**What is an Operator?**

An operator defines some function that will be performed on the data. The data on which operators work are called operands. Consider the following expression −

**7 + 5 = 12**

Here, the values 7, 5, and 12 are **operands**, while + and = are **operators**.

The major operators in TypeScript can be classified as −

* Arithmetic operators
* Logical operators
* Relational operators
* Bitwise operators
* Assignment operators
* Ternary/conditional operator
* String operator
* Type Operator

**Arithmetic Operators**

Assume the values in variables a and b are 10 and 5 respectively.

[Show Examples](https://www.tutorialspoint.com/typescript/typescript_arithmetic_operators_examples.htm)

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| + (Addition) | returns the sum of the operands | a + b is 15 |
| - (Subtraction) | returns the difference of the values | a - b is 5 |
| \* (Multiplication) | returns the product of the values | a \* b is 50 |
| / (Division) | performs division operation and returns the quotient | a / b is 2 |
| % (Modulus) | performs division operation and returns the remainder | a % b is 0 |
| ++ (Increment) | Increments the value of the variable by one | a++ is 11 |
| -- (Decrement) | Decrements the value of the variable by one | a-- is 9 |

**Relational Operators**

Relational Operators test or define the kind of relationship between two entities. Relational operators return a Boolean value, i.e., true/ false.

Assume the value of A is 10 and B is 20.

[Show Examples](https://www.tutorialspoint.com/typescript/typescript_relational_operators_examples.htm)

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| > | Greater than | (A > B) is False |
| < | Lesser than | (A < B) is True |
| >= | Greater than or equal to | (A >= B) is False |
| <= | Lesser than or equal to | (A <= B) is True |
| == | Equality | (A == B) is false |
| != | Not equal | (A != B) is True |

**Logical Operators**

Logical Operators are used to combine two or more conditions. Logical operators too return a Boolean value. Assume the value of variable A is 10 and B is 20.

[Show Examples](https://www.tutorialspoint.com/typescript/typescript_logical_operators_examples.htm)

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| && (And) | The operator returns true only if all the expressions specified return true | (A > 10 && B > 10) is False |
| || (OR) | The operator returns true if at least one of the expressions specified return true | (A > 10 || B >10) is True |
| ! (NOT) | The operator returns the inverse of the expression’s result. For E.g.: !(>5) returns false | !(A >10 ) is True |

**Bitwise Operators**

Assume variable A = 2 and B = 3

[Show Examples](https://www.tutorialspoint.com/typescript/typescript_bitwise_operators_examples.htm)

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| & (Bitwise AND) | It performs a Boolean AND operation on each bit of its integer arguments. | (A & B) is 2 |
| | (BitWise OR) | It performs a Boolean OR operation on each bit of its integer arguments. | (A | B) is 3 |
| ^ (Bitwise XOR) | It performs a Boolean exclusive OR operation on each bit of its integer arguments. Exclusive OR means that either operand one is true or operand two is true, but not both. | (A ^ B) is 1 |
| ~ (Bitwise Not) | It is a unary operator and operates by reversing all the bits in the operand. | (~B) is -4 |
| << (Left Shift) | It moves all the bits in its first operand to the left by the number of places specified in the second operand. New bits are filled with zeros. Shifting a value left by one position is equivalent to multiplying it by 2, shifting two positions is equivalent to multiplying by 4, and so on. | (A << 1) is 4 |
| >> (Right Shift) | Binary Right Shift Operator. The left operand’s value is moved right by the number of bits specified by the right operand. | (A >> 1) is 1 |
| >>> (Right shift with Zero) | This operator is just like the >> operator, except that the bits shifted in on the left are always zero. | (A >>> 1) is 1 |

**Assignment Operators**

[Show Examples](https://www.tutorialspoint.com/typescript/typescript_assignment_operators_examples.htm)

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| = (Simple Assignment) | Assigns values from the right side operand to the left side operand | C = A + B will assign the value of A + B into C |
| += (Add and Assignment) | It adds the right operand to the left operand and assigns the result to the left operand. | C += A is equivalent to C = C + A |
| -= (Subtract and Assignment) | It subtracts the right operand from the left operand and assigns the result to the left operand. | C -= A is equivalent to C = C - A |
| \*= (Multiply and Assignment) | It multiplies the right operand with the left operand and assigns the result to the left operand. | C \*= A is equivalent to C = C \* A |
| /= (Divide and Assignment) | It divides the left operand with the right operand and assigns the result to the left operand. |  |

**Note** − Same logic applies to Bitwise operators, so they will become <<=, >>=, >>=, &=, |= and ^=.