

LENGUAJEJS.COM

BASIC & FUNDAMENTAL DATATYPES

Created by @Manz (https://manz.dev/) https://lenguajejs.com/

N Number

PROPERTIES

- n .POSITIVE_INFINITY +∞
- n .NEGATIVE_INFINITY -∞
- n .MAX_VALUE largest positive
- n .MIN_VALUE smallest positive
- n .EPSILON min representable gap
- / .NaN value not show as number

METHODS

- s .toExponential(dec) → 1.24e+3
- s .toFixed(dec) → 1234.57
- s .toPrecision(p) → 1235 (4)
- s .toString(radix) → 1235 (4)

↳ (74).toString(16) → '4a' (0x4a)

- b .isFinite(n) finite number?
- b .isInteger(n) integer?
- b .isNaN(n) not a number?
- b .isSafeInteger(n) safe?
- n .parseInt(s , radix) to integer
- n .parseFloat(s , radix) to float

↳ Number.parseInt("10110",2) →22

S Set / WeakSet

PROPERTIES

- n .size return number of items

METHODS

- s .add(item) insert item ws
- b .has(item) check presence ws
- b .delete(item) remove item ws
- / .clear() purge all items

ITERATION METHODS

- i .keys() / .values() list values
- i .entries() list pair (key, value)

SET METHODS

- b .isSubsetOf(s) included in s
- b .isSupersetOf(s) contains all
- b .isDisjointFrom() no overlap?
- s .union() merge uniques
- s .intersection() shared items
- s .difference() exclude overlaps
- s .symmetricDifference() unshared

CALLBACK FOR EACH METHODS

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

F Function

PROPERTIES

- o .length get number of arguments
- s .name return name of arguments
- o .prototype prototype object

METHODS

- o .call(newthis , a1 ...) set this
- o .apply(newthis , a1 ...) ↳ array
- o .bind(newthis , a1 ...) ↳ no run

S String

PROPERTIES

- n .length string size

POSITION METHODS

- s .at(index) char at pos ~~.charAt()~~ [i]
- n .codePointAt(index) code at position
- s .fromCodePoint(n1 , n2 ...) code to char
- n .indexOf(str , from) find substr index
- n .lastIndexOf(str , from) find from end

FRAGMENTS METHODS

- s .repeat(n) repeat string n times
- s .substring(ini , end) subtext ~~substr()~~
- s .slice(ini , end) str between ini/end
- a .split(sep | regex , limit) divide string

SEARCH METHODS

- b .startsWith(str , size) check beginning
- b .endsWith(str , size) check ending
- b .includes(str , from) include substring?
- n .search(regex) search & return index
- a .match(regex) matches against string
- a .matchAll(regex) return iterator w/all
- s .replace(str | regex , newstr | func)
- s .replaceAll(str | regex , newstr | func)

MODIFY METHODS

- s .concat(str1 , str2 ...) combine text +
- s .toLowerCase()/.toLocaleLowerCase()
- s .toUpperCase()/.toLocaleUpperCase()
- s .padEnd(len , pad) add end padding
- s .padStart(len , pad) add start padding
- s .trim() remove spaces from begin/end
- s .trimStart() ↳ from end ~~trimLeft()~~
- s .trimEnd() ↳ from begin ~~trimRight()~~
- s .raw`` template string with ${vars}

UNICODE METHODS

- s .normalize(form) to standard unicode
- b .isWellFormed() check malformed string
- s .toWellFormed() fix malformed string
- n .localeCompare(str , locale , options)

M Map / WeakMap

PROPERTIES

- n .size return number of items

METHODS

- m .set(key , value) add pair (k,v) wm
- o .get(key) get value of key wm
- b .has(key) check key existence wm
- b .delete(key) remove element wm
- / .clear() remove all items

ITERATION METHODS

- i .keys() list keys i .values() list values
- i .entries() list pair (key, value)

CALLBACK FOR EACH METHODS

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

A Array

PROPERTIES

- n .length number of elements

METHODS

- o .at(n) get element at position n [n]
- b .isArray(obj) check is array
- b .includes(obj , from) include element?
- b .indexOf(obj , from) find elem. index
- b .lastIndexOf(obj , from) find from end
- s .join(sep) join elements w/ separator
- a .slice(ini , end) return array portion
- a .concat(obj1 , obj2 ...) combine arrays
- a .flat(depth) return flat array at n depth

MODIFY SOURCE ARRAY METHODS

- a .from(arraylike , (e,i,a)) convert to array
- a .fromAsync(arraylike , (e,i,a)) ↳ (async)
- a .of(obj1 , obj2 ,...) create array of objs
- a .with(pos , obj) clone with changes
- a .copyWithin(pos , ini , end) clone
- a .fill(obj , ini , end) fill array with obj
- a .toReversed() reverse array ~~.reverse()~~
- a .toSorted((a,b)) sort array ~~.sort()~~
- a .toSpliced(start , size , o1 , o2 ...)

