**MÉTODOS DE OBJETOS GLOBAIS em JAVASCRIPT**

**(funções build-in (nativas))**

**ARRAYS**

**array.concat(valor, valor…)**

Concatena arrays.

**valor**: qualquer quantidade de valores a serem concatenados

**array.every(predicado)**

Testa se um predicado é verdadeiro ou falso para cada elemento

**predicado**: uma função predicado para testar os elementos da array

**array.filter(predicado)**

Retorna elementos de array que passam um predicado

**predicado**: a função a ser chamada determina se um elemento de arrau está incluído do array retornado.

**array.forEach(função)**

Chama uma função para cada elemento do array

**função:** a função a ser chamada para cada elemento do array

**array.indexOf(valor, início)**

Pesquisa uma array

**valor**: o valor a ser procurado na array

i**nício**: o índice da array para iniciar a busca.

**array.join(separador)**

Concatena elementos da array para formar uma string

**separador**: um caractere para separar os elementos da array (, . ; / | ‘’)

**array.lastIndex(valor, início)**

Pesquisa uma array para trás

**valor**: o valor a ser pesquisado na array

**início**: o índice da array para se iniciar a pesquisa

**array.length()**

Retorna o tamanho de uma array

**array.map(função)**

Calcula novos elementos da array a partir dos antigos

**função**: a função chamada para cada elemento

**array.pop()**

Remove e retorna o último elemento de uma array

**array.push(valor)**

Anexa elementos em uma array

**valor**: o elemento a ser anexado

**array.reduce(função, inicial)**

Calcula um valor a partir dos elementos de um array

**função**: função que combina dois valores e retorna um novo valor reduzido

**inicial**: valor inicial para servir de base para a redução do array.

**array.reduceRight(função, inicial)**

Reduz uma array da direita para a esquerda

**função**: função que combina dois valores e retorna um novo valor reduzido.

**array.reverse()**

Inverte os elementos de uma array

**array.shift()**

Desloca os elementos da array para baixo.

**array.slice(início, fim)**

Retorna parte de uma array.

**início**: O índice da array no qual a fatia deve começar. Se for negativo, conta a partir do fim da array.

**fim**: O índice da array no qual a fatia deve terminar.

**array.some(predicado)**

Teste se um predicado é verdadeiro para cada elemento

**predicado**: uma função predicado para testar os elemento da array

**array.sort(ordemclass)**

Classifica os elementos de uma array.

**ordemclass**: Função para especificar a ordem de classificação.

**array.splice(início, contExcl, valor)**

Insere, remove ou substitui elementos da array.

**início**: O elemento da array no qual a inserção ou exclusão deve começar.

**conteExcl**: O número de elementos a serem incluídos da array.

**array.toLocaleString()**

Converte uma array em uma string localizada.

**array.toString()**

Converte uma array em string.

**array.unshift(valor…)**

Insere elementos no início da array.

**BOOLEAN**

**boolean.toString()**

Retorna o booleano em string.

**boolean.valueOf()**

Retorna o booleano primitivo contido no objeto Boolean.

**DATE**

**date.getDate()**

Retorna o dia do mês de um objeto Date.

**date.getDay()**

retorna o dia da semana

**date.getHours()**

Retorna o campo de horas.

**date.getMilliseconds()**

Retorna os milisegundos

**date.getMinutes()**

retorna os minutos

**date.getMonth()**

Retorna o mês

**date.getSeconds()**

Retorna os segundos

**date.getTime()**

Retorna à representação de milissegundos interna de um objeto.

**date.setDate()**

Configura o campo de dia do mês da data.

**date.setFullYear()**

Configura o campo de ano

**date.setHours()**

Configura o campo de horas

**date.setMilliseconds()**

Configura o campo de Milisegundos

**date.setMinutes()**

Configura o campo de minutos

**date.setMonth()**

Configura o campo de mês

**date.setSeconds()**

Configura o campo de segundos

**date.setTime()**

Configura o campo de data completo

**date.setYear()**

Configura o campo de ano

**date.toDateString()**

Retorna a date em string

**date.toGMTString()**

Converte um objeto Date em um String usando o fuso horário GMT

**date.toISOString()**

Converte Date em uma string no formato ISO8601

**date.toJSON()**

serializa um objeto Date usando toISOString()

**date.toLocaleDataString()**

Retorna uma string representando a parte da data, expressa no fuso horário local, usando as convenções locais de formatação de data.

