

Bitwise Operator:

Bitwise operator works on bits and perform bit-by-bit operation.

Bitwise AND Operator (&): Binary AND Operator copies a bit to the result if it exists in both operands.

Bitwise OR Operator (|): Binary OR Operator copies a bit if it exists in either operand.

Bitwise XOR Operator (^): Binary XOR Operator copies the bit if it is set in one operand but not both.

Bitwise complement (~): Binary One's Complement Operator is unary and has the effect of 'flipping' bits.

Shift left (<<): Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.

Shift right (>>): Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.

Example: Let us consider a=12 00001100 (In Binary) and b=25 00011001 (In Binary)

Bitwise AND Operator on a and b (a&b): 8 00001000 (In Binary)

Bitwise OR Operator on a and b (a|b): 29 00011101 (In decimal)

Ternary Operator:

The Ternary Operator is kind of similar to the if-else statement as it does follow the same algorithm as of if-else statement but the conditional operator takes less space and helps to write the if-else statements in the shortest way possible.

Syntax: variable=expression 1? expression 2: expression 3

Since it takes 3 operands to work it is called Ternary Operator.