ITERATION METHODS

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

CALLBACK FOR EACH METHODS

- b .every((e,i,a)) test until false
- b .some((e,i,a)) test until true
- a .map((e,i,a)) make array
- a .filter((e,i,a)) make array w/true
- o .find((e,i,a)) search element
- o .findLast((e,i,a)) ↳ from last
- n .findIndex((e,i,a)) search index
- n .findLastIndex((e,i,a)) ↳ from last
- a .flatMap((e,i,a)) map + flat(1)
- / .forEach((e,i,a)) exec for each
- o .reduce((p,e,i,a)) accumulative
- o .reduceRight((p,e,i,a)) ↳ from end

ADD/REMOVE METHODS

- o .pop() remove & return last element
- n .push(o1 , o2 ...) adds to end
- o .shift() remove & return last element
- n .unshift(o1 , o2 ...) inserts at start

n	number	d	date
/	Not-a-number	s	string
r	regular expr	f	function
b	boolean	o	object
a	array	/	undefined
?	static method	?	instance method



LENGUAJEJS.COM

MATH, REGEXP & OBJECTS

Created by @Manz (https://manz.dev/) https://lenguajejs.com/

Math

PROPERTIES

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

METHODS

- n `.abs(x)` absolute value
- n `.cbrt(x)` cube root
- n `.clz32(x)` leading zero bits (32)
- n `.exp(x)` return e^x
- n `.expm1(x)` return (e^x)-1
- n `.hypot(x1,x2...)` hypotenuse length
- n `.imul(a,b)` signed multiply
- n `.max(x1,x2...)` return max number
- n `.min(x1,x2...)` return min number
- n `.pow(base,exp)` return base^exp **
- n `.random()` float random number [0,1)
- n `.sign(x)` sign of number
- n `.sqrt(x)` square root of number

LOGARITHMS METHODS

- n `.log(x)` natural logarithm (base e)
- n `.log1p(x)` natural logarithm (1+x)
- n `.log10(x)` base 10 logarithm
- n `.log2(x)` base 2 logarithm

ROUND METHODS

- n `.ceil(x)` superior round (smallest)
- n `.floor(x)` inferior round (largest)
- n `.fround(x)` nearest single precision
- n `.f16round(x)` half precision float
- n `.round(x)` round (nearest integer)
- n `.trunc(x)` remove fractional digits

TRIGONOMETRIC MET.

- n `.acos(x)` arccosine
- n `.asin(x)` arcsine
- n `.atan(x)` arctangent
- n `.cos(x)` cosine
- n `.sin(x)` sine
- n `.tan(x)` tangent
- n `.atan2(x,y)` arctangent of ratio (x/y)

HYPERBOLIC MET.

- n `.acosh(x)`
- n `.asinh(x)`
- n `.atanh(x)`
- n `.cosh(x)`
- n `.sinh(x)`
- n `.tanh(x)`

BigInt

METHODS

- b `.asIntN(bits,str)` truncate to bits
- b `.asUintN(bits,str)` ↳ unsigned bits

JSON

METHODS

- n `.parse(str,(k,v))` parse str to object
- n `.stringify(obj,repf|wl,sp)` to str

RegExp

PROPERTIES

- n `.lastIndex` next match index
- s `.flags` active regex flags
- b `.dotAll` dot matches newline s
- b `.global` full global matching g
- b `.ignoreCase` ignore letter case i
- b `.multiline` match multiple lines m
- b `.sticky` search from lastIndex y
- b `.unicode` enable full unicode u
- b `.unicodeSets` extended unicode v
- b `.hasIndices` include index ranges d
- s `.source` raw regex source

METHODS

- b `.test(str)` check if matching
- a `.exec(str)` capture ~~`.compile()`~~

CLASSES

- `.` any character `\t` tabulator
- `\d` digit `[0-9]` `\r` carriage return
- `\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]` ⌫BS
- `\uN` char with unicode `N` `\0` NUL char

CHARACTER SETS OR ALTERNATION

- `[abc]` match any charset
- `[^abc]` match any unclosed charset
- `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
- `\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}` `x` repeated `n` times
- `x{n,}` `x` repeated at least `n` times
- `x{n,m}` `x` repeated `n`-`m` times

ASSERTIONS

- `x(?=y)` `x` before `y`
- `x(?!y)` `x` not before `y`

Error

PROPERTIES

- s `.name` name of error
- s `.message` description of error
- s `.stack` stack of error

Object

CREATION METHODS

- o `.assign()` copy properties
- o `.create()` new prototype object

METHODS

- o `.groupBy(obj,key)` group by criterion
- b `.is(o1,o2)` check exact value

OWN PROPERTIES

- o `.defineProperty()` set single descriptor
- o `.defineProperties()` ↳ set multiple
- o `.getOwnPropertyDescriptor()` fetch
- o `.getOwnPropertyDescriptors(obj)`
- o `.getOwnPropertyNames()` properties
- o `.getOwnPropertySymbols()` symbols
- o `.propertyIsEnumerable()` check
- o `.hasOwn()` own prop check
- o `.hasOwnProperty()` own prop check

PROTOTYPE MANIPULATION

- o `.getPrototypeOf()` prototype of object
- o `.setPrototypeOf(obj,)` set prototype
- b `.isPrototypeOf()` check prototype

MUTABILITY CONTROL

- o `.preventExtensions()` block additions
- b `.isExtensible()` check extensible
- o `.seal()` lock properties
- b `.isSealed()` sealed check
- o `.freeze()` deep immutability
- b `.isFrozen()` frozen check

ITERATION METHODS

- a `.keys()` list keys
- a `.values()` list values
- a `.entries()` list array of array
- o `.fromEntries()` convert entries to object

PRIMITIVE METHODS

- s `.toString()` convert to string
- s `.toLocaleString()` ↳ localization string
- o `.valueOf()` primitive value

Iterator

- i `.from(obj)` Create from object

METHODS

- i `.drop(limit)` skip some items
- i `.take(limit)` grab limited items
- o `.next()` get next value `{ value, done }`
- a `.toArray()` convert to array

ARRAY METHODS

- b `.every((e,i,a))` test until false
- a `.filter((e,i,a))` make array w/true
- o `.find((e,i,a))` search element
- a `.flatMap((e,i,a))` map + flat(1)
- / `.forEach((e,i,a))` exec for each
- a `.map((e,i,a))` make array
- o `.reduce((p,e,i,a))` accumulative
- b `.some((e,i,a))` test until true



