JAVASCRIPT

September 13th

Javascript is the programming language of the web.

Ways of outputting javascript content:

* console.log() – outputs data into the console of the browser
* alert()/window.alert() – outputs on browser window
* document.write() – outputs on HTML elements
* Using innerHtml – writing into HTML tags

document.write() should only be used for testing purposes, since using it after the HTML document has loaded will erase all other elements.

There are three ways of declaring variable in Javascript:

* var
* let
* const

let and const were introduced in 2015, so previous browsers such as Internet Explorer 11 and going back don’t support it.

We should always use const and use let only if we have to.

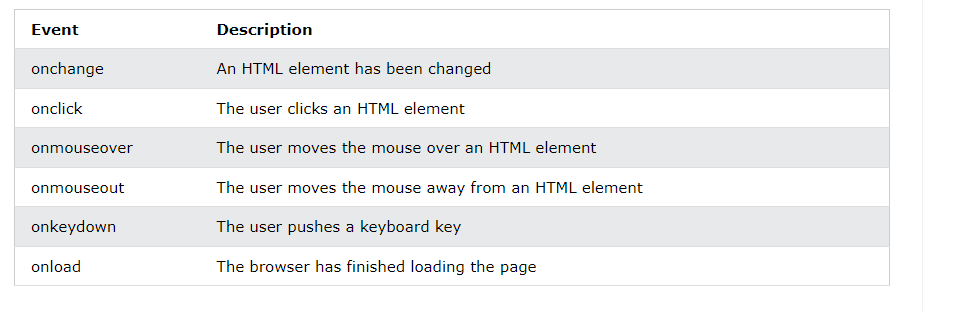
Types of operators in Javascript:

* Arithmetic such as +, \*, -, /, etc
* Assignment such as =, +=, etc
* Comparison such as ==, ===, >, <, != etc
* Logical such as &&, || etc
* Bitwise such as &, | etc
* String such as +
* Ternary such as ?
* Type such as typeof, instanceof

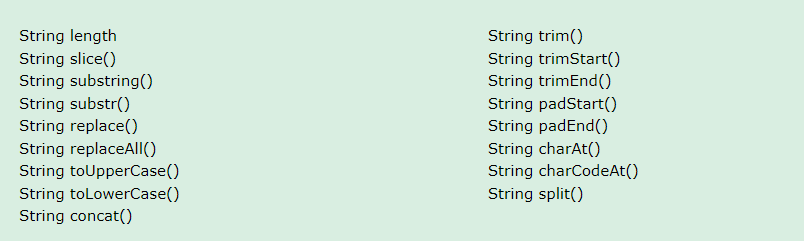
*Events*

An event is something that has happened to a html element such as clicking a button, changing the content of an input field, or completion of loading of the html page.

Javascript can react to such events.



Javascript string methods:



All string methods return a string and do not modify the original string.

String length returns the length of a string.

String slice() takes two arguments, the start and end index of the slicing (with the end index excluded)

String substring() acts the same way as slice() except that any negative arguments are treated as 0

String substr() acts the same way as slice() except that the second argument is the length of the slicing

String replace() takes two arguments. The first is the string to be replaced and the second is the replacement. It only replaces the first occurrence of the string

String replaceAll() acts as replace() except that it replaces all occurrences of the string to be replaced

String toUpperCase() and toLowerCase() are kind of self-explanatory.

String concat() takes the first argument as the item to join the strings and the second as the string to be added

String trim(), trimStart() and trimEnd() removes all white spaces from both sides, the start and the end respectively

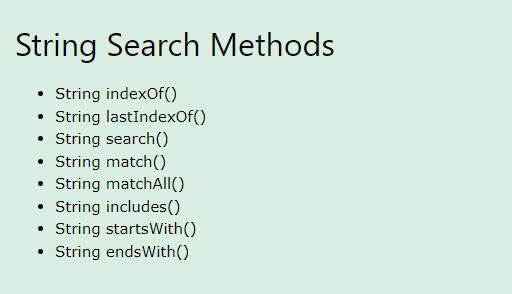
String split() takes one argument, the string item to be used for splitting. If not specified, the string is split into an array with the whole string as the first and only item.

String padStart() and padEnd() take two arguments, the number of paddings and the string to be used for padding purposes.

String charAt() and charCodeAt() take a number argument and returns the character at that index and the Unicode at that index respectively.

September 14th

More string methods in regards to searching:



String indexOf() the first argument as string to be searched and the second argument as a number of where to start the search from. String lastIndexOf() works the same way except it searches backwards. They both return the index of the start of the search string or -1 if the search string is not found.

String search() takes a string or regular expression as an argument and returns its index.

String search() and string indexOf() and lastIndexOf() return the same values but search() can only take one argument while the others cannot use regular expressions.

String match() returns string match details while string matchAll() returns an iterator.

