# Essencial para iniciantes Javascript Cheat Sheet



Aprenda JavaScript de forma prática e rápida com a nossa folha de dicas essencial para iniciantes.

Jonas Poli

# Noções básicas de Javascript

# Incluindo JavaScript em uma página HTML

```
<script type="text/javascript">
//O código JS vai aqui
</script>
```

# Chamar um arquivo JavaScript externo

```
<script src="myscript.js"></script><code></code>
```

# **Including Comments**

```
// Comentários de uma linha
```

```
/*
comentar aqui - Comentários de várias linhas
*/
```

# Variáveis

## var, const, let

var

A variável mais comum. Pode ser reatribuída, mas só pode ser acessada dentro de uma função. As variáveis definidas com var se movem para o topo quando o código é executado.

const

Não pode ser reatribuída e não é acessível antes de aparecer no código.

let

Similar a const, no entanto, uma variável let pode ser reatribuída, mas não redeclarada.

# Tipos de dados

#### Números

```
var idade = 23
```

## Tipo Variável

```
var x
```

## Texto (strings)

```
var a = "início"
```

## Operações

```
var b = 1 + 2 + 3
```

Declarações verdadeiras ou falsas

```
var c = true
```

#### Números constantes

```
const PI = 3.14
```

## Objetos

```
var nome = {firstName:"João", lastName:"Silva"}
```

# **Objetos**

```
var pessoa = {
firstName:"João",
lastName:"Silva",
idade:20,
nacionalidade:"Alemão"
};
```

# **Arrays**

var frutas = ["Banana", "Maçã", "Pera"];

# Métodos com Array

## concat()

Combina vários arrays em um só

## indexOf()

Retorna a primeira posição em que um determinado elemento aparece em um array

## join()

Combina elementos de um array em uma única string e retorna a string

## lastIndexOf()

Fornece a última posição em que um determinado elemento aparece em um array

## pop()

Remove o último elemento de um array

## push()

Adiciona um novo elemento no final

## reverse()

Inverte a ordem dos elementos em um array

# shift()

Remove o primeiro elemento de um array

## slice()

Retira uma cópia de uma porção de um array para um novo array

## sort()

Classifica os elementos em ordem alfabética

# splice()

Adiciona elementos de uma forma e posição especificadas

# toString()

Converte elementos em strings

# unshift()

Adiciona um novo elemento no início

# valueOf()

Retorna o valor primitivo do objeto especificado.

# **Operadores**

# Operadores Básicos

```
+ Adição
- Subtração
* Multiplicação
/ Divisão
( . . ) Operador de agrupamento
% Módulo (resto)
++ Incrementa números
```

## Operadores de Comparação

Decrementa números

```
== Igual a
=== Igual em valor e tipo
!= Não é igual
!== Não é igual em valor ou tipo
> Maior que
< Menor que
>= Maior ou igual a
< Menor ou igual a
? Operador ternário
```

# Operadores Lógicos

```
&& E lógico
|| Ou lógico
! Não lógico
```

# **Operadores Bitwise**

```
& E bit a bit
| OU bit a bit
~ NÃO bit a bit
^ OU exclusivo (XOR)
<< Deslocamento à esquerda bit a bit</pre>
```

- >> Deslocamento à direita bit a bit
- >>> Deslocamento para a direita sem sinal

# Funções

```
function nome(parâmetro1, parâmetro2, parâmetro3) {
// o que a função faz
}
```

## Saída de Dados

## alert()

Mostra dados em uma caixa de diálogo no navegador

## confirm()

Abre uma caixa de diálogo de sim/não e retorna verdadeiro/falso dependendo da escolha do usuário

## console.log()

Escreve informações no console do navegador, útil para depuração

## document.write()

Escreve diretamente no documento HTML

## prompt()

Cria uma caixa de diálogo para entrada do usuário

# Funções Globais

# decodeURI()

Decodifica um Identificador Uniforme de Recursos (URI) criado por encodeURI ou similar

