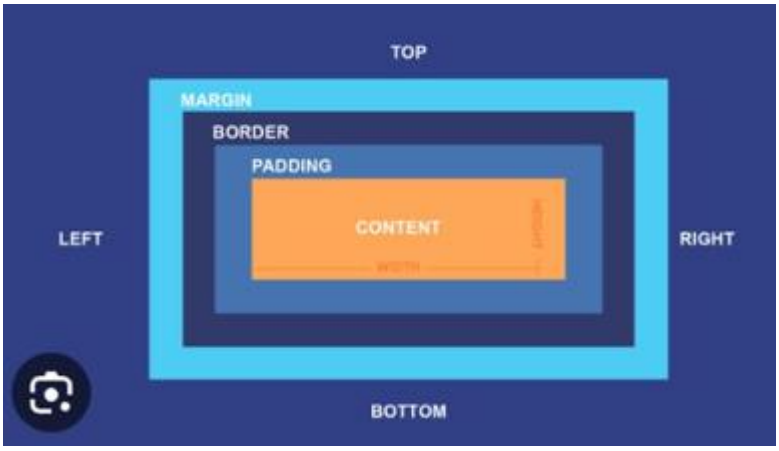

TOP
MARGIN
BORDER
PADDING
CONTENT
LEFT
RIGHT
BOTTOM

BORDER

border-top	*border-top-width* *border-style* *border-color*
border-top-color	*border-color*
border-top-style	*border-style*
border-top-width	thin | medium | thick *length*
border-width	thin | medium | thick *length*
border-radius	*border-top-right-radius* *border-bottom-right-radius* *border-bottom-left-radius* *border-top-left-radius*
border-top-right-radius	*length*
border-bottom-right-radius	*length*
border-bottom-left-radius	*length*
border-top-left-radius	*length*
box-shadow	inset | [*length*, *length*, *length*, *length* | | *<color>*] none
border-style	none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset

TRANSITIONS

transition	transition-property transition-duration transition-timing-function transition-delay
transition-delay	time
transition-duration	time
transition-property	none | all
transition-timing-function	ease | linear | ease-in | ease-out | ease-in-out | cubic-Bezier (number, number, number, number)

BOX MODEL

clear	left | right | both | none
display	none | inline | block | inlineblock | list-item | run-in | compact | table | inlinetable | table-row-group | table-header-group | tablefooter-group | table-row | table-column-group | tablecolumn | table-cell | tablecaption | ruby | ruby-base | ruby-text | ruby-base-group | ruby-text-group
float	left | right | none
height	auto *length* %
max-height	none *length* %
max-width	none *length* %
min-height	none | inherit *length* %
min-width	none | inherit *length* %
width	auto % *length*
margin	*margin margin-top* *margin-right* *margin-bottom* *margin-left*
margin-bottom	auto *length* %
margin-left	auto *length* %
margin-right	auto *length* %

BOX MODEL

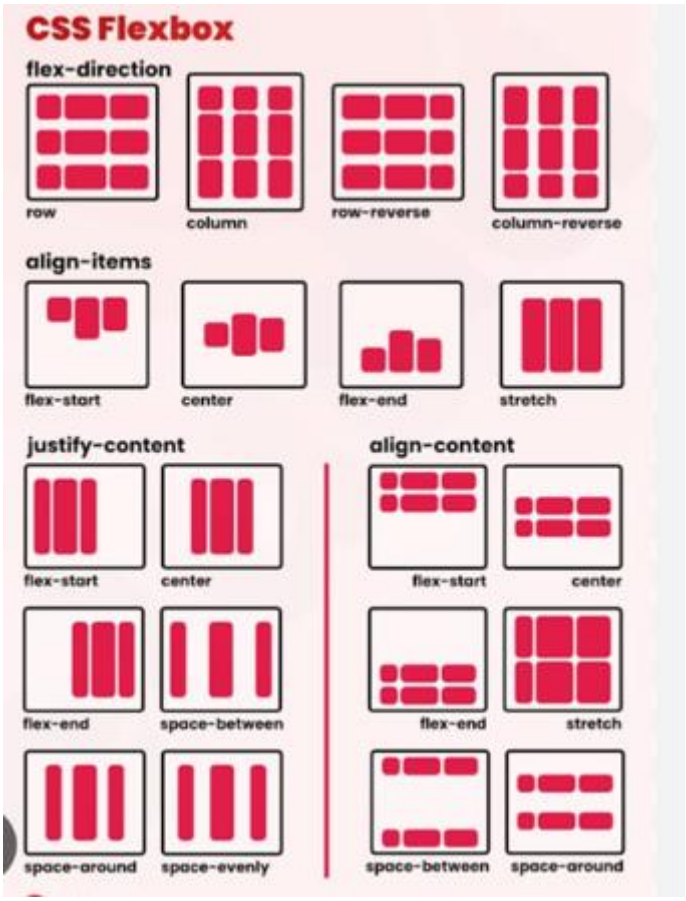
margin-top	*auto* *length* *%*
padding	*padding padding-top* *padding-right* *padding-bottom* *padding-left*
padding-bottom	*length* *%*
padding-left	*length* *%*
padding-right	*length* *%*
padding-top	*length* *%*
marquee-direction	forward | reverse
marquee-loop	infinite *number*
marquee-play-count	infinite *integer*
marquee-speed	slow | normal | fast
marquee-style	scroll | slide | alternate
overflow	visible | hidden | scroll | auto | no-display | nocontent *overflow-x* *overflow-y*
overflow-style	auto | marquee-line | marquee-block
overflow-x	visible | hidden | scroll | auto | no-display | nocontent
overflow-y	visible | hidden | scroll | auto | no-display | nocontent
rotation	*angle*
rotation-point	*position (paired value offset)*
visibility	visibility visible | hidden | collapse

