**Boolean Operators**

A logical operator (sometimes called a “Boolean operator”) in Java programming is an operator that returns a Boolean result that’s based on the Boolean result of one or two other expressions.

Sometimes, expressions that use logical operators are called “compound expressions” because the effect of the logical operators is to let you combine two or more condition tests into a single expression.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Name** | **Type** | **Description** |
| ! | Not | Unary | Returns true if the operand to the right evaluates to false. Returns false if the operand to the right is true. |
| & | And | Binary | Returns true if both of the operands evaluate to true. Both operands are evaluated before the And operator is applied. |
| | | Or | Binary | Returns true if at least one of the operands evaluates to true. Both operands are evaluated before the Or operator is applied. |
| ^ | Xor | Binary | Returns true if one — and only one — of the operands evaluates to true. Returns false if both operands evaluate to true or if both operands evaluate to false. |
| && | Conditional And | Binary | Same as &, but if the operand on the left returns false, it returns false without evaluating the operand on theright. |
| || | Conditional Or | Binary | Same as |, but if the operand on the left returns true, it returns true without evaluating the operand on the right. |

**Source Code**

public static void main(String args[]) {
 boolean a = true;
 boolean b = false;
 System.out.println("a && b = " + (a && b));
 System.out.println("a || b = " + (a||b) );
 System.out.println("!(a && b) = " + !(a && b));
 }