

1. Python. Типы данных

Автор: Александр Килинкар, © 2023

E-mail: linxam@gmail.com

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

Youtube: https://youtube.com/@it4fun_a

Сайт: kilinkarov.ru

Введение

Почти все программы работают по принципу: что-то приходит на вход программы, далее программа **это что-то изменяет**, и выдаёт результат.

Например, программа, которая редактирует фотографии, на вход получает фотографию. Затем, мы как-то редактируем эту фотографию в программе (изменяем её размер, обрабатываем фильтрами и т.д.). И полученный

результат сохраняем в виде изменённой фотографии.

Чем же манипулирует программа во время своей работы? Что она изменяет? Она изменяет **данные**. Какими бывают **данные**? Каких **типов** они бывают?

Для начала, нам хватит всего 4 типа данных. Это *целые числа*, *дробные числа*, *логический тип данных* и *строки*. Нам важно о них знать, чтобы понимать, что именно мы можем с ними делать, и как они нам могут пригодиться

Целые числа. `int`

Тип данных `int` нам интуитивно понятен из математики. Что можно делать с целыми числами? Вот, доступные нам действия.

```
3 + 4 # сложение, результат 7
6 - 4 # вычитание, результат 2
3 * 4 # умножение, результат 12
8 / 2 # деление, результат 4.0
```

```
9 // 4 # целочисленное деление, результат  
2  
9 % 4 # поиск остатка от деления,  
результат 1  
3 ** 2 # возведение в степень, результат  
9
```

Все действия с числами такие же, как в алгебре. Необычными могут показаться только два деления: целочисленное и деление с поиском остатка (еще его называют *делением по модулю*)

При целочисленном делении, первое число делится на второе, а остаток отбрасывается.

При делении с поиском остатка, наоборот, первое число делится на второе, и при этом целая часть выбрасывается, а ответом является - остаток.

Дробные числа. float

Дробные числа, или "числа с плавающей точкой". С этими числами можно применять все операторы, которые мы применяем с целыми числами.

Единственное, обратите внимание, что для отделения дробной части числа, мы используем **точку**, а не запятую. Например: 3.8

При операциях как с целыми числами, так и с дробными, операции, как и в математике, имеют свой приоритет. Например, умножение выполняется раньше сложения. Чтобы изменить порядок операций, как и в математике, мы можем использовать скобки.

Логический тип данных. bool

Значений этого типа всего два.

- **True** (истина)
- **False** (ложь)

Причём, в python эти слова пишутся *с большой буквы*.

Если к ним применить математические операторы, то истина преобразуется в 1, а ложь в 0. А значит, True + True + True будет равно 3.

В дальнейшем мы рассмотрим этот тип данных подробнее.

Строки. str

Строки это последовательность символов заключенная в кавычки. То есть, это обычный текст. И его мы должны заключать в кавычки. Кавычки могут быть одинарными и двойными. Главное, чтобы мы закрыли строку такой же кавычкой, какой открыли ее.

Например:

```
'Это строка'
```

```
"Это тоже строка"
```

```
' ' # это пустая строка, тоже бывает нужна
```

Здесь вы можете увидеть знак решетки # и за ним текст. Это **комментарий**. Python игнорирует комментарии. Они нужны нам, чтобы написать какие-то пояснения в коде, для самих себя, или для тех, кто будет читать наш код. Комментарии начинаются со знака решетки и продолжаются до конца строки кода.

Строки можно складывать друг с другом. При этом, получает другая строка, в которой первые две склеились. Например:

```
'привет' + 'друг' # получится  
'приветдруг'
```

Строки еще **можно умножать**. На целое число. В результате строка повторится столько раз. Например:

```
'Привет' * 3 # получится  
'ПриветПриветПривет'
```

Про строки еще можно много чего сказать, но пока, достаточно.