

6. Сохрани (**File ► Save**) и запусти программу (**Run ► Run Module**)
7. Проверь как она работает: программа задает вопросы, а ты отвечаешь.
8. По аналогии добавь в код еще одну переменную **city** которой будет присвоено значение, введенное пользователем при ответе на вопрос:
в каком городе ты живешь?
9. Выведи на печать фразу:
name живет в городе city
10. Вместо переменных **name** и **city** на экране будут выводиться соответствующие значения: имя, город.
11. Сохрани (**File ► Save**) и запусти программу (**Run ► Run Module** или **F5**).
12. Покажи работающую программу преподавателю:

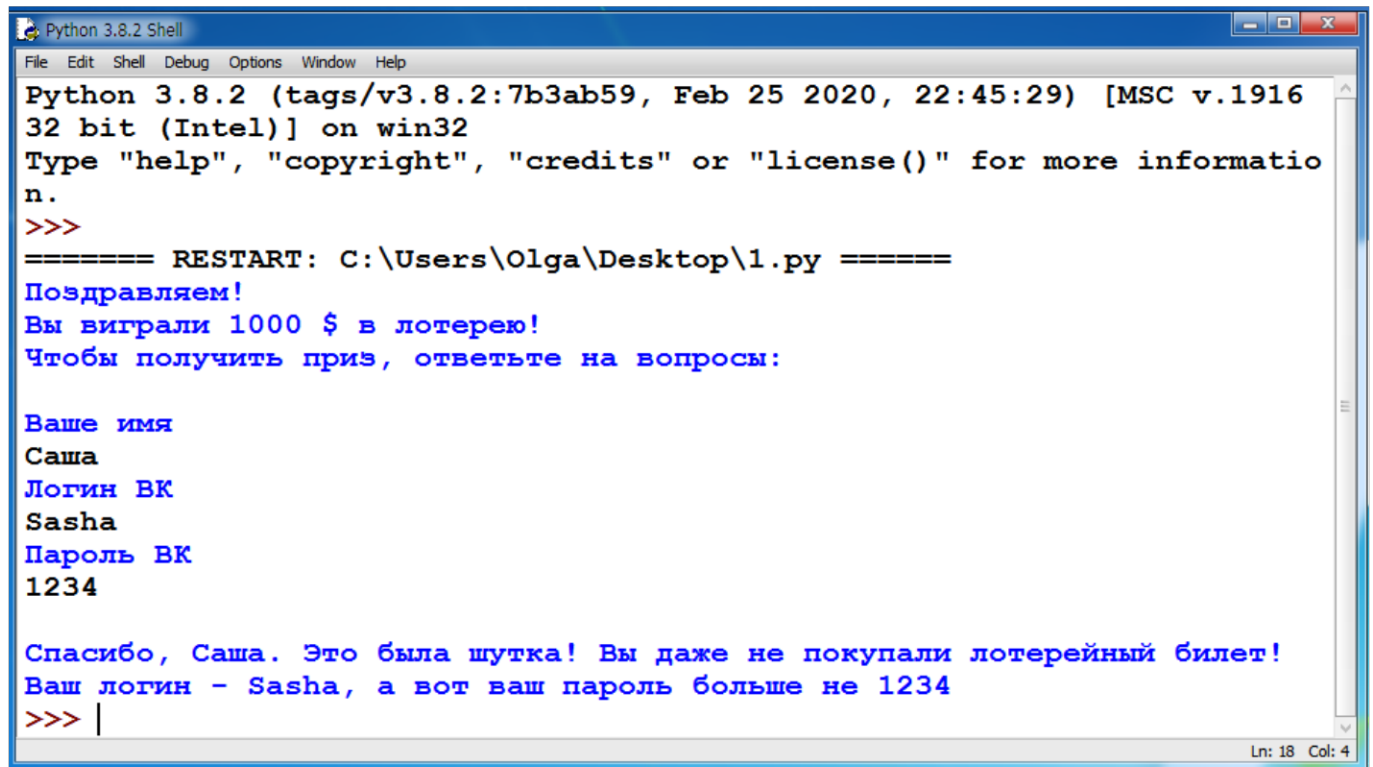
```
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18)
AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
===== RESTART: F:/ШАГ 2021/PYTHON/B2015/4/lesson4/cl
Как тебя зовут?
Алексей
Привет, Алексей
В каком классе ты учишься?
6
Алексей учится в 6 классе
В каком городе ты живешь?
Гомель
Алексей живет в городе Гомель
>>>
```

