

TECHNOLOGY



Caltech | **Center for Technology & Management Education**

Full Stack Java Developer

TECHNOLOGY



JavaScript

Operators



Learning Objectives

By the end of this lesson, you will be able to:

- 👁 Define operators and list the types of operators
- 👁 Describe operators in detail with syntaxes and example
- 👁 Explain operator precedence



A Day in the Life of a Full Stack Developer

You are working in an organization and have been assigned a website development project for an e-commerce company. The e-commerce website will consist of multiple items listed. You are supposed to create a feature that can filter the items based on their price, color, and within range.

You decide to use JavaScript to add this feature to the website. Here, you will have to compare the prices of the items with the prize specified by the user or select the item in the color requested by the user. To do so, you will have to use the comparison operator and operator precedence to filter out the items and display them to the users.

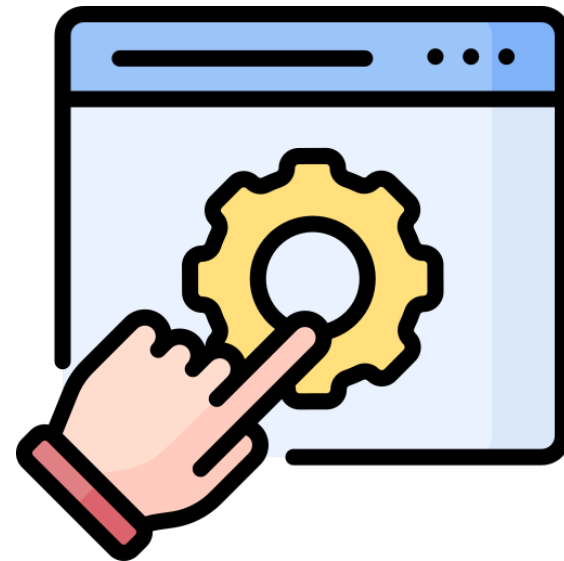
To do so, explore the concepts of operators, types of operators, and their syntax.



What Are Operators?

What Are Operators?

Operators operate the operands.



1

An operator is a program.

2

It can manipulate a value or operand.

3

It is used to perform specific mathematical and logical operations.



What Are Operators?

Operators are used to:

Perform some specific mathematical or logical computations



Compare the values and perform the arithmetic operations

Logical Operators with Non-Booleans

The logic between the variables or values is determined using logical operators.

There are two types of logical operators, they are:

&& (and)

This operator will act as true if and only if the expressions on both sides are true.



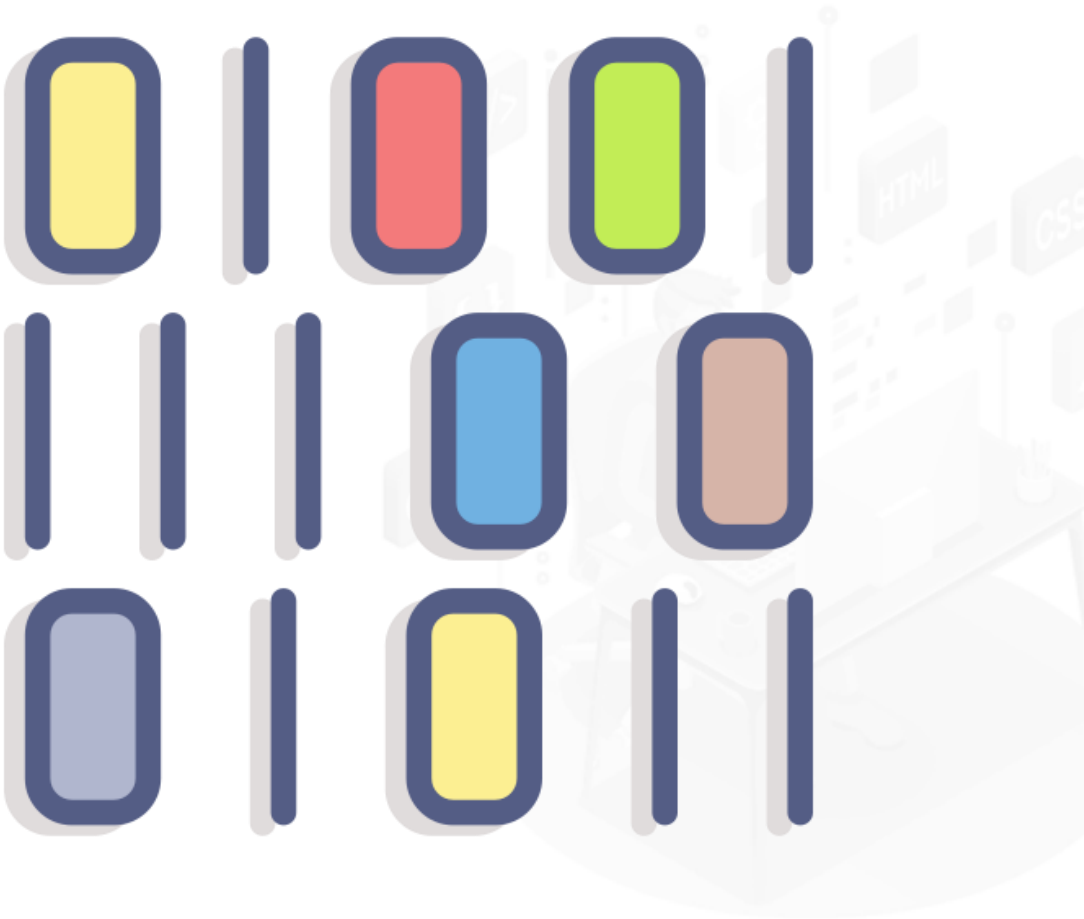
|| (or)

