

🚩 Current Skill Logical or Boolean Operators

Logical & Boolean Operators

There's a big misunderstanding surrounding the logical operators AND and OR in JavaScript. They do not return true or false! More accurately:

Operator	Usage	Description
Logical AND (&&)	a && b	Returns the first value if falsy, otherwise returns the second value whatever it may be.
Logical OR ()	a b	Returns the first value if truthy, otherwise returns the second value whatever it may be.



```
var x = 1 // x is truthy
var y = 0 // y is falsey
var z = null // z is falsey
console.log(x && y) // prints 0
console.log(z && y) // prints null
console.log(z || x) // prints 1
console.log(y || z) // prints null
```

Logical & Boolean Operators

That being said, logical operators are often used alongside comparison operators to return the evaluation of expressions instead of merely variables. This is especially handy when we're setting conditions. (We'll explore those later on, don't worry).



```
var x = 0;
var y = 10;
var z = 5;

console.log(z > x && z < y) // prints true
```



```
console.log(z > x && z <= y) // prints true
```

```
console.log(z > x && z !== y) // prints true
```

Logical & Boolean Operators



One last thing, remember the “!” operator? We said that it negates its operand, true becomes false, and false becomes true.

Well, we can use the “!” operator to **convert a value to a boolean** by negating it twice:

assuming v is **truthy**, `!!v` becomes **true**.

assuming w is **falsy**, `!!w` becomes **false**.



```
console.log(!!NaN) // prints false
```

```
console.log(!!undefined) // prints false
```

```
console.log(!!Infinity) // prints true
```



```
console.log(!!5) // prints true
```

[< Previous](#)

[next >](#)

