

# Операторы в Python.

# Что такое оператор?

Говоря простым языком, в выражении  $2 + 3$ , числа "2" и "3" называются операндами, знак "+" оператором. В языке программирования Python существуют следующие типы операторов:

Арифметические операторы

Операторы сравнения (реляционные)

Операторы присваивания

Битовые операторы

Логические операторы

Операторы членства (Membership operators)

Операторы тождественности (Identity operators)

Рассмотрим их по порядку.

# Арифметические операторы в Python:

Оператор	Описание	Примеры
+	Сложение - Суммирует значения слева и справа от оператора	15 + 5 в результате будет 20 20 + -3 в результате будет 17 13.4 + 7 в результате будет 20.4
-	Вычитание - Вычитает правый операнд из левого	15 - 5 в результате будет 10 20 - -3 в результате будет 23 13.4 - 7 в результате будет 6.4
*	Умножение - Перемножает операнды	5 * 5 в результате будет 25 7 * 3.2 в результате будет 22.4 -3 * 12 в результате будет -36
/	Деление - Делит левый операнд на правый	15 / 5 в результате будет 3 5 / 2 в результате будет 2 (В Python 2.x версии при делении двух целых чисел результат будет целое число) 5.0 / 2 в результате будет 2.5 (Чтобы получить "правильный" результат хотя бы один операнд должен быть float)
%	Деление по модулю - Делит левый операнд на правый и возвращает остаток.	6 % 2 в результате будет 0 7 % 2 в результате будет 1 13.2 % 5 в результате 3.2
**	Возведение в степень - возводит левый операнд в степень правого	5 ** 2 в результате будет 25 2 ** 3 в результате будет 8 -3 ** 2 в результате будет -9
//	Целочисленное деление - Деление в котором возвращается только целая часть результата. Часть после запятой отбрасывается.	12 // 5 в результате будет 2 4 // 3 в результате будет 1 25 // 6 в результате будет 4

# Операторы сравнения в Python:

Оператор	Описание	Примеры
<code>==</code>	Проверяет равны ли оба операнда. Если да, то условие становится истинным.	<code>5 == 5</code> в результате будет <code>True</code> <code>True == False</code> в результате будет <code>False</code> <code>"hello" == "hello"</code> в результате будет <code>True</code>
<code>!=</code>	Проверяет равны ли оба операнда. Если нет, то условие становится истинным.	<code>12 != 5</code> в результате будет <code>True</code> <code>False != False</code> в результате будет <code>False</code> <code>"hi" != "Hi"</code> в результате будет <code>True</code>
<code>&lt;&gt;</code>	Проверяет равны ли оба операнда. Если нет, то условие становится истинным.	<code>12 &lt;&gt; 5</code> в результате будет <code>True</code> . Похоже на оператор <code>!=</code>
<code>&gt;</code>	Проверяет больше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.	<code>5 &gt; 2</code> в результате будет <code>True</code> . <code>True &gt; False</code> в результате будет <code>True</code> . <code>"A" &gt; "B"</code> в результате будет <code>False</code> .
<code>&lt;</code>	Проверяет меньше ли значение левого операнда, чем значение правого. Если да, то условие становится истинным.	<code>3 &lt; 5</code> в результате будет <code>True</code> . <code>True &lt; False</code> в результате будет <code>False</code> . <code>"A" &lt; "B"</code> в результате будет <code>True</code> .
<code>&gt;=</code>	Проверяет больше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.	<code>1 &gt;= 1</code> в результате будет <code>True</code> . <code>23 &gt;= 3.2</code> в результате будет <code>True</code> . <code>"C" &gt;= "D"</code> в результате будет <code>False</code> .
<code>&lt;=</code>	Проверяет меньше или равно значение левого операнда, чем значение правого. Если да, то условие становится истинным.	<code>4 &lt;= 5</code> в результате будет <code>True</code> . <code>0 &lt;= 0.0</code> в результате будет <code>True</code> . <code>-0.001 &lt;= -36</code> в результате будет <code>False</code> .

<b>%=</b>	Делит по модулю операнды и присваивает результат левому.	$c = 5$ $a = 2$ $c \% a$ равносильно: $c = c \% a$ . $c$ будет равно 1
<b>**=</b>	Возводит в левый операнд в степень правого и присваивает результат левому операнду.	$c = 3$ $a = 2$ $c ** a$ равносильно: $c = c ** a$ . $c$ будет равно 9
<b>//=</b>	Производит целочисленное деление левого операнда на правый и присваивает результат левому операнду.	$c = 11$ $a = 2$ $c // a$ равносильно: $c = c // a$ . $c$ будет равно 5

# Побитовые операторы в Python:

Побитовые операторы предназначены для работы с данными в битовом (двоичном) формате. Предположим, что у нас есть два числа  $a = 60$ ; и  $b = 13$ . В двоичном формате они будут иметь следующий вид:

$a = 0011\ 1100$

$b = 0000\ 1101$

Оператор	Описание	Примеры
&	Бинарный "И" оператор, копирует бит в результат только если бит присутствует в обоих операндах.	(a & b) даст нам 12, которое в двоичном формате выглядит так 0000 1100
	Бинарный "ИЛИ" оператор копирует бит, если тот присутствует в хотя бы в одном операнде.	(a   b) даст нам 61, в двоичном формате 0011 1101
^	Бинарный "Исключительное ИЛИ" оператор копирует бит только если бит присутствует в одном из операндов, но не в обоих сразу.	(a ^ b) даст нам 49, в двоичном формате 0011 0001
~	Бинарный комплиментарный оператор. Является унарным (то есть ему нужен только один операнд) меняет биты на обратные, там где была единица становится ноль и наоборот.	(~a) даст в результате -61, в двоичном формате выглядит 1100 0011.
<<	Побитовый сдвиг влево. Значение левого операнда "сдвигается" влево на количество бит указанных в правом операнде.	a << 2 в результате даст 240, в двоичном формате 1111 0000
>>	Побитовый сдвиг вправо. Значение левого операнда "сдвигается" вправо на количество бит указанных в правом операнде.	a >> 2 даст 15, в двоичном формате 0000 1111