This operator will be true if the expression on either side of it is true. Otherwise, its value is false.

Bitwise Operator

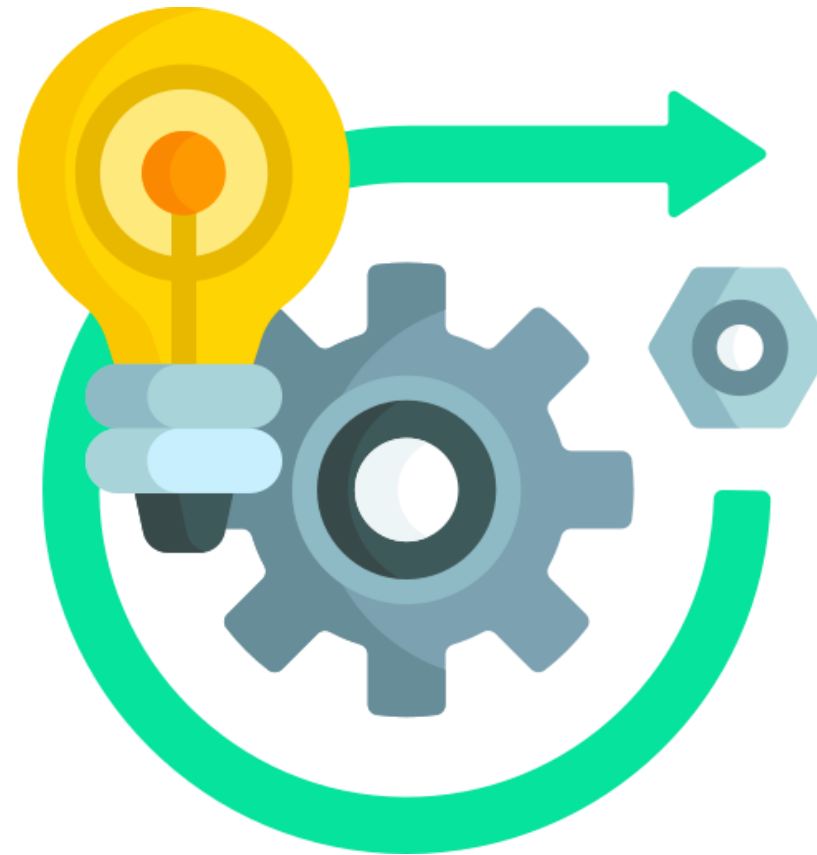
The bitwise operators treat their operands as a set of 32-bit binary digits and act.

Operators	Name	Example
&	Bitwise AND	$x \& y$
	Bitwise OR	$x y$
~	Bitwise NOT	$\sim x$
<<	Left shift	$x \ll y$
>>	Right shift	$x \gg y$
>>>	Zero fill right shift	$x \ggg y$



