

Relational operator:

List of relational operator:

1. `==/===` (equal to)
2. `!==/!===` (Not equal to)
3. `>` (greater than)
4. `>=` (greater than or equal)
5. `<` (smaller than)
6. `<=` (smaller than or equal)

Relational operator সব সময় true or false রিটার্ন করবে।

Example:

```
console.log( 2 > 1 ); // true (correct)
```

```
console.log( 2 == 1 ); // false (wrong)
```

```
console.log( 2 !== 1 ); // true (correct)
```

1.

```
Let age=18
if(age<18){
console.log("you are child")
}else{
console.log("you are not child")
}
```

2.

```
Let a = "10"
console.log(10==a) //true
console.log(10===a) //false
```

3.

```
Let age=18
if(!(age<18)){
console.log("you are child")
}else{
console.log("you are not child")
}
```

Logical operators

Javascript এ তিন ধরনের logical operators আছে:

1. **&&** (AND)
1. **||** (OR),
3. **!** (NOT).

&&(And) operator:

The AND operator is represented with two ampersands **&&**:

```
result = a && b;
```

দুইটা operand true হলে রিটার্ন true হবে। অন্যথায় false হবে:

```
1. true && true ); // true
2. ( false && true ); // false
3. ( true && false ); // false
4. ( false && false ); // false
```

An example with if:

```
let hour = 12;
let minute = 30;

if (hour == 12 && minute == 30) {
  console.log( 'The time is 12:30' );}
```

|| (OR) operator:

The “OR” operator is represented with two vertical line symbols:

```
result = a || b;
```

একটা operand true হলে রিটার্ন true হবে। অন্যথায় false হবে:

```
1. ( true || true ); // true
2. ( false || true ); // true
3. ( true || false ); // true
4. ( false || false ); // false
```

Example:

```
if (1 || 0) { // works just like if( true || false )
```

```
    console.log( 'truthy!' );
```

```
}
```

```
let hour = 9;
```

```
if (hour < 10 || hour > 18) {
```

```
    console.log( 'The office is closed.' );
```

```
}
```

We can pass more conditions:

```
let hour = 12;
```

```
let isWeekend = true;
```

```
if (hour < 10 || hour > 18 || isWeekend) {
```

```
    console.log( 'The office is closed.' ); // it is the weekend
```

```
}
```

Getting the first truthy value from a list of variables or expressions.

For instance, we have `firstName`, `lastName` and `nickName` variables, all optional (i.e. can be undefined or have falsy values).

Let's use OR `||` to choose the one that has the data and show it (or "Anonymous" if nothing set):

```
let firstName = "";
```

```
let lastName = "";
```

```
let nickName = "SuperCoder";
```

```
console.log( firstName || lastName || nickName || "Anonymous" );  
// SuperCoder
```

If all variables were falsy, "Anonymous" would show up.

Short-circuit evaluation.

Another feature of OR `||` operator is the so-called “short-circuit” evaluation.

It means that `||` processes its arguments until the first truthy value is reached, and then the value is returned immediately, without even touching the other argument.

```
true || console.log("not printed");
```

```
false || console.log("printed");
```

In the first line, the OR `||` operator stops the evaluation immediately upon seeing `true`, so the `alert` isn't run.

Sometimes, people use this feature to execute commands only if the condition on the left part is falsy.

!(NOT)

The boolean NOT operator is represented with an exclamation sign `!`.

The syntax is pretty simple:

```
result = !value;
```

The operator accepts a single argument and does the following:

1. Converts the operand to boolean type: `true/false`.
2. Returns the inverse value.

For instance:

```
( !true ); // false
```

```
( !false ); // true
```

Example:

```
Let age=18  
if(!(age<18)){  
  console.log("you are child")  
}else{  
  console.log("you are not child")  
}
```