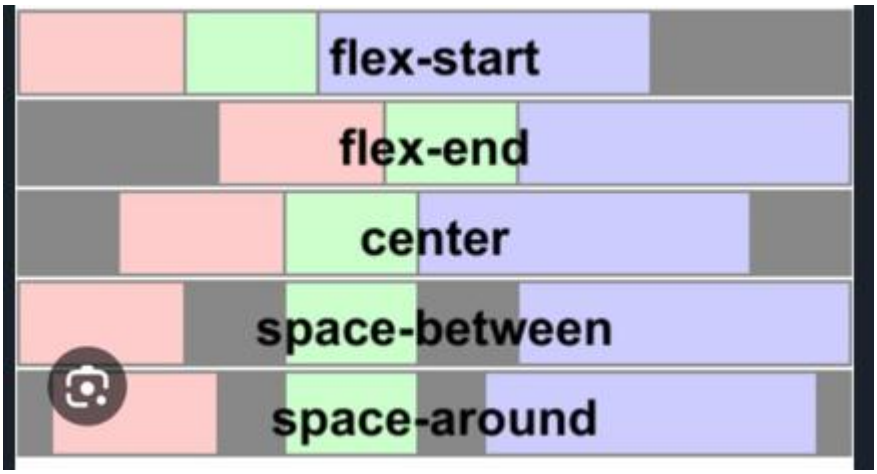
FONT

font	*font-style* *font-variant* *font-weight* *font-size/line-height* *font-family* caption | icon | menu | message-box | smallcaption | status-bar
font-family	*family-name* *generic-family* inherit
font-size	xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger | inherit *length* *%*
font-size-adjust	none| inherit *number*
font-stretch	normal | wider | narrower | ultra-condensed | extracondensed | condensed | semi-condensed | semiexpanded | expanded | extra-expanded | ultraexpanded | inherit
font-style	normal | italic | oblique | inherit
font-variant	normal | small-caps | inherit
font-weight	normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit

COLOR

color	inherit *color*
opacity	inherit *number*


CSS Flexbox
flex-direction
row
column
row-reverse
column-reverse
align-items
flex-start
center
flex-end
stretch
justify-content
flex-start
center
flex-end
space-between
space-around
space-evenly
align-content
flex-start
center
flex-end
stretch
space-between
space-around


flex-start
flex-end
center
space-between
space-around

Number() = 42

PROPERTIES

- .POSITIVE_INFINITY +∞ equivalent
- .NEGATIVE_INFINITY -∞ equivalent
- .MAX_VALUE largest positive value
- .MIN_VALUE smallest positive value
- .EPSILON diff between 1 & smallest >1
- .NaN not-a-number value

METHODS

- .toExponential(dec) exp. notation
- .toFixed(dec) fixed-point notation
- .toPrecision(p) change precision
- .isFinite(n) check if number is finite
- .isInteger(n) check if number is int.
- .isNaN(n) check if number is NaN
- .parseInt(s, radix) string to integer
- .parseFloat(s, radix) string to float

Regexp() = /.+/ig

PROPERTIES

- .lastIndex index to start global regexp
- .flags active flags of current regexp
- .global flag g (search all matches)
- .ignoreCase flag i (match lower/upper)
- .multiline flag m (match multiple lines)
- .sticky flag y (search from lastIndex)
- .unicode flag u (enable unicode feat.)
- .source current regexp (w/o slashs)

METHODS

- .exec(str) exec search for a match
- .test(str) check if regexp match w/str

CLASSES

. any character	\t tabulator
\d digit [0-9]	\r carriage return
\D no digit [^0-9]	\n line feed
\w any alphanumeric char [A-Za-z0-9_]	
\W no alphanumeric char [^A-Za-z0-9_]	
\s any space char (space, tab, enter...)	
\S no space char (space, tab, enter...)	
\xN char with code N	[\b] backspace
\uN char with unicode N	\0 NUL char

CHARACTER SETS OR ALTERNATION

- [abc] match any character set
- [^abc] match any char. set not enclosed
- a|b match a or b

BOUNDARIES

^ begin of input	$ end of input
\b zero-width word boundary	
\B zero-width non-word boundary	

GROUPING

- (x) capture group (?:x) no capture group
- \n reference to group n captured

QUANTIFIERS

- x* preceding x 0 or more times {0,}
- x+ preceding x 1 or more times {1,}
- x? preceding x 0 or 1 times {0,1}
- x{n} n ocurrences of x
- x{n,} at least n ocurrences of x
- x{n,m} between n & m ocurrences of x

ASSERTIONS

String() = 'text'