Operator Precedence

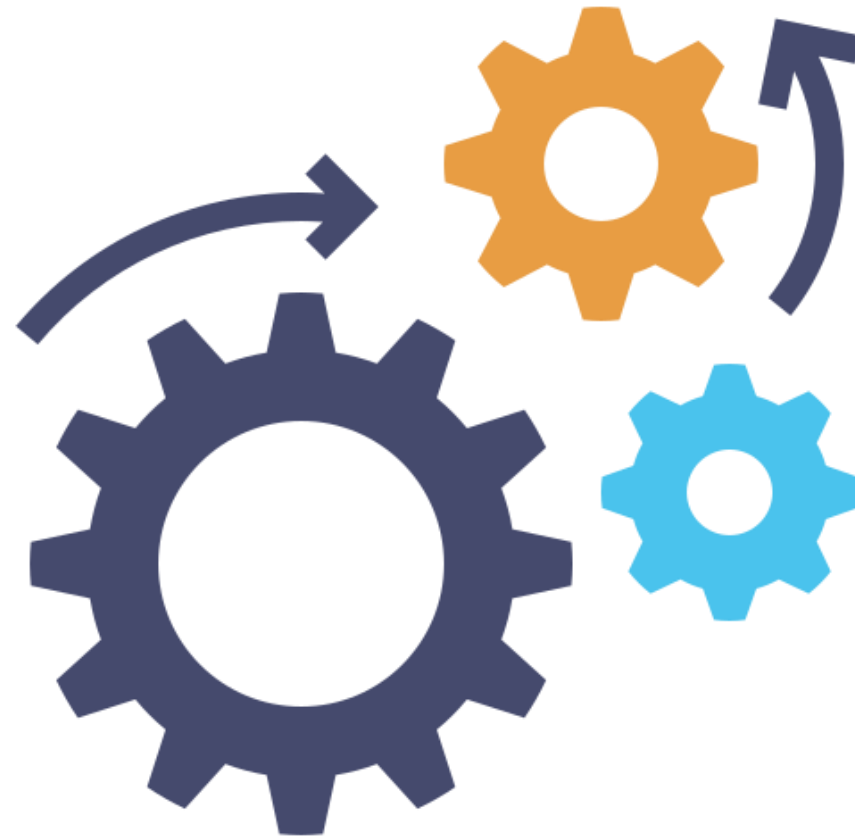
Higher precedence operators become the operands of lower precedence operators.



Types of Operators

Types of Operators

Types of operators are:

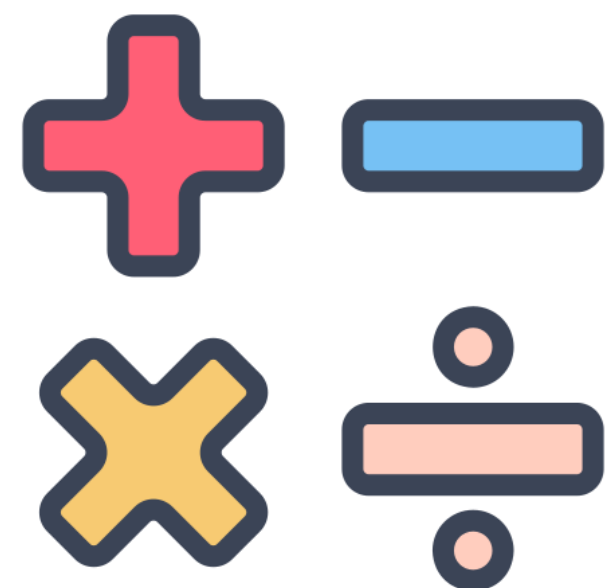


- 1 Arithmetic operators
- 2 Assignment operators
- 3 Comparison operators
- 4 Equality operators
- 5 Ternary operators
- 6 Logical operators
- 7 Bitwise operators



Arithmetic Operators

All arithmetic operations on numbers are performed by the arithmetic operators.