LENGUAJEJS.COM

TEMPORAL, DATE & TIMES

Created by @Manz (https://manz.dev/) https://lenguajejs.com/

D Date (legacy)

METHODS

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

UNIT GETTERS / SETTERS .GETUTC* / .SETUTC*

- n .get/.setFullYear(y , m , d) (yyyy)
- n .get/.setMonth(m , d) (0-11)
- n .get/.setDate(d) (1-31)
- n .get/.setHours(h , m , s , ms) (0-23)
- n .get/.setMinutes(m , s , ms) (0-59)
- n .get/.setSeconds(s , ms) (0-59)
- n .get/.setMilliseconds(ms) (0-999)
- n .getDay() return day of week (0-6)
- n .getTimezoneOffset() mins (-840-840)

LOCALE & TIMEZONE METHODS

- s .toLocaleDateString(locale , options)
- s .toLocaleTimeString(locale , options)
- s .toLocaleString(locale , options)
- s .toUTCString() return UTC date
- s .toDateString() return American date
- s .toTimeString() return American time
- s .toISOString() return ISO8601 date
- s .toJSON() return date ready for JSON

Units

UNITS

n .year	2025	n .hour	13
n .month	6	n .hoursInDay	24
n .day	7	n .minute	30
n .dayOfWeek	6	n .second	0
n .dayOfYear	158	n .millisecond	0
n .daysInMonth	30	n .microsecond	0
n .daysInYear	365	n .nanosecond	0

OTHER UNITS

- s .calendarId "iso8601"
- s .monthCode "M06"
- n .yearOfWeek 2025
- n .monthsInYear 12
- n .weekOfYear 23
- n .daysInWeek 7
- s .era "bce"
- n .eraYear 2022
- b .inLeapYear false

TIMESTAMP PROPERTIES

- n .epochMilliseconds 1766261720000
- b .epochNanoseconds 17662617200...0n

TIMEZONE PROPERTIES

- s .timeZoneId "Europe/London"
- s .offset "+01:00"
- n .offsetNanoseconds 0

T Temporal

TYPES

- o .PlainTime hh:mm:ss
- o .PlainYearMonth yyyy-mm
- o .PlainMonthDay mm-dd
- o .PlainDate yyyy-mm-dd
- o .PlainDateTime yyyy-mm-ddThh:nn:ss
- o .ZonedDateTime yyyy-mm-ddThh:nn:ss[z]

METHODS

- o .from(date | opt) create date object
- n .compare(date1 , date2) (-1,0,1)

CALC METHODS

- o .add(date | opt) add duration to date
- o .subtract(date | opt) sub dur. to date
- o .since(date | opt) difference from date
- o .until(date | opt) difference to date
- b .equals(date | opt) check equality
- o .round(date | opt) truncate date

MODIFY METHODS

- o .with(d | opt) set fields
- o .withCalendar(d | opt) set calendar
- o .withPlainTime(d | opt) set time
- o .withTimeZone(d | opt) set timezone

CONVERT METHODS

- o .toInstant() → get Instant date
- o .toJSON() → serialize (ISO)
- o .toLocaleString() → get local format
- o .toPlainDate() → PlainDate
- o .toPlainDateTime() → PlainDateTime
- o .toPlainTime() → PlainTime
- o .toPlainMonthDay() → PlainMonthDay
- o .toPlainYearMonth() → PlainYearMonth
- o .toZonedDateTime() → ZonedDateTime

TIMEZONE METHODS

- o .getTimeZoneTransition(dir | opt)
- o .startOfDay(date | opt) beginning day

T Temporal Now

TYPES

- o .Now get current date/time

METHODS

- o .instant() → get Instant date
- o .plainDateISO() → PlainDate
- o .plainDateTimeISO() → PlainDateTime
- o .plainTimeISO() → PlainTime
- o .zonedDateTimeISO() → ZonedDateTime
- s .timeZoneId() get current timezone

Temporal examples

TEMPORAL EXAMPLES

```
Temporal.Now.plainDateISO()      2026-12-20
Temporal.PlainDate.from("2026-12-20")
date.add("P2D")                  2026-12-22
```

TD Temporal Duration

TYPES

- o .Duration P_Y_M_W_DT_H_M_S

METHODS

- o .from(date | opt) create date obj
- n .compare(date1 , date2) (-1,0,1)

DURATION PROPERTIES

- b .blank duration is zero ?
- n .years get year (defined)
- n .months get months (defined)
- n .weeks get weeks (defined)
- n .days get days (defined)
- n .hours get hours (defined)
- n .minutes get minutes (defined)
- n .seconds get seconds (defined)
- n .milliseconds get millisecs
- n .microseconds get microsecs
- n .nanoseconds get nanosecs
- n .sign sign of duration (-1,0,1)

DURATION METHODS

- o .abs() get positive duration
- o .negated() invert duration (negate)
- o .round(unit | opt) truncate to unit
- o .add(duration) duration + duration
- o .subtract(duration) dur1 - dur2
- o .toJSON() get JSON-format duration
- o .toLocaleString() human duration
- o .total(unit | opt) calc in unit
- o .with(units) modify units

TI Temporal Instant

TYPES

- o .Instant yyyy-mm-ddThh:nn:ssZ

PROPERTIES

- n .epochMilliseconds get timestamp
- b .epochNanoseconds get timestamp

METHODS

- o .add(date | opt) add duration
- o .subtract(date | opt) sub duration
- o .since(date | opt) diff from date
- o .until(date | opt) diff to date
- b .equals(date | opt) check equality
- o .round(date | opt) trunc date

CONVERT METHODS

- o .toJSON() → serialize (ISO)
- o .toLocaleString() get local format
- o .toZonedDateTime() ZonedDateTime

