



Menu ▼

Log in



HTML

CSS



# C Operators

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

## Operators

Operators are used to perform operations on variables and values.

In the example below, we use the **+** **operator** to add together two values:

### Example

```
int myNum = 100 + 50;
```

Try it Yourself »

Although the **+** operator is often used to add together two values, like in the example above, it can also be used to add together a variable and a value, or a variable and another variable:

### Example

```
int sum1 = 100 + 50;           // 150 (100 + 50)
int sum2 = sum1 + 250;         // 400 (150 + 250)
```

```
int sum3 = sum2 + sum2;    // 800 (400 + 400)
```

[Try it Yourself »](#)

C divides the operators into the following groups:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Bitwise operators

## Arithmetic Operators

Arithmetic operators are used to perform common mathematical operations.

Operator	Name	Description	Example	Try it
+	Addition	Adds together two values	x + y	<a href="#">Try it »</a>
-	Subtraction	Subtracts one value from another	x - y	<a href="#">Try it »</a>
*	Multiplication	Multiplies two values	x * y	<a href="#">Try it »</a>
/	Division	Divides one value by another	x / y	<a href="#">Try it »</a>
%	Modulus	Returns the division remainder	x % y	<a href="#">Try it »</a>
++	Increment	Increases the value of a variable by 1	++x	<a href="#">Try it »</a>
--	Decrement	Decreases the value of a variable by 1	--x	<a href="#">Try it »</a>

## Assignment Operators

Assignment operators are used to assign values to variables.

In the example below, we use the **assignment** operator ( `=` ) to assign the value **10** to a variable called **x**:

## Example

```
int x = 10;
```

Try it Yourself »

The **addition assignment** operator ( `+=` ) adds a value to a variable:

## Example

```
int x = 10;  
x += 5;
```

Try it Yourself »

A list of all assignment operators:

Operator	Example	Same As	Try it
=	x = 5	x = 5	Try it »
+=	x += 3	x = x + 3	Try it »
-=	x -= 3	x = x - 3	Try it »
*=	x *= 3	x = x * 3	Try it »
/=	x /= 3	x = x / 3	Try it »
%=	x %= 3	x = x % 3	Try it »
&=	x &= 3	x = x & 3	Try it »

=	x  = 3	x = x   3	<a href="#">Try it »</a>
^=	x ^= 3	x = x ^ 3	<a href="#">Try it »</a>
>>=	x >>= 3	x = x >> 3	<a href="#">Try it »</a>
<<=	x <<= 3	x = x << 3	<a href="#">Try it »</a>

## Comparison Operators

Comparison operators are used to compare two values.

**Note:** The return value of a comparison is either true ( **1** ) or false ( **0** ).

In the following example, we use the **greater than** operator ( **>** ) to find out if 5 is greater than 3:

### Example

```
int x = 5;
int y = 3;
printf("%d", x > y); // returns 1 (true) because 5 is greater than 3
```

[Try it Yourself »](#)

A list of all comparison operators:

Operator	Name	Example	Try it
==	Equal to	x == y	<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	

[Try it »](#)

&lt;=

Less than or equal to

x &lt;= y

[Try it »](#)

## Logical Operators

Logical operators are used to determine the logic between variables or values:

Operator	Name	Description	Example	Try it
&&	Logical and	Returns true if both statements are true	x < 5 && x < 10	<a href="#">Try it »</a>
	Logical or	Returns true if one of the statements is true	x < 5    x < 4	<a href="#">Try it »</a>
!	Logical not	Reverse the result, returns false if the result is true	!(x < 5 && x < 10)	<a href="#">Try it »</a>

## Sizeof Operator

The memory size (in bytes) of a data type or a variable can be found with the **sizeof** operator:

### Example

```
int myInt;
float myFloat;
double myDouble;
char myChar;

printf("%lu\n", sizeof(myInt));
printf("%lu\n", sizeof(myFloat));
printf("%lu\n", sizeof(myDouble));
printf("%lu\n", sizeof(myChar));
```

[Try it Yourself »](#)

Note that we use the `%lu` format specifier to print the result, instead of `%d`. It is because the compiler expects the `sizeof` operator to return a `long unsigned int` (`%lu`), instead of `int` (`%d`). On some computers it might work with `%d`, but it is safer to use `%lu`.

---

## C Exercises

### Test Yourself With Exercises

#### Exercise:

Fill in the blanks to multiply `10` with `5`, and print the result:

```
int x = 10;
int y = 5;
printf("    ", x    y);
```

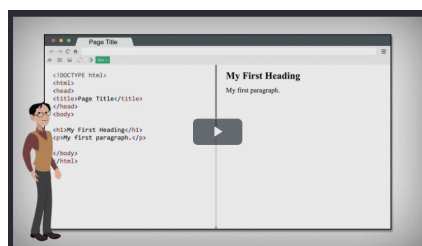
[Submit Answer »](#)[Start the Exercise](#)

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

## ADVERTISEMENT

## NEW

We just launched  
W3Schools videos

[Explore now](#)

## COLOR PICKER



Get certified  
by completing  
a course today!



Get started

## CODE GAME



Play Game

ADVERTISEMENT





ADVERTISEMENT

ADVERTISEMENT

[Report Error](#)[Spaces](#)[Pro](#)[Get Certified](#)

---

## Top Tutorials

[HTML Tutorial](#)  
[CSS Tutorial](#)  
[JavaScript Tutorial](#)  
[How To Tutorial](#)  
[SQL Tutorial](#)  
[Python Tutorial](#)  
[W3.CSS Tutorial](#)  
[Bootstrap Tutorial](#)  
[PHP Tutorial](#)  
[Java Tutorial](#)  
[C++ Tutorial](#)  
[jQuery Tutorial](#)

## Top References

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

## Top Examples

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

[XML Examples](#)  
[jQuery Examples](#)

## Get Certified

[HTML Certificate](#)  
[CSS Certificate](#)  
[JavaScript Certificate](#)  
[Front End Certificate](#)  
[SQL Certificate](#)  
[Python Certificate](#)  
[PHP Certificate](#)  
[jQuery Certificate](#)  
[Java Certificate](#)  
[C++ Certificate](#)  
[C# Certificate](#)  
[XML Certificate](#)

---

[FORUM](#) | [ABOUT](#)

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

Copyright 1999-2022 by Refsnes Data. All Rights Reserved.  
W3Schools is Powered by W3.CSS.

