

Using Booleans in Conditions & More on Text Comparisons

Understanding the "Condition"

Always keep in mind that **condition** in

```
1. if (condition) { ... }
```

simply has to be a **boolean value**.

Often, you'll **generate** such a boolean value with the help of `===`, `>`, `<` etc. **All these operators yield boolean values** (without changing the variables/ values you're using them on).

Since **if** only wants a boolean, you of course **don't have to use such an operator**. If you already got a variable that holds a boolean, you can use it without any extra operator.

Example:

```
1. const isLoggedIn = true;
2. if (isLoggedIn) {
3.     // This code will execute because isLoggedIn is true => A valid condition
4. }
```

You could write

```
1. const isLoggedIn = true;
2. if (isLoggedIn === true) {
3.     ...
4. }
```

but that would be **redundant**. You'd generate another new boolean where you already got one.

You can use the **!** operator to negate ("invert") the value:

```
1. const isLoggedIn = true;
2. if (!isLoggedIn) {
3.     // This code will NOT execute because isLoggedIn is true but ! inverts it
    (in this check)
4. } else {
5.     // This would execute because !isLoggedIn yields false => else block
    executes
6. }
```

Again, that would be similar to:

```
1. const isLoggedIn = true;
2. if (isLoggedIn !== true) {
3.     // This would NOT execute
4. } else {
5.     // This would execute because isLoggedIn is true and hence !== true
    yields false
```

6. }

But again, that would be redundant.

More on Text (String) Comparisons

Strings can also be compared with greater than (>) or lower/ smaller than (<) operators.

JavaScript compares strings based on standard lexicographical ordering, using Unicode values.

That means that **b** is greater than **a** for example.

JavaScript always looks at the first character and only considers other characters if the first character is similar. In addition, capital characters are considered to be smaller than lowercase characters.

See these examples:

1. `'ab' > 'aa' // true`
2. `'a' > 'B' // true`
3. `'a' > 'b' // false`