Задание 3* joke.py

1. Создай новый файл и назови его **joke.py**

(*joke* в переводе с английского *шутка*)

2. Напиши скрипт, в результате которого в оболочке будет выводиться:



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916
32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more informatio
n.
>>>
===== RESTART: C:\Users\Olga\Desktop\1.py =====
Поздравляем!
Вы выиграли 1000 $ в лотерею!
Чтобы получить приз, ответьте на вопросы:

Ваше имя
Саша
Логин ВК
Sasha
Пароль ВК
1234

Спасибо, Саша. Это была шутка! Вы даже не покупали лотерейный билет!
Ваш логин - Sasha, а вот ваш пароль больше не 1234
>>> |
```

Подсказка.

Объяви три переменных:

```
name,
login,
password.
```

Присвой им значения с использованием функции **input()** .

Чтобы пользователь начинал вводить данные с новой строки используй **\n**

Чтобы напечатать пустую строку воспользуйтесь функцией **print()** .

Для того, чтобы в предложениях знаки препинания располагались по правилам русского языка, т.е. сразу после слова, а не через пробел, примени в нужных местах знак склеивания "+" вместо запятой.

Задание 4 math1.py

1. Создай новый файл **math1.py** в папке **Lesson4**.
2. Напиши следующий код:

```
# Математика
print("Введи число: ")
Number1 = input()
print("Введи еще одно число: ")
Number2 = input()
print (Number1+Number2)
```

3. Запусти программу (Run ► Run Module)
4. Введи поочередно числа 2 и 3

Результат 2+3 равен 23? Это же неправда. Или правда?

Проблема в том, что здесь все числа были введены функцией **input()**, которая применяется для символов или строк.

5. Добавь в код еще строчку на операцию деления, умножения или вычитания:

```
print (Number1 / Number2)
print (Number1 - Number2)
print (Number1 * Number2)
```

6. Сохрани изменения (**File ► Save**) и запусти программу (**Run ► Run Module** или **F5**).

Все остальные операторы, кроме сложения приводят лишь к сообщениям об ошибках. Потому что другие операторы *строковых типов* (**'str'**) данных вообще не поддерживают.

```
Введи число:
2
Введи еще одно число:
3
23
Traceback (most recent call last):
  File "F:\ШАГ 2021\PYTHON\B2015\4\lesson4\math1.py", line 7, in <module>
    print (Number1 / Number2)
TypeError: unsupported operand type(s) for /: 'str' and 'str'
>>> |
```

Нам нужны **числа** для вычислений!

Но как превратить их из строкового типа данных в числовой?

Взглянем на небольшой пример обработки чисел.

`int()` – функция целочисленного типа данных.

Она превращает значение переменной (строку) в целое число (сокращение *int* происходит от слова *integer*, что в переводе с английского означает «целое число»).

Здесь вложены две функции, одна в другую: `input()` принимает ввод символов с клавиатуры, а `int()` превращает введенные (строковые) символы в целые числа. Другими словами, `int()` принимает значение другой функции, указанной в скобках.

Для лучшего понимания, выполни задание 5

Задание 5 `math2.py`

1. Открой файл `math1.py`, если ты его закрыл.
2. Сохрани этот файл под новым именем `math2.py` в папке **Lesson4**. Для этого нажми **File ► Save As ► math2.py**
3. Файл `math1.py` автоматически закроется, а вместо него будет открыт `math2.py` с такой же программой.
4. Работать будем в файле `math2.py`
5. Измени исходный код (я выделила жирным) и запусти программу несколько раз. Попробуй все математические операторы.

