|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Python Operators | |  |  |  |
|  |  |  |  |  |
| Operators are used to perform operations on variables and values. | | | |  |
|  |  |  |  |  |
| In the example below, we use the + operator to add together two values: | | | |  |
|  |  |  |  |  |
| [ExampleGet your own Python Server](https://www.w3schools.com/python/python_server.asp) | |  |  |  |
| print(10 + 5) | |  |  |  |
| [Run example »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper) |  |  |  |  |
|  |  |  |  |  |
| Python divides the operators in the following groups: | | |  |  |
|  |  |  |  |  |
| Arithmetic operators | |  |  |  |
| Assignment operators | |  |  |  |
| Comparison operators | |  |  |  |
| Logical operators | |  |  |  |
| Identity operators | |  |  |  |
| Membership operators | |  |  |  |
| Bitwise operators | |  |  |  |
|  |  |  |  |  |
| Python Arithmetic Operators | | |  |  |
|  |  |  |  |  |
| Arithmetic operators are used with numeric values to perform common mathematical operations: | | | | |
|  |  |  |  |  |
| **Operator** | **Name** | **Example** | **Try it** |  |
| + | Addition | x + y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_add) |  |
| - | Subtraction | x - y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_sub) |  |
| \* | Multiplication | x \* y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_mult) |  |
| / | Division | x / y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_div) |  |
| % | Modulus | x % y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_mod) |  |
| \*\* | Exponentiation | x \*\* y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_exp) |  |
| // | Floor division | x // y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_floordiv) |  |
|  |  |  |  |  |
| Python Assignment Operators | | |  |  |
|  |  |  |  |  |
| Assignment operators are used to assign values to variables: | | |  |  |
|  |  |  |  |  |
| **Operator** | **Example** | **Same As** | **Try it** |  |
| = | x = 5 | x = 5 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass1) |  |
| += | x += 3 | x = x + 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass2) |  |
| -= | x -= 3 | x = x - 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass3) |  |
| \*= | x \*= 3 | x = x \* 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass4) |  |
| /= | x /= 3 | x = x / 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass5) |  |
| %= | x %= 3 | x = x % 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass6) |  |
| //= | x //= 3 | x = x // 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass7) |  |
| \*\*= | x \*\*= 3 | x = x \*\* 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass8) |  |
| &= | x &= 3 | x = x & 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass9) |  |
| |= | x |= 3 | x = x | 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass10) |  |
| ^= | x ^= 3 | x = x ^ 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass11) |  |
| >>= | x >>= 3 | x = x >> 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass12) |  |
| <<= | x <<= 3 | x = x << 3 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_ass13) |  |
|  |  |  |  |  |
| ADVERTISEMENT | |  |  |  |
|  |  |  |  |  |
| Python Comparison Operators | | |  |  |
|  |  |  |  |  |
| Comparison operators are used to compare two values: | | |  |  |
|  |  |  |  |  |
| **Operator** | **Name** | **Example** | **Try it** |  |
| == | Equal | x == y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare1) |  |
| != | Not equal | x != y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare2) |  |
| > | Greater than | x > y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare4) |  |
| < | Less than | x < y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare5) |  |
| >= | Greater than or equal to | x >= y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare6) |  |
| <= | Less than or equal to | x <= y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_compare7) |  |
|  |  |  |  |  |
| Python Logical Operators | | |  |  |
|  |  |  |  |  |
| Logical operators are used to combine conditional statements: | | | |  |
|  |  |  |  |  |
| **Operator** | **Description** | **Example** | **Try it** |  |
| and | Returns True if both statements are true | x < 5 and  x < 10 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_logical1) |  |
| or | Returns True if one of the statements is true | x < 5 or x < 4 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_logical2) |  |
| not | Reverse the result, returns False if the result is true | not(x < 5 and x < 10) | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_logical3) |  |
|  |  |  |  |  |
| Python Identity Operators | | |  |  |
|  |  |  |  |  |
| Identity operators are used to compare the objects, not if they are equal, but if they are actually the same object, with the same memory location: | | | | |