**date.toLocaleString()**

Converte um objeto Date em uma string, usando o fuso horário local e as convenções de formatação locais de data.

**date.toLocaleTimeString()**

Retorna uma string representando a parte da hora da data, no fuso horário local.

**date.toString()**

Converte Date em String.

**date.toTimeString()**

Retorna uma string representando a parte da hora da data

**date.toUTCString()**

Converte Date em String, usando hora universal

**date.valueOf:**

Converte objeto Date em seu formato interno de milisegundos.

**Date.now()**

Retorna a hora atual

**Date.parse()**

Analisa um representação de string de uma data e hora e retorna a representação interna dess data em milisegundos.

**Date.UTC()**

Retorna a representação da data e hora UTC especificadas em milisegundos.

**ERROR**

**error.toString()**

retorna uma string definida pela implementação representando o Error.

**error.message()**

Mensagem de erro legível para o usuário

**error.name()**

O tipo de um erro

**error.toString()**

Converte um objeto error em uma string

**escape(s)**

Codifica uma string

s: string a ser codificada

**eval(código)**

Executa o código JS a partir de uma string

**código**: Uma string contendo a expressão JS a ser avaliada ou as instruções a serem executadas

**FUNCTION**

**function.apply(objthis, args)**

Chama a função como método de um objeto, passando uma array de argumentos.

**objthis**: O objeto no qual a função será aplicada. No corpo da função, objthis se torna o valor da palavra-chave this. se esse argumento é null, o objeto global é usado.

**args**: Uma array de argumentos para a função

**function.bind(o, args)**

Retorna uma nova função que chama esta como método do objeto especificado, com os argumentos especificados.

**o:** O objeto ao qual essa função deve estar vinculada

**args**: zero ou mair valores de argumentos que também estarão vinculados

**function.call(objthis, args)**

Retorna uma representação de string da função. Chama uma função como método de um objeto.

**objthis**: O objeto no qual a função será aplicada. No corpo da função, objthis se torna o valor da palavra-chave this. se esse argumento é null, o objeto global é usado.

**args**: Uma array de argumentos para a função

**function.caller()**

A função que chamou esta.

**function.lenght()**

O número de argumentos declarados

**function.prototype()**

O protótipo de uma classe de objetos

**function.toString()**

Converte uma função em uma string

**isFinite(n)**

Verifica se um número é finito.

**n:** número ser verificado

**isNaN(n)**

Verifica se é not a number

**N**: valor a ser verificado

**JSON**

**json.parse(s, reviver)**  
Analisa uma string formatada com JSON

**s**: string a ser analisada

**reviver**: Uma função que pode transformar valores analisados

**json.stringify(o, filtro, recuo)**

Serializa um objeto, array, ou valor primitivo.

**o:** o objeto, array ou valor primitivo a ser convertido em uma string Json.

**filtro**: uma função que pode substituir valores antes da transformação em strings ou num array contendo os nomes de propriedades a serem transformadas em strings.

