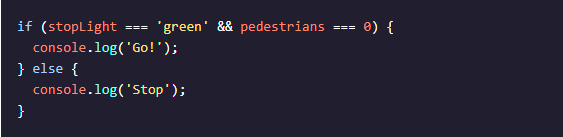
Logical Operators

Working with conditionals means that we will be using booleans, true or false values. In JavaScript, there are operators that work with boolean values known as *logical operators*. We can use logical operators to add more sophisticated logic to our conditionals. There are three logical operators:

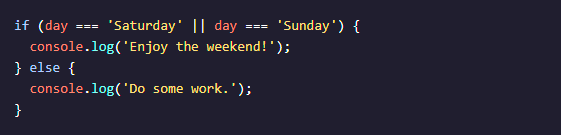
* the *and* operator (&&)
* the *or* operator (||)
* the *not* operator, otherwise known as the *bang* operator (!)

When we use the && operator, we are checking that two things are true:

;  
}

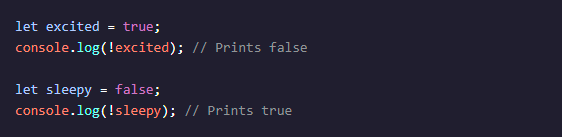
When using the && operator, both conditions *must* evaluate to true for the entire condition to evaluate to true and execute. Otherwise, if either condition is false, the && condition will evaluate to false and the else block will execute.

If we only care about either condition being true, we can use the || operator:

;}

When using the || operator, only one of the conditions must evaluate to true for the overall statement to evaluate to true. In the code example above, if either day === 'Saturday' or day === 'Sunday' evaluates to true the if‘s condition will evaluate to true and its code block will execute. If the first condition in an || statement evaluates to true, the second condition won’t even be checked. Only if day === 'Saturday' evaluates to false will day === 'Sunday' be evaluated. The code in the else statement above will execute only if both comparisons evaluate to false.

The ! *not operator* reverses, or *negates*, the value of a boolean:



Essentially, the ! operator will either take a true value and pass back false, or it will take a false value and pass back true.

Logical operators are often used in conditional statements to add another layer of logic to our code.