

Step 3

Step on Stepik: <https://stepik.org/lesson/2232/step/3>

Использование переменных просто необходимо если мы хотим написать программу которая использует пользовательский ввод. Допустим мы хотим написать программу, которая будет не просто считать чему равно дважды два, а которая будет просить пользователя ввести два числа, читать их и вычислять их произведение. В такой задаче нам будет сложно обойтись без переменных. Как можно решить эту задачу? Давайте заведем две переменные и запишем в них значения, введенных пользователем чисел. После этого посчитаем произведение этих двух переменных. Но для начала посмотрим, как вообще можно осуществлять пользовательски ввод.

Ввод данных

```
> input() # читает строку с клавиатуры
> input('Введите данные')

> s = input() # записать в переменную s
пользовательский ввод
> a = int(input()) # прочитать строку с
клавиатуры и преобразовать в число

Вывод данных: print('some output')
```



Для этого в Python существует простая функция - **input**, которая читает строку, введенную с клавиатуры. Если во время выполнения программы встречается функция **input**, программа приостанавливает свою работу и ожидает ввода данных от пользователя до тех пор, пока он не нажмет клавишу **<Enter>**. Дальше программа возобновляет свою работу. В качестве результата функции возвращается строка, которую пользователь ввел.

```
In [2]: input()
Hello!!!

Out[2]: 'Hello!!!'

In [ ]:
```

В функцию **input** можно передать строку в качестве параметра.
Например:

```
In [*]: input("Введите данные ")
Введите данные В этом окне мы вводим данные

In [ ]:
```

В этом случае с начала выведется пригласительное сообщение - "Введите данные", а уж после этого программа передаст управление пользователю, чтобы он ввел свои данные с клавиатуры.

Сохранить пользовательский ввод можно следующим образом:

```
In [6]: s = input()
Сохраним этот текст в переменную s

In [8]: s
Out[8]: 'Сохраним этот текст в переменную s'

In [ ]:
```

тем самым мы заводим переменную **s** и записываем в нее результат того, что ввел пользователь. Поскольку функция **input** всегда возвращает строку, а нам необходимо число, нам нужно явно преобразовать строку в число. Преобразование осуществляется с помощью функции преобразования типа - **int**. В итоге наша инструкция будет выглядеть так:

```
In [9]: s = int(input())
```

```
5
```

```
In [10]: type(s)
```

```
Out[10]: int
```

```
In [ ]: |
```

Для вывода данных используется функция, которую мы уже упоминали – **print**. Она принимает в качестве параметра произвольный объект и выводит его в строковом представлении. В функцию **print** можно передать несколько параметров через запятую. Например: **print(a, s)** выведет на экран строковые представления значений переменных **a** и **s**, разделенные пробелом.

```
In [11]: print("Этот текст выведется функцией print")
```

```
Этот текст выведется функцией print
```

```
In [12]: a = "a"  
s = 5  
print(a, s)
```

```
a 5
```

```
In [ ]: |
```