Just like in any programming language, comparison operators play a vital role in the building of a program. Comparison operators, such as == or === in JavaScript (there are many more) help programmers compare statements which ultimately tell the computer what to do given each statement. These statements at their core are most often true or false. A common way to use these operators is within the concept of Iteration. Iteration in programming is simply a way to continue a set of instructions, providing that a given condition is met. For instance, If a is true then run this code. If a is false, then run this other code.

In JavaScript the most common comparison operators are ==, ===, !=, !==, <, >, <=, >=. Lets discuss these further in detail.

The equality operators are inequality (==) and strict equality (===). These two operators are similar however, we must look at the subtle but distinct difference. When we use ==, we are not looking at comparing the data type of our variable, object etc. In this case we can compare a string to an int due to JS converting the string 3 to and int 3. Thus if we said:

var x = “3”;

var y = 3;

If (x == y) { console.log(“TRUE”)}

We would expect that the JS output would be TRUE as the equality of == is going to convert the string to an int.

This would not be the case for our triple equals or strict equality (===). The strict equality looks at type because it performs type casting. If we use our same variables as above we would not get TRUE printed out.

!= is considered to be one of our inequality operators. In essence it represents the syntax of, “does not equal” in common language. It says that if some statement does not equal another, then perform whatever code. It is a more direct way of comparing inequality vs just relying on a false statement of equality. It also does not look at type as in our equality (==) statement.

!== is considered to be our strict inequality operator and it acts like our inequality statement but it performs type casting. In other words it acts just like our strict equality (===) operator and will need to be comparing like types in addition to values.

Our greater than > and less than < operators are most often used when comparing number values. We read them from left to right and are fairly straight forward. A common place to use these is also with iterating. The greater than operator compares if the left operand is larger than the right operand and the less than operator compares if the left operator is less than the right operand.

When we add equality to our greater than >= or less than <= operator we are including the actual number of comparisons. This is common when we iterate with loops. Say we would like to print 10 numbers to our document. If we use a while loop and we wrote:

var i = 1;

while (i < 10) {

document.writeln(i);

++i;

}

We would see an output of 1-9. This is because the less than < operator will not include 10. So we need to use our <= operator.

var i = 1;

while (i <= 10) {

document.writeln(i);

++i;

}

This is what will give us our output of 1-10