Operator Precedence All Versions

Python operators have a set **order of precedence** , which determines what operators are evaluated first in a potentially ambiguous expression. For instance, in the expression 3 * 2 + 7, first 3 is multiplied by 2, and then the result is added to 7, yielding 13. The expression is not evaluated the other way around, because * has a higher precedence than +.

Below is a list of operators by precedence, and a brief description of what they (usually) do.

Examples

Improvements requested:

Simple Operator Precedence Examples in python.

Python follows PEMDAS rule. PEMDAS stands for Parentheses, Exponents, Multiplication and Division, and Addition and Subtraction

Here's an example of operator precedence in action using the Python docs table in the Remarks section below.

```
>>> (a + b) * c / d       # (30 * 15) / 5
90

>>> ((a + b) * c) / d     # ((30 * 15) / 5
90

>>> (a + b) * (c / d);    # (30) * (15 / 5)
90

>>> a + (b * c) / d;      #  20 + (150 / 5)
50

>>> (a + b) * c ** d;     #  30 * (15 ^ 5)
50
```

Syntax

Parameters

Remarks

From the Python documentation:

> The following table summarizes the operator precedences in Python, from lowest precedence (least binding) to highest precedence (most binding). Operators in the same box have the same precedence. Unless the syntax is explicitly given, operators are binary. Operators in the same box group left to right (except for comparisons, including tests, which all have the same precedence and chain from left to right and exponentiation, which groups from right to left).

Operator	Description
lambda	Lambda expression
if – else	Conditional expression
or	Boolean OR
and	Boolean AND
not x	Boolean NOT
in, not in, is, is not, <, <=, >, >=, <>, !=, ==	Comparisons, including membership tests and identity tests
|	Bitwise OR
^	Bitwise XOR

Operator	Description
&	Bitwise AND
<<, >>	Shifts
+, -	Addition and subtraction
*, /, //, %	Multiplication, division, remainder [8]
+x, -x, ~x	Positive, negative, bitwise NOT
**	Exponentiation [9]
x[index], x[index:index], x(arguments...), x.attribute	Subscription, slicing, call, attribute reference
(expressions...), [expressions...], {key: value...}, expressions...	Binding or tuple display, list display, dictionary display, string conversion