

# 3. Python.

## Преобразование типов данных

---

Автор: Александр Килинкарров, © 2023

E-mail: [linxam@gmail.com](mailto:linxam@gmail.com)

Telegram: [@linxam](https://t.me/_linxam)

Youtube: [https://youtube.com/@it4fun\\_a](https://youtube.com/@it4fun_a)

Сайт: [kilinkarov.ru](http://kilinkarov.ru)

---

Итак, попробуем написать небольшую программу, которая будет складывать два числа. Числа спросим у пользователя.

```
num1 = input('Введи первое число: ')\nnum2 = input('Введи второе число: ')\n\nprint('Результат:', num1 + num2)
```

Предположим, что пользователь запустил нашу программу:

```
Введи первое число: 2  
Введи второе число: 3  
Результат: 23
```

Пользователь ввел числа 2 и 3, и надеялся получить 5. А вместо этого программа напечатала 23. Почему? Потому что, `input` приносит в программу **строку** (тип данных **str**). А строки, как мы знаем, можно складывать, и при этом они склеиваются. Вот мы и получили в результате **строку** 23. То есть, это не число, а строка, состоящая из символов 2 и 3. Как же нам быть? Ведь нам хочется складывать **числа**.

## Преобразование типов данных

Для того, чтобы получить из строки число, нам нужно **преобразовать** её в число. И тут нам

нужно вспомнить названия типов данных, которые мы уже знаем: **int**, **float**, **bool**, **str**.

## int

Для преобразования в целое число, у нас есть **функция int**. Пишется она так: **int()**

В скобки ей кладём то, что хотим преобразовать в целое число. Если у *int* получится преобразовать, то мы получим целое число. Если же не получится, то возникнет ошибка.

Рассмотрим такой код:

```
x = input('Введи число: ')\n x = int(x)\n print(x + 10)
```

В первой строчке кода x - строка. После второй строчки x уже стал числом. И к нему можно спокойно прибавить 10

Ещё примеры:

```
x = '3' # это строка
x = int(x) # теперь x это целое число

x = '234asd'
x = int(x) # тут возникнет ошибка, потому
что int не знает как поступить с этой
строкой
```

Функция `int` может работать не только со строками, но и с другими типами данных. Что будет, если ей дать дробное число? Она отбросит дробную часть, и оставит целую.

Вот так:

```
x = 2.9
x = int(x)
print(x) # напечатается 2, так как
осталась только целая часть числа
```

Что будет, если функции `int` дать логическое (булево) значение? Истина станет единицей, а ложь нулём.

Например:

```
x = True
x = int(x)
print(x) # Напечатается 1
```

## float

Функция **float** работает похоже на **int**. Она нам дает на выходе - дробные числа.

Единственное, она при преобразовании строк, разрешает нам точку внутри строки.

Примеры:

```
x = 123 # сейчас x - целое число
x = float(x) # сейчас x равен 123.0

x = True # сейчас x имеет тип данных bool
x = float(x) # а сейчас он - float, и
равен 1.0
```

```
x = '123.9' # сейчас x - строка
x = float(x) # а сейчас он, число (float)
```

## bool

Функция **bool** преобразует всё, что мы ей дадим, в истину или ложь.

Если преобразуем числа, целые или дробные, то любое число, отличное от нуля, преобразуется в истину. Ноль - преобразуется в ложь.

```
x = 123 # x целое число
x = bool(x) # x теперь истина

x = 0 # x целое число
x = bool(x) # x теперь ложь

x = 0.123 # x дробное число
x = bool(x) # x теперь истина

x = 0.0 # x дробное число
x = bool(x) # x теперь ложь
```

Как функция **bool** работает со строками?

Любая строка, в которой есть, хотя бы, один символ - становится истиной. И только *пустая строка* - становится ложью.

Примеры:

```
x = 'привет' # x - строка
x = bool(x) # x становится истиной, True

x = 'False' # x - строка
x = bool(x) # x становится истиной, True.
Ну и что, что в строке написано False,
ведь она не пустая, в ней есть символы.
Поэтому x стал истиной, True

x = '' # x строка. Причём пустая. В ней
нет символов
x = bool(x) # x стал ложью. Так как в
строке не было ни одного символа
```

**str**

Функция **str** превращает всё в строку. Она всеядная.

```
x = 123 # x число
x = str(x) # x теперь строка '123'

x = 123.9 # x число
x = str(x) # x теперь строка '123.9'

x = True # x истина
x = str(x) # x теперь строка 'True'

x = False # x ложь
x = str(x) # x теперь строка 'False'
```

## Пример

Теперь давайте вернемся к примеру, написанному вначале. Тогда мы, сложив 2 и 3 получили 23. Исправим это.

```
num1 = input('Введи первое число: ')
num2 = input('Введи второе число: ')
num1 = int(num1) # делаем num1 целым
```



числом

```
num2 = int(num2) # делаем num2 целым
```

числом

```
print('Результат:', num1 + num2) #
```

получаем реальную сумму. В математическом смысле

Если присмотреться повнимательнее, то мы поймём, что этот код можно сократить.

```
num1 = int(input('Введи первое число: '))
```

```
num2 = int(input('Введи второе число: '))
```

```
print('Результат:', num1 + num2)
```

Здесь мы те строки, которые приносит нам **input** сразу же кладем в скобки функции **int**, и в результате, в переменные **num1** и **num2** сразу же записываются целые числа.

Ну а теперь, поэкспериментируйте сами с кодом. Подумайте, как нам быть, если хотим

принять от пользователя дробные числа.  
Например, его вес в кг.

Как нам быть, если нам нужно спросить что-то такое у пользователя нашей программы, что должно быть строкой? Экспериментируйте, пробуйте.