PROPERTIES

- .length string size

METHODS

- .charAt(index) char at position [i]
- .charCodeAt(index) unicode at pos.
- .fromCharCode(n1, n2...) code to char
- .concat(str1, str2...) combine text +
- .startsWith(str, size) check beginning
- .endsWith(str, size) check ending
- .includes(str, from) include substring?
- .indexOf(str, from) find substr index
- .lastIndexOf(str, from) find from end
- .search(regex) search & return index
- .localeCompare(str, locale, options)
- .match(regex) matches against string
- .repeat(n) repeat string n times
- .replace(str|regex, newstr|func)
- .slice(ini, end) str between ini/end
- .substr(ini, len) substr of len length
- .substring(ini, end) substr fragment
- .split(sep|regex, limit) divide string
- .toLowerCase() string to lowercase
- .toUpperCase() string to uppercase
- .trim() remove space from begin/end
- .raw`` template strings with ${vars}

Date()

METHODS

- .UTC(y, m, d, h, i, s, ms) timestamp
- .now() timestamp of current time
- .parse(str) convert str to timestamp
- .setTime(ts) set UNIX timestamp
- .getTime() return UNIX timestamp

UNIT SETTERS (ALSO .setUTC() METHODS)

- .setFullYear(y, m, d) set year (yyyy)
- .setMonth(m, d) set month (0-11)
- .setDate(d) set day (1-31)
- .setHours(h, m, s, ms) set hour (0-23)
- .setMinutes(m, s, ms) set min (0-59)
- .setSeconds(s, ms) set sec (0-59)
- .setMilliseconds(ms) set ms (0-999)

UNIT GETTERS (ALSO .getUTC() METHODS)

- .getDate() return day (1-31)
- .getDay() return day of week (0-6)
- .getMonth() return month (0-11)
- .getFullYear() return year (yyyy)
- .getHours() return hour (0-23)
- .getMinutes() return minutes (0-59)
- .getSeconds() return seconds (0-59)
- .getMilliseconds() return ms (0-999)

LOCALE & TIMEZONE METHODS

- .getTimezoneOffset() offset in mins
- .toLocaleDateString(locale, options)
- .toLocaleTimeString(locale, options)
- .toLocaleString(locale, options)
- .toUTCString() return UTC date
- .toDateString() return American date
- .toTimeString() return American time

Array() = [1, 2, 3]

PROPERTIES

- .length number of elements

METHODS

- .isArray(obj) check if obj is array
- .includes(obj, from) include element?
- .indexOf(obj, from) find elem. index
- .lastIndexOf(obj, from) find from end
- .join(sep) join elements w/separator
- .slice(ini, end) return array portion
- .concat(obj1, obj2...) return joined array

MODIFY SOURCE ARRAY METHODS

- .copyWithin(pos, ini, end) copy elems
- .fill(obj, ini, end) fill array with obj
- .reverse() reverse array & return it
- .sort(cf(a,b)) sort array (unicode sort)
- .splice(ini, del, o1, o2...) del&add elem

ITERATION METHODS

- .entries() iterate key/value pair array
- .keys() iterate only keys array
- .values() iterate only values array

CALLBACK FOR EACH METHODS

- .every(cb(e,i,a), arg) test until false
- .some(cb(e,i,a), arg) test until true
- .map(cb(e,i,a), arg) make array
- .filter(cb(e,i,a), arg) make array w/true
- .find(cb(e,i,a), arg) return elem w/true
- .findIndex(cb(e,i,a), arg) return index
- .forEach(cb(e,i,a), arg) exec for each
- .reduce(cb(p,e,i,a), arg) accumulative
- .reduceRight(cb(p,e,i,a), arg) from end

ASSIGNMENTS METHODS

- .pop() remove & return last element
- .push(o1, o2...) add element & return length
- .shift() remove & return first element
- .unshift(o1, o2...) add element & return len

Boolean() = true / false

no own properties or methods

Function() = function(a, b) { ... }

PROPERTIES

- .length return number of arguments
- .name return name of function
- .prototype prototype object

METHODS

- .call(newthis, arg1, arg2...) change this
- .apply(newthis, arg1) with args array
- .bind(newthis, arg1, arg2...) bound func

number	date
NaN (not-a-number)	regular expresion
string	function
boolean (true/false)	object
array	undefined

only available on ECMAScript 6

- static (ex: Math.random())

Math

PROPERTIES

- .E Euler's constant
- .LN2 natural logarithm of 2
- .LN10 natural logarithm of 10
- .LOG2E base 2 logarithm of E
- .LOG10E base 10 logarithm of E
- .PI ratio circumference/diameter
- .SQRT1_2 square root of 1/2
- .SQRT2 square root of 2

METHODS

- .abs(x) absolute value
- .cbrt(x) cube root
- .clz32(x) return leading zero bits (32)
- .exp(x) return e^x
- .expm1(x) return e^x-1
- .hypot(x1, x2...) length of hypotenuse
- .imul(a, b) signed multiply
- .log(x) natural logarithm (base e)
- .log1p(x) natural logarithm (1+x)
- .log10(x) base 10 logarithm
- .log2(x) base 2 logarithm
- .max(x1, x2...) return max number
- .min(x1, x2...) return min number
- .pow(base, exp) return base^exp
- .random() float random number [0,1)
- .sign(x) return sign of number
- .sqrt(x) square root of number

