**Лекция 3.**

**Числа**

Числа в Python бывают трёх типов: целые, с плавающей точкой и комплексные.

* Примером целого числа может служить 2.
* Примерами чисел с плавающей точкой (или “плавающих” для краткости) могут быть 3.33 и 3.33Е-4. Обозначение E показывает степени числа 10.
* Примеры комплексных чисел: (-5+4j) и (5.5-4.4j)

**Строки**

Строка – это последовательность символов. Чаще всего строки – это просто некоторые наборы слов. После создания строки её больше нельзя изменять.

Слова могут быть как на английском языке, так и на любом другом, поддерживаемом стандартом Unicode, что означает почти на любом языке мира.

Способы ввода строки:

**A=’text’**

**A=”text”**

**A=input()**

В первых двух случаях переменная A станет строкой, содержащей текст text, причем одинарные кавычки ничем не будут отличаться от двойных. Во втором случае текст надо будет вводить с клавиатуры, причем даже в случае ввода чисел, переменная по-прежнему будет являться строкой, а не числом.

**>>> input()**

**1234**

**'1234'**

**>>> input()**

**Hello World!**

**'Hello World!'**

Чтобы строка ‘1234’ стала числом, используйте команду int.

**>>> A=int(input())**

**>>> A**

**1234**

**Операторы**

Большинство предложений (логических строк) в программах содержат выражения. Простой пример выражения: 2+3. Выражение можно разделить на операторы и операнды.

Операторы – это некий функционал, производящий какие-либо действия, который может быть представлен в виде символов, как например +, или специальных зарезервированных слов. Операторы могут производить некоторые действия над данными, и эти данные называются операндами. В нашем случае 2 и 3 – это операнды.

**Операторы и их применение**

|  |  |  |  |
| --- | --- | --- | --- |
| Оператор | Название | Объяснение | Примеры |
| + | Сложение | Суммирует два объекта | 3 + 5 даст 8; 'a' + 'b' даст 'ab' |
| - | Вычитание | Даёт разность двух чисел; если первый операнд отсутствует, он считается равным нулю | -5.2 даст отрицательное число, а 50 - 24 даст 26. |
| \* | Умножение | Даёт произведение двух чисел или возвращает строку, повторённую заданное число раз. | 2 \* 3 даст 6. 'la' \* 3 даст 'lalala'. |
| \*\* | Возведение в степень | Возвращает число х, возведённое в степень y | 3 \*\* 4 даст 81 (т.е. 3 \* 3 \* 3 \* 3) |
| / | Деление | Возвращает частное от деления x на y | 4 / 3 даст 1.3333333333333333. |
| // | Целочисленное деление | Возвращает неполное частное от деления | 4 // 3 даст 1. |
| % | Деление по модулю | Возвращает остаток от деления | 8 % 3 даст 2. -25.5 % 2.25 даст 1.5. |
| << | Сдвиг влево | Сдвигает биты числа влево на заданное количество позиций. (Любое число в памяти компьютера представлено в виде битов - или двоичных чисел, т.е. 0 и 1) | 2 << 2 даст 8. В двоичном виде 2 представляет собой 10. Сдвиг влево на 2 бита даёт 1000, что в десятичном виде означает 8. |
| >> | Сдвиг вправо | Сдвигает биты числа вправо на заданное число позиций. | 11 >> 1 даст 5. В двоичном виде 11 представляется как 1011, что будучи смещённым на 1 бит вправо, даёт 101, а это, в свою очередь, ни что иное как десятичное 5 |
| & | Побитовое И | Побитовая операция И над числами | 5 & 3 даёт 1. |
| | | Побитовое ИЛИ | Побитовая операция ИЛИ над числами | 5 | 3 даёт 7 |
| ^ | Побитовое ИСКЛЮЧИТЕЛЬНО ИЛИ | Побитовая операция ИСКЛЮЧИТЕЛЬНО ИЛИ | 5 ^ 3 даёт 6 |
| ~ | Побитовое НЕ | Побитовая операция НЕ для числа xсоответствует -(x+1) | ~5 даёт -6. |
| < | Меньше | Определяет, верно ли, что x меньше y. Все операторы сравнения возвращают True или False. Обратите внимание на заглавные буквы в этих словах. | 5 < 3 даст False,  а 3 < 5 даст True.  Можно составлять произвольные цепочки сравнений: 3 < 5 < 7 даёт True. |
| > | Больше | Определяет, верно ли, что x больше y | 5 > 3 даёт True. Если оба операнда - числа, то перед сравнением они оба преобразуются к одинаковому типу. В противном случае всегда возвращается False. |
| <= | Меньше или равно | Определяет, верно ли, что x меньше или равно y | x = 3; y = 6; x <= y даёт True. |
| >= | Больше или равно | Определяет, верно ли, что x больше или равно y | x = 4; y = 3; x >= 3 даёт True. |
| == | Равно | Проверяет, одинаковы ли объекты | x = 2; y = 2; x == y даёт True. x = 'str'; y = 'stR'; x == y даёт False. x = 'str'; y = 'str'; x == y даёт True. |
| != | Не равно | Проверяет, верно ли, что объекты не равны | x = 2; y = 3; x != y даёт True. |
| not | Логическое НЕ | Если x равно True, оператор вернёт False. Если же x равно False, получим True. | x = True; not x даёт False. |
| and | Логическое И | x and y даёт False, если x равно False , в противном случае возвращает значение y | x = False; y = True; x and y возвращает False, поскольку x равно False. В этом случае Python не станет проверять значение y, так как уже знает, что левая часть выражения ‘and’ равняется False, что подразумевает, что и всё выражение в целом будет равно False, независимо от значений всех остальных операндов. Это называется укороченной оценкой булевых (логических) выражений. |
| or | Логическое ИЛИ | Если x равно True, в результате получим True, в противном случае получим значение y | x = True; y = False; x or y даёт True. Здесь также может производиться укороченная оценка выражений. |