

Relational Operators

In JavaScript, relational operators are used to comparing two values and return a boolean value of true or false. The following are the relational operators in JavaScript:

1. Greater than (>): This operator returns true if the left operand is greater than the right operand, and false otherwise. For example:

```
console.log(5 > 3); // true  
console.log(2 > 7); // false  
console.log(5 > 5); // false
```
2. Less than (<): This operator returns true if the left operand is less than the right operand, and false otherwise. For example:

```
console.log(4 < 6); // true  
console.log(8 < 2); // false  
console.log(2 < 2); // false
```
3. Greater than or equal to (>=): This operator returns true if the left operand is greater than or equal to the right operand, and false otherwise. For example:

```
console.log(5 >= 5); // true  
console.log(3 >= 8); // false  
console.log(5 >= 3); // true
```
4. Less than or equal to (<=): This operator returns true if the left operand is less than or equal to the right operand, and false otherwise. For example:

```
console.log(2 <= 2); // true  
console.log(7 <= 4); // false  
console.log(-2 <= 2); // true
```
5. Equality (==): This operator returns true if the operands are equal, and false otherwise. The operands are coerced to the same type before comparison. For example:

```
console.log(2 == '2'); // true  
console.log(true == 1); // true  
console.log(false == ''); // true  
console.log(false == 0); // true  
console.log(false == null); // false  
console.log(false == undefined); // false  
console.log(null == undefined); // true  
console.log(true == true); // true
```

== operator is also called as equals operator.

6. Inequality (!=): This operator returns true if the operands are not equal, and false otherwise. The operands are coerced to the same type before comparison. For example:
- ```
console.log(2 != '2'); // false
console.log(true != 1); // false
console.log(false != ""); // false
console.log(false != true); // true
console.log(true != 10); // true
console.log(null != undefined); // false
```
- != operator is also called as not equal operator.**
7. Strict equality (===): This operator returns true if the operands are equal and of the same type, and false otherwise. For example:
- ```
console.log(2 === '2'); // false
console.log(true === 1); // false
console.log(true === true); // true
console.log(20 === 20); // true
console.log(true === true); // true
```
- === operator is also called as identical operator.**
8. Strict inequality or Not identical (!==): This operator returns true if the operands are not equal and/or not of the same type, and false otherwise. For example:
- ```
console.log(2 !== '2'); // true
console.log(true !== 1); // true
console.log(false !== ''); // true
```
- !== operator is also called as not identical operator.**