



Python 3.6.2 documentation

Ausblenden Suchen Zurück Vorwärts Startseite Schriftart Drucken Optionen

Inhalt Index Suchen Favoriten

Zu suchendes Schlüsselwort:

operator

// operator
//= augmented assignment
/= augmented assignment
2to3
:package namespace portion
< operator
<< operator
<<= augmented assignment
<= operator
<protocol>_proxy
= assignment statement
== operator
> operator
>= operator
>> operator
>>= augmented assignment
>>> @

Anzeigen

| Operation | Result | Notes | Full documentation |
|------------------------------|---|--------|------------------------|
| $x + y$ | sum of x and y | | |
| $x - y$ | difference of x and y | | |
| $x * y$ | product of x and y | | |
| x / y | quotient of x and y | | |
| $x // y$ | floored quotient of x and y | (1) | |
| $x \% y$ | remainder of x / y | (2) | |
| $-x$ | x negated | | |
| $+x$ | x unchanged | | |
| <code>abs(x)</code> | absolute value or magnitude of x | | <code>abs()</code> |
| <code>int(x)</code> | x converted to integer | (3)(6) | <code>int()</code> |
| <code>float(x)</code> | x converted to floating point | (4)(6) | <code>float()</code> |
| <code>complex(re, im)</code> | a complex number with real part re , imaginary part im . im defaults to zero. | (6) | <code>complex()</code> |
| <code>c.conjugate()</code> | conjugate of the complex number c | | |
| <code>divmod(x, y)</code> | the pair $(x // y, x \% y)$ | (2) | <code>divmod()</code> |
| <code>pow(x, y)</code> | x to the power y | (5) | <code>pow()</code> |
| $x ** y$ | x to the power y | (5) | |



Python 3.6.2 documentation

Ausblenden Suchen Zurück Vorwärts Startseite Schriftart Drucken Optionen

Inhalt Index Suchen Favoriten

Zu suchendes Schlüsselwort:

operator

&
operator
&=
augmented assignment
in expression lists
in function calls
operator
statement
**
in dictionary displays
in function calls
operator
statement
**=
augmented assignment
+=
augmented assignment
+
operator
+=
augmented assignment
-
operator
-create <tarfile> <source1> ... <sourceN>
tarfile command line option
-details
inspect command line option
-extract <tarfile> [<output_dir>]
tarfile command line option
-help
command line option
trace command line option
inspect dir/dir

Anzeigen

4.4.1. Bitwise Operations on Integer Types

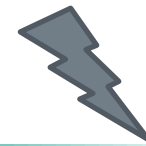
Bitwise operations only make sense for integers. Negative numbers are treated as their 2's complement value (this assumes that there are enough bits so that no overflow occurs during the operation).

The priorities of the binary bitwise operations are all lower than the numeric operations and higher than the comparisons; the unary operation `~` has the same priority as the other unary numeric operations (`+` and `-`).

This table lists the bitwise operations sorted in ascending priority:

| Operation | Result | Notes |
|---------------------------|--|--------|
| <code>x y</code> | bitwise <i>or</i> of <i>x</i> and <i>y</i> | |
| <code>x ^ y</code> | bitwise <i>exclusive or</i> of <i>x</i> and <i>y</i> | |
| <code>x & y</code> | bitwise <i>and</i> of <i>x</i> and <i>y</i> | |
| <code>x << n</code> | <i>x</i> shifted left by <i>n</i> bits | (1)(2) |
| <code>x >> n</code> | <i>x</i> shifted right by <i>n</i> bits | (1)(3) |
| <code>~x</code> | the bits of <i>x</i> inverted | |

Notes:



| Bit Position Berechnung Wert Wert | Bits | | | | | | | |
|---|---------------------------|-------|-------|-------|-------|---------------|---------------|---------------|
| | 4 | 3 | 2 | 1 | 0 | -1 | -2 | -3 |
| | 2^4 | 2^3 | 2^2 | 2^1 | 2^0 | 2^{-1} | 2^{-2} | 2^{-3} |
| | 16 | 8 | 4 | 2 | 1 | $\frac{1}{2}$ | $\frac{1}{4}$ | $\frac{1}{8}$ |
| Beispiel | 9 | 0 | 1 | 0 | 0 | 1 | | |
| | $(9 \ll 1) = 18$ | 1 | 0 | 0 | 1 | 0 | | |
| | $(18 \gg 2) = 4$ | 0 | 0 | 1 | 0 | 0 | | |
| | 6 | 0 | 0 | 1 | 1 | 0 | | |
| | $4 \& 6 = 4$ | 0 | 0 | 1 | 0 | 0 | | |
| | $4 6 = 6$ | 0 | 0 | 1 | 1 | 0 | | |
| | $4 \wedge 6 = 2$ | 0 | 0 | 0 | 1 | 0 | | |
| | $4 \wedge 6 \wedge 6 = 4$ | 0 | 0 | 1 | 0 | 0 | | |

| Bitweise Verknüpfungen | | |
|---|--|---|
| UND | ODER | XOR |
| $\begin{array}{c cc} \& & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$ | $\begin{array}{c cc} & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array}$ | $\begin{array}{c cc} \wedge & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array}$ |

NOT

| | |
|--------|---|
| \sim | |
| 0 | 1 |
| 1 | 0 |

$\sim 1 = -2$

