

PROPRIEDADES

- n** .POSITIVE_INFINITY $+\infty$
- n** equivalente .NEGATIVE_INFINITY $-\infty$
- n** equivalente .MAX_VALUE maior valor
- n** positivo .MIN_VALUE menor valor positivo
- n** EPSILON diferença entre 1 e menor $> 1/n$.NaN
- n** valor não numérico

MÉTODOS	
S	exp. notação
S	notação de ponto fixo
S	altera a precisão
verifica se o número é finito	<code>b.isInteger(n)</code>
verifica se o número é int.	<code>b.isNaN(n)</code> verifica se
o número é NaN	<code>n.parseInt(s, radix)</code> string
para inteiro	<code>n.parseFloat(s, radix)</code> string para
float	

PROPRIEDADES	
n	índice para iniciar a regexp global
s	flag g (pesquisar todas as correspondências)
i	flag i (corresponde inferior/superior)
m	flag m (corresponder múltiplas linhas)
b	flag b (pesquisar em lastIndex)
u	flag u (habilitar recurso unicode)

MÉTODOS

a **.exec(str)** exec procura por uma correspondência verifica se regexp corresponde

AULAS

- . qualquer caractere **lt** tabulador **ld** dígito **[0-9]** **lr** retorno de carro **ld** sem dígito **[^0-9]** **ln** avanço de linha **lw** qualquer caractere alfanumérico **[A-Za-z0-9_]** **W** sem caracteres alfanuméricos **[^A-Za-z0-9_]** **ls** qualquer caractere de espaço (espaço, tabulação, enter...) **lS** sem espaço char (espaço, tabulação, enter...) **lxN** char com código **N** **[b]** backspace **luN** char com unicode **N** **l0** NUL char

CONJUNTOS DE PERSONAGENS OU ALTERAÇÃO

[abc] corresponde a qualquer conjunto de caracteres
[^abc] corresponde a qualquer caractere. conjunto não incluído
alb corresponde a a ou b

LIMITES	
^ início da entrada	\$ fim da entrada
\b limite de palavra com largura zero	
\B limite sem palavra de largura zero	

AGRUPAMENTO

(x) grupo de captura (? : x) nenhum grupo de captura
ln referência ao grupo n capturado

QUANTIFICADORES

- x^* precedendo x 0 ou mais vezes $\{0, \infty\}$ x^+ precedendo x 1 ou mais vezes $\{1, \infty\}$ $x^?$ precedendo x 0 ou 1 vez $\{0, 1\}$ $x\{n\}$ n ocorrências de x $x\{n, \infty\}$ pelo menos n ocorrências de x $x\{n, m\}$ entre n & m ocorrências de x

Asserções

x(?=y) x (somente se **x** for seguido por **y**) **x(?!y)** (somente se **x** não for seguido por **y**)

PROPRIEDADES	
n	tamanho da string
MÉTODOS	
s	char na posição
n [i]	<code>.charCodeAt(index)</code> unicode na pos.
n	cp na posição
s	<code>n2...</code> código para char
s	<code>fromCodePoint(n1, n2...)</code> cp para char

s `str2...)` combinar texto

b `startsWith(str, size)` verifica o início `b.endsWith(str, size)` verifica o final `b.includes(str, from)`

n `indexOf(str, from)` encontrar

i `indexOf(str, from)` encontrar

d `indexOf(str, from)` encontrar

n `indexOf(str, from)` encontrar

a `indexOf(str, from)` encontrar

(regex) retorna iterador com todos

(form) unicode normaliza `(len, pad)`

adiciona preenchimento final `s.padStart(len, pad)`

adiciona preenchimento inicial `s.padEnd(len, pad)`

repete a string n vezes `s.repeat(n)`

substitui a string `s.replace(str, regex, newstr)`

func) str entre ini/end
slen) substr de comprimento
ede lenend) fragmento de
ssubstr .split(sep | regex, limite) divide a string
astring para minúscula
string para maiúscula
sremove espaço do início/fim
remove espaço do finalremove
ide com strremove
espaço das strings de modelo start com \$
(vars)