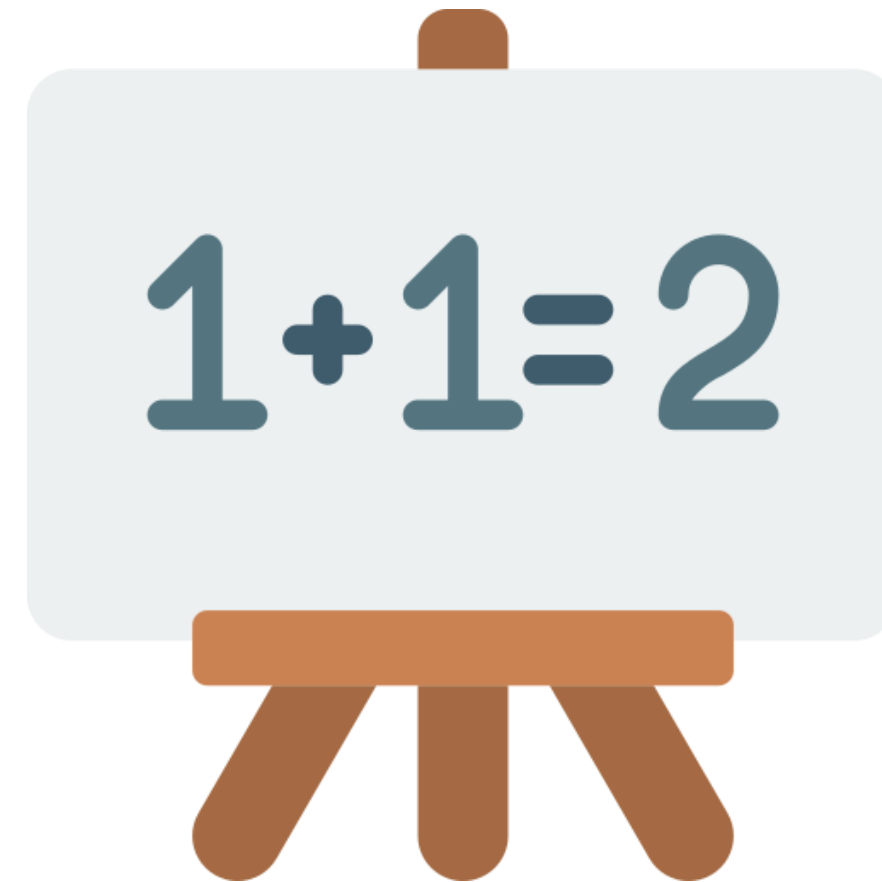
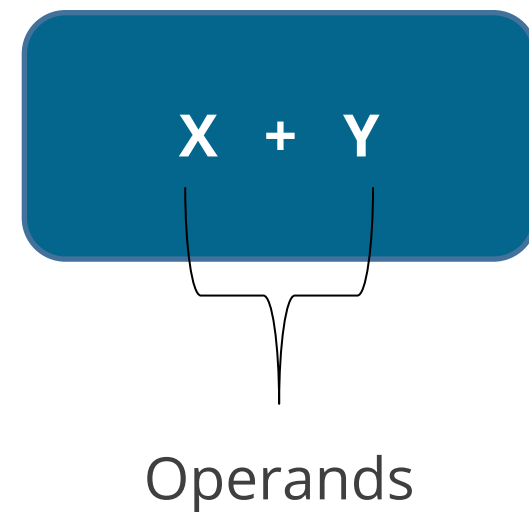


Arithmetic operators	Symbol
Addition	+
Subtraction	-
Division	/
Multiplication	*



Arithmetic Operators

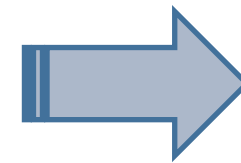
The numbers in the arithmetic operation are called operands.



Arithmetic Operators

Example:

```
<!DOCTYPE html>
<html>
<body>
<h3>Arithmetic Operators</h3>
<p>Addition of two numbers</p>
<p id="test"></p>
<script>
let a = 40 +60;
Document.getElementById("test").innerHTML = a;
</script>
</html>
```



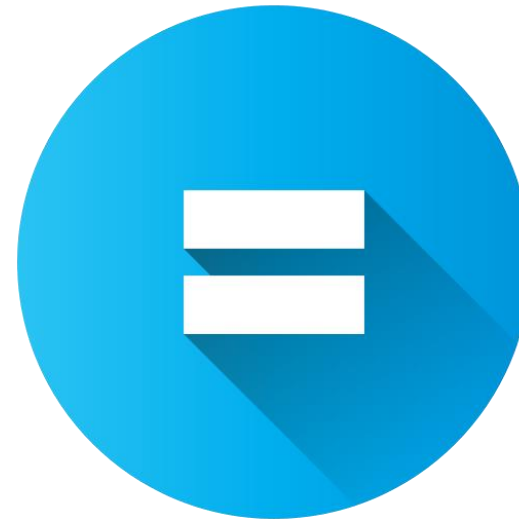
Output

```
Arithmetic Operators
Addition of two numbers
100
```


Assignment Operators

Assignment operators are used to assign values to variables.

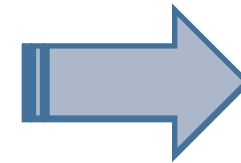
The assignment operator's symbol is
(=).