**recuo**: um argumento especificando uma string de recuo ou o número de espaços a usar para o recuo.

**MATH**

**math.abs(x)**

Calcula um valor absoluto

**math.acos(x)**

Calcula um arco-cosseno

**math.asin(x)**

Calcula um arco seno

**math.atan(x)**

calcula um arco-tangente

**math.atan2(y, x)**

Calcula o ângulo o eixo x até um ponto

**y:** coordenada y do ponto

**x:** coordenada x do ponto

**math.ceil(x)**

arredonda para cima

**math.cos(x)**

Calcula um cosseno

**math.e(x)**

Calcula o expoente de e

**math.floor(x)**

Arredonda para baixo

**math.LN10**

A constante matemática de log 10

**math.LN2**

A constante matemática de log 2

**math.log(x)**

Calcula um logaritmo natural

**x**: Quaquer número ou expressão

**math.LOG10E**

A constante matemática log10e

**math.max(args)**

Retorna o maior argumento

**args:** zero ou mais valores

**math.min(args)**

retorna o menor argument

**math.pow(x, y)**

Calcula a potência de um número

**x**: o número a ser elevado à potência

**y**: a potência

**math.random()**

Retorna um número aleatório

**math.round(x)**

Arredonda para o inteiro mais próximo

**math.sin(x)**

calcula um seno

**math.sqrt(x)**

Calcula uma raix quadrada

**math.tan()**

Calcula tangente

**NUMBER**

**number.toLocaleString()**

Converte um número em uma string formatada pela localidade

**number.toPrecision(precisão)**

Formata os dígitos significativos de um número

**precisão**: O número de dígitos significativos a aparecer na string retornada.

**numer.toString(raiz)**

Converte um numero em uma string

**raiz**: Argumento especificadndo a raiz entre 2 e 36 na qual o número deve ser representado.

**number.valueOf()**

retorna o valor numérico primitivo

**OBJECT**

**object.create(proto, descritores)**

Cria um objeto com o protótipo e as propriedades especificados

**proto**: O protótipo do objeto recentemente criado ou null

descritores: um objeto que mapeia nomes de propriedades em descritores de propriedade

**object.defineProperties(o)**

Cria ou configura várias propriedades de objeto.

**o:** objeto no qual as propriedades serão criadas ou configuradas.

**object.defineProperty(o, name, desc)**

Cria ou configura uma propriedade de objeto

**o:** objeto no qual uma propriedade será criada ou configurada

**nome**: nome da propriedade a ser criada ou configurada

**desc**: objeto descritor de propriedade descrevendo a nova propriedade ou as alterações feitas em uma propriedade já existente.

**object.freeze(o)**

Torna um objeto imutável

**o:** objeto a ser congelado

**object.getOwnPropertyDescriptor(o, nome)**

Consulta atributos de propriedade

**o:** o objeto consultado

nome: nome da propriedade ou índice de array a consultar

**object.getOwnPropertyNames(o)**

Retorna os nomes de propriedades não herdadas

**o:** o objeto

**object.getPrototype(o)**

Retorna o protótipo de um objeto

**o:** o objeto

**object.hasOwnProperty()**

Verifica se uma propriedade é herdada

**object.isExtensible(o)**

Novas propriedades podem ser adicionadas em um objeto

**object.isFrozen(o)**

Verifica se objeto é imutável

**object.isSealed(o)**

Verifica se propriedades podem ser adicionadas ou excluídas de um objeto

**object.keys(o)**

Retorna nomes de propriedades próprias enumeráveis.

