

# Теоретический материал

Теоретический материал к разделу "Типы данных"

Сайт: [Дистанционная подготовка](#)  
Курс: Д. П. Кириенко. Программирование на языке Python (школа 179 г. Москвы)  
Book: Теоретический материал  
Printed by: maung myo  
Date: Воскресенье 4 Март 2018, 01:14

## Table of Contents

---

[Интерактивный калькулятор](#)

[Типы данных](#)

[Преобразование типов](#)

Данный курс будет посвящен изучению программирования с использованием языка **Python**. Это — современный язык программирования, работающий на всех распространенных операционных системах.

В настоящее время существует две версии языка Python: более старая, но пока ещё более распространенная версия 2 и современная версия 3. Они не вполне совместимы друг с другом: программа, написанная для одной версии языка может оказаться невыполнимой для другой версии. Но в основном обе версии очень похожи.

Мы будем использовать версию 3 данного языка, некоторые из используемых примеров не будут работать с версией 2. Последняя версия языка, доступная в сентябре 2011 года — 3.2.2, именно её необходимо установить дома, скачав данную версию с сайта [www.python.org](http://www.python.org).

Запустить интерпретатор python можно из командной строки:

```
$ python3
```

Будьте внимательны — команда python запустит интерпретатор версии 2, с которым мы работать не будем. В системе Windows можно использовать пункт меню “Python (command line)”

Вы увидите примерно следующее приглашение командной строки:

```
Python 3.1.2 (r312:79147, Jun 12 2010, 15:29:06)
[GCC 4.4.3 20100316 (ALT Linux 4.4.3-alt2)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Смело вводите команды и наслаждайтесь результатом. А что можно вводить? Несколько примеров:

[illegible]

Первая команда вычисляет сумму двух чисел, вторая команда вычисляет 2 в степени 100, третья команда выполняет операцию **конкатенации** для строк, а четвертая команда печатает строку 'ABC', повторенную 100 раз.

Хотите закончить работу с питоном? Введите команду `exit()` (именно так, со скобочками, так как это — **функция**), или нажмите `Ctrl+D`.

## Типы данных

---

Итак, мы видим, что Питон умеет работать как минимум с двумя видами данных — числами и строками. Числа записываются последовательностью цифр, также перед числом может стоять знак минус, а строки записываются в одинарных кавычках. 2 и '2' — это разные объекты, первый объект — число, а второй — строка. Операция + для целых чисел и для строк работает по-разному: для чисел это сложение, а для строк — конкатенация.

Кроме целых чисел есть и другой класс чисел: действительные (вещественные числа), представляемые в виде десятичных дробей. Они записываются с использованием десятичной точки, например, 2.0. В каком-то смысле, 2 и 2.0 имеют равные значения, но это — разные объекты. Например, можно вычислить значения выражения 'ABC' \* 10 (повторить строку 10 раз), но нельзя вычислить 'ABC' \* 10.0.

Определить тип объекта можно при помощи функции type:

```
>>> type(2)
<class 'int'>
>>> type('2')
<class 'str'>
>>> type(2.0)
<class 'float'>
```

Обратите внимание — type является функцией, аргументы функции указываются в скобках после ее имени.

Вот список основных операций для чисел:

A + B — сумма;

A - B — разность;

A \* B — произведение;

A / B — частное;

A \*\* B — возведение в степень. Полезно помнить, что квадратный корень из числа x — это x \*\* 0.5, а корень степени n это x \*\* (1 / n).

Есть также унарный вариант операции -, то есть операция с одним аргументом. Она возвращает число, противоположное данному. Например: -A.

В выражении может встречаться много операций подряд. Как в этом случае определяется порядок действий? Например, чему будет равно 1 + 2 \* 3 \*\* 1 + 1? В данном случае ответ будет 8, так как сначала выполняется возведение в степень, затем — умножение, затем — сложение.

Более общие правила определения приоритетов операций такие:

1. Выполняются возведения в степень **справа налево**, то есть 3 \*\* 3 \*\* 3 это  $\backslash(3^{\{(3^3)\}}\backslash)$ .
2. Выполняются унарные минусы (отрицания).
3. Выполняются умножения и деления слева направо. Операции умножения и деления имеют одинаковый приоритет.
4. Выполняются сложения и вычитания слева направо. Операции сложения и вычитания имеют одинаковый приоритет.

Основные операции над строками:

A + B — конкатенация;

A \* n — повторение n раз, значение n должно быть целого типа.

## Преобразование типов

---

Иногда бывает полезно целое число записать, как строку. И, наоборот, если строка состоит из цифр, то полезно эту строку представить в виде числа, чтобы дальше можно было выполнять арифметические операции с ней. Для этого используются функции, одноименные с именем типа, то есть `int`, `float`, `str`. Например, `int('123')` вернет целое число 123, а `str(123)` вернет строку '123'.

Пример:

```
>>> str(2 + 2) * int('2' + '2')  
'44444444444444444444444444444444'
```

Результатом будет строка из числа 4, повторенная 22 раза.