```
# Математика
print("Введи число: ")
Number1 = int(input())
print("Введи еще одно число: ")
Number2 = int(input())
print (Number1+Number2)
print (Number1*Number2)
print (Number1/Number2)
print (Number1-Number2)
```

6. Сохрани изменения (**File ► Save**) и запусти программу (**Run ► Run Module**).
7. Если все работает, покажи результат преподавателю:

```
Введи число:
2
Введи еще одно число:
3
5
0.6666666666666666
-1
6
>>>
```


Тип float

Обрати внимание на операцию деления. Как видим, операция деления выводит правильный результат, даже если два целых числа фактически делятся на нецелое число. Но если мы попытаемся ввести число с плавающей точкой, возникнет проблема.

Попробуем ввести любое число, например, **2.5** и выполнить любую операцию.

В этом примере Python не способен ничего сделать с числом с плавающей точкой, таким как **2.5**, потому что оно не является целым числом, вот и возникает ошибка. Чтобы решить проблему и избежать сообщения об ошибке, нужно использовать другой тип чисел, отличный от целых (**int**).

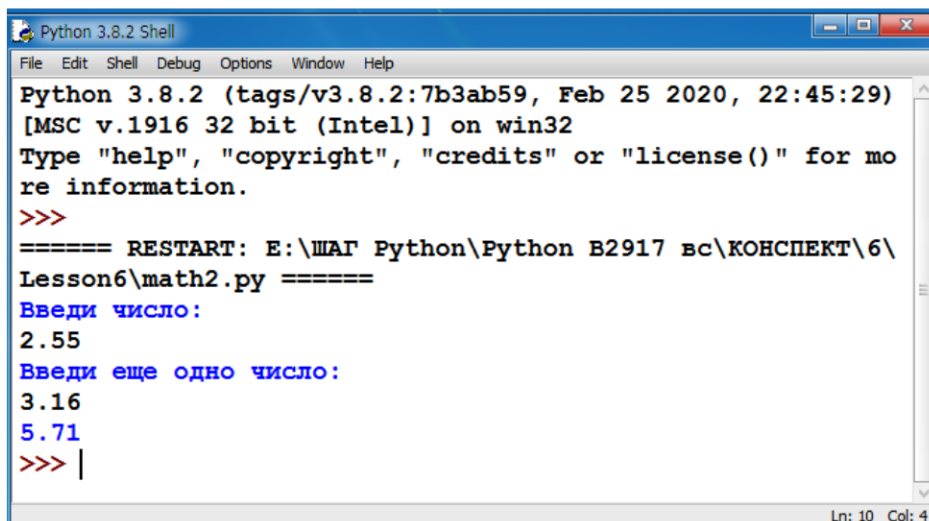
Для этого используется тип **float**, означающий число с плавающей точкой. Посмотрим, как это работает, выполнив следующее задание.

Задание 6 math3.py

1. Открой файл **math2.py**, если ты его закрыл.
2. Сохрани этот файл под новым именем **math3.py** в папке **Lesson4**. Для этого нажми **File ► Save As ► math3.py**
3. Измени строки кода, написанного ниже, в своей программе:

```
# Математика
print("Введи число: ")
Number1 = float(input())
print("Введи еще одно число: ")
Number2 = float(input())
print (Number1+Number2)
print (Number1*Number2)
print (Number1/Number2)
print (Number1-Number2)
```

4. Запусти программу (Run ►Run Module)
5. Убедись, что, программа работает с числами с плавающей точкой.



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29)
[MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
===== RESTART: E:\ШАГ Python\Python B2917 вс\КОНСПЕКТ\6\
Lesson6\math2.py =====
Введи число:
2.55
Введи еще одно число:
3.16
5.71
>>> |
```