ROUND METHODS

- .ceil(x) superior round (smallest)
- .floor(x) inferior round (largest)
- .fround(x) nearest single precision
- .round(x) round (nearest integer)
- .trunc(x) remove fractional digits

TRIGONOMETRIC METHODS

- .acos(x) arccosine
- .acosh(x) hyperbolic arccosine
- .asin(x) arcsine
- .asinh(x) hyperbolic arcsine
- .atan(x) arctangent
- .atan2(x, y) arctangent of quotient x/y
- .atanh(x) hyperbolic arctangent
- .cos(x) cosine
- .cosh(x) hyperbolic cosine
- .sin(x) sine
- .sinh(x) hyperbolic sine
- .tan(x) tangent
- .tanh(x) hyperbolic tangent

JSON

METHODS

- .parse(str, tf(k,v)) parse string to object
- .stringify(obj, repf|wl, sp) convert to str

Error()

PROPERTIES

- .name return name of error
- .message return description of error

EvalError(), InternalError(), RangeError(), URIError(), ReferenceError(), SyntaxError(), TypeError()

Object() = {key: value, key2: value2}

PROPERTIES

- .constructor return ref. to object func.

METHODS

- .assign(dst, src1, src2...) copy values
- .create(proto, prop) create obj w/prop
- .defineProperties(obj, prop)
- .defineProperty(obj, prop, desc)
- .freeze(obj) avoid properties changes
- .getOwnPropertyDescriptor(obj, prop)
- .getOwnPropertyNames(obj)
- .getOwnPropertySymbols(obj)
- .getPrototypeOf(obj) return prototype
- .is(val1, val2) check if are same value
- .isExtensible(obj) check if can add prop
- .isFrozen(obj) check if obj is frozen
- .isSealed(obj) check if obj is sealed
- .keys(obj) return only keys of object
- .preventExtensions(obj) avoid extend
- .seal(obj) prop are non-configurable
- .setPrototypeOf(obj, prot) change prot

INSTANCE METHODS

- .hasOwnProperty(prop) check if exist
- .isPrototypeOf(obj) test in another obj
- .propertyIsEnumerable(prop)
- .toString() return equivalent string
- .toLocaleString() return locale version
- .valueOf() return primitive value

Promise()

METHODS

- .all(obj) return promise
- .catch(onRejected(s)) = .then(undef,s)
- .then(onFulfilled(v), onRejected(s))
- .race(obj) return greedy promise (res/rej)
- .resolve(obj) return resolved promise
- .reject(reason) return rejected promise

Proxy()

Reflect same methods (not func)

METHODS

- .apply(obj, arg, arglist) trap function call
- .construct(obj, arglist) trap new oper
- .defineProperty(obj, prop, desc)
- .deleteProperty(obj, prop) trap delete
- .enumerate(obj) trap for...in
- .get(obj, prop, rec) trap get property
- .getOwnPropertyDescriptor(obj, prop)
- .getPrototypeOf(obj)
- .has(obj, prop) trap in operator
- .ownKeys(obj)
- .preventExtensions(obj)
- .set(obj, prop, value) trap set property
- .setPrototypeOf(obj, proto)

globals

METHODS

- eval(str) evaluate javascript code
- isFinite(obj) check if is a finite number
- isNaN(obj) check if is not a number
- parseInt(s, radix) string to integer
- parseFloat(s, radix) string to float
- encodeURIComponent(URI) = to %3D
- decodeURIComponent(URI) %3D to =

Set()

WeakSet only obj as items

PROPERTIES

- .size return number of items

METHODS

- .add(item) add item to set ws
- .has(item) check if item exists ws
- .delete(item) del item & return if del ws
- .clear() remove all items from set

ITERATION METHODS

- .entries() iterate items
- .values() iterate only value of items

CALLBACK FOR EACH METHODS

- .forEach(cb(e,i,a), arg) exec for each

Map()

WeakMap only obj as keys

PROPERTIES

- .size return number of elements

METHODS

- .set(key, value) add pair key=value wm
- .get(key) return value of key wm
- .has(key) check if key exist wm
- .delete(key) del elem. & return if ok wm
- .clear() remove all elements from map

ITERATION METHODS

- .entries() iterate elements
- .keys() iterate only keys
- .values() iterate only values

CALLBACK FOR EACH METHODS

- .forEach(cb(e,i,a), arg) exec for each

Symbol()

PROPERTIES

- .iterator specifies default iterator
- .match specifies match of regexp
- .species specifies constructor function

METHODS

- .for(key) search existing symbols
- .keyFor(sym) return key from global reg

Generator() = function* () { ... }

METHODS

- .next(value) return obj w/{value,done}
- .return(value) return value & true done
- .throw(except) throw an error

Others

BASIC TIPS

- var declare variable
- let declare block scope local variable
- const declare constant (read-only)
- func(a=1) default parameter value
- func(...a) rest argument (spread operator)
- (a) => { ... } function equivalent (fat arrow)
- `string ${a}` template with variables
- 0bn binary (2) number n to decimal
- 0on octal (8) number n to decimal
- 0xn hexadecimal (16) number n to decimal
- for (i in array) { ... } iterate array, i = index
- for (e of array) { ... } iterate array, e = value
- class B extends A () { } class sugar syntax

window = Browser global object

PROPERTIES

- b .closed check if window is closed
- n .devicePixelRatio ratio vertical size pix
- b .fullScreen check if window is fullscreen
- n .innerWidth width size (incl. scrollbar)
- n .innerHeight height size (incl. scrollbar)
- n .outerWidth width size (incl. browser)
- n .outerHeight height size (incl. browser)
- n .length number of frames
- s .name inner name of window
- s .status bottom statusbar text

