Comparison Operators: <, >, <=, >=, ==, ===, !=, !==

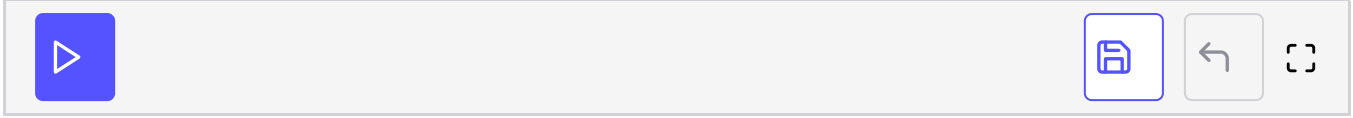
Learn even more operators. We'll go over less than, greater than, equal to, and all of their variations. We'll go into how the double and triple-equal signs differ and which one we should use.

> We have more operators to cover, but I promise this is it!

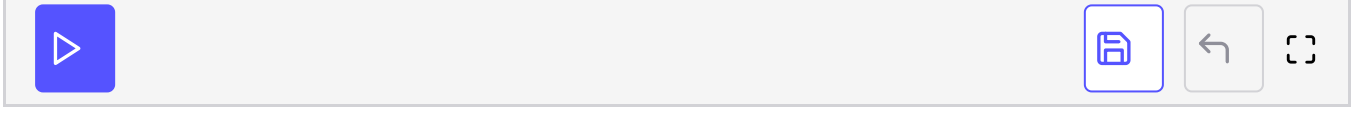
`<` | `>`

These two operators are used to compare the value of numbers. They result in `true` or `false` and are often used in if-statements. `>` means greater than and `<` means less than.

```
if(10 > 5) {
    console.log('Condition is true!');
} else {
    console.log('Condition is false.');
}
```



```
if(10 < 5) {
    console.log('Condition is true!');
} else {
    console.log('Condition is false.');
}
```



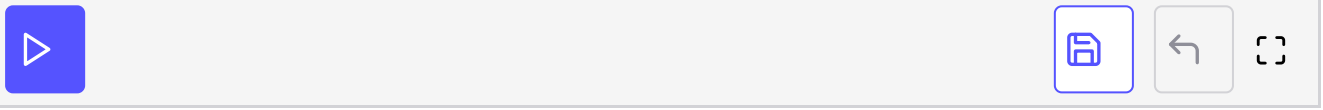
`<=` | `>=`

These mean “less than or equal to” and “greater than or equal to” respectively.

```
if(10 >= 5) {
    console.log('Condition is true!');
} else {
    console.log('Condition is false.');
```

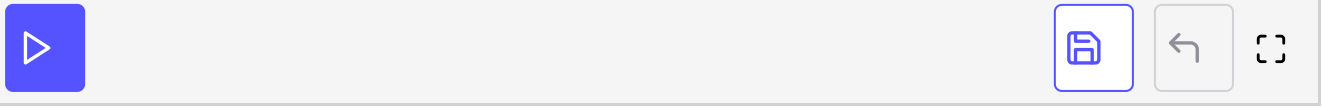
```
}

if(5 >= 5) {
    console.log('Condition is true!');
} else {
    console.log('Condition is false.');
}
```



```
if(5 <= 10) {
    console.log('Condition is true!');
} else {
    console.log('Condition is false.');
}

if(5 <= 5) {
    console.log('Condition is true!');
} else {
    console.log('Condition is false.');
}
```



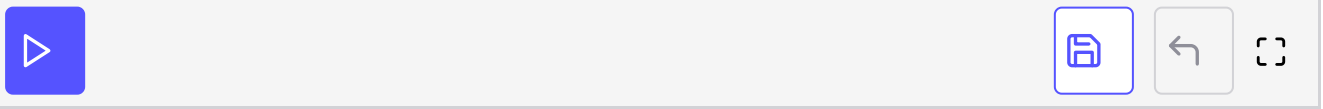
`==` | `===`

These two are used to check if items are equal to each other. Let's start with `===`, the triple-equals.

`===`

This operator checks if two items are the same. It results in `true` if the items are identical and `false` if not.

```
console.log(4 === 4);         // -> true
console.log('abc' === 'abc'); // -> true
console.log(false === false); // -> true
console.log(17 === '17');     // -> false
```



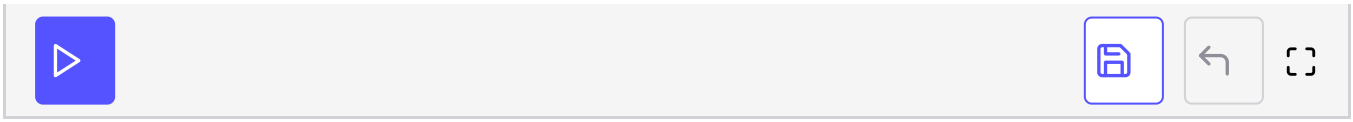
`==`

The double-equal is similar. It will do the same thing as the triple equal, but has an extra “feature”.

If we try to compare two values of different types (for example, a number and a string), the double-equal will attept to coerce the values to be the same type. One or both values may be coerced.

After it attempts coercion, it will return `true` or `false`.

```
console.log(4 == '4'); // -> true
console.log(true == 1); // -> true
console.log(false == 0); // -> true
console.log('' == false); // -> true
```



The double-equal sign is symmetric. That is, if `a == b` is true, then `b == a` is also true, for all values `a` and `b`.

Always Use `===`

We rarely ever want to check if two values can be coerced to the same thing. Situations like that don’t appear often. We almost always want to make sure two values are exactly equal.

This is why it’s a common best practice to always use the triple-equal. Using a double-equal when we want to check two values can lead to unexpected results due to coercion.

With triple-equal, we don’t have to worry about it. My advice is to use the triple-equal wherever possible.

`!=` | `!==`

These two operators are similar to `==` and `===`. They do the same thing, except they return the *opposite* value.

Spelling it Out

- Where `==` returns `true`, `!=` returns `false`
- Where `==` returns `false`, `!=` returns `true`
- Where `===` returns `true`, `!==` returns `false`

- Where `===` returns `false`, `!==` returns `true`

Again, we should always use `!==` over `!=`. These are used in if-statements frequently.