CONVERT DATE TO TEMPORAL

```
const d = new Date();          legacy date
Temporal.Instant               date → Temporal
.fromEpochMilliseconds(d)
```



LENGUAJEJS.COM

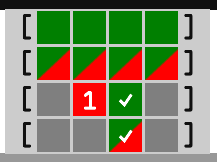
PROMISES, FETCH & ESM MODULES

Created by @Manz (https://manz.dev/) https://lenguajejs.com/

P Promise

STATIC

- p `.all(list)` wait all resolve
- p `.allSettled(list)` wait finish all (res/rej)
- p `.any(list)` wait first resolve
- p `.race(list)` wait first finish (resolve or reject)



OTHERS METHODS

DIRECT

- p `.withResolvers()` same → `new Promise((res,rej)=>...)`
- p `.resolve(value)` create direct resolved promise ✓
- p `.reject(value)` create direct rejected promise ✖

INSTANCE

- p `.then(resolve,reject)` handle result (ok) or rejected (error)
- p `.then(resolve)` runs when promise is fullfilled (success)
- p `.catch(reject)` runs when promise is rejected (error)
- p `.finally(end)` always run when finish (resolved or rejected)

ASYNC/AWAIT

`async function name() { ... }` prepare function return promise

`const data = await name()` wait promise, block execution

M Modules (Import/Export)

MODULES SPECIFIER

`import ... from "./file.js";` load from relative file URL

`import ... from "name";` bare import (node_modules/, maps...)

`import ... from "@/name/file.js";` alias import (vite, typescript...)

`import ... from "node:package";` import built-in node package

`import ... from "https://web.com/file.js";` load from absolute URL

STATIC IMPORTS — PRE-RUNNING TIME

MAIN IMPORTS

`import { name } from "./file.js";` import data from module

`import { name as rename } from "./file.js";` ↳ and rename

`import { m1 , m2 , ... } from "./file.js";` ↳ multiple data

`import * as name from "./file.js";` ↳ import all in object

`import "./file.js";` no import data, only run external code

DEFAULT IMPORT

`import name from "./file.js";` import default data with name

DYNAMIC IMPORT — RUNNING TIME

p `import("./file.js")` dynamic import file (running time)

STATIC EXPORTS

`export { m1 , m2 , ... }` export multiple data (node-style)

`export ...` export element (prefix, direct-style)

`export default ...` export data as default import of module

IMPORT META

s `import.meta.url` get absolute URL of module

s `import.meta.resolve(name)` resolve bare import to full URL

R Resource Modules

STATIC IMPORT

`import name from "./file.json" with { type: "json" }` parse json

`import styles from "./file.css" with { type: "css" }` parse css

`import image from "./file.png" with { type: "bytes" }` parse raw

DYNAMIC IMPORTS

p `import("./file.json", { with: { type: "json" }})` ↳ parse json

p `import("./file.css", { with: { type: "css" }})` ↳ parse css

p `import("./file.png", { with: { type: "bytes" }})` ↳ parse raw

F Fetch

METHODS

p `fetch(url | request , opt)` HTTP request ~~`XMLHttpRequest`~~

p `fetchLater(url | request , opt)` deferred HTTP request

R Request / Response

PROPERTIES

RESPONSE

- b `.ok` response was successful (`200-299`)
- b `.redirected` url is redirected
- n `.status` HTTP code error
- s `.statusText` status related for code error
- s `.type` `basic`, `cors`, `default`, `error`...
- o `.body` content (`string`, `FormData`, `Blob`...)
- b `.bodyUsed` `.body` has been used
- o `.headers` http header `Headers` or `Object`
- s `.url` absolute full http url

REQUEST

ONLY REQUEST PROPERTIES

- s `.cache` → `default`, `reload`, `no-cache`, `no-store`...
- s `.credentials` → `omit`, `same-origin`, `include`
- s `.destination` content type → `json`, `script`, `iframe`...
- s `.duplex` enable request body streaming: `half`
- s `.integrity` hash integrity value (ej: `sha256-...`)
- b `.isHistoryNavigation` request is a history navigation
- b `.isReloadNavigation` request is a reload/refresh
- b `.keepalive` browser will keep request alive
- s `.method` request HTTP method: `GET`, `POST`, `HEAD`...
- s `.mode` → `cors`, `no-cors`, `same-origin`, `navigate`...
- s `.redirect` mode → `follow`, `error`, `manual`.
- s `.referrer` previous url of user navigation
- s `.referrerPolicy` → `no-referrer`, `strict-origin`...
- o `.signal` set `AbortSignal` to abort request

METHODS

RESPONSE

REQUEST

- p `.arrayBuffer()` → binary data as buffer (raw data)
- p `.blob()` → raw file-like object (store images, files...)
- p `.bytes()` → return raw bytes (low-level byte handle)
- p `.clone()` → duplicate response object (reuse)
- p `.formData()` → parse form data (handle html form)
- p `.json()` → parse JSON content (json to object)
- p `.text()` → return as text (plain text string)

STATIC METHODS

- o `.error()` create error response (failed network requests)
- o `.json(data,opt)` get json response (send json data)
- o `.redirect(url,code)` redirect to URL (http redirection)

A AbortController

PROPERTIES

o `.signal` AbortSignal object

METHODS

/ `.abort(reason)` abort async operation

ABORTCONTROLLER EXAMPLE

`const controller = new AbortController();` create abort

`fetch("...", { signal: controller.signal })` assign signal

`controller.abort();` send signal to abort async operation

LENGUAJEJS.COM

DOCUMENT, DOM & EVENTS

Created by @Manz (https://manz.dev/) https://lenguajejs.com/

d document (Properties)