APPLICATION PROPERTIES

- o .applicationCache offline resources API
- o .console console browser API
- o .crypto cryptographic API
- o .history session page history API
- o .location information about URL API
- o .localStorage storage for site domain
- o .sessionStorage storage until closed
- o .navigator information about browser
- o .performance data about performance

SCREEN PROPERTIES

- o .screen information about screen
- n .screenX horizontal pos browser/screen
- n .screenY vertical pos browser/screen
- n .pageXOffset horizontal pixels scrolled
- n .pageYOffset vertical pixels scrolled

WINDOW PROPERTIES

- o .opener window that opened this window
- o .parent parent of current window/frame
- o .self this window (equal to .window)
- o .top top window of current win/frame

METHODS

- s .btoa(str) encode string to base64
- s .atob(str) decode base64 string to text
- .focus() request send window to front
- .blur() remove focus from window
- o .getSelection(id) return Selection object
- .postMessage(msg, dst, transf) send
- o .open(uri, name, options) open popup
- .stop() stop window loading
- b .find(str, case, back, wrap, word, fr, d)
- .print() open print document window

ANIMATION METHODS

- n .requestAnimationFrame(cb(n))
- .cancelAnimationFrame(reqID)

TIMER METHODS

- n .setTimeout(f(a...), ms, a...) delay&run
- .clearTimeout(id) remove timeout
- n .setInterval(f(a...), ms, a...) run every
- .clearInterval(id) remove interval

SCROLL METHODS

- .scrollBy(x, y) scroll x,y pixels (relative)
- .scrollTo(x, y) scroll x,y pixels (absolute)
- .moveBy(x, y) move window by x,y (rel)
- .moveTo(x, y) move window to x,y (abs)
- .resizeBy(x, y) resize win by x,y (rel)
- .resizeTo(w, h) resize win to WxX (abs)

STYLE SHEET METHODS

- o .getComputedStyle(elem, pseudoelem)
- a .matchMedia(mediaq) match CSSMQ

screen = Info about screen / resolution

PROPERTIES

- n .availTop top-from space available
- n .availLeft left-from space available
- n .availWidth width space available
- n .availHeight height space available
- n .width screen width resolution
- n .height screen height resolution
- n .colorDepth screen color depth (bits)
- n .pixelDepth screen pixel depth (bits)

METHODS

- b .lockOrientation(mode|modearray)
- b .unlockOrientation() remove locks

console = unofficial console browser API

METHODS

- .assert(cond, str1|obj1...) set a assert
- .count(str) count (show number times)
- .dir(obj) show object (expanded debug)
- .group() open new message group
- .groupCollapsed() open new group coll.
- .groupEnd() close previous group
- .table(array|obj, colnames) show table
- .trace() show code trace
- .timeStamp(str) put time on timeline

PERFORMANCE METHODS

- .profile(name) start performance profile
- .profileEnd(name) stop perf. profile
- .time(name) start performance timer
- .timeEnd(name) stop perf. timer

LOG LEVEL METHODS

- .log(str1|obj1...) output message
- .info(str1|obj1...) output information
- .warn(str1|obj1...) output warning
- .error(str1|obj1...) output error

window = global interaction func.

METHODS

USER INTERACTION METHODS

- .alert(str) show message (ok button)
- s .prompt(str, def) ask answer to user
- b .confirm(str) show message (ok, cancel)

history = page history on tab

PROPERTIES

- n .length number of pages in historytab
- n .state return state top history stack

METHODS

- .back() go prev page (same as .go(-1))
- .forward() go next page (same as .go(1))
- .go(n) go n page (positive or negative)
- .pushState(obj, title, url) insert state
- .replaceState(obj, title, url) repl. state

storage localStorage / sessionStorage

PROPERTIES

- n .length number of items in storage

METHODS

- s .key(n) return key name on position n
- s .getItem(key) return value of item key
- .setItem(key, value) set or update key
- .removeItem(key) delete item with key
- .clear() delete all items for current site

performance = info about performance

PROPERTIES

- o .navigation info about redir/type nav.
- o .timing info about latency-load perf.