Задание 7* pizza.py

1. Создай новый файл **pizza.py** в папке **Lesson4**.
2. Напиши следующий код (номера строк вводить не нужно!!!!):

```
1. #простой калькулятор стоимости пиццы
2. #Спросить человека, сколько пицц он хочет:
3. kol_pizza = int(input("Сколько пицц вы хотите? \n"))
4. #Запросить стоимость пиццы по меню:
5. price_pizza = float(input("Сколько стоит пицца? \n"))
6. #Подсчитать общую стоимость пиццы
7. summa = kol_pizza * price_pizza
8. #Показать пользователю общую сумму к оплате:
9. print() # пустая строка
10. print("Ваш заказ: ", kol_pizza, 'шт. пиццы' )
11. print("По цене ", price_pizza, " за одну пиццу")
12. print("К оплате", summa, "руб.")
```

3. Запусти программу (**Run ► Run Module**)
4. Проверь как она работает.
5. Измени код, таким образом, чтобы программ считала чаевые.
6. Добавьте после седьмой строчки кода переменную **tips**, которая будет спрашивать у пользователя о том, сколько бы он хотел оставить официанту «на чай».

```
tips = float(input('сколько вы хотите оставить официанту "на чай"? \n'))
```

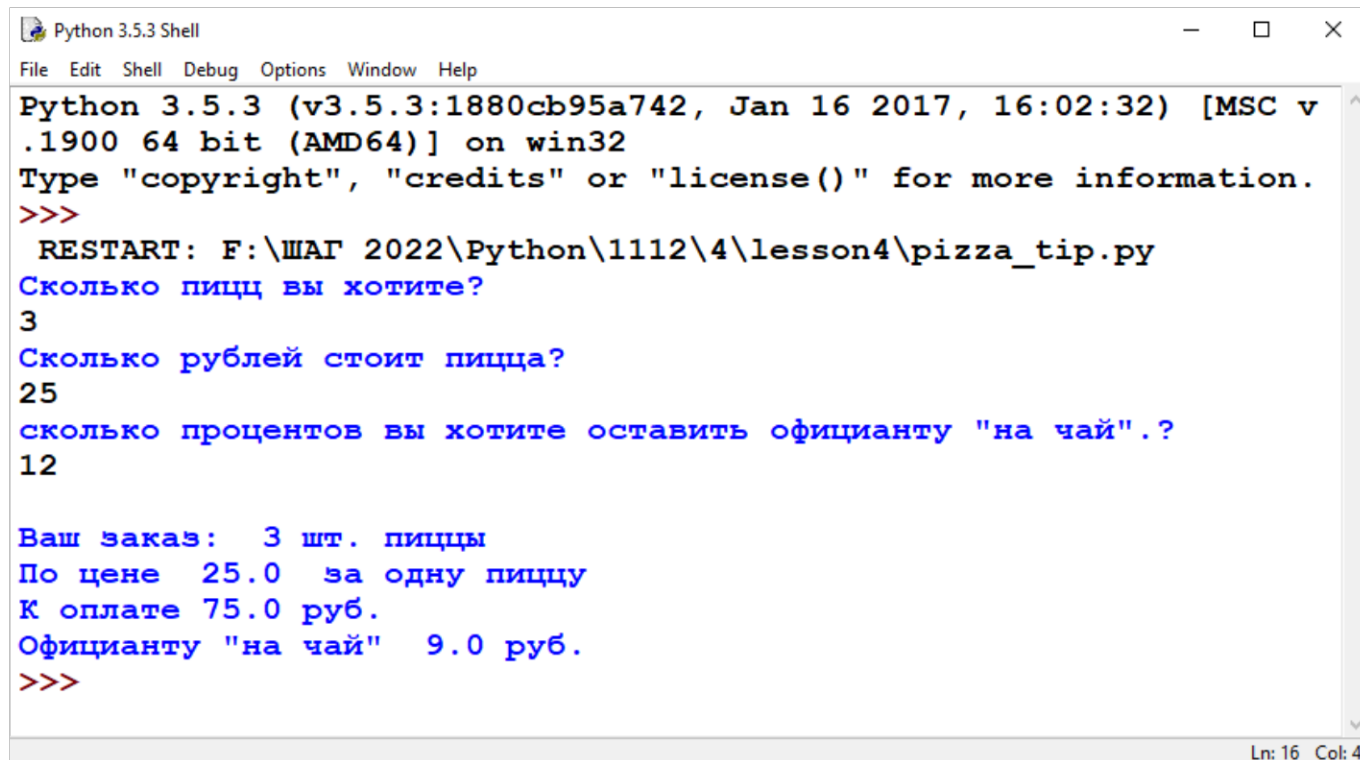
7. Часто оставляют 10% ... 15% от общей суммы.
8. Если ты помнишь из математики, то процент, можно рассчитать так:
указанное число/100

