CHEAT SHEET

JAVASCRIPT WEB DEVELOPMENT

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

n 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

- s .toExponential(dec) exp. notation
- s .toFixed(dec) fixed-point notation
- s .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

r Regexp() = /.+/ig

PROPERTIES

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

METHODS

- a .exec(str) exec search for a match
- b .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

- x(?=y) x (only if x is followed by y)
- x(?!y) x (only if x is not followed by y)

s String() = 'text'

PROPERTIES

- n .length string size

METHODS

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

d Date()

METHODS

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

UNIT GETTERS / SETTERS (ALSO .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)

LOCALE & TIMEZONE METHODS

- n .getTimezoneOffset() offset in mins
- 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

a Array() = [1, 2, 3]

PROPERTIES

- n .length number of elements

METHODS

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

MODIFY SOURCE ARRAY METHODS

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

ITERATION METHODS

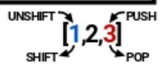
- ai .entries() iterate key/value pair array
- ai .keys() iterate only keys array
- ai .values() iterate only values array

CALLBACK FOR EACH METHODS

- b .every(cb(e,i,a), arg) test until false
- b .some(cb(e,i,a), arg) test until true
- a .map(cb(e,i,a), arg) make array
- a .filter(cb(e,i,a), arg) make array w/true
- o .find(cb(e,i,a), arg) return elem w/true
- n .findIndex(cb(e,i,a), arg) return index
- a .flatMap(cb(e,i,a), arg) map + flat(1)
- ▨ .forEach(cb(e,i,a), arg) exec for each
- o .reduce(cb(p,e,i,a), arg) accumulative
- o .reduceRight(cb(p,e,i,a), arg) from end

ADD/REMOVE METHODS

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



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

PROPERTIES

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

METHODS

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

n number	d date
∅ NaN (not-a-number)	r regular expresion
s string	f function
b boolean (true/false)	o object
a array	▨ undefined

available on ECMAScript 2015 or higher

- ⓝ static (ex: Math.random())
- n non-static (ex: new Date().getDate())

argument required
argument optional