

JAVASCRIPT OPERATOR CHEATSHEET

Arithmetic Operators

| Operator | Description | Example |
|----------|--|---|
| + | Adds two numeric operands. | <code>2 + 3 // 5</code> |
| - | Subtract right operand from left operand. | <code>3 - 2 // 1</code> |
| * | Multiply two numeric operands. | <code>2 * 3 // 6</code> |
| / | Divide left operand by right operand. | <code>10 / 5 // 2</code> |
| % | Returns remainder of two operands. | <code>5 % 3 // 2</code> |
| ++ | Increase operand value by one. | <pre>let x = 5; console.log(++x, x); // 6, 6 let y = 5; console.log(y++, x); // 5, 6</pre> |
| -- | Decrease value by one. | <pre>let x = 5; console.log(--x, x); // 4, 4 let y = 5; console.log(y--, x); // 5, 4</pre> |
| ** | Raise one operand to the power of the other operand. | <code>5 ** 3 // 125</code> |

Assignment Operators

| Operator | Description | Longform | Example |
|----------|--|------------------------|-------------------------------|
| = | Adds two numeric operands. | | <pre>let x = 20; x = 20</pre> |
| += | Sums up left and right operand values and assigns the result to the left operand. | <code>x = x + 5</code> | <code>x += 5 // 25</code> |
| -= | Subtract right operand value from the left operand value and assigns the result to the left operand. | <code>x = x - 5</code> | <code>x -= 5 // 15</code> |
| *= | Multiply left and right operand values and assigns the result to the left operand. | <code>x = x * 5</code> | <code>x *= 5 // 100</code> |
| /= | Divide left operand value by right operand value and assign the result to the left operand. | <code>x = x / 5</code> | <code>x /= 5 // 4</code> |
| %= | Get the remainder of left operand divide by right operand and assign resulted remainder to the left operand. | <code>x = x % 5</code> | <code>x %= 5 // 0</code> |

Comparison Operators

| Operator | Description | Example |
|----------|---|--|
| == | Returns true if the operands are equal | <pre>let x = 5; x == '5' // true</pre> |
| != | Returns true if the operands are not equal | <code>x != '6' // true</code> |
| === | Returns true if the operands are equal and of the same type | <code>x === 5 // true</code> |
| !== | Returns true if the operands are the same type but are not equal or are different types | <code>x !== 5 // false</code> |
| > | Returns true if the left operand is greater than the right operand | <code>x > 5 // false</code> |
| >= | Returns true if left operand is greater than or equal to the right operand | <code>x >= 5 // true</code> |
| < | Returns true if the left operand is less than the right operand | |
| <= | Returns true if the left operand is less than or equal to the right operand | <code>x <= 5 // true</code> |

Logical Operators

| Operator | Description | Example |
|----------|---|---|
| && | Returns true if both the operands are true, else returns false | <pre>3 > 2 && 2 > 1 // true 3 < 2 && 2 > 1 // false</pre> |
| | Returns true if either of the operands is true; returns false if both are false | <pre>3 < 2 2 > 1 // true 3 < 2 2 < 1 // false</pre> |
| ! | Returns true if the operand is false and vice-versa. | <pre>!true // false !false // true</pre> |

Ternary Operators

| | |
|-------------|---|
| Operator | condition ? expressionIfTrue : expressionIfFalse |
| Description | If the condition is true, the first expression will be executed. Otherwise, the second expression will be executed. |
| Example | <code>(2 === 2) ? 'x' : 'y'; // 'x'</code> |

typeof Operator

| | |
|-------------|--|
| Operator | typeof |
| Description | Returns a string indicating the data type |
| Example | <pre>console.log(typeof true); // 'boolean' console.log(typeof 300); // 'number' console.log(typeof 'Pete'); // 'string'</pre> |