Указанное число	Формула	Результат
1%	1/100	0,01
2%	2/100	0,02
⋮	⋮	⋮
10%	10/100	0,1
11%	11/100	0,11
12%	12/100	0,12
⋮	⋮	⋮
99%	99/100	0,99
100%	100/100	1,0

Таким образом, когда пользователь ответит на вопрос «Сколько процентов вы дадите официанту на чай?» тебе нужно общую стоимость умножить на это число и разделить на 100.

Поэтому после двенадцатой строчки кода нужно добавить функцию

```
print('Официанту "на чай" ', summa*tips/100, "руб.")
```



```
Python 3.5.3 Shell
File Edit Shell Debug Options Window Help
Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 16:02:32) [MSC v
.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: F:\ШАГ 2022\Python\1112\4\lesson4\pizza_tip.py
Сколько пицц вы хотите?
3
Сколько рублей стоит пицца?
25
сколько процентов вы хотите оставить официанту "на чай".?
12

Ваш заказ: 3 шт. пиццы
По цене 25.0 за одну пиццу
К оплате 75.0 руб.
Официанту "на чай" 9.0 руб.
>>>
```

Задание 8** dlina.py

В англоязычных странах широко используются меры расстояний в милях, футах и дюймах.

В нашей стране для измерения длины применяют километры, метры, сантиметры и миллиметры.

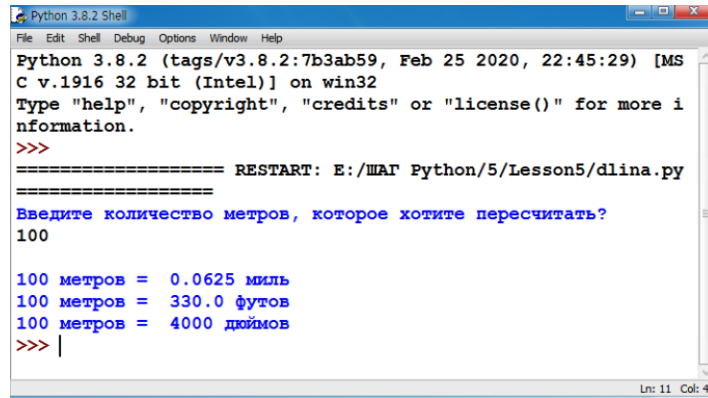
1 миля	~ 1,6 км
1 фут	~ 0,3 м
1 дюйм	~ 2,5 см

Напишите программу, которая запрашивает у пользователя количество метров, а затем на дисплей выводит это значение в милях, футах, дюймах.

