Operators

Arithmetic Operators

+ (addition)
Returns the sum of two expressions.

- (subtraction)
Returns the difference of two expressions.

*** (multiplication)**
Returns the product of two expressions.

**** (power)**
Returns the value of a numeric expression raised to a specified power.

/ (division)
Returns the quotient of two expressions.

// (floor division)
Returns the integral part of the quotient.

% (modulus)
Returns the decimal part (remainder) of the quotient.

Assignment Operators

= (simple assignment)
Assigns a value to a variable(s).

+= (increment assignment)
Adds a value and the variable and assigns the result to that variable.

-= (decrement assignment)
Subtracts a value from the variable and assigns the result to that variable.

***= (multiplication assignment)**
Multiplies the variable by a value and assigns the result to that variable.

/= (division assignment)
Divides the variable by a value and assigns the result to that variable.

****= (power assignment)**
Raises the variable to a specified power and assigns the result to the variable.

%= (modulus assignment)
Computes the modulus of the variable and a value and assigns the result to that variable.

//= (floor division assignment)
Floor divides the variable by a value and assigns the result to that variable.

Relational Operators

== (equal)
Returns a Boolean stating whether two expressions are equal.

!= (not equal)
Returns a Boolean stating whether two expressions are not equal.

> (greater than)
Returns a Boolean stating whether one expression is greater than the other.

>= (greater than or equal)
Returns a Boolean stating whether one expression is greater than or equal the other.

< (less than)
Returns a Boolean stating whether one expression is less than the other.

<= (less than or equal)
Returns a Boolean stating whether one expression is less than or equal the other.

Boolean Operators

and
Returns the first operand that evaluates to *False* or the last one if all are *True*.

or
Returns the first operand that evaluates to *True* or the last one if all are *False*.

not
Returns a boolean that is the reverse of the logical state of an expression.

Conditional Operator

if else
Returns either value depending on the result of a Boolean expression.

Identity

is
Returns a Boolean stating whether two objects are the same.

Membership

in
Returns a Boolean stating whether the object is in the container.

Deletion

`del`_
Removes object.

Callables Operators

*** (tuple packing)**
Packs the consecutive function positional arguments into a tuple.

**** (dictionary packing)**
Packs the consecutive function keyword arguments into a dictionary.

*** (tuple unpacking)**
Unpacks the contents of a tuple into the function call.

**** (dictionary unpacking)**
Unpacks the contents of a dictionary into the function call.

@ (decorator)
Returns a callable wrapped by another callable.

() (call operator)
Calls a callable object with specified arguments.

lambda
Returns an anonymous function.

Bitwise Operators

& (bitwise AND)
Returns the result of bitwise AND of two integers.

| (bitwise OR)
Returns the result of bitwise OR of two integers.

^ (bitwise XOR)
Returns the result of bitwise XOR of two integers.

<< (left shift)
Shifts the bits of the first operand left by the specified number of bits.

>> (right shift)
Shifts the bits of the first operand right by the specified number of bits.

~ (bitwise complement)
Sets the 1 bits to 0 and 1 to 0 and then adds 1.

Bitwise Assignment Operators

&= (bitwise AND assignment)
Performs bitwise AND and assigns value to the left operand.

|= (bitwise OR assignment)
Performs bitwise OR and assigns value to the left operand.

^= (bitwise XOR assignment)
Performs bitwise XOR and assigns value to the left operand.

<<= (bitwise right shift assignment)
Performs bitwise left shift and assigns value to the left operand.

>>= (bitwise left shift assignment)
Performs bitwise right shift and assigns value to the left operand.

Misc

; (statement separator)
Separates two statements.

(line continuation)
Breaks the line of code allowing for the next line continuation.

. (attribute access)
Gives access to an object's attribute.

String and Sequence Operators

+ (concatenation)
Returns a concatenation of two sequences.

*** (multiple concatenation)**
Returns a sequence self-concatenated specified amount of times.

% (string formatting operator)
Formats the string according to the specified format.

Sequence Assignment Operators

+= (concatenation assignment)
Concatenates the sequence with the right operand and assigns the result to that sequence.

***= (multiple concatenation assignment)**
Multiple concatenates the sequence and assigns the result to that sequence.