**Operators**

There are operators, and there are operands.

Example:

1 + 1

The plus (+) sign is an OPERATOR, because it performs some action. The 1's are OPERANDS, because they are the things being acted, or operated, upon.

You already know several JavaScript operators:

**+** (plus)  
**-** (minus)  
**\*** (multiply)  
**/** (divide)

Here are some more operators, called COMPARISON operators, which you are probably also familiar with:

**<** (less than)  
**>** (greater than)  
**<=** (less than or equal to)  
**>=** (greater than or equal to)

You have already encountered the "gets" (=) operator, also called the "assignment" operator. As you know, this operator is NOT used for comparisons; rather, it takes whatever is on its right side, and places that information into whatever is on its left side.

Example:

var george = 1;

To make comparisons for equality, you need to use the EQUALITY operator, which is represented by two equals signs together:

**==** (is equal to)

If you want to test whether something is NOT equal, you would use the INEQUALITY operator, which is represented by an exclamation point and an equals sign together:

**!=** (is not equal to)

Example:

if (theSky == "gray") putOnYourOvercoat();

If the sky is gray, put on your overcoat.

Example:

if (theWater != "warm") turnOnTheHotWater();

If the water is not warm, turn on the hot water.

Both of the above JavaScript examples use something called BRANCH logic; we'll talk more about JavaScript logic in the next section. I would also like to point out that I have created variable and function names which reflect PURPOSE, and, when used, sound very much like an English sentence; you should strive for this sort of clarity when writing JavaScript code.

Here are some more operators:

Logical AND:

**&&** (and)

Logical OR:

**||** (or)

Example:

if (theSky == "blue" && theTemperature >= 65) takeOffCoatAndTakeWalkOutside();

If the sky is blue AND the temperature is greater than or equal to 65 degrees Fahrenheit, take off coat and take walk outside. The condition above will only be met if BOTH the sky is blue AND the temperature is greater than or equal to 65; otherwise, nothing will happen.

Example:

if (theSky == "gray" || theRain == "falling") stayHomeWithABook();

If the sky is gray OR the rain is falling, stay home with a book. The condition above will be met if EITHER the sky is gray OR the rain is falling, which allows you more potential opportunities to stay home with a good book.

Again, I'll discuss JavaScript branch logic syntax in more detail in the next section.

JavaScript also provides something called the INCREMENT and DECREMENT operators, which either add or subtract 1 from something.

**++** (increment)  
**--** (decrement)

Example:

var myCounter = 0;

myCounter++;

// myCounter is now equal to 1.

myCounter--;

// myCounter is now equal to 0.

myCounter--;

// myCounter is now equal to -1.

Without the increment and decrement operators, we would be forced to say:

var myCounter = 0;

myCounter = myCounter + 1;

// myCounter is now equal to 1.

myCounter = myCounter - 1;

// myCounter is now equal to 0.

myCounter = myCounter - 1;

// myCounter is now equal to -1.

The increment and decrement counters provide a shorthand for adding 1 to a variable or subtracting 1 from a variable. This is extremely useful when counting mechanical repetitions using LOOP logic, another form of JavaScript logic which we will introduce in the next section.

The last operator we will discuss is called the "assignment with addition" operator, or the "add by value" operator.

**+=** (add by value)

This is another shorthand operator, like the increment/decrement operators. If we say "a += b", it is the same as saying "a = a + b". With the "by value" or "assignment with operation" operators, you get two actions for the price of one! The first operation is performed between the two operands, then the result of that operation is put into the first operand, leaving the second (right-hand) operand unchanged.

Example:

var fred = 1;

var ethel = 2;

fred += ethel;

In the above example, the variable fred starts out equal to 1, and the variable ethel starts out equal to 2. After the "add by value" operator works on the two variables, fred is now equal to 3, while ethel is still equal to 2. Again, this is because:

fred += ethel;

is the same as saying:

fred = fred + ethel;

This operator is VERY important in JavaScript because it is used to CONCATENATE, or join, strings together. You'll need to do this in order to create HTML text on-the-fly using JavaScript.

Example:

var myName = "Michael";

var myText = "<p>Hi there, ";

myText += myName;

myText += "! How nice to see you!</p>";

In the above example, I am using the "add by value" operator to concatenate my HTML string together, adding in the user name stored in the variable myName, to create one whole paragraph of HTML code. If this were a real piece of JavaScript code, I would probably have gotten the user's name from a prompt or out of a form text INPUT field, and stored that information, placing it into HTML pages (using SCRIPT tags and the document.write() method) at key points to personalize the site for the user.

There are a lot of other operators, but these are most (though not all) of the essential ones. For more information about operators, see *"JavaScript: The Definitive Guide"*, Chapter 5.

**Summary of Operators**

**+** (plus)  
**-** (minus)  
**\*** (multiply)  
**/** (divide)

**=** (gets, assignment)  
**==** (equality, is equal to)  
**!=** (inequality, is not equal to)

**<** (less than)  
**>** (greater than)  
**<=** (less than or equal to)  
**>=** (greater than or equal to)

**&&** (and, logical AND)  
**||** (or, logical OR)

**++** (increment, adds 1 to operand)  
**--** (decrement, subtracts 1 from operand)

**+=** (assignment with addition, add by value)