

Difference Between Bitwise and Logical Operators

www.differencebetween.com

Key Difference - Bitwise vs Logical Operators

In [programming](#), there are situations to perform mathematical computations. An operator is a symbol of [programming languages](#) to perform specific logical or mathematical functions on a value or a variable. There are various operators in programming languages. Some of them are arithmetic operators, relational operators, logical operators, bitwise operators and assignment operators. Arithmetic operators support mathematical operations such as addition (+), subtraction (-), division (/), multiplication (*), modulus (%), incrementing (++) and decrement (--). Relation operators are >, >=, <, <=, == or !=. These operators help to find the relationship of operands. Assignment operators assign values from right side operand to left side operand. Bitwise operators are &, |, ^. Logical operators are &&, ||, !. This article discusses the difference between bitwise and logical operators. The **key difference** between Bitwise and Logical operators is that **Bitwise operators work on bits and perform bit by bit operations while logical operators are used to make a decision based on multiple conditions.**

What are Bitwise Operators?

Bitwise operators work on bits and perform bit by bit operation. In computations such as addition, subtraction, multiplication, division etc. the values are converted into [binaries](#). Those operations are performed on bit level. Bit-level processing is used to increase speed and to save power. Some examples of Bitwise operators are as follows. The & represents bitwise AND. The | represents bitwise OR. The ^ represents bitwise exclusive OR. The ~ is the complement. The << symbol is to represent the left shift while >> symbol represents the right shift.

Bitwise AND operation is as follows. When x and y are operands, and x has value 0, and y has value 0, then bitwise AND is 0. When x is 0 and y is 1, then the bitwise AND is 0. If x is 1 and y is 0, then the bitwise AND is 0. When both x and y have 1, the bitwise AND is 1. The output will be 1 only if both operands contain the value 1. Assume 20 and 25 as two values. The binary of 20 is 10100. The binary of 25 is 11001. Bitwise AND of these two numbers is 10000. When performing bit by bit AND operation, the value one only comes when both operands contain one.

Bitwise OR operation is as follows. When x and y are operands, and x has value 0 and y has value 0, then bitwise OR is 0. When x is 0 and y is 1, then the output is 1. When x is 1 and y is 0, the output is 1. When both x and y are having value 1, the output is 1. From two operands, if either one operand is 1, then the Bitwise OR is 1. Assume 20 and 25 as two values. The binary of 20 is 10100. The binary of 25 is 11001. Bitwise OR of the 20 and 25 is 11101.

Bitwise XOR operator will give 1 if both values are different. When x and y operands are zeros, then the Bitwise XOR is 0. When x is 0 and y is 1, the output is 1. When x is 1 and y is 0, then the output is 1. When both x and y are 1, then the output is 0. The Bitwise XOR of 20 and 25 is 01101. The ~ symbol is to take the complement of the value. The binary value of 20 is 10100. The complement is $\sim 20 = 01011$. It is to convert ones to zeros and to convert the zeros to ones.

The << is the binary left shift operator. The left operands value is moved left by the number of bits specified by the right operand. In example $5 \ll 1$, the binary value of 5 is 0101. $0101 \ll 1$ is 1010. The >> is the binary right shift operator. The left operands value is moved right by the number of bits specified by the right operand. As an example, $5 \gg 1$, $0101 \gg 1$ is 0010.

What are Logical Operators?

The logical operators are used to make a decision based on multiple conditions. The && symbol represents the logical AND. The || symbol represents the logical OR. The ! symbol represents the logical NOT. In logical AND, if both operands are non zero, then the condition becomes true. In logical OR, if both operands are non zero, then the condition becomes true. The ! operator can reverse the logical status of the operand. If a condition is true, then Logical NOT operator will make it false. The true represents value 1, and falsely represents value 0.

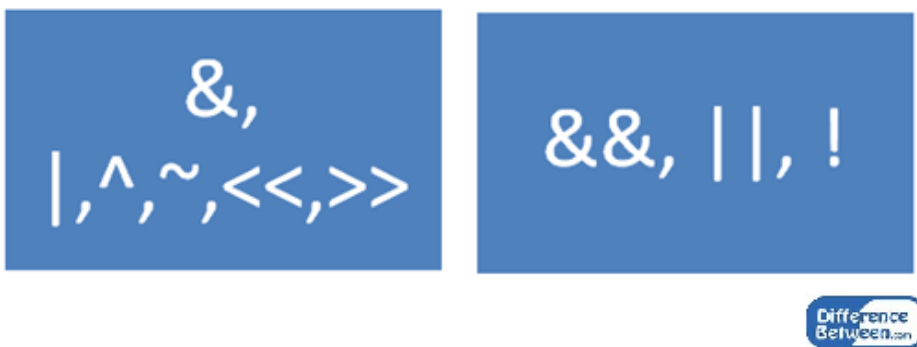


Figure 01: Bitwise and Logical Operators

When variable x is holding the value 1 and variable y is holding the value 0, the logical AND that is $(x \&\& y)$ is false or 0. The logical OR that is $(x \|\| y)$ will give true or 1. The NOT operator reverses the logical status. When x is having value 1, then $!x$ is 0. When y is having value 0, then $!y$ is 1.

What is the Similarity Between Bitwise and Logical Operators?

- Both are operators in programming to perform specific logical or mathematical functions on a value or a variable.

What is the Difference Between Bitwise and Logical Operators?

Bitwise vs Logical Operators	
Bitwise operator is the type of operator provided by the programming language to perform computations.	Logical Operator is a type of operator provided by the programming language to perform logic-based operations.
Functionality	
Bitwise operators work on bits and perform bit by bit operations.	Logical operators are used to making a decision based on multiple conditions.
Themes	
Bitwise operators are &, , ^, ~, <<, >>.	Logical operators are &&, , !

Summary - Bitwise vs Logical Operators

In programming, it is necessary to perform mathematical and logical operations. They can be achieved using operators. There are various types of operators. This article discussed the difference between two operators such as bitwise operators and logical operators. The difference between Bitwise and Logical operators is that Bitwise operators work on bits and perform bit by bit operations while logical operators are used to make a decision based on multiple conditions.

Reference:

1.tutorialspoint.com. "C Operators." [The Point. Available here](#)

How to Cite this Article?

APA: Difference Between Bitwise and Logical Operators.(2018 February 06). Retrieved (date), from <http://differencebetween.com/difference-between-lignin-and-vs-cellulose/>

MLA: "Difference Between Bitwise and Logical Operators" Difference Between.Com. 06 February 2018. Web.

Chicago: "Difference Between Bitwise and Logical Operators." Difference Between.Com. <http://differencebetween.com/difference-between-lignin-and-vs-cellulose/> accessed (accessed [date]).



Copyright © 2010-2018 Difference Between. All rights reserved.