**object.preventExtensions(o)**

Não permite novas propriedades em um objeto

**object.propertyIsEnumerable(nomeprop)**

Diz se a propriedade será vsta por um laço for

**nomeprop**: uma string contendo o nome de uma propriedade de objeto

**object.seal(o)**

Impede a adição ou exclusão de propriedades

**object.toLocaleString()**

Retorna a representação de string localizada de um objeto

**object.toString()**

Define a representação de string de um objeto

**object.valueOf()**

O valor primitivo do objeto especificado

**parseFloat(s)**

Converte uma string em um número

**s**: string a ser convertida

**parseInt(s)**

Converte um string em número inteiro

**s**: string a ser convertida

**REGEX**

**regexp.exec(string)**

Comparação de padrões de uso geral

**string**: a string a ser pesquisada

**regexp.global()**

Diz se uma expressão regular coincide globalmente

**regexp.ignoreCase**

Diz se uma expressão regular não diferencia letras maiúsculas e minúsculas

**regexp.lastIndex**

Diz a posição inicial da próxima coincidência

**regexp.source**

O texto da expressão regular

**regexp.test(string)**

Testa se uma string corresponde a um padrão

**string**: a string a ser testada

**regexp.toString()**

converte uma regex em string

**STRING**

**string.charAt(n)**

Obtém o n-ésimo caractere de uma string

**n:** índice do caractere

**string.charCodeAt(n)**

Obtém o n-ésimo código de caractere de uma string

**n**: o índice do caractere

**string.concat(valor…)**

Concatena strings

**valor**: um ou mais valores a serem concatenados

**string.fromCharCode(c…)**

Cria uma string a partir de codificações de caractere

**c:** Zero ou mais inteiros especificando as codificações unicode dos caracteres na string

**string.indexOf(substring, início)**

Pesquisa uma string

**substring**: a substring a ser pesquisada dentro da string

**início**: Inteiro especificando a posição dentro da string

**string.lastIndexOf(substring, início)**

**substring**: a substring a ser pesquisada dentro da string

**início**: inteiro especificando a posiçaõ dentro da string

**string.length**

Comprimento da string

**string.localeCompare(alvo)**

Compara uma string com outra, usando a ordenação específica da localidade

**alvo**: string a ser comparada

**string.match(regex)**

Encontra uma ou mais correspondência de expressão regular

**regex**: um objeto regex especificando o padrão a ser correspondido.

**string.replace(regex, subst)**

Substitui substrings correspondentes a uma regex

**regex**: objeto regex especificando o padrão a ser substituído

**subst**: uma string especificando o texto substituto ou uma função que é chamada para gerar o texto substituto.

**string.search(regex)**

Pesquisa uma expressão regular

**regex**: objeto regex especificando o padrão procurado.

**string.slice(início, fim)**

Extrai uma substring

**início**: O índice da string onde a fatia deve começar

**fim**: o índice onde deve terminar

**string.split(delimitador, limite)**

Decompõe a string em uma array de strings

**delimitador**: string ou regex na qual a string é decomposta

**limite**: inteiro especificando o comprimento máximo da array retornada.

**string.substr(início, comprimento)**

Extrai uma substring

**início**: posição inicial da substring

**comprimento**: número de caracteres da substring

**string.substring(de, para)**

Retorna uma substring de uma string

**de**: inteiro especificando a posição dentro da string do primeiro caractere da substring desejada

**para**: um inteiro especificando a posição do último caractere da substring desejada.

**string.toLocaleLowerCase()**

Converte uma string para letras minúsculas

**string.toLocaleUpperCase()**

Converte para letras maiúsculas

**string.toLowerCase()**

Converte para minúsculas

**string.toString()**

retorna a string

**string.toUpperCase()**

Converte uma string para maiúsculas

**string.trim()**

Retira espaços brancos à esquerda e à direita

**string.valueOf()**

Retorna a string

**WINDOW**

**window.alert(string)**

window.atob(string, a\_para\_b)

window.blur()

window.btoa(string, b\_para\_a)

window.clearInterval(long alça)

window.clearTimeout(long alça)

window.close()

window.confirm(string, msg)

window.focus()

window.getComputedStyle(element elt [string pseudoElt])

window.open([string url], [string alvo], [string recursos], [string substituição])

window.postMessage(any msg, string origemAlvo, [MsgPort [ ] portas])

window.print()

window.prompt(string msg, [string padrão])

window.scroll(long x, long y)

window.scrollBy(long x, long y)

window.scrollTo(long x, long y)

window.setInterval(function, unsigned long intervalo, any args…)

window.setTimeout(function, unsigned long tempolimite, any args…)

window.showModalDialog(string url, [any args])

window.document.adoptNode(nó)

window.document.close()

window.document.createComment(string dados)

window.document.createDocumentFragment()

window.document.createElement(string nomeLocal)

window.document.createElementNS(string nameSpace, string nomeQualificado)

window.document.createevent(string interfaceEvento)

window.document.createProcessinginstruction(string destino, string dados)

window.document.createTextNode(string dados)

window.document.elementFromPoint(floatx, float y)

window.document.execCommand(string idComando, [boolean iuExibição, [string valor]])

window.document.getElementById(string idElemento)

window.document.getElementByClassName(string nomesClasse)

window.document.getElementByName(string nomeElemento)

window.document.getElementByTagName(string nomeQualificado)

window.document.getElementByTagNameNS(string namespace, string nomeLocal)

window.document.hasFocus()

window.document.importNode(node nó, boolean profundidade)

window.document.open(string url, string nome, string recursos, [boolean substituição])

window.document.open([string tipo], [string substituição])

window.document.queryCommandEnabled(string IdComando)

window.document.queryCommandIndeterm(string idComando)

window.document.queryCommandState(string idcomando)

window.document.queryCommandSupported(string idComando)

window.document.queryCommandValue(string idComando)

window.document.querySelector(string seletores)

window.document.querySelectorAll(string seletores)

window.document.write(string texto…)

window.document.writeln(string texto…)

element.blur()

.