## decodeURIComponent()

Decodifica um componente de URI

## encodeURI()

Codifica um URI em UTF-8

## encodeURIComponent()

O mesmo, mas para componentes de URI

## eval()

Avalia código JavaScript representado como uma string

## isFinite()

Determina se um valor passado é um número finito

## isNaN()

Determina se um valor é NaN ou não

## Number()

Retorna um número convertido de seu argumento

## parseFloat()

Analisa um argumento e retorna um número de ponto flutuante

## parseInt()

Analisa seu argumento e retorna um número inteiro

# Laços de repetição

```
for (antes do laço; condição do laço; executado após o laço) {
  // o que fazer durante o laço
}
```

#### for

A forma mais comum de criar um laço em Javascript

## while

Define as condições sob as quais um laço é executado

## do while

Semelhante ao laço while, no entanto, ele é executado pelo menos uma vez e verifica no final se a condição é atendida para executar novamente

## break

Usado para interromper e sair do ciclo em determinadas condições

#### continue

Pula partes do ciclo se certas condições forem atendidas

# Instruções If-Else

```
Se (condição) {
// O que fazer se a condição for satisfeita
} else {
// O que fazer se a condição não for satisfeita
}
```

# **Strings**

```
var pessoa = "João";
```

# Caracteres de Escape

```
\' - Aspas simples
\" - Aspas duplas
\\ - Barra invertida
\b - Backspace
\f - Feed de Formulário
\n - Nova linha
\r - Retorno de carro
\t - Tabulação horizontal
\v - Tabulação vertical
```

# Métodos de String

#### charAt()

Retorna um caractere em uma posição específica dentro de uma string.

#### charCodeAt()

Retorna o unicode do caractere naquela posição.

#### concat()

Concatena (une) duas ou mais strings em uma.

#### fromCharCode()

Retorna uma string criada a partir da sequência especificada de unidades de código UTF-16.

#### indexOf()

Fornece a posição da primeira ocorrência de um texto especificado dentro de uma string.

## lastIndexOf()

Mesmo que indexOf(), mas com a última ocorrência, procurando de trás para frente.

#### match()

Recupera as correspondências de uma string em relação a um padrão de pesquisa.

## replace()

Encontra e substitui um texto específico em uma string.

## search()

Executa uma pesquisa por um texto correspondente e retorna sua posição.

#### slice()

Extrai uma seção de uma string e a retorna como uma nova string.

## split()

Divide um objeto de string em uma matriz de strings em uma posição especificada.

#### substr()

Semelhante a slice(), mas extrai uma substring dependente de um número especificado de caracteres.

## substring()

Também semelhante a slice(), mas não pode aceitar índices negativos.

#### toLowerCase()

Converte strings em minúsculas.

#### toUpperCase()

Converte strings em maiúsculas.

## valueOf()

Retorna o valor primitivo (que não tem propriedades nem métodos) de um objeto de string.

# Regex (Expressões Regulares)

#### Modificadores de Padrão

- e Avaliar substituição
- i Executar correspondência insensível a maiúsculas/minúsculas
- g Executar correspondência global
- m Executar correspondência em múltiplas linhas
- s Tratar strings como uma única linha
- x Permitir comentários e espaços em branco no padrão
- U Padrão não guloso

#### Colchetes

- [abc] Encontrar qualquer um dos caracteres entre os colchetes
- [^abc] Encontrar qualquer caractere que não esteja nos colchetes
- [0-9] Usado para encontrar qualquer dígito de 0 a 9
- [A-z] Encontrar qualquer caractere de A maiúsculo a z minúsculo
- (a|b|c) Encontrar qualquer uma das alternativas separadas por |