METHODS

- n .now() high precision timestamp

navigator = info about browser

PROPERTIES

- b .cookieEnabled browser cookies on?
- n .doNotTrack DNT privacy enabled?
- o .geolocation user-info geolocation
- s .language language in browser
- n .maxTouchPoints max on device
- b .onLine browser work in online mode?
- s .userAgent identify browser of user

METHODS

- n .vibrate(n|pattern) use device vibration

location = info about current URL

PROPERTIES

- s .href full document url
- s .protocol https://www.emezeta.com/
- s .username https://user:pass@www
- s .password https://user:pass@www
- s .host https://emezeta.com:81/
- s .hostname https://emezeta.com:81/
- s .port https://emezeta.com:81/
- s .pathname http://emezeta.com/42/
- s .hash http://emezeta.com/#contacto
- s .search http://google.com/?q=emezeta
- o .searchParams search params object
- s .origin source origin of document url

onClick="..." (HTML) .onclick =(JS func) 'click' (Listener)

e events (only popular events)

MOUSE EVENTS

- e .onClick
- e .onDblClick
- e .onMouseDown
- e .onMouseUp
- e .onMouseEnter
- e .onMouseLeave
- e .onMouseMove
- e .onMouseOver
- e .onMouseOut
- e .onWheel

KEYBOARD EVENTS

- e .onKeyDown
- e .onKeyUp
- e .onKeyPress

LOAD/OBJECT EVENTS

- e .onDOMContentLoaded
- e .onLoad
- e .onAbort
- e .onError
- e .onResize
- e .onScroll
- e .onBeforeUnload
- e .onUnload

FORM/FIELDS EVENTS

- e .onBlur
- e .onFocus
- e .onChange
- e .onInput
- e .onInvalid
- e .onSelect
- e .onReset
- e .onSubmit

ANIMATION/TRANSITION EVENTS

- e .onDragEnter
- e .onDragLeave
- e .onDragStart
- e .onDragEnd
- e .onDragOver
- e .onDrag
- e .onDrop

ANIMATION/TRANSITION EVENTS

- e .onAnimationStart
- e .onAnimationEnd
- e .onAnimationIteration
- e .transitionEnd

Emezeta.com

document = Document object

PROPERTIES

- .characterSet document charset
- .compatMode quirks or standard mode
- .cookie return all cookies doc string
- .designMode return design mode status
- .dir return direction text: "rtl" or "ltr"
- .doctype return document type (DTD)
- .domain return document domain
- .documentURI return document URL
- .lastModified return date/time modific.
- .origin return document's origin
- .readyState return current load status
- .referrer return previous page (referrer)
- .title return document title
- .URL return HTML document URL
- .location information about URL

ELEMENTS PROPERTIES

- .activeElement focused element
- .body return body element
- .currentScript return active script
- .defaultView return window element
- .documentElement first element (root)
- .head return head element
- .scrollingElement first scrollable elem.

DOCUMENT ARRAY PROPERTIES

- .anchors array of images elements
- .applets array of applets elements
- .embeds array of embeds elements
- .forms array of forms elements
- .images array of images elements
- .links array of links elements
- .plugins array of plugins elements
- .scripts array of scripts elements

STYLESHEET PROPERTIES

- .styleSheets array of style files elem
- .preferredStyleSheetSet preferred css
- .selectedStyleSheetSet selected css

METHODS

- .adoptNode(node) adopt from ext doc
- .createAttribute(name) create Attr obj
- .createDocumentFragment()
- .createElement(tag) create Element obj
- .createEvent(type) create Event object
- .createRange() create Range object
- .createTextNode(text) create TextNode
- .enableStyleSheetsForSet(name)
- .importNode(node, desc) import copy
- .getElementById(id) find elem with id
- .getElementsByName(name) w/ name
- .getSelection(id) return Selection object

ClientRect() = Coords of element

PROPERTIES

- .top top coord of surrounding rect
- .right right coord of surrounding rect
- .bottom bottom coord of surrounding r.
- .left left coord of surrounding rect
- .width width coord of surrounding rect
- .height height coord of surrounding r.

Emezeta.com

Element() = Element object

PROPERTIES

- .accessKey if exist, shortcut key
- .attributes array of Attr objects
- .classList DOMTokenList of classes
- .className classes list to string
- .id id string of element
- .name name string of element
- .tagName HTML tag of element

POSITION, SIZE AND SCROLL PROPERTIES

- .clientTop top border width element
- .clientLeft left border width element
- .clientWidth inner width element
- .clientHeight inner height element
- .scrollTop top-position in document
- .scrollLeft left-position in document
- .scrollWidth width of element
- .scrollHeight height of element

GET/SET HTML CODE PROPERTIES

- .innerHTML get/set HTML inside elem
- .outerHTML get/set HTML (incl. elem)

METHODS

- .closest(selec) closest ancestor
- .getElementsByClassName(class)
- .getElementsByTagName(tag)
- .querySelector(selec) return first elem
- .querySelectorAll(selec) return elems
- .matches(selec) match with this elem?
- .insertAdjacentHTML(posstr, html)

ATTRIBUTE METHODS

- .hasAttributes() exists attributes?
- .hasAttribute(name) exist attribute?
- .getAttribute(name) return value
- .removeAttribute(name) del attribute
- .setAttribute(name, value) set attrib.