MÉTODOS

- n** **m, d, h, i, s, ms** timestamp
- timestamp da hora atual converter
- ar em timestamp definir timestamp
- UNIX () retornar carimbo de data/
- hora UNIX

GETTERS/SETTERS DE UNIDADE		(TAMBÉM .getUTC() / .setUTC())
n	m, d)	(aaaa)
n		(0-11)
n	d) .get / .setDate(d)	(1-31)
n	m, s, ms)	(0-23)
n	s, ms)	(0-59)
n	ms)	(0-59)
n		(0-999)
n	retorna o dia da semana	(0-6)

MÉTODOS DE LOCAL E FUSO HORÁRIO	
n	deslocamento em minutos
s	options)
s	options)
s	options)
s	retorna a data UTC
s	retorna a data americana
toTimeString()	retorna o horário americano
s	retorna a data ISO8601
s	retorna a data pronta para JSON





PROPRIEDADES	
n	número de elementos
MÉTODOS	
b	verifica se obj é um array
b	includes(obj) inclui o elemento?
n	from encontre o elemento. índice
n	from encontrar a partir do final
s	juntar elementos com separador
a	end retornar a parte do array
a	obj2...) retornar o array unido
flat(profundidade)	retorna matriz plana em n profundida

- MODIFICAR MÉTODOS DE MATRIZ DE ORIGEM
- a** copyWithin(pos, ini, end) copiar elementos
- a** fill(obj, ini, end) preencher array com obj
- a** invert() inverter array e retorná-lo
- a** sort(array (classificação unicode))
- a** del, o1, o2...) del&adicionar elemento

MÉTODOS DE ITERAÇÃO

- `ai.entries()` iterar array de par chave/valor `ai.keys()`
- iterar apenas array de chaves `ai.values()`
- iterar apenas array de valores

CHAMADA PARA CADA MÉTODO	
b	arg) teste até falso
b	arg) teste até verdadeiro
a	arg) make array
a	arg) make array w/true
o	find(cb(e,i,a)) arg) return elem w/true
n	findIndex(cb(e,i,a)) arg) retorna índice
a	flatMap(cb(e,i,a)) arg) map +
▒	flatMap(1).forEach(cb(e,i,a)) arg) exec para cada
r	reduzir (cb (p, e, i, a), arg) acumulativo
(b	(cb (p, e, i, a), arg) do final

ADICIONAR/REMOVER MÉTODOS	
	remove e retorna o último elemento
	<code>.push(o1, o2...)</code> adiciona elemento e retorna comprimento
	remove e retorna o primeiro elemento
	<code>.unshift(o1, o2...)</code> adiciona elemento e retorna len

- PROPRIEDADES
 - `returna o número de argumentos .name`
 - `returna o nome da função`
 - `objeto protótipo .prototype`
- MÉTODOS
 - `arg1, arg2...)` altere isso
 - `arg1)` com matriz de args
 - `arg1, arg2...)` função vinculada

n número	d data
NaN (não-um-número)	e expressão regular
s string	f objeto de
b booleano (verdadeiro/ falso)	o função
a array	? indefinido

estático (ex: `Math.random()`) não
nestático (ex: `new Date().getDate()`)
 argumento obrigatório
 argumento opcional