#### Metacaracteres

- . Encontrar um único caractere, exceto nova linha ou caractere de término de linha
- \w Caractere de Palavra
- \W Caractere não-palavra
- \d Um dígito
- \D Um caractere não-dígito
- \s Caractere de espaço em branco
- \S Caractere não-espaço em branco
- \b Encontrar uma correspondência no início/fim de uma palavra
- \B Uma correspondência que não esteja no início/fim de uma palavra
- \0 Caractere NUL
- \n Um caractere de nova linha
- \f Caractere de formulário
- \r Caractere de retorno de carro
- \t Caractere de tabulação
- \v Caractere de tabulação vertical
- \xxx O caractere especificado por um número octal xxx

\xdd Caractere especificado por um número hexadecimal dd \uxxxx O caractere Unicode especificado por um número hexadecimal xxxx

## Quantificadores

- n+ Corresponde a qualquer string que contenha pelo menos um n
- n\* Qualquer string que contenha zero ou mais ocorrências de n
- n? Uma string que contenha zero ou uma ocorrência de n
- n{X} String que contém uma sequência de X n's
- n{X,Y} Strings que contém uma sequência de X a Y n's
- $n\{X,\}$  Corresponde a qualquer string que contenha uma sequência de pelo menos X n's
- n\$ Qualquer string com n no final dela
- ^n String com n no começo dela
- ?=n Qualquer string que seja seguida por uma string específica n
- ?!n String que não seja seguida por uma string específica n

# Números e Matemática

## Propriedades do Número

# MAX\_VALUE

O valor numérico máximo representável em JavaScript

## MIN VALUE

O menor valor numérico positivo representável em JavaScript

#### NaN

O valor "Não-é-um-Número"

#### **NEGATIVE INFINITY**

O valor negativo infinito

## POSITIVE INFINITY

O valor positivo infinito

## Métodos do Número

## toExponential()

Retorna uma string com um número arredondado escrito em notação exponencial

## toFixed()

Retorna uma string de um número com um número especificado de casas decimais

## toPrecision()

String de um número escrito com um comprimento especificado

## toString()

Retorna um número como uma string

## valueOf()

Retorna um número como um número

## Propriedades Matemáticas

E O número de Euler

LN2 O logaritmo natural de 2 LN10 Logaritmo natural de 10 LOG2E Logaritmo de base 2 de E LOG10E Logaritmo de base 10 de E

PI O número PI

SQRT1\_2 A raiz quadrada de 1/2 SQRT2 A raiz quadrada de 2

## Métodos Matemáticos

#### abs(x)

Retorna o valor absoluto (positivo) de x

#### acos(x)

O arcocoseno de x, em radianos

#### asin(x)

Arcoseno de x, em radianos

#### atan(x)

A tangente inversa de x como um valor numérico

## atan2(y, x)

Arco tangente do quociente de seus argumentos

## ceil(x)

Valor de x arredondado para o inteiro mais próximo acima

#### cos(x)

O cosseno de x (x está em radianos)

## exp(x)

Valor de Ex

## floor(x)

O valor de x arredondado para o inteiro mais próximo abaixo

## log(x)

O logaritmo natural (base E) de x

## $max(x, y, z, \ldots, n)$

Retorna o número com o valor mais alto

## $min(x, y, z, \ldots, n)$

Retorna o número com o valor mais baixo

## pow(x, y)

X elevado à potência y

## random()

Retorna um número aleatório entre 0 e 1

#### round(x)

O valor de x arredondado para o inteiro mais próximo

#### sin(x)

O seno de x (x está em radianos)

## sqrt(x)

Raiz quadrada de x

Investigat Chart Chart Lives Delic Périns 4

#### tan(x)

A tangente de um ângulo

# Lidando com Datas

## **Definindo Datas**

#### Date()

Cria um novo objeto de data com a data e hora atuais.

## Date (2024, 5, 21, 3, 23, 10, 0)

Cria um objeto de data personalizado. Os números representam o ano, mês, dia, hora, minutos, segundos e milissegundos. Você pode omitir qualquer um deles, exceto ano e mês.