ELEMENT/RECT (POSITION AND SIZE) METHODS

- .getBoundingClientRect() return pos.
- .getClientRects() return pos/size array

Event() = Event on action

PROPERTIES

- .bubbles true=bubble, false=captures
- .cancelable event is cancelable?
- .currentTarget current element
- .defaultPrevented preventDefault() call
- .detail additional event info
- .eventPhase current stage (0-3)
- .isTrusted user action or dispatched
- .target reference to dispatched object
- .timeStamp time when was created
- .type type of event

METHODS

- .preventDefault() cancel event
- .stopImmediatePropagation()
- .stopPropagation() prevent being called

EventTarget (use over elements)

METHODS

- .addEventListener(ev, cb(ev), capt)
- .removeEventListener(ev, cb(ev), capt)
- .dispatchEvent(ev)

Attr() = Attribute object

PROPERTIES

- .name name of element attribute
- .value value of element attribute

DOMTokenList() = List of classes

PROPERTIES

- .length number of items

METHODS

- .contains(item) check if item exists
- .add(item) add item to list
- .item(n) return item number n
- .remove(item) del item from list
- .toggle(item) del item if exist, add else

Node() = Minor element (elem. or text)

PROPERTIES

- .baseURI absolute base URL of node
- .namespaceURI namespace of node
- .nodeName name of node
- .nodeType 1=element, 2=text, 9=doc
- .nodeValue value of node
- .prefix namespace prefix of node
- .textContent text of node and children

NAVIGATION PROPERTIES

- .childNodes children nodes collection
- .firstChild first children (include text)
- .lastChild last children (include text)
- .nextSibling immediate next node
- .previousSibling immediate prev node
- .parentElement immediate parent elem
- .parentNode immediate parent node
- .ownerDocument return document

METHODS

- .appendChild(node) add node to end
- .cloneNode(child) duplicate node
- .compareDocumentPosition(node)
- .contains(node) node is descendant?
- .hasChildNodes() node has childs?
- .insertBefore(newnode, node)
- .isDefaultNamespace(nsURI)
- .isEqualNode(node) check if are equal
- .lookupNamespaceURI() ret namesp.
- .lookupPrefix() return prefix for a ns
- .normalize() normalize-form children
- .removeChild(node) del node & return
- .replaceChild(newnode, oldnode)

ChildNode()

METHODS

- .remove() remove specified node

ParentNode()

PROPERTIES

- .childElementCount number of children
- .children children elements
- .firstElementChild first children elem.
- .lastElementChild last children elem.

NonDocumentTypeChildNode()

PROPERTIES

- .nextElementSibling next element
- .previousElementSibling prev element

HTML Cheat Sheet

HTML Basic Tags

<!DOCTYPE> : Defines type of the document.
<html> ... </html> : Root of an HTML document.
<head> ... </head> : Container for all the metadata about an HTML document
<title> ... </title> : Defines title for the document.
<meta /> : Defines metadata like character set, viewport, keywords, page description, author etc...
<base /> : Specifies default URL for all links on a page.
<link /> : Defines link to external sources.
<style> ... </style> : Defines style for a document.
<script> ... </script> : Defines client-side scripts.
<noscript> ... </noscript> : Specifies an alternate content to be displayed if the browser doesn't support scripts.
<body> ... </body> : Represents the main body of a document.

HTML Tables

<table> ... </table> : Table
<tr> ... </tr> : Table Row
<th> ... </th> : Table Header Cell
<td> ... </td> : Table Data
<caption> ... </caption> : Table Caption
<thead> ... </thead> : Group of header contents
<tbody> ... </tbody> : Group of body contents
<tfoot> ... </tfoot> : Group of footer contents
<colgroup> ... </colgroup> : Group of columns
<col /> : Column in a <colgroup>

HTML Lists

 ... : Unordered List
 ... : List Item
 ... : Ordered List
<dl> ... </dl> : Description List
<dt> ... </dt> : Terms of the description list
<dd> ... </dd> : Describe each term of the description list

HTML Frames, Audio, Video & Others

<iframe> ... </iframe> : Inline Frame
<audio> ... </audio> : Audio
<video> ... </video> : Video
<source /> : Resources for <audio>, <video> and <picture> elements
<track /> : Text track for <audio> and <video> elements
<embed /> : Container for external applications
<object> ... </object> : Embedded Object
<param /> : Parameter for <object>

HTML Text Formatting

 ... : Paragraph

 : Single line break
<hr /> : Horizontal Rule
 ... : Bold Text
 ... : Italic Text
<small> ... </small> : Smaller Text
<mark> ... </mark> : Marked / Highlighted Text
<del> ... </del> : Deleted Text
<ins> ... </ins> : Inserted Text
 ... : Important Text
 ... : Emphasized Text
_{...} : Subscript
^{...} : Superscript
<blockquote> ... </blockquote> : Block Quotation
<q> ... </q> : Inline Quotation
<abbr> ... </abbr> : Abbreviations / Acronyms.
<address> ... </address> : Address
<cite> ... </cite> : Citation
<bdo> ... </bdo> : Changes text direction.
<code> ... </code> : Programming Code
<pre> ... </pre> : Preformatted Text
<samp> ... </samp> : Sample output of a program
<kbd> ... <kbd> : Keyboard Input
<var> ... </var> : Variable
<dfn> ... </dfn> : Definition
<meter> ... </meter> : Scalar Measurement
<progress> ... </progress> : Progress bar
<u> ... </u> : Underlined Text

HTML Forms

<form> ... </form> : HTML form
<input /> : Input Field
<label> ... </label> : Label for <input> element
<button> ... </button> : Clickable button
<textarea> ... </textarea> : Multiline Text Input
<select> ... </select> : Drop down list
<option> ... </option> : An option in drop down list
<optgroup> ... </optgroup> : Groups related options in dropdown list
<fieldset> ... </fieldset> : Groups related elements in a form
<legend> ... </legend> : Caption for <fieldset> element
<datalist> ... </datalist> : List of options
<output> ... </output> : Result

