**JavaScript Operator Precedence Order**

The below table lists the complete order of precedence for operators by JavaScript.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator Precedence** | **Operator** | **Operation** | **Order of Evaluation** |
| 1 | () | Parentheses | Left to right |
|  | [] | Array subscript | Left to right |
|  | function() | function call | Left to right |
| 2 | ! | Logical NOT | Right to left |
|  | ~ | Bitwise NOT | Right to left |
|  | – | Unary Negation | Right to left |
|  | ++ | Increment | Right to left |
|  | – – | Decrement | Right to left |
|  | typeof | Typeof | Right to left |
|  | new | New | Right to left |
|  | void | Void | Right to left |
|  | delete | Delete | Right to left |
| 3 | \* | Multiplication | Left to right |
|  | / | Division | Left to right |
|  | % | Modulus | Left to right |
| 4 | + | Addition | Left to right |
|  | – | Subtraction | Left to right |
| 5 | <<, >>, >>> | Bitwise sift | Left to right |
| 6 | <, <=, >, >= | Comparison (Relational) | Left to right |
| 7 | ==, !=, ===, !== | Equality | Left to right |
| 8 | & | Bitwise AND | Left to right |
| 9 | ^ | Bitwise XOR | Left to right |
| 10 | | | Bitwise OR | Left to right |
| 11 | && | Logical AND | Left to right |
| 12 | || | Logical OR | Left to right |
| 13 | ?: | Conditional | Right to left |
| 14 | =, +=, -=, \*=, /=, %=, <<=, >>=, &=, ^=, |= | Assignment | Right to left |
| 15 | , | Comma | Left to right |

From this table, it is clear that JavaScript performs multiplication before addition. Therefore, understanding the precedence operator in JavaScript is important, otherwise, your statements might generate unexpected results.