Front-end development-

JavaScript

JavaScript Home

* Javascript is written inbetween the <script> and </script> tag
* Function = the thing that happens when the javascript kicks in
* Put javascript reference in the src in <script> for external files

Output

Javascript can be displayed in many ways

* innerHTML =HTML content
* document.write() = used for testing purposes. USING THIS AFTER THE HTML DOCUMENT HAS LOADED WILL DELETE ALL EXISTING HTML
* window.alert()= for alert boxes
* console.log() = for debugging

// statements

* values, operators, expressions, keywords and comments
* ;= separates each statement
* Break codes afer operators if its longer than 80 letters
* Use code blocks in {}

Literals= “” or .

Variable = var

Operators =+-\*/

Assign value to an operator using the “ = “ sign

var x, y;  
x = 5 + 6;  
y = x \* 10;

"John" + " " + "Doe" = John Doe

// = comment or /\* between \*/

Javascript names begin with \_ or $

firstName = type of lower camel case

comments

// = single line comments

/\* = start of multi-line comment

\*\= end of multi-line comment

Variables

== is the equals sign

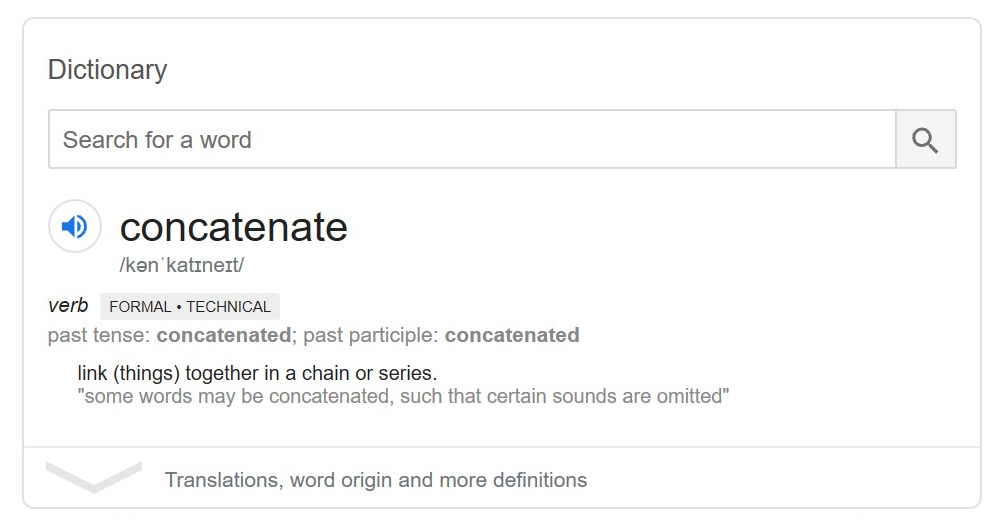
= means equal to

// declare all variables at the beginning of a script

Separate variables using commas

Redeclaring a javascript variable does not get rid of the value

If you put a string of words or numbers in a var equation e.g 1 + 2 + 3 this would give 6 because there is no “ “ signs. If you did “1” + 2 + 3 this would give 15. Anything following the “” gets concatenated.