HTML Headings

<h1> ... </h1> : Heading One
<h2> ... </h2> : Heading Two
<h3> ... </h3> : Heading Three
<h4> ... </h4> : Heading Four
<h5> ... </h5> : Heading Five
<h6> ... </h6> : Heading Six

HTML Comments

<!-- Single Line comment -->

```
<!--
    Multiple
    Lines
    comment
-->
```

HTML Links

<a> ... : Hyperlink

HTML Images

<img /> : Image
<map> ... </map> : Image Map with clickable area
<area /> : Clickable Area in <map>
<picture> ... </picture> : Group of pictures
<figure> ... </figure> : Figures / Diagrams / Illustrations
<figcaption> ... </figcaption> : Caption for <figure>
<canvas> ... </canvas> : Graphics
<svg> ... </svg> : SVG graphics

HTML Layouts

 ... : Division / Section / Block Container for HTML elements
 ... : Inline container for HTML elements
<header> ... </header> : Header content of a document
<main> ... </main> : Main content of a document
<footer> ... </footer> : Footer content of a document
<nav> ... </nav> : Navigation Links
<section> ... </section> : Section in a document
<article> ... </article> : Article in a document
<aside> ... </aside> : Side content in a document
<details> ... </details> : Additional details that user can hide or view on demand.
<summary> ... </summary> : Heading for <details> element
<dialog> ... </dialog> : Dialog Box

BASIC HTML

<!DOCTYPE>	Defines the document
<html>	Defines HTML documents
<head>	Contains metadata / information for the document
<title>	Defines a title for the document
<body>	Defines HTML documents body
<h1>to<h6>	Defines HTML headings
<p>	Defines a paragraph
 	Insert a single line break
<hr>	Defines a thematic break in an HTML page
<!--->	Defines a comment

FORMATTING

<abbr>	Defines an abbreviation or an acronym
<code>	Defines a piece of computer code
	Defines emphasized text
<mark>	Defines marked / highlighted text
<pre>	Defines preformatted text
<small>	Defines smaller text
<progress>	Represents the progress of a task
<blockquote>	Defines a section that is quoted from another source
<b>	Defines bold text
<q>	Defines a short quotation

FORMS AND INPUTS

<form>	Defines an HTML form for user input
<input>	Defines an input control
<textarea>	Defines a multiline input control (text area)
<button>	Defines a clickable button
<select >	Defines a drop-down list
<option>	Defines an option in a drop-down list
<optgroup>	Defines a group of related options in a drop-down list
<label>	Defines a label for an <input> element
<fieldset>	Groups related elements in a form
<legend >	Defines a caption for a <fieldset> element

IMAGES

<img>	Defines an image
<map>	Defines a client-side image map
<area>	Defines an area inside an image map
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content
<svg>	Defines a container for SVG graphics
<picture>	Defines a container for multiple image resources
<canvas>	Used to draw graphics, on the fly, via scripting (usually JavaScript)

PSEUDO SELECTORS & ELEMENTS

Mouse Over Selector	a:hover { }
Active Link Selector	a:active { }
Focus Selector	input:focus { }
Visited Links Selector	a:visited { }
Link Selector	.class:link { }
First Line Selector	p:first-line { }
First Letter Selector	p:first-letter { }
Checked elements selector	input:checked { }
Disabled elements selector	input:disabled{ }
Enabled elements selector	input:enabled { }
Not a Specified Element Selector	:not(p) {}

BOX PROPERTIES

Box Sizing	box-sizing: border-box | content-box
Border Style	border-style: none | hidden| dotted | dashed | solid| double | groove | ridge inset | outset

SELECTORS

Universal Selector	*{ }
ID Selectors	#{ }
Class Selector	.class { }
Type Selector	h1, h2, h3{ }
Adjacent Sibling Selector	h1 + p { }
Child Selector	ul > li { }
General Sibling Selector	h1 ~ p { }
Descendant Selector	p a { }
Attribute Selector	div[attribute="SomeValue"] { }

TEXT STYLING

Text-decoration	text-decoration: none | underline |overline | line-through
Text Justify	text-justify: auto | inter-word | inter-character | none| initial | inherit
Text Overflow	text-overflow: clip| ellipsis | string | initial | inherit

POSITION

Position	position: static |relative | absolute | fixed | sticky
Position Properties	top | right | bottom | left
Float Element	float: left | right | none
Clear Floating Elements	clear: none | left | right | both
Z Index	z-index: 3 | auto | inherit

PSEUDO SELECTORS & ELEMENTS

First Child Selector	p:first-child { }
Last Child Selector	p:last-child { }
Only Child Selector	p:only-child { }
nth-child Selector	p:nth-child { }
Before Element	.class:before { }
After Element	.class:after { }
First Element of its Parent Selector	p:first-of-type { }
Selector Elements that have no Children Selector	p:empty { }

BACKGROUND

Background Image	background-image: url()
Background Color	background-color: #2AA9E0
Background Repeat	background-repeat: repeat-x| repeat-y | repeat,| space | round | no-repeat
Background Attachment	background-attachment: scroll |fixed | local | initial | inherit
Background Position	background-position: top | right bottom | left | center