PARSE STATIC METHODS

- o .parseHTML(html , opt) strict parse
- o .parseHTMLUnsafe(html , opt) soft parse

PROPERTIES

- s .title document title
- o .doctype document type header
- s .characterSet doc charset ~~.charset~~
- s .compatMode quirks or standard mode
- s .cookie all cookies document
- s .currentScript current run script / null
- s .documentURI full URL ~~.URL~~
- s .referrer previous user page
- s .dir text direction
- s .lastModified date/time modification

LOADING STATE

- b .prerendering prerender loading state
- s .readyState ready state

API STATES

- s .visibilityState page visibility ~~.hidden~~
- b .fullscreenEnabled fullscreen API
- b .pictureInPictureEnabled PiP API

STYLES AND FONTS

- o .fonts Font Loading API
- o .styleSheets style sheets
- a .adoptedStyleSheets adopted styles

ELEMENTS

- o .defaultView default view: window
- e .documentElement refer <html> element
- e .body refer <body> element
- e .head refer <head> element
- e .scrollingElement current scrolling tag
- e .activeElement current focus html tag

FF FontFace

PROPERTIES

- n .size number of @fontface rules
- s .status loading or loaded
- p .ready all fonts loaded

METHODS

- o .add(font) add a new font
- b .check(cssrule , limit) check render font
- / .clear() remove all fonts added with JS
- b .delete(font)
- b .has(cssrule) exist font?
- p .load(cssrule , limit) force load font
- / .forEach((e,i,a)) loop fonts

ITERATORS

- i .values() iterate fonts
- i .entries() iterate entries

s SanitizerConfig

- a .elements allowed elements

PARAMETERS

- a .elements allowed elements
- a .attributes allowed attributes
- a .removeElements removed elements
- a .removeAttributes removed attributes
- a .replaceWithChildrenElements unwrap

ENABLE FEATURES

- b .comments allow comments
- b .dataAttributes allow data-*

d document (Methods)

METHODS

SEARCH FROM POINT

- o .caretPositionFromPoint(x , y) node
- e .elementFromPoint(x , y) first element
- a .elementsFromPoint(x , y) all elements
- s .evaluate(xpath , doc) XPath search

API METHODS

- p .exitFullscreen() close fullscreen API
- p .exitPictureInPicture() close PiP API
- / .exitPointerLock() unlock pointer
- b .hasFocus() page is focused?

ANIMATIONS

- a .getAnimations() list css animations
- p .startViewTransition() animate pages

SEARCH DOM

- e .getElementById(id) search first element
- a .getElementsByClassName(class) class
- a .getElementsByName(name) name
- a .getElementsByTagName(tag) <tag> NS
- e .querySelector(css) first element
- a .querySelectorAll(css) all elements
- o .getSelection() get text selection data

STORAGE ACCESS METHODS

- p .hasStorageAccess()
- p .hasUnpartitionedCookieAccess()
- p .requestStorageAccess()

CREATION METHODS

- e .createAttribute(name) → Attr NS
- e .createDocumentFragment() → Fragment
- e .createElement(tag , opt) → Tag NS
- o .createRange() → Range
- e .createTextNode(text) → Text
- e .createComment(text) → Comment
- s .adoptNode(node) move node to outside
- s .importNode(node , deep) clone node
- s .append(node ,...) ~~.writeln()~~
- s .prepend(node ,...) ~~.write()~~

A Attr

PROPERTIES

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

DTL DOM Token List

PROPERTIES

- n .length number of items
- s .value all items concatenated

METHODS

- b .contains(name) item exist?
- / .add(name ,...) add item to list
- s .item(n) get item number n
- / .remove(name) delete item
- b .toggle(name) conmute item
- b .replace(old , now) replace item
- b .supports(name) item support?
- / .forEach((e,i,a)) create iterator

ITERATORS

- i .keys() get names of tokens
- i .values() get values of tokens
- i .entries() get pair name-values

S Selection

PROPERTIES

- s .type selection type
- s .direction selection dir
- n .rangeCount range count
- b .isCollapsed

NODES

- e .baseNode base node
- e .anchorNode start node
- e .extentNode end node
- e .focusNode focus node

OFFSETS

- n .baseOffset base offset
- n .anchorOffset start offset
- n .extentOffset end offset
- n .focusOffset focus offset

R Range

PROPERTIES

- b .collapsed collapsed range

NODES

- e .commonAncestorContainer
- e .startContainer start node
- e .endContainer end node

OFFSETS

- n .startOffset start offset
- n .endOffset end offset

```
Document.parseHTMLUnsafe(`<div class="container"></div>`)          unsafe and permissive document or fragment code
Document.parseHTML(`<div class="container"></div>`)                strict and safe sanitize document or fragment code
Document.parseHTMLUnsafe(`<div class="container"></div>`, { removeElements: ["script"] })          ↳ custom options
```

LENGUAJEJS.COM

ELEMENTS, DOM & NODES

Created by @Manz (https://manz.dev/)

https://lenguajejs.com/

E Element (Properties)

PROPERTIES

- s `.accessKey` set accessKey attribute
- m `.attributes` get attribute/value list
- s `.className` value of class attribute
- o `.classList` classes control helper
- s `.id` get `id` attribute value
- s `.name` get `name` attribute value
- s `.tagName` get uppercase tag name
- s `.localName` get lowercase tag name
- s `.namespaceURI` namespace tag
- s `.prefix` namespace prefix (xml)
- o `.part` css part control helper

DOM HIERARCHY

- n `.childElementCount` child number
- a `.children` child element list
- e `.firstElementChild` first child
- e `.lastElementChild` last child
- e `.previousElementSibling` prev tag
- e `.nextElementSibling` next tag

