



HTML

CSS

MORE ▼



# Python Operators

[< Previous](#)[Next >](#)

## Python Operators

Operators are used to perform operations on variables and values.

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   | <a href="#">Try it »</a> |
| -        | Subtraction    | x - y   | <a href="#">Try it »</a> |
| *        | Multiplication | x * y   | <a href="#">Try it »</a> |
| /        | Division       | x / y   | <a href="#">Try it »</a> |
| %        | Modulus        | x % y   | <a href="#">Try it »</a> |

|           |                |                     |                          |
|-----------|----------------|---------------------|--------------------------|
| <b>**</b> | Exponentiation | <code>x ** y</code> | <a href="#">Try it »</a> |
| <b>//</b> | Floor division | <code>x // y</code> | <a href="#">Try it »</a> |

## Python Assignment Operators

Assignment operators are used to assign values to variables:

| Operator         | Example                    | Same As                       | Try it                   |
|------------------|----------------------------|-------------------------------|--------------------------|
| <b>=</b>         | <code>x = 5</code>         | <code>x = 5</code>            | <a href="#">Try it »</a> |
| <b>+=</b>        | <code>x += 3</code>        | <code>x = x + 3</code>        | <a href="#">Try it »</a> |
| <b>-=</b>        | <code>x -= 3</code>        | <code>x = x - 3</code>        | <a href="#">Try it »</a> |
| <b>*=</b>        | <code>x *= 3</code>        | <code>x = x * 3</code>        | <a href="#">Try it »</a> |
| <b>/=</b>        | <code>x /= 3</code>        | <code>x = x / 3</code>        | <a href="#">Try it »</a> |
| <b>%=</b>        | <code>x %= 3</code>        | <code>x = x % 3</code>        | <a href="#">Try it »</a> |
| <b>//=</b>       | <code>x //= 3</code>       | <code>x = x // 3</code>       | <a href="#">Try it »</a> |
| <b>**=</b>       | <code>x **= 3</code>       | <code>x = x ** 3</code>       | <a href="#">Try it »</a> |
| <b>&amp;=</b>    | <code>x &amp;= 3</code>    | <code>x = x &amp; 3</code>    | <a href="#">Try it »</a> |
| <b> =</b>        | <code>x  = 3</code>        | <code>x = x   3</code>        | <a href="#">Try it »</a> |
| <b>^=</b>        | <code>x ^= 3</code>        | <code>x = x ^ 3</code>        | <a href="#">Try it »</a> |
| <b>&gt;&gt;=</b> | <code>x &gt;&gt;= 3</code> | <code>x = x &gt;&gt; 3</code> | <a href="#">Try it »</a> |
| <b>&lt;&lt;=</b> | <code>x &lt;&lt;= 3</code> | <code>x = x &lt;&lt; 3</code> | <a href="#">Try it »</a> |

## Python Comparison Operators

Comparison operators are used to compare two values:

| Operator  | Name  | Example             | Try it                   |
|-----------|-------|---------------------|--------------------------|
| <b>==</b> | Equal | <code>x == y</code> | <a href="#">Try it »</a> |

|    |                          |        |                          |
|----|--------------------------|--------|--------------------------|
| != | Not equal                | x != y | <a href="#">Try it »</a> |
| >  | Greater than             | x > y  | <a href="#">Try it »</a> |
| <  | Less than                | x < y  | <a href="#">Try it »</a> |
| >= | Greater than or equal to | x >= y | <a href="#">Try it »</a> |
| <= | Less than or equal to    | x <= y | <a href="#">Try it »</a> |

## 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      | <a href="#">Try it »</a> |
| or       | Returns True if one of the statements is true           | x < 5 or x < 4        | <a href="#">Try it »</a> |
| not      | Reverse the result, returns False if the result is true | not(x < 5 and x < 10) | <a href="#">Try it »</a> |

## 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     | <a href="#">Try it »</a> |
| is not   | Returns True if both variables are not the same object | x is not y | <a href="#">Try it »</a> |

# 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     | <a href="#">Try it »</a> |
| not in   | Returns True if a sequence with the specified value is not present in the object | x not in y | <a href="#">Try it »</a> |

## Python Bitwise Operators

Bitwise operators are used to compare (binary) numbers:

| Operator | Name                 | Description   |
|----------|----------------------|---|
| &        | AND                  | Sets each bit to 1 if both bits are 1   |
|          | OR                   | Sets each bit to 1 if one of two bits is 1  |
| ^        | XOR                  | Sets each bit to 1 if only one of two bits is 1   |
| ~        | NOT                  | Inverts all the bits  |
| <<       | Zero fill left shift | Shift left by pushing zeros in from the right and let the leftmost bits fall off                        |
| >>       | Signed right shift   | Shift right by pushing copies of the leftmost bit in from the left, and let the rightmost bits fall off |

## Test Yourself With Exercises

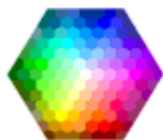
### Exercise:

Multiply **10** with **5** , and print the result.

```
print(10 5)
```

[Submit Answer »](#)[Start the Exercise](#)[◀ Previous](#)[Next ▶](#)

## COLOR PICKER



## HOW TO

- Tabs
- Dropdowns
- Accordions
- Side Navigation
- Top Navigation
- Modal Boxes
- Progress Bars
- Parallax
- Login Form
- HTML Includes
- Google Maps
- Range Sliders
- Tooltips
- Slideshow

[Filter List](#)

[Sort List](#)

SHARE



CERTIFICATES

[HTML](#)

[CSS](#)

[JavaScript](#)

[SQL](#)

[Python](#)

[PHP](#)

[jQuery](#)

[Bootstrap](#)

[XML](#)

[Read More »](#)

---

[REPORT ERROR](#)

[PRINT PAGE](#)

[FORUM](#)

[ABOUT](#)

---

## Top Tutorials

[HTML Tutorial](#)

[CSS Tutorial](#)

[JavaScript Tutorial](#)

[How To Tutorial](#)

[SQL Tutorial](#)

[Python Tutorial](#)

[W3.CSS Tutorial](#)

[Bootstrap Tutorial](#)  
[PHP Tutorial](#)  
[jQuery Tutorial](#)  
[Java Tutorial](#)  
[C++ Tutorial](#)

## Top References

[HTML Reference](#)  
[CSS Reference](#)  
[JavaScript Reference](#)  
[SQL Reference](#)  
[Python Reference](#)  
[W3.CSS Reference](#)  
[Bootstrap Reference](#)  
[PHP Reference](#)  
[HTML Colors](#)  
[jQuery Reference](#)  
[Java Reference](#)  
[Angular Reference](#)

## Top Examples

[HTML Examples](#)  
[CSS Examples](#)  
[JavaScript Examples](#)  
[How To Examples](#)  
[SQL Examples](#)  
[Python Examples](#)  
[W3.CSS Examples](#)  
[Bootstrap Examples](#)  
[PHP Examples](#)  
[jQuery Examples](#)  
[Java Examples](#)  
[XML Examples](#)

## Web Certificates

[HTML Certificate](#)  
[CSS Certificate](#)  
[JavaScript Certificate](#)  
[SQL Certificate](#)  
[Python Certificate](#)  
[jQuery Certificate](#)  
[PHP Certificate](#)  
[Bootstrap Certificate](#)  
[XML Certificate](#)

[Get Certified »](#)

---

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2020 by Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

