Comparisons and Decisions

JavaScript

Relational Operators

We can compare numerical values (two non-booleans in, one boolean out), to see if one is greater or less than the other, using relational operators. Give them a try in the Web Console!

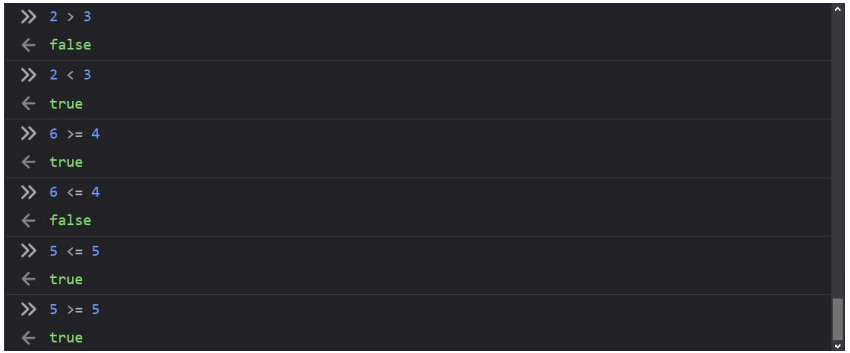
These expressions will return to you either true or false, depending on if the statement is true or false.

< “Less than”

> “Greater than”

<= “Less than, or equal to”

>= “Greater than, or equal to”



Order of Operations

Just as there is an order of operations in math, JavaScript also uses an order of operations, except JavaScript includes a few more types of operators. For mathematics in JavaScript, PEMDAS describes the order of operations followed. Therefore, parenthesis are first evaluated, followed by exponents, then multiplication, then division, then addition, and finally subtraction. For more information on the other types of operators in JavaScript, and their order of precedence, please see the MDN Operator Precedence Documentation and scroll down to the “Table” portion of the article. Note that the higher the precedence number, the higher the precedence for the operation overall.

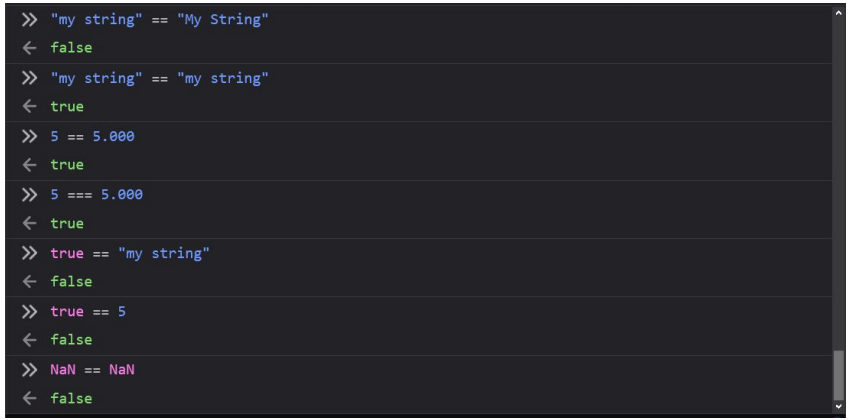
Relational Operators cont.

Equality operators can be a little more tricky, as JavaScript is loose with data-types.

Run lots of experiments to see which expressions evaluate to true or false!

== Equality operator, checks if the values on either side appear to be equivalent.

=== Identity operator, is similar but always considers values of different types to be different.



Logical Operators

Logical operators are used for comparing two boolean values (two booleans in, one boolean out).

Regardless of the two values being compared, they'll be implicitly converted to the nearest boolean equivalent (true or false).

The first few times you use logical operators, they can be a bit of a thought exercise. Later on, they can get a bit complex too as you can nest them!

Try to keep logical operator comparisons grouped within parentheses to make the order of operations more obvious.

The logical operators you'll see most often include...

&& And, true if both operands are true.

|| Or, true if either operand is true.

```
>> true && true
<- true
>> true && false
<- false
>> false && false
<- false
>> true || true
<- true
>> true || false
<- true
>> false || false
<- false
```

Logical Operators cont.

If you want to invert a boolean value, you can prepend a “not” operator.

```
>> !true
<- false
>> !false
<- true
```