

PYTHON OPERATOR SUMMARY

The Python operators are listed in groups of decreasing precedence in the table below. The horizontal lines in the table indicate a change in operator precedence. Operators with higher precedence bind more strongly than those with lower precedence. For example, $x + y * z$ means $x + (y * z)$ because the $*$ operator has higher precedence than the $+$ operator. Looking at the table below, you can tell that x and y or z means $(x \text{ and } y)$ or z because the `or` operator has lower precedence.

The *associativity* of an operator indicates whether it groups left to right, or right to left. All operators in Python have left to right associativity except exponentiation, which has right to left associativity. For example, the $-$ operator binds left to right. Therefore, $x - y - z$ means $(x - y) - z$. But the $**$ operator binds right to left, and $x ** y ** z$ means $x ** (y ** z)$.

Operator	Description	Reference Location
[]	Index	Section 2.4.4, Section 6.1.2
[:]	Slice operator	Special Topic 6.3
()	Function call or method call	Section 2.2.4
.	Method or instance variable access	Section 2.4.5, Section 9.2
**	Exponentiation	Section 2.2.2
+ (<i>unary</i>)	Positive	Section 2.2.1
- (<i>unary</i>)	Negative	Section 2.2.1
~ (<i>unary</i>)	Bitwise <i>not</i>	Appendix E
*	Multiplication	Section 2.2.1
/	Real division	Section 2.2.1
//	Floor division	Section 2.2.3
%	Integer remainder	Section 2.2.3
*	Sequence replication	Section 2.4.2
+	Addition	Section 2.2.1
+	Sequence concatenation	Section 2.4.2
-	Subtraction	Section 2.2.1

A-2 Appendix A Python Operator Summary

Operator	Description	Reference Location
<<	Bitwise shift left	Appendix E
>>	Bitwise shift right	Appendix E
&	Bitwise <i>and</i>	Appendix E
^	Bitwise exclusive <i>or</i>	Appendix E
	Bitwise <i>or</i>	Appendix E
in	Membership test: in container	Section 3.8, Section 6.3.5
not in	Membership test: not in container	Section 3.8, Section 6.3.5
is	Identity test: is an alias	Section 9.10.1
is not	Identity test: is not an alias	Section 9.10.1
<	Less than	Section 3.2
<=	Less than or equal	Section 3.2
>	Greater than	Section 3.2
>=	Greater than or equal	Section 3.2
!=	Not equal	Section 3.2
==	Equal	Section 3.2
not	Boolean <i>not</i>	Section 3.7
and	Boolean “short circuit” <i>and</i>	Section 3.7
or	Boolean “short circuit” <i>or</i>	Section 3.7
if - else	Conditional expression	Special Topic 3.1
lambda	Anonymous function	Not covered