

GSET - Intro to Programming with Python

Tristan Hill

Tennessee Technological University

Summer 2023

Module 3 - Operators and Expressions

Module 3 - Operators and Expressions

- Operators in Python
- Arithmetic Operators
- Operator Precedence
- What is an Expression?
- Tutorial 2 - Quadratic Equation - Solution

Operators in Python

There are *many different* operators used in computer programming.

Python Operations, Operators, and Operator Functions

Operation	Syntax	Function
Addition	<code>a + b</code>	<code>add(a, b)</code>
Concatenation	<code>seq1 + seq2</code>	<code>concat(seq1, seq2)</code>
Containment Test	<code>obj in seq</code>	<code>contains(seq, obj)</code>
Division	<code>a / b</code>	<code>truediv(a, b)</code>
Division	<code>a // b</code>	<code>floordiv(a, b)</code>
Bitwise And	<code>a & b</code>	<code>and_(a, b)</code>
Bitwise Exclusive Or	<code>a ^ b</code>	<code>xor(a, b)</code>
Bitwise Inversion	<code>~ a</code>	<code>invert(a)</code>
Bitwise Or	<code>a b</code>	<code>or_(a, b)</code>
Exponentiation	<code>a ** b</code>	<code>pow(a, b)</code>
Identity	<code>a is b</code>	<code>is_(a, b)</code>
Identity	<code>a is not b</code>	<code>is_not(a, b)</code>
Indexed Assignment	<code>obj[k] = v</code>	<code>setitem(obj, k, v)</code>
Indexed Deletion	<code>del obj[k]</code>	<code>delitem(obj, k)</code>
Indexing	<code>obj[k]</code>	<code>getitem(obj, k)</code>
Left Shift	<code>a << b</code>	<code>lshift(a, b)</code>
Modulo	<code>a % b</code>	<code>mod(a, b)</code>

Multiplication	<code>a * b</code>	<code>mul(a, b)</code>
Matrix Multiplication	<code>a @ b</code>	<code>matmul(a, b)</code>
Negation (Arithmetic)	<code>- a</code>	<code>neg(a)</code>
Negation (Logical)	<code>not a</code>	<code>not_(a)</code>
Positive	<code>+ a</code>	<code>pos(a)</code>
Right Shift	<code>a >> b</code>	<code>rshift(a, b)</code>
Slice Assignment	<code>seq[i:j] = values</code>	<code>setitem(seq, slice(i, j), values)</code>
Slice Deletion	<code>del seq[i:j]</code>	<code>delitem(seq, slice(i, j))</code>
Slicing	<code>seq[i:j]</code>	<code>getitem(seq, slice(i, j))</code>
String Formatting	<code>s % obj</code>	<code>mod(s, obj)</code>
Subtraction	<code>a - b</code>	<code>sub(a, b)</code>
Truth Test	<code>obj</code>	<code>truth(obj)</code>
Ordering	<code>a < b</code>	<code>lt(a, b)</code>
Ordering	<code>a <= b</code>	<code>le(a, b)</code>
Equality	<code>a == b</code>	<code>eq(a, b)</code>
Difference	<code>a != b</code>	<code>ne(a, b)</code>
Ordering	<code>a >= b</code>	<code>ge(a, b)</code>
Ordering	<code>a > b</code>	<code>gt(a, b)</code>

reference: docs.python.org

Operators in Python

Commonly used operators

Category	Operators	Example
Assignment	=	
Arithmetic	+ - * / % **	
Relational	== != > < >= <=	
Logical	! &&	
Bitwise	& ^ ~ << >>	

The functions fall into categories that perform object comparisons, logical operations, mathematical operations and sequence operations -python.org.

Arithmetic Operators

Mathematical Operations in Python

Operator	Name	Example
+	Addition	$x+y$
-	Subtraction	$x-y$
*	Multiplication	$x*y$
/	Division	x/y
%	Modulus	$x\%y$
**	Exponentiation	$x**y$
//	Floor Division	$x//y$

Arithmetic Operators

Which one of these is not like the other?

= + - * / % **

Why?

Operator Precedence

The interpreter must assume an order to perform operations. The *operator precedence* for python is shown in the table. The top of the table has highest precedence.

Operator	Description
<code>(expressions...),</code>	Binding or parenthesized expression, list display, dictionary display, set display
<code>[expressions...], {key: value...}, {expressions...}</code>	
<code>x[index], x[index:index], x(arguments...), x.attribute</code>	Subscription, slicing, call, attribute reference
<code>await x</code>	Await expression
<code>**</code>	Exponentiation [5]
<code>+, -, ~</code>	Positive, negative, bitwise NOT
<code>*, @, /, //, %</code>	Multiplication, matrix multiplication, division, floor division, remainder [6]
<code>+, -</code>	Addition and subtraction
<code><<, >></code>	Shifts
<code>&</code>	Bitwise AND
<code>^</code>	Bitwise XOR
<code> </code>	Bitwise OR
<code>in, not in, is, is not, <, <=, >, >=, !=, ==</code>	Comparisons, including membership tests and identity tests
<code>not x</code>	Boolean NOT
<code>and</code>	Boolean AND
<code>or</code>	Boolean OR
<code>if - else</code>	Conditional expression
<code>lambda</code>	Lambda expression
<code>:</code>	Assignment expression

Operator Precedence

Question: What was the concept of operator precedence called in mathematics class?

Answer:

What is an Expression?

An expression is a sequence of operators and their operands, that specifies a computation.

Tutorial 2 - Quadratic Equation - Solution