JavaScript Variables

Variables are "containers" for storing information.
JavaScript variables are used to hold values or expressions.
A variable can have a short name, like x, or a more descriptive name, like carname.
Rules for JavaScript variable names:

- Variable names are case sensitive (y and Y are two different variables)
- Variable names must begin with a letter or the underscore character

Declaring (Creating) JavaScript Variables

```
var x;

var carname;
```

Assigning Values to Variables

```
var x=5;

var carname="Scorpio";
```

The Lifetime of JavaScript Variables

When you declare a variable within a function, the variable can only be accessed within that function.
When you exit the function, the variable is destroyed. These variables are called local variables. You can
have local variables with the same name in different functions, because each is recognized only by the
function in which it is declared.
If you declare a variable outside a function, all the functions on your page can access it. The lifetime of
these variables starts when they are declared, and ends when the page is closed.

DataTypes

• Numbers - are values that can be processed and calculated. You don't enclose them in quotation
marks. The numbers can be either positive or negative.

•

Strings - are a series of letters and numbers enclosed in quotation marks. JavaScript uses the string
literally; it doesn't process it. You'll use strings for text you want displayed or values you want
passed along.

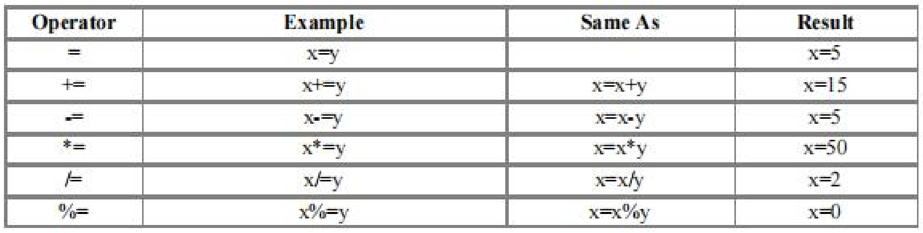
• Boolean (**true**/**false**) - lets you evaluate whether a condition meets or does not meet specified
criteria.

• Null - is an empty value. **null** is not the same as 0 -- 0 is a real, calculable number, whereas **null** is
the ***absence of any value***.

JavaScript Operators

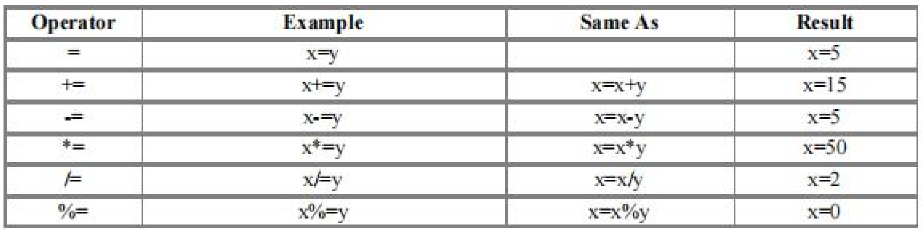
Assignment operators
Comparison operators
Arithmetic operators
Bitwise operators
Logical operators
BigInt operators
String operators
Conditional (ternary) operator
Comma operator
Unary operators
Relational operators

ARITHMETIC:-



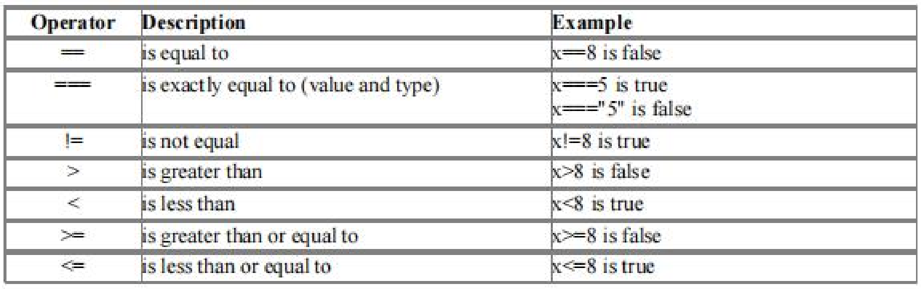
Operator	Example	Same As	Result
=	x=y		x=5
+=	x+=y	x=x+y	x=15
-=	x-=y	x=x-y	x=5
*=	x*=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=0

ASSIGNMENT:-



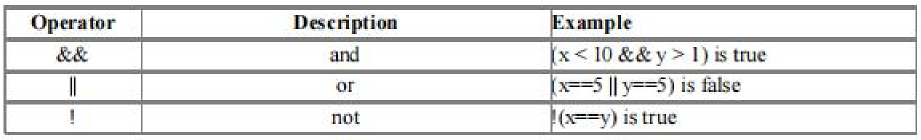
Operator	Example	Same As	Result
=	x=y		x=5
+=	x+=y	x=x+y	x=15
-=	x-=y	x=x-y	x=5
*=	x*=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=0

COMPARISON:-



Operator	Description	Example
==	is equal to	x==8 is false
===	is exactly equal to (value and type)	x===5 is true x==="5" is false
!=	is not equal	x!=8 is true
>	is greater than	x>8 is false
<	is less than	x<8 is true
>=	is greater than or equal to	x>=8 is false
<=	is less than or equal to	x<=8 is true

LOGICAL:-



Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
||	or	(x==5 || y==5) is false
!	not	!(x==y) is true

JavaScript Controlling(Looping) Statements

Loops in JavaScript are used to execute the same block of code a specified number of times or while a specified condition is true.

JavaScript Loops

Very often when you write code, you want the same block of code to run over and over again in a row.
Instead of adding several almost equal lines in a script we can use loops to perform a task like this.
In JavaScript there are two different kind of loops:

- **for** - loops through a block of code a specified number of times
- **while** - loops through a block of code while a specified condition is true

Conditional Statements

Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.
In JavaScript we have the following conditional statements:

- **if statement** - use this statement if you want to execute some code only if a specified condition is true
- **if...else statement** - use this statement if you want to execute some code if the condition is true and another code if the condition is false
- **if...else if....else statement** - use this statement if you want to select one of many blocks of code to be executed
- **switch statement** - use this statement if you want to select one of many blocks of code to be executed

JavaScript Functions

A function (also known as a *method*) is a self-contained piece of code that performs a particular
"function". You can recognise a function by its format - it's a piece of descriptive text, followed by open
and close brackets.A function is a reusable code-block that will be executed by an event, or when the
function is called.
To keep the browser from executing a script when the page loads, you can put your script into a function.
A function contains code that will be executed by an event or by a call to that function.
You may call a function from anywhere within the page (or even from other pages if the function is
embedded in an external .js file)

How to Define a Function

```
function functionname(var1,var2,...,varX)
{
some code
}
```