1 м	= 100 см
	= 0,001 км

Программу сохраните под именем **dlina.py**

Результат будет выглядеть примерно так, как у меня на скриншоте:



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MS
C v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more i
nformation.
>>>
===== RESTART: E:/ШАГ Python/5/Lesson5/dlina.py =====
Введите количество метров, которое хотите пересчитать?
100

100 метров = 0.0625 миль
100 метров = 330.0 футов
100 метров = 4000 дюймов
>>> |
```

Задание 9** my_car.py

Напиши программу **my_car.py** которая просит пользователя ввести:

- Количество бензина в баке автомобиля.
- Расход в литрах на 100 км.
- Цена 1 литра бензина.

Затем вычисли и выведите на экран:

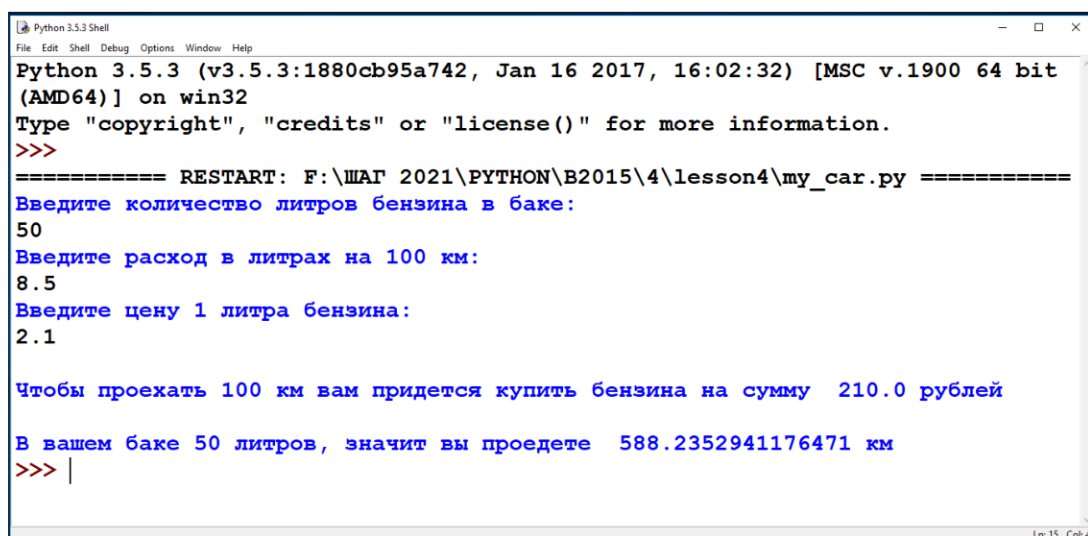
Стоимость бензина на 100 км

Сколько километров можно проехать на имеющемся в баке бензине.

Программу сохраните под именем.

Например, при вводе данных, у пользователя-автомобилиста может быть объем бака 50 литров, расход 8,5 литр на 100 км. Цена девяносто второго бензина примерно 2,1 рубля.

У меня получилось так:



```
Python 3.5.3 Shell
File Edit Shell Debug Options Window Help
Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 16:02:32) [MSC v.1900 64 bit
(AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:\ШАГ 2021\PYTHON\B2015\4\lesson4\my_car.py =====
Введите количество литров бензина в баке:
50
Введите расход в литрах на 100 км:
8.5
Введите цену 1 литра бензина:
2.1