SHADOW DOM

- o `.shadowRoot` get shadow DOM
- s `.slot` get name slot
- e `.assignedSlot` get `<slot>`

ELEMENT SIZE/POS

- s `.clientWidth`
- s `.clientHeight`
- s `.clientTop`
- s `.clientLeft`
- s `.currentCSSZoom` css zoom value

ELEMENT SCROLL

- s `.scrollWidth`
- s `.scrollHeight`
- s `.scrollTop`
- s `.scrollLeft`

HTML PROPERTIES

- s `.innerHTML` get/set html code
- s `.outerHTML` ↳ with wrapper tag

MODERN HTML METHODS ~~.INNERHTML~~

- s `.getHTML( opt )` get html code
- / `.setHTML( html , opt )` safe html
- / `.setHTMLUnsafe( html , opt )`

SO Shadow Options

PROPERTIES

- s `.mode` `"open"` or `"closed"`
- b `.clonable` with `.cloneNode()`
- b `.delegatesFocus` focus inside
- b `.serializable` included in export
- s `.slotAssignment` how slot match

SD Shadow DOM

SHADOW DOM CREATION

```
const options = { mode: "open" }
element.attachShadow( options );
```

ELEMENTS

`element`	main DOM (hidden)
`element.shadowRoot`	shadow DOM

E Element (Methods)

METHODS

- o `.animate( keyframes , time | opt )`
- a `.getAnimations()` all anims of element
- b `.checkVisibility()` element is visible?
- m `.computedStyleMap()` get computed styles
- o `.attachShadow( options )` Create Shadow DOM
- b `.matches( css )` check if match with css selector

DOM RECT METHODS

- o `.getBoundingClientRect()` Element DOMRect
- a `.getClientRects()` DOMRect list

ATTRIBUTE METHODS

- b `.hasAttribute( name )` check if exist attrib NS
- b `.hasAttributes()` check if has at least 1 attribute
- a `.getAttributeNames()` get list of attributes
- s `.getAttribute( name )` get attribute value NS
- / `.setAttribute( name , value )` set attrib value NS
- o `.getAttributeNode( name )` get `Attr` node NS
- o `.setAttributeNode( attr )` set `Attr` node NS
- b `.toggleAttribute( name , force )` add/del attrib
- / `.removeAttribute( name )` delete attribute NS

SCROLL METHODS

- / `.scrollBy( x , y )` / `.scrollBy( opt )` relative
- / `.scrollTo( x , y )` / `.scrollTo( opt )` absolute
- / `.scrollIntoView( top | opt )` visual scroll
- / `.scrollIntoViewIfNeeded( center )` ↳ conditional

SEARCH CHILD ELEMENTS

- a `.getElementsByClassName( class )` find by class
- a `.getElementsByTagName( tag )` find by tag NS
- e `.querySelector( css )` first tag match by css
- a `.querySelectorAll( css )` all tags match by css

SEARCH PARENT ELEMENTS

- e `.closest( css )` first ancestor match by css

ELEMENT API

- / `.after( node ,...)` insert node **after** element
- / `.append( node ,...)` insert node as child **at end**
- / `.before( node ,...)` insert node **before** element
- / `.prepend( node ,...)` insert node as child **at start**
- / `.remove()` remove element from document

REPLACE/CHILD ELEMENT API

- / `.replaceChildren( node ,...)` replace child
- / `.replaceWith( node ,...)` replace node element
- / `.moveBefore( child , target )` → before target

ADJACENT API

- / `.insertAdjacentHTML( pos , html )` insert html
- / `.insertAdjacentElement( pos , el )` insert elem
- / `.insertAdjacentText( pos , text )` insert textnode

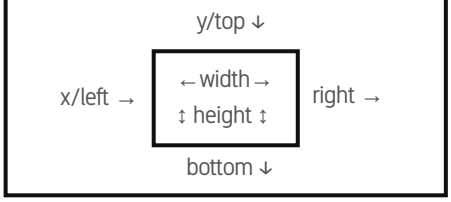
API METHODS

- / `.setPointerCapture( id )` capture all event
- b `.hasPointerCapture( id )` element captured?
- / `.releasePointerCapture( id )` release capture
- p `.requestPointerLock( opt )` infinite mouse moves
- p `.requestFullscreen( opt )` enable fullscreen

DR DOMRect

PROPERTIES

- n `.x` ~~`.left`~~ horizontal start position
- n `.y` ~~`.top`~~ vertical start position
- n `.width` rectangle width
- n `.height` rectangle height
- n `.bottom` vertical end position
- n `.right` horizontal end position



N Nodes

PROPERTIES

- s `.baseURI` document base URL
- s `.nodeName` tag uppercase name
- n `.nodeType` node kind
- a `.childNodes` all children
- o `.firstChild` first child node
- o `.lastChild` last child node
- o `.previousSibling` prev node
- o `.nextSibling` next node
- o `.ownerDocument` document root
- e `.parentElement` parent tag
- e `.parentNode` parent node
- b `.isConnected` inserted in DOM?
- s `.textContent` node text value

METHODS

- e `.cloneNode( deep )` clone tree
- e `.getRootNode( opt )` root parent
- b `.contains( node )` check node
- b `.hasChildNodes()` has children
- b `.isEqualNode( node )` content
- b `.isSameNode( node )` node
- s `.lookupNamespaceURI( prefix )`
- s `.lookupPrefix( namespace )`
- / `.normalize()` merge text nodes

NODE API

- e `.appendChild( node )` add node
- e `.removeChild( node )` del node
- e `.replaceChild( new , old )`
- e `.insertBefore( new , ref )`

