

Приоритетность. Операторы и их применение

Оператор	Название	Объяснение	Примеры
<code>+</code>	Сложение	Суммирует два объекта	<code>3 + 5</code> даст <code>8</code> ; <code>'a' + 'b'</code> даст <code>'ab'</code>
<code>-</code>	Вычитание	Даёт разность двух чисел; если первый операнд отсутствует, он считается равным нулю	<code>-5.2</code> даст отрицательное число, а <code>50 - 24</code> даст <code>26</code> .
<code>*</code>	Умножение	Даёт произведение двух чисел или возвращает строку, повторённую заданное число раз.	<code>2 * 3</code> даст <code>6</code> . <code>'la' * 3</code> даст <code>'lalala'</code> .
<code>**</code>	Возведение в степень	Возвращает число <code>x</code> , возведённое в степень <code>y</code>	<code>3 ** 4</code> даст <code>81</code> (т.е. <code>3 * 3 * 3 * 3</code>)
<code>/</code>	Деление	Возвращает частное от деления <code>x</code> на <code>y</code>	<code>4 / 3</code> даст <code>1.3333333333333333</code> .
<code>//</code>	Целочисленное деление	Возвращает неполное частное от деления	<code>4 // 3</code> даст <code>1</code> .
<code>%</code>	Деление по модулю	Возвращает остаток от деления	<code>8 % 3</code> даст <code>2</code> . <code>-25.5 % 2.25</code> даст <code>1.5</code> .

<<	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)	$2 \ll 2$ 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.	$11 \gg 1$ 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	$5 \wedge 3$ даёт 6
~	Побитовое НЕ	Побитовая операция НЕ для числа x соответствует $\neg(x+1)$	~ 5 даёт -6.
<	Меньше	Определяет, верно ли, что x меньше y. Все операторы сравнения возвращают True и False [1]. 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 <code>x</code> больше <code>y</code>	<code>5 > 3</code> даёт <code>True</code> . Если оба операнда - числа, то перед сравнением они оба преобразуются к одинаковому типу. В противном случае всегда возвращается <code>False</code> .
<=	Меньше или равно	Определяет, верно ли, что <code>x</code> меньше или равно <code>y</code>	<code>x = 3; y = 6; x <= y</code> даёт <code>True</code> .
>=	Больше или равно	Определяет, верно ли, что <code>x</code> больше или равно <code>y</code>	<code>x = 4; y = 3; x >= 3</code> даёт <code>True</code> .
==	Равно	Проверяет, одинаковы ли объекты	<code>x = 2; y = 2; x == y</code> даёт <code>True</code> . <code>x = 'str'; y = 'stR'; x == y</code> даёт <code>False</code> . <code>x = 'str'; y = 'str'; x == y</code> даёт <code>True</code> .
!=	Не равно	Проверяет, верно ли, что объекты не равны	<code>x = 2; y = 3; x != y</code> даёт <code>True</code> .
not	Логическое НЕ	Если <code>x</code> равно <code>True</code> , оператор вернёт <code>False</code> . Если же <code>x</code> равно <code>False</code> , получим <code>True</code> .	Open in Python Editor <code>x = True; not x</code> даёт <code>False</code> .
and	Логическое И	Open in Python Editor <code>x and y</code> даёт <code>False</code> , если <code>x</code> равно <code>False</code> , в противном случае возвращает значение <code>y</code>	Open in Python Editor <code>x = False; y = True; x and y</code> возвращает <code>False</code> , поскольку <code>x</code> равно <code>False</code> . В этом случае Python не станет проверять значение <code>y</code> , так как уже знает, что левая часть выражения 'and' равняется <code>False</code> , что подразумевает, что и всё выражение в целом будет равно <code>False</code> , независимо от значений всех остальных операндов. Это называется укороченной оценкой булевых (логических) выражений.
or	Логическое ИЛИ	Если <code>x</code> равно <code>True</code> , в результате получим <code>True</code> , в противном случае получим значение <code>y</code>	Open in Python Editor <code>x = True; y = False; x or y</code> даёт <code>True</code> . Здесь также может производиться укороченная оценка выражений.