String includes() returns true if the search string is available, else false

The same goes for string startsWith() and endsWith().

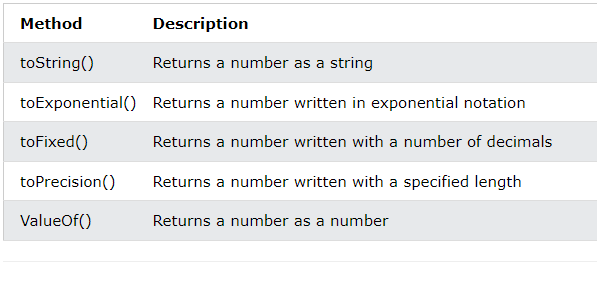
September 15th

*Numbers*

Javascript numbes are only accurate up to 15 digits.

A string can be converted to a number using Number() function. A number can be converted to a string using the toString() function, which can take the base of the number as an argument (to base 2 for binary, to base 8 for octal and to base 16 for hexagonal).

Number methods include:



Number toString() returns a number as a string.

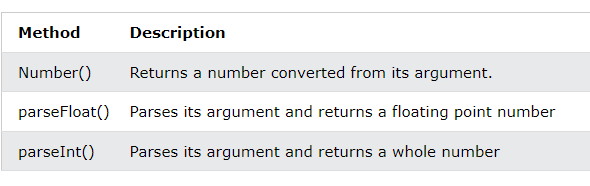
Number toExponential() returns a number written in exponential notation

Number toFixed() truncates a number to the specified number of decimal places.

Number toPrecision() returns a specified number of digits.

ValueOf() returns a number as a number.

There are three Javascript methods that can be used to convert a variable to a number:



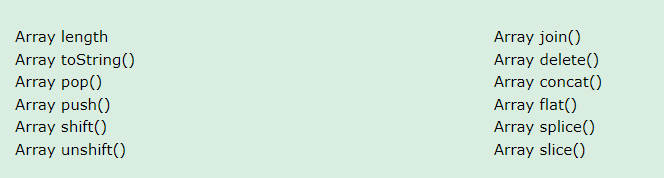
The above methods return a number version of the arguments.

*Arrays*

JavaScript arrays are like objects with indexes as keys.

Type of array returns object. To check if it is really an array we can either use array instanceof Array or Array.isArray(array) which both return a Boolean value.

Common array methods include:



Array pop() removes the last index of the array.

Array push() adds the arguments it takes to the end of an array

Array shift() is like pop() but from the beginning of the array. Array unshift() is like push() but at the beginning of the array.

Array join() joins the array items into a string and can take a joining string as an argument.

Array flat() flattens the array if the array has other arrays within it

Array concat() joins arrays into a longer array.

delete array[i] deletes the item at index i and leaves the space undefined.

Array splice() takes 2 or more arguments. The first is the position of the item to be replaced and the second is the number of items to be removed. Any other argument after that will be what will be inserted at the specified position( the first argument). It can be used to remove items from an array.

Array slice() takes either one or two arguments, the start index of slicing and the start and end index of slicing.

September 16th

The array sort() method sorts the items as strings by default. A function is required in order to sort numbers.

The Math.max.apply(null, array) returns the highest value in the array. The format goes for min.

Objects within an array can also be sorted using the same format.

The array forEach(function) takes a function as an argument, calling it across all items.

The array map(function) returns a new array with the function argument performed on all items

The array filter(function) takes a function that returns a Boolean value and creates a new array of items that satisfy the function.

The methods above take a function as an argument, which takes three arguments, the value, the index and the array.

The array reduce(function) and reduceRight(function) take a function as an argument, that takes four arguments, the total, the value, the index and the array.

The array every(function) and some(function) check if all items or any item respectively, satisfy a condition.

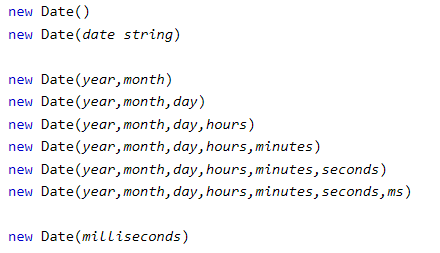
The array from() method creates an array out of a string argument.

The array find(function) and findIndex(function) returns the first value and the index of the first value respectively, that satisfy a condition.

September 17th

JavaScript Date objects are not “moving” like the clock in our device/computer.