## Date ("2024-06-23")

Declaração de data como uma string.

## Obtendo Valores de Data e Hora

#### getDate()

Obtém o dia do mês como um número (1-31).

#### getDay()

Obtém o dia da semana como um número (0-6).

## getFullYear()

Obtém o ano como um número de quatro dígitos (yyyy).

## getHours()

Obtém a hora (0-23).

## getMilliseconds()

Obtém o milissegundo (0-999).

#### getMinutes()

Obtém o minuto (0-59).

## getMonth()

Obtém o mês como um número (0-11).

## getSeconds()

Obtém o segundo (0-59).

## getTime()

Obtém os milissegundos desde 1º de janeiro de 1970.

#### getUTCDate()

Obtém o dia do mês na data especificada de acordo com o horário universal (também disponível para dia, mês, ano, hora, minutos etc.).

## parse

Analisa uma representação de string de uma data e retorna o número de milissegundos desde 1º de janeiro de 1970.

## Definindo Partes da Data

#### setDate()

Define o dia como um número (1-31).

#### setFullYear()

Define o ano (opcionalmente mês e dia).

#### setHours()

Define a hora (0-23).

## setMilliseconds()

Define os milissegundos (0-999).

#### setMinutes()

Define os minutos (0-59).

#### setMonth()

Define o mês (0-11).

#### setSeconds()

Define os segundos (0-59).

#### setTime()

Define o tempo (milissegundos desde 1º de janeiro de 1970).

## setUTCDate()

Define o dia do mês para uma data especificada de acordo com o horário universal (também disponível para dia, mês, ano, hora, minutos etc.).

# Nó do DOM

# Propriedades do nó

#### attributes

Retorna uma coleção dinâmica de todos os atributos registrados em um elemento

#### baseURI

Fornece a URL base absoluta de um elemento HTML

#### childNodes

Retorna uma coleção de todos os nós filhos de um elemento

#### firstChild

Retorna o primeiro nó filho de um elemento

#### lastChild

Retorna o último nó filho de um elemento

## nextSibling

Retorna o próximo nó no mesmo nível da árvore de nós

#### nodeName

Retorna o nome de um nó

## nodeType

Retorna o tipo de um nó

#### nodeValue

Define ou retorna o valor de um nó

#### ownerDocument

O objeto de documento de nível superior para este nó

#### parentNode

Retorna o nó pai de um elemento

## previousSibling

Retorna o nó imediatamente anterior ao atual

#### textContent

Define ou retorna o conteúdo textual de um nó e seus descendentes

## Métodos de nó

## appendChild()

Adiciona um novo nó filho a um elemento como último nó filho

## cloneNode()

Clona um elemento HTML

## compareDocumentPosition()

Compara a posição de documento de dois elementos

#### getFeature()

Retorna um objeto que implementa as APIs de um recurso especificado

#### hasAttributes()

Retorna true se um elemento tiver quaisquer atributos, caso contrário false

## hasChildNodes()

Retorna true se um elemento tiver quaisquer nós filhos, caso contrário false

## insertBefore()

Insere um novo nó filho antes de um nó filho especificado existente

## isDefaultNamespace()

Retorna true se um determinado namespaceURI for o padrão, caso contrário false

#### isEqualNode()

Verifica se dois elementos são iguais

## isSameNode()

Verifica se dois elementos são o mesmo nó

## isSupported()

Retorna true se um recurso especificado for suportado no elemento

## lookupNamespaceURI()

Retorna o namespaceURI associado a um determinado nó

## lookupPrefix()

Retorna uma DOMString contendo o prefixo para um determinado namespaceURI, se presente