P Position

PROPERTIES

- s `"beforebegin"` [■ |]
- s `"afterbegin"` [■ |]
- s `"beforeend"` [| ■]
- s `"afterend"` [|] ■



LENGUAJEJS.COM

LANGUAGE, TIMERS, URL & GLOBALS

Created by @Manz (https://manz.dev/) https://lenguajejs.com/

Main Javascript details

DATA STRUCTURES

`const` constant variable (no reassign)
`let` variable (block scope) ~~`var`~~
`using` auto-dispose variable
`class` Name `extends` Base `...` sugar syntax
`super` base class `this` ref this class `new` instance
o `structuredClone(` struct `,` opt `)` deep clone

CONDITIONALS

`if ... / else if ... / else ...`
`switch ... / case ... / default ...`
`condition ? A : B` ternary operator

FUNCTIONS

`function name(a = 1) ...` default parameter value
`function name(...a) ...` rest argument (spread)
`const name = (a) => ...` fat arrow (no this)

LOOPS

`for (` start `,` condition `,` incr `)` traditional loop
`for (` item `in` array `)` loop keys
`for (` item `of` array `)` loop values
`while (` condition `) ...` repeat loop
`do ... while (` condition `)` ↳ always iteration #0

OPERATORS

`!` Logical NOT	`!false` → `true`
`&&` Logical AND	`true && "manz"` → `"manz"`
`||` Logical OR	`0 || "manz"` → `"manz"`
`??` nullish coalescing	`0 ?? "manz"` → `0`
`||=` logical OR assign	`A ||= B` → `A = A || B`
`??=` nullish assign	`A ??= B` → `A = A ?? B`
`?.` Optional	`A?.life` → `A && A.life`

NUMBER SYSTEM

`42` → 42 (decimal) `0b` `0b10` → 2 (binary)
`0o` `0o20` → 16 (octal) `0x` `0x32` → 50 (hexadecimal)

TIPS

`` `string ${a}` `` string template (variables + multiline)

Timers

METHODS

BASIC

n `setTimeout(` function `,` time `)` run after time ms
/ `clearTimeout(` id `)` cancel timer with id
n `setInterval(` function `,` time `)` run every time ms
/ `clearInterval(` id `)` cancel timer with id

HIGH PERFORMANCE

n `requestAnimationFrame(` function `)` run auto
/ `cancelAnimationFrame(` id `)` cancel timer w/ id
n `requestIdleCallback(` function `,` opt `)` when idle
/ `cancelIdleCallback(` id `)` cancel timer with id

Symbol

METHODS

p `.for(` key `)` create symbol from key (global)
s `.keyFor(` sym `)` search key from symbol

URL

METHODS

https:// user : 1234 @ sub. manz. dev :80 /path / index.html #anchor :~:text=word

URL

s `.protocol` https:	s `.hostname` sub.manz.dev
s `.username` user	s `.host` sub.manz.dev:80
s `.password` 1234	s `.origin` https://sub.manz.dev
s `.port` 80	s `.pathname` /path/index.html
s `.hash` #anchor	s `.search` full query string parameters
s `.href` absolute full URL	o `.searchParams` query string object helper

URL SEARCH PARAMS PROPERTIES & METHODS

n `.size` total parameters
b `.has(` name `)` exist param?
s `.get(` name `)` get param
/ `.sort()` sort params
/ `.append(` n `,` v `)` add
/ `.delete(` name `)` delete
a `.getAll(` name `)` multiple
/ `.set(` n `,` v `)` set param
/ `.forEach(` (v,n,o) `)` iterate query string parameters

ITERATORS

i `.keys()`
i `.values()`
i `.entries()`

Globals

REFERENCE

`globalThis` ref this window = ~~`window`~~

PROPERTIES

SIZES

n `.innerWidth` n `.innerHeight`
n `.outerWidth` n `.outerHeight`

SCREEN & SCROLL

o `.visualViewport` ~~`.screen`~~ data
n `.screenX` ~~`.screenLeft`~~ X offset
n `.screenY` ~~`.screenTop`~~ Y offset
n `.scrollX` quantity horizontal scroll
n `.scrollY` quantity vertical scroll

OTHERS

b `.closed` is this window closed?
o `.crypto` cryptographic services
o `.location` current URL information
o `.navigator` data & API browser
o `.caches` cache storage control API

METHODS

CLASSIC

/ `.alert(` msg `)` show message
b `.confirm(` msg `)` show question
s `.prompt(` msg `,` def `)` asks text
/ `.print()` show printer dialog
o `.open(` url `,` target `,` opt `)`
/ `.close()` close window
/ `.focus()` focus window
s `.atob(` base64 `)` → to binary
s `.btoa(` binary `)` → to base64

OTHERS

o `.createImageBitmap(` img `,` opt `)`
p `.getScreenDetails()` screen data
o `.getSelection()` get selected text
o `.matchMedia(` MQ `)` check MQ
o `.getComputedStyle(` el `,` pseudo `)`

Console

METHODS

LOG LEVEL METHODS

/ `.log(` str | obj `)` show message
/ `.info(` str | obj `)` ↳ info message
/ `.warn(` str | obj `)` ↳ warning msg
/ `.error(` str | obj `)` ↳ error msg

PERFORMANCE METHODS

/ `.profile(` name `)` start profile
/ `.profileEnd(` name `)` stop profile
/ `.time(` name `)` start benchmark
/ `.timeEnd(` name `)` stop benchmark

OTHERS

/ `.assert(` cond `,` str | obj `...)` test
/ `.count(` str `)` count number times
/ `.dir(` obj `)` extended prop debug
/ `.group()` open new group
/ `.groupCollapsed()` ↳ collapsed
/ `.groupEnd()` close previous group
/ `.table(` array | obj `,` cols `)` format
/ `.trace()` show code trace
/ `.timeStamp(` str `)` put on timeline