Eme

JAVASCRIPT DESENVOLVIMENTO WEB

JS

Criado por @Manz (<https://twitter.com/Manz>)<https://lenguajejs.com/>

Matemática
PROPRIEDADES
E Constante de Euler .LN2 logaritmo natural de 2 .LN10 logaritmo natural de 10 .LOG2E logaritmo de base 2 de E .LOG10E logaritmo de base 10 de E razão circunferência/ diâmetro .SQRT1_2 raiz quadrada de 1/2 .SQRT2 raiz quadrada de 2
MÉTODOS
abs(x) valor absoluto cbrt(x) raiz cúbica clz32(x) retorna zero bits iniciais (32) .exp(x) retorna ex .expm1(x) retorna ex -1 hypot(x1, x2...) comprimento da hipotenusa imul(a, b) sinal multiplicado log(x) logaritmo natural (base e) log1p(x) logaritmo natural (1+x) log10(x) logaritmo de base 10 log2(x) logaritmo de base 2 max(x1, x2...) retorna o número máximo min(x1, x2...) retorna o número mínimo pow(base, exp) retorna float número aleatório sign(x) retorna o sinal do número sqrt(x) raiz quadrada do número
MÉTODOS REDONDOS
ceil(x) arredondamento superior floor(x) arredondamento inferior (maior) round(x) arredondamento toFixed(x) precisão única toPrecision(x) arredondamento toInteger() remove fração e arredonda para inteiro mais próximo truncate(x) remove fração e arredonda para zero
MÉTODOS TRIGONOMÉTRICOS
acos(x) arco cosseno asin(x) arco seno atan(x) arco tangente atanh(x) arco tangente hiperbólico cos(x) cosseno sin(x) seno tan(x) tangente tanh(x) tangente hiperbólica expm1(x) retorna ex -1 log1p(x) logaritmo natural (1+x) log2(x) logaritmo de base 2 log10(x) logaritmo de base 10 max(x1, x2...) retorna o número máximo min(x1, x2...) retorna o número mínimo pow(base, exp) retorna float número aleatório sign(x) retorna o sinal do número sqrt(x) raiz quadrada do número

BigInt()
BigInt(n) = 9007199254740992n
Modulos Módulo ES
MÉTODOS
import {m1, m2 como nome} de "/file.js" import obj de "/file.js" importação padrão export {m1, m2} exportar objs como módulos import ("/file.js") dinâmico importar

JSON
manipulação de arquivo json
MÉTODOS
JSON.stringify(obj, replacer, space) analisa string para objeto JSON.parse(str) converte para str

Erro()
PROPRIEDADES
name retorna o nome do erro message retorna a descrição do erro

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

Objeto()
= {chave: valor, chave2: valor2}
PROPRIEDADES
__proto__ retorna ref. para objeto func.
MÉTODOS
assign(dst, src1, src2...) copiar valores create(prop) criar obj com prop defineProperty(obj, prop, desc) define propriedade defineProperties(obj, props) define propriedades delete(obj, prop) remove propriedade freeze(obj) congela objeto getOwnPropertyDescriptor(obj, prop) retorna descrição de propriedade getOwnPropertyNames(obj) retorna nomes de propriedades getOwnPropertySymbols(obj) retorna símbolos de propriedades getPrototypeOf(obj) retorna protótipo is(obj, val) verifica se são o mesmo isExtensible(obj) verifica se pode adicionar propriedades isFrozen(obj) verifica se obj está congelado isSealed(obj) verifica se obj está selado keys(obj) retorna chaves do objeto preventExtensions(obj) evita estender prototypeOf(obj, prot) altera protótipo setPrototypeOf(obj, prot) altera protótipo set(obj, prop, value) define propriedade structuredClone(obj) clona objeto estruturado toLocaleString(obj) retorna string equivalente toJSON(obj) retorna versão local valueOf(obj) retorna valor primitivo
MÉTODOS DE INSTÂNCIA
hasOwnProperty(prop) verifica se existe isPrototypeOf(prot, obj) teste em outro obj isFrozen() retorna string equivalente isSealed() retorna versão local isExtensible() retorna valor primitivo

Promessa()
funções assíncronas / aguardam assíncronas
MÉTODOS
all(obj) promessa de retorno p .catch(onRejected(s)) then(undef, s) p .finally(onFinish()) retorno then(onFulfilled(v), onRejected(s)) retorna promessa gananciosa race(obj) retorna promessa gananciosa resolve(obj) retorna promessa resolvida reject(reason) retorna promessa rejeitada