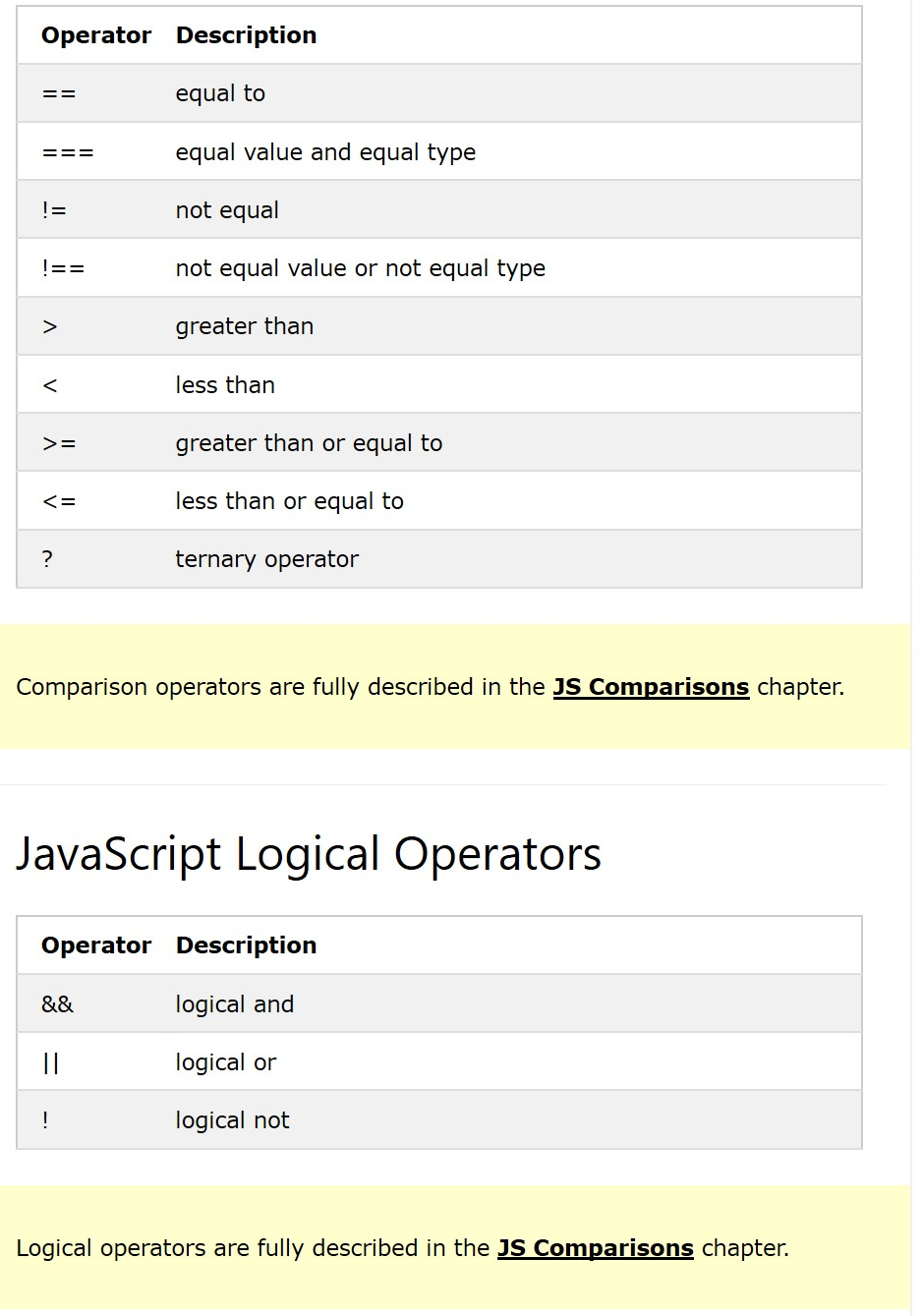


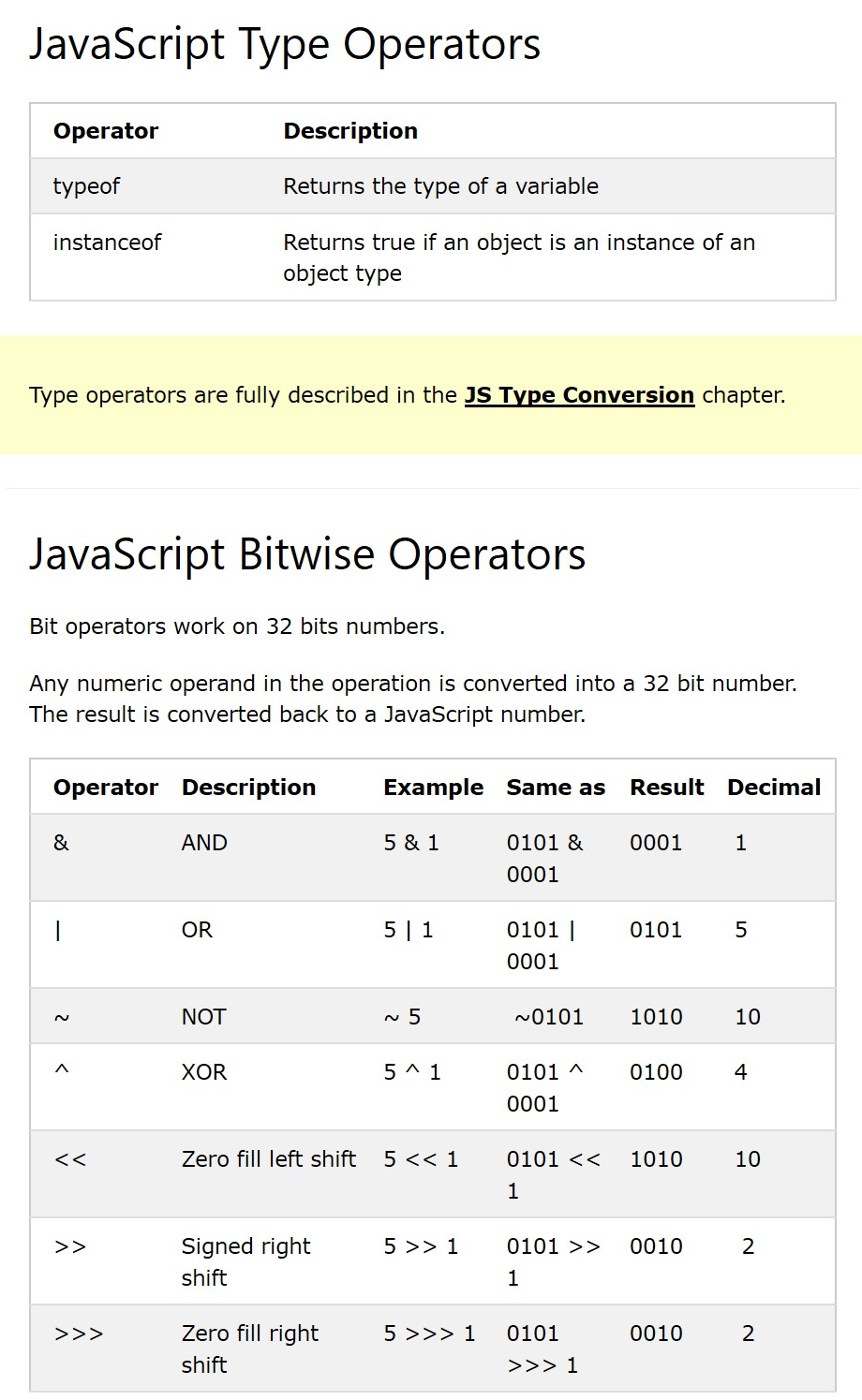
Operators

|  |  |
| --- | --- |
| **Operator** | **Description** |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| \*\* | Exponentiation ([ES2016](https://www.w3schools.com/js/js_es6.asp)) |
| / | Division |
| % | Modulus (Division Remainder) |
| ++ | Increment |
| -- | Decrement |

|  |  |  |
| --- | --- | --- |
| **Operator** | **Example** | **Same As** |
| = | x = y | x = y |
| += | x += y | x = x + y |
| -= | x -= y | x = x - y |
| \*= | x \*= y | x = x \* y |
| /= | x /= y | x = x / y |
| %= | x %= y | x = x % y |
| \*\*= | x \*\*= y | x = x \*\* y |

If you add a number and a string you will end up with a string regardless of “” or not.





Arithmetic

Var x = 5;

Var y = 2;

Var z = x%y; // the answer to this is 1. Because 5/2=2remainder1. % only gives the remainder so the answer is 1.

++ = add one

-- = minus 1

\*\* = squared

* If x = 5

Math.pow(x,2) = same as doing 5\*\*2

Assignment operators

\*\*= is not stable in all browsers so do not use it.

Data Types

var x = 16 + 4 + "Volvo";

this results in 20Volvo

var x = "Volvo" + 16 + 4;

this results in 164

if you want quotes the quotes inside the script, the quotation marks cant be the same as the ones on the outside. E.g “ cant be put with ‘.

Booleans = true or false

Arrays = [ ] and items are separated by commas

Objects = {} can be a person with many desctiptions or properties

Typeof = return the type of variable the variable is

Undefined and empty are not the same thing.

Null = an object but it is nothing.

Empty an object by setting it to undefined.

Primitive data value = single simple data value with no additional properties and methods.

Typeof operator can return either function (for functions) or object (objects arrays and null)

Functions

* Function name ()
* Code to be executed in put into {}
* Object and return
* Local = can only be accessed from within the function
* Since local variables are only recognized inside their functions, variables with the same name can be used in different functions.
* Local variables are created when a function starts, and deleted when the function is completed.

Objects

Events – when something happens in html e.g the link is clicked

Strings – zero or more characters in quotes

* \” = used to add quotes in a string.

Sting methods = length, indexOf(), lastIndexOf(), search(), slic(start, end), substring(start, end) and substr(start, Length), toUpperCase(), toLowerCase(), concat(), trim(), replace(), String.prototype

s

NaN = not a number

Increment ++

Decrement –

\*\* power of.

\*\* is the same as Math.pow(x,y)

Arrays are items separated by commas []

Objects are written in {}

Primitive data value= single simple data value with no additional properties and methods. Primitives can be strings, numbers, Booleans or undefined.

Function= block of code designed to perform a particular task.

Local variables can only be accessed from within a function. They are created when a function starts and deleted when the function has ended.