

MLClass. "Прикладной анализ данных".

Курс "Инструментарий Data Science".

**Преподаватель ФКН НИУ ВШЭ Кашницкий
Юрий**



Урок 2. Основы языка Python

Часть 2. Числа

Числа в Python бывают разные:

- Целые числа (int): 122, -4, 999999999999, 0o177, 0x9ff, 0b101010
- Вещественные числа (float): 1.0, 3.14, .5, 4E21, 4.0e21
- Комплексные числа (complex): 3 + 4j, 3.0 + 4.0j,
- Числа фиксированной точности% decimal.Decimal('0.1')
- Рациональные числа: fractions.Fraction(3, 4)

Операции с целыми и вещественными числами

Оператор/ функция	Описание
$X + Y$	Сложение
$X - Y$	Вычитание
$X * Y$	Умножение
X / Y	Деление
$X // Y$	Деление с округлением вниз (до ближайшего меньшего)
$X \% Y$	Остаток от деления
<code>divmod(X,Y)</code>	Пара ($X // Y$, $X \% Y$)
$-X$	Смена знака числа
<code>abs(X)</code>	Модуль числа
$X ** Y$	Возведение в степень
<code>pow(X, Y[, Z])</code>	X в степени Y , [X в степени Y по модулю Z : вычисление остатка от деления числа X в степени Y на число Z (модуль)]
<code>X.as_integer_ratio()</code>	Возвращает пару целых чисел, чье отношение равно вещественному X
<code>X.is_integer()</code>	Проверка - является ли значение вещественного X целым числом
<code>int([строка-число], [основание системы счисления])</code>	Преобразование к целому числу в десятичной системе счисления числа в системе от 2 до 36 (по умолчанию в десятичной)
<code>bin(X)</code>	Преобразование целого числа в двоичную строку
<code>hex(X)</code>	Преобразование целого числа в шестнадцатеричную строку
<code>oct(X)</code>	Преобразование целого числа в восьмеричную строку

Целые числа

In [21]:

```
print(3+2)
```

5

In [22]:

```
print(3-2)
```

1

In [23]:

```
print(3*2)
```

6

In [24]:

```
print(3/2)
```

1

In [25]:

```
print(3**2)
```

9

Можно использовать скобки для управления порядком операций

In [26]:

```
standard_order = 2+3*4  
print(standard_order)  
my_order = (2+3)*4  
print(my_order)
```

14

20

Вещественные числа

In [27]:

```
print(0.1+0.1)
```

0.2

Но бывает и так:

In [28]:

```
print(0.1 + 0.2)
```

0.3

Это связано с внутренним представлением десятичных чисел в двоичной системе, может появиться и при других операциях.

In [29]:

```
print(3*0.1)
```

0.3

Целые числа в Python 2 и Python 3

В Python 2 результат деления двух чельх чисел - тоже целое, а в Python 3 - float.

In [30]:

```
# Python 2.7  
print 4/2
```

2

В Python 3 результат был бы 2.0

In [31]:

```
# Python 2.7  
print 3/2
```

1

В Python 3 результат был бы 1.5

Комплексные числа

In [6]:

```
print "|{0}| = {1}".format(complex(2, 3), abs(complex(2, 3)))  
print "{0} * {1} = {2}".format(complex(1, 1), complex(2, 3), complex(1, 1)  
* complex(2, 3))
```

```
| (2+3j) | = 3.60555127546  
(1+1j) * (2+3j) = (-1+5j)
```

Числа фиксированной точности и рациональные числа

Для решения проблем, связанных с точностью представления простых вещественных чисел введены вещественные числа с фиксированной точностью и рациональные числа (числа, представленные дробью, то есть парой целых чисел – числителем и знаменателем).

In [9]:

```
print(1.1/3)
```

0.3666666666667

In [10]:

```
from decimal import Decimal  
from decimal import getcontext  
getcontext().prec = 2  
print(Decimal('1.10') / 3)
```

0.37

In [11]:

```
print(7/71)
print(7/71*71 == 7)
```

0
False

In [12]:

```
from fractions import Fraction
print(Fraction(7, 71) * 71 == 7)
```

True

Приоритет арифметических операций в Python

Операция	Описание
**	Возведение в степень.
~, +, -	Комплиментарный оператор.
*, /, %, //	Умножение, деление, деление по модулю, целочисленное деление.
+, -	Сложение и вычитание.
>>, <<	Побитовые сдвиги.
&	Побитовое и.
^,	Побитовое исключающее или, побитовое или.
<, <=, >, >=	Сравнение.
==, <>, !=	Проверка на равенство.
=, -=, +=, *= и т.п.	Операторы присваивания.
is, is not	Проверка на тождественность.
in, not in	Проверка на принадлежность.
and, or, not	Логические операторы.

Полезные ссылки

- [сайт Python \(http://python.org/\)](http://python.org/)

Основной сайт Python, содержит исчерпывающую документацию.

- [StackOverflow \(http://stackoverflow.com/\)](http://stackoverflow.com/)

StackOverflow - популярный форум для вопросов и ответов о программировании. Решения для очень многих ошибок, появляющихся в процессе программирования, можно найти, используя поисковик и StackOverflow.

- [CodeAcademy \(https://www.codecademy.com/\)](https://www.codecademy.com/)

CodeAcademy - это простое интерактивное введение во многие языки, включая Python.