Proxy()
Refletir os mesmos métodos (não func)
MÉTODOS
apply(obj, arg, arglist) chamada de função construct(obj, arglist) trap new defineProperty(obj, prop, desc) trap deleteProperty(obj, prop) trap delete(obj, prop) trap enumerate(obj) trap get(obj, prop, rec) trap get property getOwnPropertyDescriptor(obj, prop) trap getPrototypeOf(obj) trap has(obj, prop) trap in operador ownKeys(obj) trap preventExtensions(obj) trap set(obj, prop, value) propriedade do conjunto de armadilhas setPrototypeOf(obj, prot) trap

Símbolo()
PROPRIEDADES
iterator especifica o iterador padrão match especifica a correspondência de regexp species especifica a função construtora
MÉTODOS
for(key) pesquisa símbolos existentes keyFor(sym) retorna chave do registro global

Definir()
WeakSet apenas obj como itens
PROPRIEDADES
size número de retorno de itens
MÉTODOS
add(item) adiciona item ao conjunto has(item) verifica se o item existe delete(item) remove item clear() remove todos os itens do conjunto
MÉTODOS DE ITERAÇÃO
entries() itera itens values() itera apenas o valor dos itens
CHAMADA PARA CADA MÉTODO
forEach(cb(e,i,a), arg) exec para cada

Mapa()
WeakMap apenas obj como chaves
PROPRIEDADES
size retorna o número de elementos
MÉTODOS
set(key, value) add pair key=value get(key) retorna o valor da chave has(key) verifica se a chave existe delete(key) remove elemento & retorna se existe clear() remove todos os elementos do mapa
MÉTODOS DE ITERAÇÃO
entries() itera elementos keys() itera apenas chaves values() itera apenas valores
CHAMADA PARA CADA MÉTODO
forEach(cb(e,i,a), arg) exec para cada

Gerador()
= função* () { ... }
MÉTODOS
next(value) return obj w/{value,done} return(value) return value & true done .throw(except) gera um erro

globais
inclui isFinite, isNaN, parseInt e parseFloat
MÉTODOS
eval(str) avalia o código javascript encodeURIComponent(URI) = para %3D decodeURIComponent(URI) %3D para =

Outros
DECLARAÇÕES VARIÁVEIS/CONSTANTES
var bloco variável deixe o escopo do bloco const declarar constante (somente leitura)
OPERADORES
?? operador coalescente nulo ? encadeamento opcional
DICAS RÁPIDAS
func(a=1) valor do parâmetro padrão func(...a) argumento rest (operador spread) (a) func(...a) { ... } função seta gorda (sem isto) `string` `\${a}` modelo com variáveis 0bn binário (2) n para decimal 0on octal (8) n para decimal 0xn hexadecimal (16) n para decimal para (i na matriz) { ... } iterar matriz, i = índice para (e da matriz) { ... } matriz iterativa, e = valor classe B estende A () {} classe sugar sintaxe

Enezeta.com

	desempenho	= informações sobre desempenho
.navigation propriedades		
.	navigation	informações sobre redir/digitar nav.
o		informações sobre desempenho de carga de lat
MÉTODOS		
n		carimbo de data/hora de alta precisão
	navegador	= informações sobre o navegador
PROPRIEDADES		
.	cookieOs cookies do navegador estão ativados?	
b		Privacidade DNT habilitada?
o		informações do usuário
e	geolocalização .idioma idioma no	
v	navegador	max no
d	dispositivo	navegador funciona no modo
online?		identifica o navegador do usuário
MÉTODOS		
n	pattern)	usa vibração do dispositivo
	localização	= informações sobre o URL atual
PROPRIEDADES		
s		url completo do documento
s		https://www.emezeta.com/
r	https://user:pass@www	https://
t	osmipass@wmpo	https:// emezeta.com:81/
s		https://emezeta.com:81/
l	https://emezeta.com:81/	do caminho http://
m	emezeta.com/42/	http://emezeta.com/
c	contacto	http://google.com/?
p	emezeta.searchParams parâmetros de pesquisa	
no	ronometro	origem origem do URL do documento
j		
ó		
y		
	eventos	(apenas eventos populares)
EVENTOS DE RATO		
e		x e
e		us
e		Mc
e		Iou
e		e.
-		
EVENTOS DE TECLADO		
e.		e.
e		-
onKeyPress		
DE CARGA/OBJETO		
e		e.
e		-
e	.or	
e	JnI	
e		
FORMULÁRIO/EVENTOS DE CAMPOS		
e		e
e		e
e		e
e		e
EVENTOS DE ANIMAÇÃO/TRANSIÇÃO		
e		Le
e		ve
e		
EVENTOS DE ANIMAÇÃO/TRANSIÇÃO		
e		Ar
e		nsi