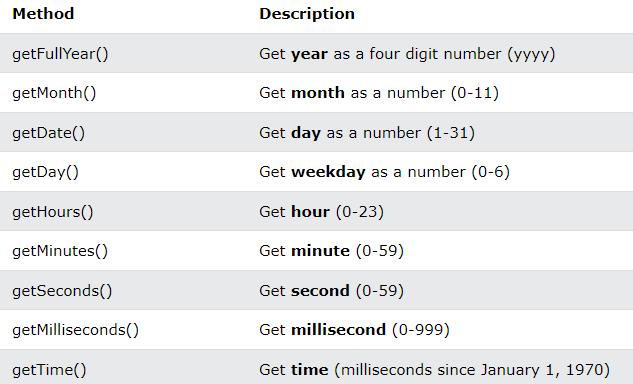
The following are the nine ways of creating a date object:



A date object has the following methods:

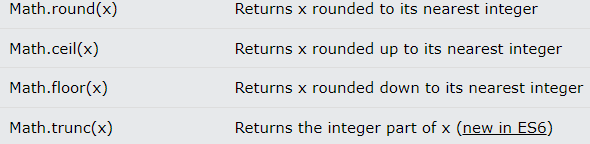
* Date toDateString() that returns a better readable date format
* Date toUTCString() that returns a Time-Zone date format
* Date toISOString() that returns a ISO standards date.

Date object get methods include:



*Math*

Common math methods include:



The methods above are self-explanatory.

The math random() method returns a random number between 0 and 1, (0 included and 1 excluded). We can use that concept to find a random number between either two numbers or a maximum number.

*Conditional statements and loops*

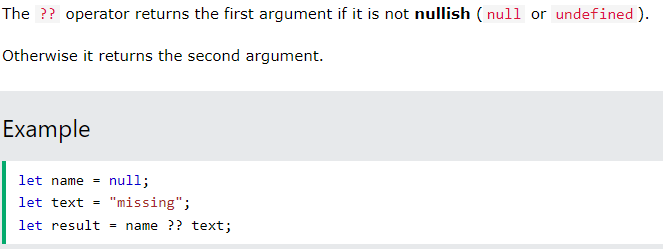
There are four conditional statements in JavaScript, if, else if, else and switch.

September 18th

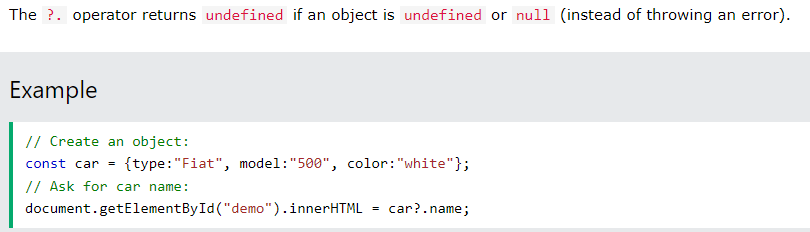
The ? operator can be used to make comparisons in place, returning one of two conditions.



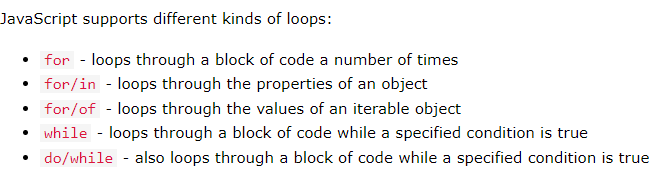
The ?? operator returns a statement if a condition is nullish, else, it returns the item.



The ?. can be used to check object attributes without throwing an error.



*Loops*



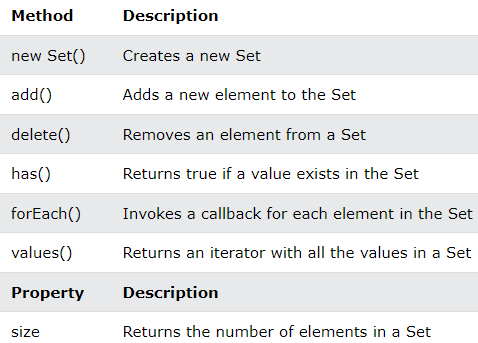
The break keyword is used to stop an iteration. The continue keyword is used to move to the next iteration.

Iterables are items that can be iterated over, such as strings and arrays.

JavaScript objects are not directly iterable.

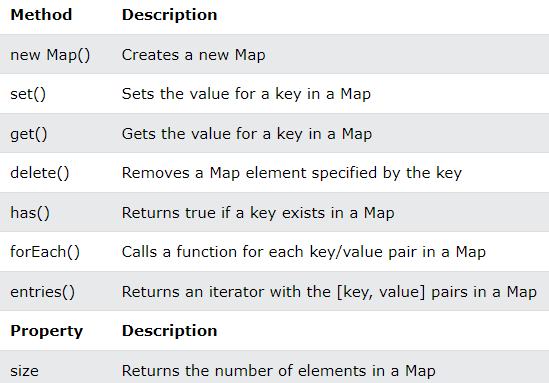
*Sets*

A set is a collection of unique values.



A map is an item with key-value pairs. They are just the same as dictionaries in python.

