Examples

```
3 > 2 && 2 > 1
// returns true

3 > 2 > 1
// returns false because 3 > 2 is true, then true is converted to 1 in inequality operators, therefore true >
// Adding parentheses makes things clear: (3 > 2) > 1
```

Table

The following table is ordered from highest (21) to lowest (1) precedence.

Precedence	Operator type	Associativity	Individual operators
21	Grouping	n/a	`( … )`
20	Member Access	left-to-right	`… . …`
	Computed Member Access	left-to-right	`… [ … ]`
	`new` (with argument list)	n/a	`new … ( … )`
	Function Call	left-to-right	`… ( … )`
	Optional chaining	left-to-right	`?.`
19	`new` (without argument list)	right-to-left	`new …`
18	Postfix Increment	n/a	`… ++`
	Postfix Decrement		`… --`
17	Logical NOT	right-to-left	`! …`
	Bitwise NOT		`~ …`
	Unary Plus		`+ …`
	Unary Negation		`- …`
	Prefix Increment		`++ …`
	Prefix Decrement		`-- …`
	`typeof`		`typeof …`
	`void`		`void …`
	`delete`		`delete …`
	`await`		`await …`
16	Exponentiation	right-to-left	`… ** …`
15	Multiplication	left-to-right	`… * …`
	Division		`… / …`
	Remainder		`… % …`
14	Addition	left-to-right	`… + …`
	Subtraction		`… - …`

13	Bitwise Left Shift	left-to-right	`… << …`
	Bitwise Right Shift		`… >> …`
	Bitwise Unsigned Right Shift		`… >>> …`
12	Less Than	left-to-right	`… < …`
	Less Than Or Equal		`… <= …`
	Greater Than		`… > …`
	Greater Than Or Equal		`… >= …`
	`in`		`… in …`
	`instanceof`		`… instanceof …`
11	Equality	left-to-right	`… == …`
	Inequality		`… != …`
	Strict Equality		`… === …`
	Strict Inequality		`… !== …`
10	Bitwise AND	left-to-right	`… & …`
9	Bitwise XOR	left-to-right	`… ^ …`
8	Bitwise OR	left-to-right	`… | …`
7	Nullish coalescing operator	left-to-right	`… ?? …`
6	Logical AND	left-to-right	`… && …`
5	Logical OR	left-to-right	`… || …`
4	Conditional	right-to-left	`… ? … : …`
3	Assignment	right-to-left	`… = …`
			`… += …`
			`… -= …`
			`… **= …`
			`… *= …`
			`… /= …`
			`… %= …`
			`… <<= …`
			`… >>= …`
			`… >>>= …`
			`… &= …`
			`… ^= …`
			`… |= …`
2	`yield`	right-to-left	`yield …`
	`yield*`		`yield* …`
1	Comma / Sequence	left-to-right	`… , …`

Last modified: Mar 4, 2020, by MDN contributors