Precedence	Operator type	Associativity	Individual operators
19	Grouping	n/a	`( … )`
18	Member Access	left-to-right	`… . …`
	Computed Member Access	left-to-right	`… [ … ]`
	new (with argument list)	n/a	`new … ( … )`
17	Function Call	left-to-right	`… ( … )`
	new (without argument list)	right-to-left	`new …`
16	Postfix Increment	n/a	`… ++`
	Postfix Decrement	n/a	`… --`
15	Logical NOT	right-to-left	`! …`
	Bitwise NOT	right-to-left	`~ …`
	Unary Plus	right-to-left	`+ …`
	Unary Negation	right-to-left	`- …`
	Prefix Increment	right-to-left	`++ …`
	Prefix Decrement	right-to-left	`-- …`
	typeof	right-to-left	`typeof …`
	void	right-to-left	`void …`
	delete	right-to-left	`delete …`
14	Exponentiation	right-to-left	`… ** …`
	Multiplication	left-to-right	`… * …`
	Division	left-to-right	`… / …`
	Remainder	left-to-right	`… % …`
13	Addition	left-to-right	`… + …`
	Subtraction	left-to-right	`… - …`
12	Bitwise Left Shift	left-to-right	`… << …`
	Bitwise Right Shift	left-to-right	`… >> …`
	Bitwise Unsigned Right Shift	left-to-right	`… >>> …`
11	Less Than	left-to-right	`… < …`
	Less Than Or Equal	left-to-right	`… <= …`
	Greater Than	left-to-right	`… > …`
	Greater Than Or Equal	left-to-right	`… >= …`
	in	left-to-right	`… in …`
	instanceof	left-to-right	`… instanceof …`

10	Equality	left-to-right	… == …
	Inequality	left-to-right	… != …
	Strict Equality	left-to-right	… === …
	Strict Inequality	left-to-right	… !== …
9	Bitwise AND	left-to-right	… & …
8	Bitwise XOR	left-to-right	… ^ …
7	Bitwise OR	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 …
1	Spread	n/a	... …
0	Comma / Sequence	left-to-right	… , …