JAVASCRIPT DESENVOLVIMENTO WEB

JS

Criado por @Manz (<https://twitter.com/Manz>)<https://lenguajejs.com/>**documento** = Objeto de documento

PROPRIEDADES	
.characterSet	documento
charset	.compatMode peculiaridades ou modo
padrão	.cookie retorna todos os cookies
doc string	.designMode retorna status do modo de design
.dir	retorna texto de direção:
"rtl" ou "ltr"	.doctype retorna tipo de documento (DTD)
.domain	retorna domínio de documento
.documentURI	retorna URLs documentos
.lastModified	retorna modificação de data/hora
.origin	retorna a origem do documento
.readyState	retorna o status de carregamento atual
.referrer	retorna a página anterior
.referrerPolicy	retorna URL usado para documento HTML
.URL	informações sobre URL
PROPRIEDADES DOS ELEMENTOS	
.body	elemento focado
.currentScript	retorna o elemento do script
.defaultView	retorna o elemento da janela
.documentElement	primeiro elemento (raiz)
.head	primeiro elemento rolável
PROPRIEDADES DA MATRIZ DE DOCUMENTOS	
.anchors	array de elementos de imagens
.applets	array de elementos de applets
.embeds	array de elementos de incorporados
.forms	array de elementos de formulários
.images	array de elementos de imagens
.links	array de elementos de links
.plugins	array de elementos de plugins
.scripts	array de elementos de scripts
PROPRIEDADES DA FOLHA DE ESTILO	
.styleSheets	matriz de arquivos de estilo
.css	preferido
.selected	css selecionado
MÉTODOS	
.adoptNode(node)	adota do ext
.createAttribute(name)	create Attr
.createDocumentFragment()	create Element(tag)
.createElement(tag)	create Element obj
.createEvent(type)	create Event object
.createRange()	create Range object
.createTextNode(text)	create TextNode
.enableStyleSheetsForSet(name)	importNode(node, desc)
.importNode(node, desc)	importa cópia
.getElementById(id)	encontra elemento com id específico

Elemento() = Objeto elemento

PROPRIEDADES	
.accessKey	se existir, tecla de atalho
.matrix	matriz de objetos Attr
.DOMTokenList	de classes
.list	lista de classes para string
.id	string do elemento
.name	nome string do elemento
.Tag	HTML do elemento
PROPRIEDADES DE POSIÇÃO, TAMANHO E ROLAGEM	
.n	elemento de largura da borda superior
.clientWidth	elemento de largura da borda esquerda
.clientHeight	elemento de largura interna
.scrollLeft	posição superior no documento
.scrollTop	posição esquerda no documento
.scrollHeight	altura do elemento
OBTÉM/DEFINIR PROPRIEDADES DO CÓDIGO HTML	
.getHTML()	obtem/configura HTML dentro do elemento
.setHTML(html)	obtem/configura HTML (incl. elemento)
MÉTODOS	
.closest(selector)	ancestral mais próximo
.getElementsByClassName(class)	getElementsByName(tag)
.getElementsByTagName(tag)	retorna o primeiro elemento
MÉTODOS DE ATRIBUTOS	
.hasAttribute(name)	existem atributos?
.getAttribute(name)	existe atributo?
.removeAttribute(name)	valor de retorno
.setAttribute(name, value)	define o atributo
MÉTODOS DE CLIENTRECT (POSIÇÃO E TAMANHOS)	
.getBoundingClientRect()	retorna posição.
.getClientRects()	retorna array pos/tamanho