Чтобы проехать 100 км вам придется купить бензина на сумму 210.0 рублей

В вашем баке 50 литров, значит вы проедете 588.2352941176471 км
>>> |
```

Задание 10** zakaz.py

Программа на компьютере в книжном магазине для расчета стоимости заказа новой партии книг из типографии должна выполнить следующие действия:

- выполнить ввод с клавиатуры стоимости одной книги;
- выполнить ввод с клавиатуры количества заказанных книг;
- вычислить общую стоимость книг;
- подсчитать налог с продажи (7,5 процентов от общей стоимости);
- подсчитать расходы на транспортировку (15 копеек за 1 книгу);

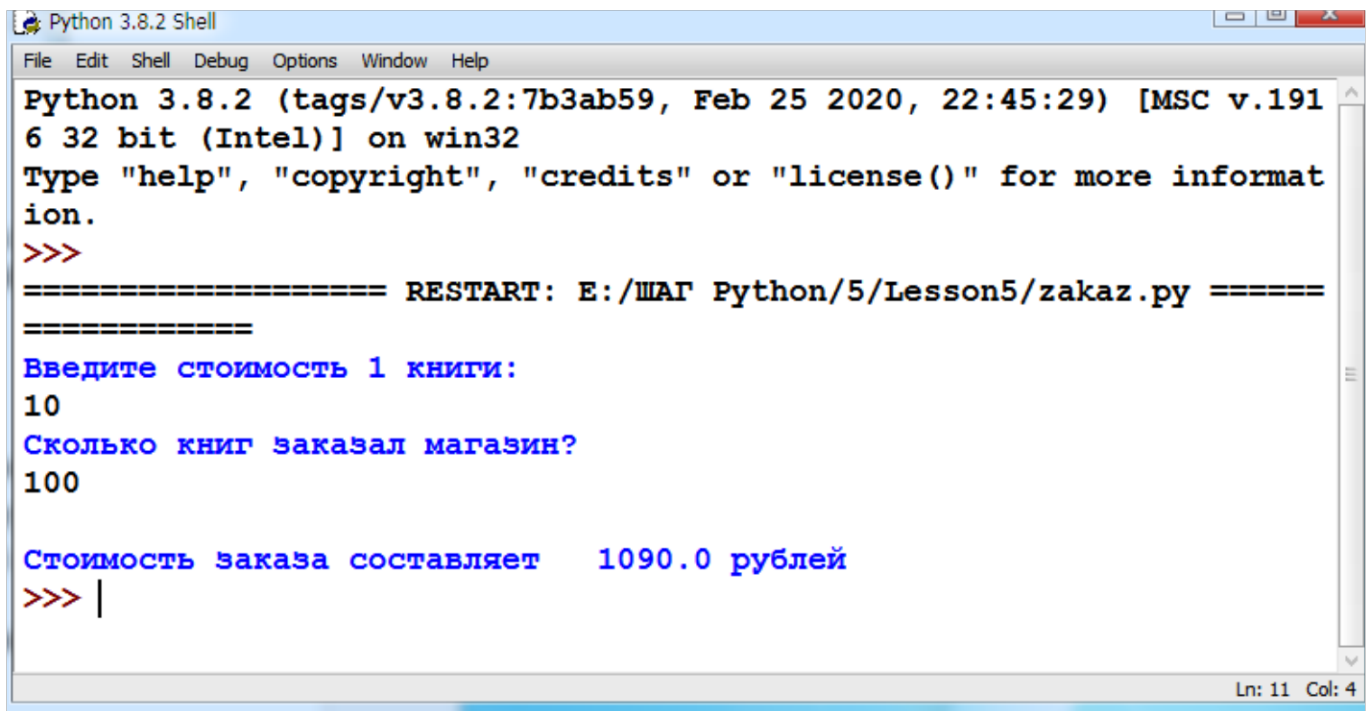
Стоимость заказа будет определяться суммой, состоящей из общей стоимости книг, налога с продажи всех книг и расходов на транспортировку.

Необходимо вывести на дисплей стоимость заказа в рублях в виде:

Стоимость заказа составляет рублей.

Примечание. Стоимость книги и количество книг вводит пользователь по запросу программы. Файл программы назови: **zakaz.py**.

У меня так:



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.191
6 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more informat
ion.
>>>
===== RESTART: E:/ШАГ Python/5/Lesson5/zakaz.py =====
=====
Введите стоимость 1 книги:
10
Сколько книг заказал магазин?
100

Стоимость заказа составляет    1090.0 рублей
>>> |
```