|  |  |  |  |  |
| **Operator** | **Description** | **Example** | **Try it** |  |
| is | Returns True if both variables are the same object | x is y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_identity1) |  |
| is not | Returns True if both variables are not the same object | x is not y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_identity2) |  |
|  |  |  |  |  |
| Python Membership Operators | | |  |  |
|  |  |  |  |  |
| Membership operators are used to test if a sequence is presented in an object: | | | | |
|  |  |  |  |  |
| **Operator** | **Description** | **Example** | **Try it** |  |
| in | Returns True if a sequence with the specified value is present in the object | x in y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_membership1) |  |
| not in | Returns True if a sequence with the specified value is not present in the object | x not in y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_membership2) |  |
|  |  |  |  |  |
| Python Bitwise Operators | | |  |  |
|  |  |  |  |  |
| Bitwise operators are used to compare (binary) numbers: | | |  |  |
|  |  |  |  |  |
| **Operator** | **Name** | **Description** | **Example** | **Try it** |
| & | AND | Sets each bit to 1 if both bits are 1 | x & y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_and) |
| | | OR | Sets each bit to 1 if one of two bits is 1 | x | y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_or) |
| ^ | XOR | Sets each bit to 1 if only one of two bits is 1 | x ^ y | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_xor) |
| ~ | NOT | Inverts all the bits | ~x | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_not) |
| << | Zero fill left shift | Shift left by pushing zeros in from the right and let the leftmost bits fall off | x << 2 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_left_shift) |
| >> | Signed right shift | Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off | x >> 2 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_oper_right_shift) |
|  |  |  |  |  |
| Operator Precedence | |  |  |  |
|  |  |  |  |  |
| Operator precedence describes the order in which operations are performed. | | | |  |
|  |  |  |  |  |
| Example | |  |  |  |
|  |  |  |  |  |
| Parentheses has the highest precedence, meaning that expressions inside parentheses must be evaluated first: | | | | |
|  |  |  |  |  |
| print((6 + 3) - (6 + 3)) | |  |  |  |
| [Run example »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_parentheses) |  |  |  |  |
| Example | |  |  |  |
|  |  |  |  |  |
| Multiplication \* has higher precedence than addition +, and therefor multiplications are evaluated before additions: | | | | |
| print(100 + 5 \* 3) | |  |  |  |
| [Run example »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_multiplication) |  |  |  |  |
|  |  |  |  |  |
| The precedence order is described in the table below, starting with the highest precedence at the top: | | | | |
|  |  |  |  |  |
| **Operator** | **Description** | **Try it** |  |  |
| () | Parentheses | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_parentheses) |  |  |
| \*\* | Exponentiation | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_exponent) |  |  |
| +x  -x  ~x | Unary plus, unary minus, and bitwise NOT | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_bitwise_not) |  |  |
| \*  /  //  % | Multiplication, division, floor division, and modulus | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_multiplication) |  |  |
| +  - | Addition and subtraction | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_subtraction) |  |  |
| <<  >> | Bitwise left and right shifts | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_shift) |  |  |
| & | Bitwise AND | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_bitwise_and) |  |  |
| ^ | Bitwise XOR | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_bitwise_xor) |  |  |
| | | Bitwise OR | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_bitwise_or) |  |  |
| ==  !=  >  >=  <  <=  is  is not  in  not in | Comparisons, identity, and membership operators | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_like) |  |  |
| not | Logical NOT | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_not) |  |  |
| and | AND | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_and) |  |  |
| or | OR | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_precedence_or) |  |  |
|  |  |  |  |  |
| If two operators have the same precedence, the expression is evaluated from left to right. | | | | |
|  |  |  |  |  |
| Example | |  |  |  |
|  |  |  |  |  |
| Addition + and subtraction - has the same precedence, and therefor we evaluate the expression from left to right: | | | | |
|  |  |  |  |  |
| print(5 + 4 - 7 + 3) | |  |  |  |