# Логические операторы в Python:

Оператор	Описание	Примеры
<b>and</b>	Логический оператор "И". Условие будет истинным если оба операнда истина.	True and True равно True. True and False равно False. False and True равно False. False and False равно False.
<b>or</b>	Логический оператор "ИЛИ". Если хотя бы один из операндов истинный, то и все выражение будет истинным.	True or True равно True. True or False равно True. False or True равно True. False or False равно False.
<b>not</b>	Логический оператор "НЕ". Изменяет логическое значение операнда на противоположное.	not True равно False. not False равно True.

# Операторы членства в Python:

В добавок к перечисленным операторам, в Python присутствуют, так называемые, операторы членства, предназначенные для проверки на наличие элемента в составных типах данных, таких, как строки, списки, кортежи или словари:

Оператор	Описание	Примеры
<b>in</b>	Возвращает истину, если элемент присутствует в последовательности, иначе возвращает ложь.	"cad" in "cadillac" вернет True. 1 in [2,3,1,6] вернет True. "hi" in {"hi":2,"bye":1} вернет True. 2 in {"hi":2,"bye":1} вернет False (в словарях проверяется наличие в ключах, а не в значениях).
<b>not in</b>	Возвращает истину если элемента нет в последовательности.	Результаты противоположны результатам оператора in.

# Операторы тождественности в Python:

Операторы тождественности сравнивают размещение двух объектов в памяти компьютера.

Оператор	Описание	Примеры
<b>is</b>	Возвращает истину, если оба операнда указывают на один объект.	<code>x is y</code> вернет истину, если <code>id(x)</code> будет равно <code>id(y)</code> .
<b>is not</b>	Возвращает ложь если оба операнда указывают на один объект.	<code>x is not y</code> , вернет истину если <code>id(x)</code> не равно <code>id(y)</code> .

# Приоритет операторов в Python:

В следующей таблице описан приоритет выполнения операторов в Python от наивысшего (выполняется в первую очередь) до наинизшего.

Оператор	Описание
<b>**</b>	Возведение в степень
<b>~ + -</b>	Комплиментарный оператор
<b>* / % //</b>	Умножение, деление, деление по модулю, целочисленное деление.
<b>+ -</b>	Сложение и вычитание.
<b>&gt;&gt; &lt;&lt;</b>	Побитовый сдвиг вправо и побитовый сдвиг влево.
<b>&amp;</b>	Бинарный "И".
<b>^  </b>	Бинарный "Исключительное ИЛИ" и бинарный "ИЛИ"
<b>&lt;= &lt; &gt; &gt;=</b>	Операторы сравнения
<b>&lt;&gt; == !=</b>	Операторы равенства
<b>= %= /= //= -= += *= **=</b>	Операторы присваивания
<b>is is not</b>	Тождественные операторы
<b>in not in</b>	Операторы членства
<b>not or and</b>	Логические операторы

## Арифметические операторы в Python:

```
print(10 + 20)
print(10 * 20)
print(20 - 10)
print(20 / 10)
```

## Операторы сравнения в Python:

```
print(10 == 10)
print(12 != 5)
print(20 > 10)
print(20 >= 20)
```

## Операторы присваивания в Python:

```
hello = "Здравствуйте."
hello2 = hello
print(hello2)
```

## Логические операторы в Python:

```
print(True and True)
print(True and False)
print(False and True)
print(False and False)
```

```
print(True or True)
print(True or False)
print(False or True)
print(False or False)
```

```
print(not True)
print(not False)
```

## Операторы членства в Python:

```
print("cad" in "cadillac")
print(1 in [2, 3, 1, 6])
print("hi" in {"hi": 2, "bye": 1})
print(2 in {"hi": 2, "bye": 1})
```

```
print("cad" not in "cadillac")
print(1 not in [2, 3, 1, 6])
print("hi" not in {"hi": 2, "bye": 1})
print(2 not in {"hi": 2, "bye": 1})
```

```
text = input()
if "плох" not in text:
    print("В тексте нет ничего
плохого")
else:
    print("Найдены негативные слова")
```

Операторы тождественности в Python:  
Операторы тождественности сравнивают  
размещение двух объектов в памяти  
компьютера.

Оператор	Описание	Примеры
<b>is</b>	Возвращает истину, если оба операнда указывают на один объект.	x is y вернет истину, если id(x) будет равно id(y).
<b>is not</b>	Возвращает ложь если оба операнда указывают на один объект.	x is not y, вернет истину если id(x) не равно id(y).

```
x = 1
y = 1
print(x is y)
```

```
x = 1
y = 2
print(x is not y)
```

# Приоритет операторов в Python:

В следующей таблице описан приоритет выполнения операторов в Python от наивысшего (выполняется в первую очередь) до наимизшего.

Оператор	Описание
**	Возведение в степень
~ + -	Комплиментарный оператор
* / % //	Умножение, деление, деление по модулю, целочисленное деление.
+ -	Сложение и вычитание.
>> <<	Побитовый сдвиг вправо и побитовый сдвиг влево.
&	Бинарный "И".
^	Бинарный "Исключительное ИЛИ" и бинарный "ИЛИ"
<= < > >=	Операторы сравнения
<> == !=	Операторы равенства
= %= /= //= -= += *= **=	Операторы присваивания
is is not	Тождественные операторы
in not in	Операторы членства
not or and	Логические операторы