

Язык программирования

Python

03 Работа с числами



Знакомство с IDLE

Для введения в программирование на языке Python будем использовать интегрированную среду разработки IDLE.

Подсветка синтаксиса — это все, что нам нужно для счастья, и в IDLE она есть.

Подсветка синтаксиса облегчает чтение исходного кода программы путем выделения синтаксических конструкций различными цветами и шрифтами.

```
>>> a = input().upper() # Комментарий
Подсветка синтаксиса
>>> ф
Traceback (most recent call last):
  File "<pyshell#18>", line 1, in <module>
    ф
NameError: name 'ф' is not defined
>>> а
'ПОДСВЕТКА СИНТАКСИСА'
>>>
```

Комментарий

Комментарий — пояснение к исходному коду программы, написанное внутри этого кода.

- Комментарии не нужны интерпретатору, они нужны программисту.
- Комментарии в Python начинаются с символа решетки #, конец комментария — это конец физической строки.

Физическая строка — это то, что мы воспринимаем за одну строку.

Логическая строка — это то, что интерпретатор воспринимает за одну строку.

```
# Это комментарий
a = 1 # Это тоже комментарий
b = '# А это не комментарий, это строка'
```

Выражение

Давайте поработаем с интерпретатором.

- Запустим IDLE.
- После основного приглашения можно вводить команды.

Работа в интерактивном режиме может напоминать работу с калькулятором: вы вводите выражение и получаете значение (результат вычисления выражения).

Выражение — это оператор (инструкция), результат выполнения которого возвращает значение.

Выражения состоят из:

- операций (сложение, вычитание и так далее);
- операндов — это данные, над которыми выполняются операции;
- *вызовов функций.*

Число

Каждое значение в Python имеет тип. Рассмотрим типы, относящиеся к числам.

Числа в Python представлены следующими типами данных:

- `int` — это класс целых чисел (например, 1, 2, 100).
- `float` — это класс чисел с плавающей точкой (например, 1.3, 3.14159).
- *`complex` — это класс комплексных чисел (не в этом курсе).*
- *другие типы (не в этом курсе).*

Целое число

Целые числа в Python имеют тип `int` (от англ. `integer` — целое число).

```
>>> 2 + 2
```

```
4
```

```
>>> 99 * 99
```

```
9801
```

Деление (операция `/`) всегда возвращает `float`.

```
>>> 34 / 8
```

```
4.25
```

Целочисленное деление (операция `//`) позволяет получить целочисленный результат (дробная часть отбрасывается).

```
>>> 34 // 8
```

```
4
```

Число с плавающей точкой

Число с плавающей точкой — это экспоненциальная форма представления вещественных чисел.

Вещественные числа (например, 1.2, 3.14159) в Python имеют тип float (число с плавающей точкой).

Целая и дробная части разделяются точкой, а не запятой.

- 1.2 # Это число с плавающей точкой
- 1,2 # Это 2 целых числа, написанных через запятую

Вещественные числа также могут выглядеть следующим образом:

```
>>> .7 * 2. # То же, что и 0.7 * 2.0
1.4
```

Число с плавающей точкой

```
>>> 2.0 + 3.5  
5.5
```

```
>>> 2.1 + 1.3  
3.4000000000000004
```

Такой результат связан с тем, что несколько десятичных значений разделяют одно и то же приближение двоичного значения.

Арифметические операции

Операции, используемые в арифметических выражениях:

- **Сложение**

```
>>> 2 + 2
```

```
4
```

```
>>> 1. + 0.23
```

```
1.23
```

- **Вычитание**

```
>>> 10 - 15
```

```
-5
```

```
>>> 1.2 - 0.6
```

```
0.6
```

Арифметические операции

- Умножение

```
>>> 2 * 3
```

```
6
```

```
>>> 0.3 * 0.5
```

```
0.15
```

- Деление (результат операции — число с плавающей точкой)

```
>>> 9 / 3
```

```
3.0
```

- Целочисленное деление (результат операции — целое число)

```
>>> 9 // 3
```

```
3
```

Арифметические операции

Получение остатка от деления

```
>>> 5 % 2
```

```
1
```

```
>>> 16 % 4
```

```
0
```

Возведение в степень

```
>>> 2 ** 10
```

```
1024
```

```
>>> 1024 ** 10
```

```
1267650600228229401496703205376
```

Значение арифметического выражения

Если в качестве операндов некоторого арифметического выражения используются только целые числа, то результат тоже будет целое число. Исключением является операция деления, результатом которой является вещественное число.

При совместном использовании целых и вещественных чисел, результат будет вещественным.

```
>>> 1 + 2
```

```
3
```

```
>>> 1.0 + 2.0
```

```
3.0
```

```
>>> 1 + 2.0
```

```
3.0
```

Нижнее подчеркивание

У нижнего подчеркивания (`_`) в Python есть несколько применений:

1. Хранение значения последнего выражения

```
>>> 1024 ** 10
1267650600228229401496703205376
```

Представим, что мы забыли это число, например, умножить на 2. Переписывать это значение будет неприятным занятием, также возможны опечатки.

Чтобы избавить себя от страданий, воспользуемся нижним подчеркиванием.

```
>>> _ * 2
2535301200456458802993406410752
```

Нижнее подчеркивание содержит значение предыдущего выражения.

Выражения «`_ * 2`» и «`1267650600228229401496703205376 * 2`» равносильны.

Нижнее подчеркивание

2. Отделение цифр числа друг от друга

Отделение числовых разрядов друг от друга повышает читаемость длинных чисел.

Например:

```
>>> 1_000_000_000
```

Классы единиц, тысяч, миллионов, миллиардов; итого 1 миллион. Такая запись читается намного удобнее, чем 1000000000, где приходится «глазами» делить число на классы числовых разрядов.

Также на группы разрядов можно разбивать числа, представленные в двоичной и шестнадцатеричной системах счисления (например, 0b_1010_1010 и 0x_dead_beef соответственно).

У нижнего подчеркивания есть и другие способы применения, но о них в другой раз :) (не в этом курсе).

Приоритет арифметических операций

При использовании нескольких арифметических операций в выражении их выполнение производится в соответствии с их приоритетом. В начале выполняются операции с большим приоритетом.

Арифметические операции в порядке убывания их приоритета выполнения:

- $()$
- $**$
- $*, /, //, \%$
- $+, -$

Выполнение арифметических операций происходит слева направо. Исключением является операция возведения в степень ($**$), выполняемая справа налево.

`>>> 2 ** 4 ** 8 # То же, что и 2 ** (4 ** 8)`

Итоги

Мы познакомились с такими понятиями, как комментарий, выражение, значение выражения, операция, операнд, целое число, число с плавающей точкой, арифметические операции, приоритет операции,

а также рассмотрели 2 способа использования нижнего подчеркивания и попрактиковались в работе с интерпретатором в интерактивном режиме.