Crypto

METHODS

o `.getRandomValues(` typedarray `)`
s `.randomUUID()` generate unique id
o `.subtle` ↴

SUBTLE

METHODS

p `.encrypt()`	p `.decrypt()`
p `.deriveBits()`	p `.deriveKey()`
p `.generateKey()`	p `.digest()`
p `.importKey()`	p `.exportKey()`
p `.wrapKey()`	p `.unwrapKey()`
p `.sign()`	p `.verify()`



LENGUAJEJS.COM

EVENTS, LOCALES & STORAGE

Created by @Manz (https://manz.dev/) https://lenguajejs.com/

E EventTarget

METHODS

- / .addEventListener(name , (ev) , opt)
- / .removeEventListener(name , (ev) , opt)
- b .dispatchEvent(event)

EVENTLISTENER OPTIONS

b	.capture dispatched DOM (parent → child)	false
b	.once only one-time event	false
b	.passive never call .preventDefault()	false
o	.signal set AbortSignal to remove listener	

E Event / CustomEvent

PROPERTIES

s	.type name of event	
b	.bubbles later, event move (child → parents)	false
b	.cancelable event is cancelable?	false
b	.composed can pass through shadow dom	false
b	.defaultPrevented action already prevented	false
o	.detail additional event info, only CustomEvent	
n	.eventPhase current event stage	(0-3)
b	.isTrusted user action or dispatched	false
o	.currentTarget active listener element	null
o	.target event origin element	
n	.timeStamp time when was created	

METHODS

- a .composedPath() element list (propagation path)
- / .preventDefault() cancel default event action
- / .stopImmediatePropagation() stop all listeners
- / .stopPropagation() stop event bubbling
- o .when(name , opt) create ev. stream Observable

O Observable API

METHODS

ARRAY-LIKE METHODS

- b .every((expr))
- b .some((expr))
- a .filter((expr))
- o .find((expr))
- / .forEach((expr))
- a .map((expr))
- a .flatMap((expr))
- o .reduce((acc, start))

OBSERVABLE METHODS

- o .subscribe(observer , opt) consume emitted ev.
- o .take(n) process only first n
- o .takeUntil(expr) process to n
- o .drop(n) ignore first n items
- o .switchMap(func) switch to new stream
- o .inspect((expr)) peek values/debug
- o .first()
- o .last()
- a .toArray()

ERRORS METHODS

- o .catch(f) handle errors
- o .finally(f) cleanup

B BroadcastChannel API

PROPERTIES

- s .name name of the channel

METHODS

- / .postMessage(msg) send data
- / .close(msg)

EventList

EVENT NAMES

MOUSE / POINTER	MOUSE	click
		dblclick
		mousedown
		mouseup
		mousemove
		mouseenter
		mouseleave
		mouseleave
		mouseover
		mouseout
		contextmenu
	POINTER	pointerdown
		pointerup
		pointermove
		pointerenter
		pointerleave
		pointerover
		pointerout
		pointercancel
	SCR	wheel
		scroll
KEYBOARD / TOUCH	KEYB	keydown
		keyup
		~~keypress~~
	CLIP	copy
		cut
		paste
	TOUCH	touchstart
		touchmove
		touchend
		touchcancel
DRAG & DROP		drag dragstart
		dragend dragenter
		dragleave
		dragover drop
FORMS		focus blur
		focusin reset
		focusout input
		change submit
ANIMATIONS		animationstart
		animationend
		animationiteration
		transitionstart
		transitionend
		transitionrun
		transitioncancel
OTHERS		load unload
		beforeunload
		error resize
		DOMContentLoaded

I Intl

INTL API OBJECTS

- o .NumberFormat(code , opt) number format
- o .DateTimeFormat(code , opt) date/time
- o .DurationFormat(code , opt) duration
- o .RelativeTimeFormat(code , opt) reltime
- o .ListFormat(code , opt) locale list data
- o .Collator(code , opt) string comparator
- o .DisplayNames(code , opt) locale names
- o .PluralRules(code , opt) plural categories
- o .Segmenter(code , opt) text segmentation
- o .Locale(code , opt) locale identifier object

METHODS

- o .resolvedOptions() locale and data details

FORMATTING METHODS

- s .format(n) locale format number
- a .formatToParts(n) ↳ to array

RANGE METHODS — ONLY NUMBER & DATETIME

- s .formatRange(start , end) range
- a .formatRangeToParts(start , end) ↳ split

COLLATOR METHODS

- n .compare(str1 , str2) compares two strings

DISPLAYNAMES METHODS

- s .of(code) return locale name

PLURALRULES METHODS

- s .select(n) get plural name
- s .selectRange(start , end) get plural ranges

SEGMENTER METHODS

- o .segment(str) locale segments string

LOCALES METHODS

- a .getCalendars()
- a .getCollations()
- a .getTimeZones()
- a .getHourCycles()
- a .getNumberingSystems()
- o .getTextInfo()
- o .minimize()
- o .maximize()
- o .getWeekInfo()

S Local/Session Storage

PROPERTIES

- n .length number of items in storage

METHODS

- s .key(n) get key name on position n
- s .getItem(key) get value of key key
- / .setItem(key , value) set value
- / .removeItem(key) remove key
- / .clear() remove all items (current site)

G GeneratorFunction

EXAMPLE — { VALUE: ..., DONE: FALSE }

```
function* gen() {
  yield 1;
  yield 2;
  yield 3;
}
const fg = gen();
fg.next(); // 1
fg.next(); // 2
fg.next(); // 3
fg.next(); // undefined
```