

Python Expressions

An expression is a combination of values, variables, operators and calls to functions. Expressions need to be evaluated. If you ask Python to print an expression, the interpreter evaluates the expression and displays the result.

Order of Evaluation - lowest to highest

- 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
- & : Bitwise AND
- <<, >> : Shifts
- +, - : Addition and Subtraction
- *, /, //, % : Multiplication, Division, Floor Division and Remainder
- +x, -x, ~x : Positive, Negative, Bitwise NOT
- ** : Exponentiation
- 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, set display

Operators with same precedence listed on same row. Evaluation of operators with same precedence is left to right.

Use parentheses to force order of evaluation. Using parentheses can also make an expression more clear and readable.