Assignment Operators

Example:

```
<!DOCTYPE html>
<html>
<body>
<h2>JavaScript Assignment Operators (=)</h2>
<p id="test"></p>
<script>
  let a = 60;
  document.getElementById("test").innerHTML = a;
</script>
</body>
</html>
```



Output

```
JavaScript Assignment
Operators (=) 60
```

Comparison Operators

The comparison operators compare values and return true or false. They are:

Less than (<)

Greater than (>)

Less than or equal to
(<=)


Greater than or equal
to (>=)

Equal to (==)

Not equal to (!=)

Comparison Operators


The list of comparison operators are:



Less than (<): Returns true if the value on the left side is less than the right-hand value; else, it returns the false value




Greater than (>): Returns true if the value on the left side is greater than the value on the right; else, it returns the false value




Less than or equal to (<=): Returns true if the value on the left side is less than or equal to the right side; else, it returns the false value


Comparison Operators



Greater than or equal to (\geq): Returns true if the value of the left side exceeds or equals the value on the right side; otherwise, it returns the false value



Equals to ($===$): Returns true if the value on the left side is equal to the value on the right side; else, it returns the false value

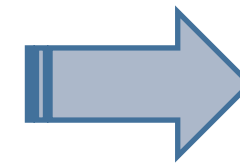


Not equal to ($!==$): Returns true if the value on the left side is not the same as the amount on the right; otherwise, it returns the false value

Comparison Operators

Example:

```
<!DOCTYPE html>
<html>
<body>
<h3>Comparison example</h3>
<p>The comparison operators compare value and return
the Boolean value</p>
<script>
  let a = 10;
  let exp = a === 11;
  document.getElementById("test").innerHTML = exp;
</script>
</body>
</html>
```



Output

```
Comparison example
The comparison operators
compare values and return the
Boolean value.
false
```

Equality Operators

The equality operator compares two expressions or values.

If the result is true, the expressions are equal; otherwise, the expressions are not equal.

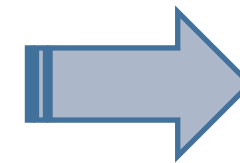
```
x = y;
```



Equality Operators

Example:

```
<!DOCTYPE html>
<html>
<body>
<p id="test"></p>
<script>
  var x = 10;
  document.getElementById("test").innerHTML = (x ==
10);
</script>
</body>
</html>
```



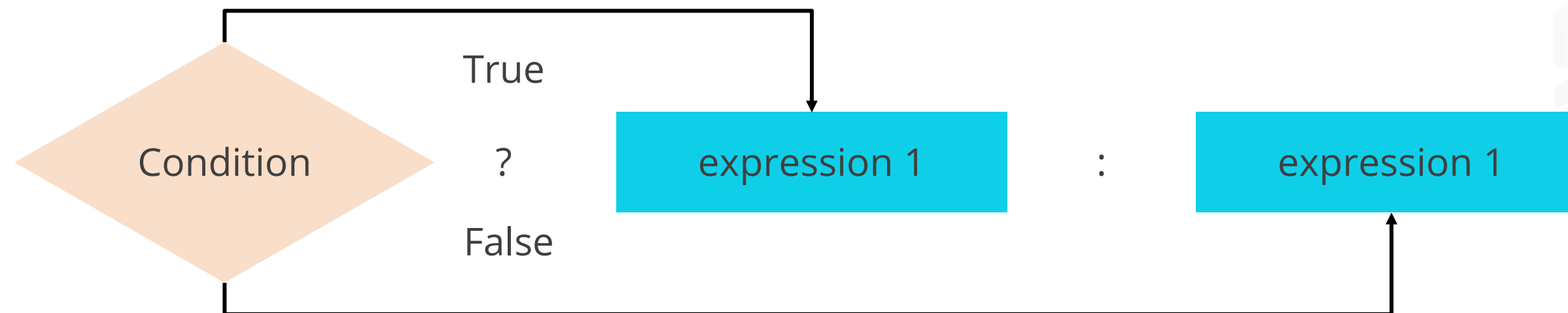
Output

true

Ternary Operators

The conditional ternary operator accepts three operands.

```
condition? expression 1: expression 2
```



Key Takeaways

- An operator is a program that is used to perform some specific mathematical or logical computations.
- The bitwise operators treat their operands as a set of 32-bit binary digits and act.
- The comparison operators compare values and return true or false.
- The equality operator (==) compares two expressions or values.
- Higher precedence operators become the operands of lower precedence operators.



TECHNOLOGY

Thank You