Atributo() = Objeto de atributo

PROPRIEDADES	
.name	nome do atributo do elemento
.value	valor do atributo do elemento
DOMTokenList () = Lista de aulas	
PROPRIEDADES	
.length	número de itens
MÉTODOS	
.contains(token)	verifica se o item existe
.add(token)	adiciona item à lista
.remove(token)	del item da lista
.toggle(token)	del item se existir, adicione else

Nó() = Elemento secundário (elem. ou texto)

PROPRIEDADES	
.baseURI	URL base absoluta do nó
.namespaceURI	namespace do nó
.localName	nome do nó
.value	1 = elemento, 2 = texto, 9 = doc
.prefix	prefixo do namespace do nó
.textContent	texto do nó e dos filhos
PROPRIEDADES DE NAVEGAÇÃO	
.childNodes	coleção de nós filhos
.firstChild	primeiros filhos (incluindo texto)
.lastChild	últimos filhos (incluindo texto)
.nextElementSibling	próximo nó imediato
.previousElementSibling	nó anterior imediato
.parentElement	elemento pai imediato
.parentNode	nó pai imediato
.document	documento de retorno
MÉTODOS	
.append()	adiciona nó ao final
.cloneNode()	nó duplicado
.contains(node)	o nó é descendente? O nó tem filhos?
.isEqualNode(node)	verifica se são iguais
.normalize()	ret nomesp.
.normalize()	retorna o prefixo para um
.normalize()	filhos de forma normalizada
.replaceChild(newNode, oldNode)	del node & return

FilhoNode()

MÉTODOS	
.cloneNode()	clonando nó
.cloneNode(true)	clonando com especificações

ParentNode() p

PROPRIEDADES	
.childNodes	número de filhos
.children	elementos filhos
.firstChild	primeiro elemento filhos.
.lastChild	último elemento filho.

NonDocumentTypeChildNode()

PROPRIEDADES	
.nextElementSibling	próximo elemento
.previousElementSibling	elemento anterior

ClienteRect() = Coordenadas do elemento

PROPRIEDADES	
.top	coordenada superior do retângulo
.left	coordenada esquerda do retângulo
.right	coordenada direita do retângulo
.bottom	coordenada inferior do retângulo
.width	largura do recto
.height	altura do recto
MÉTODOS	
.getBoundingClientRect()	retorna posição.
.getClientRects()	retorna array pos/tamanho

Evento() = Evento em ação

PROPRIEDADES	
.bubbles	true=bubble, false=captures
.cancelable	evento é cancelável?
.currentTarget	elemento
.defaultPrevented	preventDefault()
.detail	informações
.target	referência ao objeto despachado
.timeStamp	hora em que foi criado
.type	tipo de evento
MÉTODOS	
.preventDefault()	cancela evento
.stopImmediatePropagation()	stopPropagation()
MÉTODOS	
.addEventListener(ev, cb(ev), {capture})	addEventListener(ev, cb(ev), {capture})
.removeEventListener(ev, cb(ev), {capture})	removeEventListener(ev, cb(ev), {capture})
.dispatchEvent(ev)	dispatchEvent(ev)

EventTarget (usar sobre elementos)

MÉTODOS	
.addEventListener(ev, cb(ev), {capture})	addEventListener(ev, cb(ev), {capture})
.removeEventListener(ev, cb(ev), {capture})	removeEventListener(ev, cb(ev), {capture})
.dispatchEvent(ev)	dispatchEvent(ev)