Map methods include:

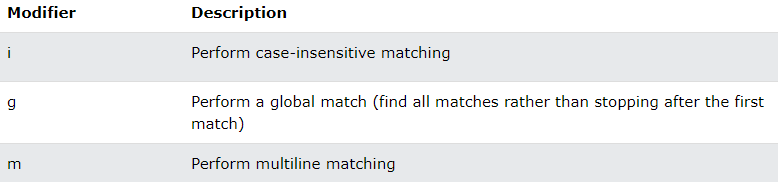


*Regular expressions*

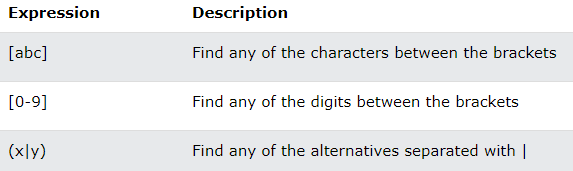
The syntax for a regular expression in JavaScript is:

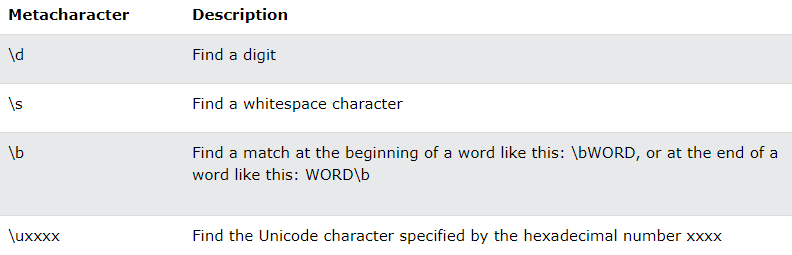


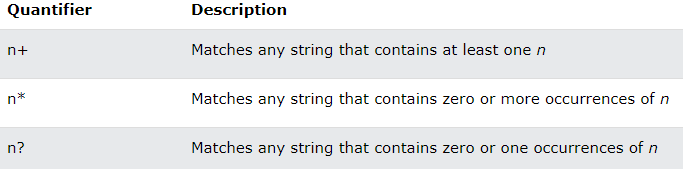
JavaScript regular expressions are often used with the string methods search() and replace().



The match() method, when used with a regular expression, returns details of a match. If we use the global modifier, it returns an array of all the matches. Hence, this can be used to find the count of a match.



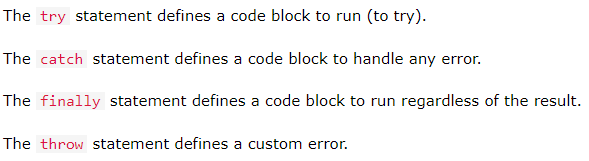




The regex test() method returns true if a match is found, else false

The regex exec() method returns the matched string, just like match() without a global modifier.

*Errors*



September 20th

*Scope*

There are three types of scopes in JavaScript:

* Global
* Function
* Block

Variables defined inside a block scope {} can only be used within the block, unless they are declared using var.

Variables declared within a function have a local scope, meaning they can only be used within the function, even if declared using var.

Variables declared on the global scope, or rather the JavaScript environment can be used within other functions or blocks. If you assign a variable to a value without declaring it, it automatically becomes global.

In “strict mode”, assigning values to a variable without declaring them does not automatically make them global.

Do not create global variables unless you intend to. Global variables and functions can overwrite window variables or functions.

The lifetime of a variable begins when it is declared. Global variables are deleted when the window or browser is closed and local variables are deleted when the function or block is closed or completed.

*Hoisting*

Hoisting is the default way that JavaScript uses to automatically move all variable declarations to the top. This only works for var. Hoisting only moves declarations to the top and does not do the same for initializations.

*Arrow functions*

Arrow functions are like lambda functions in python. They are anonymous functions that are usually of one statement that can be used in line with a statement.

*Class*

A JavaScript class is a template for creating JavaScript objects.

We should always use a constructor function when we create a class in order to define the base class variables.

*Modules*

Classes and functions can be imported from other JavaScript files. In order for an import to take place, the function or class has to be exported from its source.



If it is a default export (There can only be one default export), it should be exported without the curly braces.



Afterwards, the file can be imported in the normal way as



Or as a default export as



*JSON*

This stands for JavaScript object notation. It is a format that JavaScript uses to store or transport data. They kind of represent a JavaScript object with both attributes and values as strings or like python dictionaries with both keys and values as strings.

To convert a JSON file into an object, we use JSON.parse(json file)

To convert an object into a JSON file, we use JSON.stringify(object)

JSON files are basically strings in data type

*Best Practices*

* Avoid global variables
* Avoid using new Object to create variables
* Always declare variables and do it on top of the scope
* Always initialize variables
* Beware of automatic type conversions such as string and number additions
* Always use === for comparisons
* Use parameter defaults
* Always end a switch statement with a default case