#### normalize()

Une nós de texto adjacentes e remove nós de texto vazios em um elemento

#### removeChild()

Remove um nó filho de um elemento

## replaceChild()

Substitui um nó filho em um elemento

#### Métodos de elemento

## getAttribute()

Retorna o valor do atributo especificado de um nó de elemento

#### getAttributeNS()

Retorna o valor da string do atributo com o namespace e nome especificados

## getAttributeNode()

Obtém o nó de atributo especificado

## getAttributeNodeNS()

Retorna o nó de atributo para o atributo com o namespace e nome especificados

#### getElementsByTagName()

Fornece uma coleção de todos os elementos filhos com o nome da tag especificado

#### getElementsByTagNameNS()

Retorna uma HTMLCollection dinâmica de elementos com um determinado nome de tag pertencente ao namespace especificado

## hasAttribute()

Retorna true se um elemento tiver quaisquer atributos, caso contrário false

## hasAttributeNS()

Fornece um valor true/false indicando se o elemento atual em um determinado namespace tem o atributo especificado

## removeAttribute()

Remove um atributo especificado de um elemento

## removeAttributeNS()

Remove o atributo especificado de um elemento dentro de um determinado namespace

## removeAttributeNode()

Remove um nó de atributo especificado e retorna o nó removido

## setAttribute()

Define ou altera o atributo especificado para um valor especificado

## setAttributeNS()

Adiciona um novo atributo ou altera o valor de um atributo com o namespace e nome especificados.

#### setAttributeNode()

Define ou altera o nó de atributo especificado.

## setAttributeNodeNS()

Adiciona um novo nó de atributo com namespace a um elemento.

# Trabalhando com o Navegador

# Propriedades da Janela

#### closed

Verifica se uma janela foi fechada ou não e retorna true ou false

#### defaultStatus

Define ou retorna o texto padrão na barra de status de uma janela

#### document

Retorna o objeto documento para a janela

#### frames

Retorna todos os elementos <iframe> na janela atual

## history

Fornece o objeto Histórico para a janela

## innerHeight

A altura interna da área de conteúdo de uma janela

#### innerWidth

A largura interna da área de conteúdo

#### length

Descubra o número de elementos <iframe> na janela

#### location

Retorna o objeto de localização para a janela

#### name

Define ou retorna o nome de uma janela

## navigator

Retorna o objeto Navegador para a janela

#### opener

Retorna uma referência à janela que criou a janela

## outerHeight

A altura externa de uma janela, incluindo barras de ferramentas/scrollbars

#### outerWidth

A largura externa de uma janela, incluindo barras de ferramentas/scrollbars

## pageXOffset

Número de pixels pelos quais o documento atual foi rolado horizontalmente

## pageYOffset

Número de pixels pelos quais o documento foi rolado verticalmente

#### parent

A janela pai da janela atual

#### screen

Retorna o objeto Tela para a janela

#### screenLeft

A coordenada horizontal da janela (em relação à tela)

#### screenTop

A coordenada vertical da janela

#### screenX

Igual a screenLeft, mas necessário para alguns navegadores

## screenY

Igual a screenTop, mas necessário para alguns navegadores

#### self

Retorna a janela atual

#### status

Define ou retorna o texto na barra de status de uma janela

#### top

Retorna a janela do navegador mais alta

## Métodos da Janela

## alert()

Mostra uma caixa de alerta com uma mensagem e um botão OK

## blur()

Remove o foco da janela atual

## clearInterval()

Limpa um temporizador definido com setInterval()

## clearTimeout()

Limpa um temporizador definido com setTimeout()

## close()

Fecha a janela atual

#### confirm()

Mostra uma caixa de diálogo com uma mensagem e um botão OK e Cancelar

## focus()

Define o foco na janela atual

#### moveBy()

Move uma janela em relação à sua posição atual

#### moveTo()

Move uma janela para uma posição especificada

## open()

Abre uma nova janela do navegador

## print()

Imprime o conteúdo da janela atual

#### prompt()

Mostra uma caixa de diálogo que solicita entrada do visitante

## resizeBy()

Redimensiona a janela pelo número especificado de pixels

## resizeTo()

Redimensiona a janela para uma largura e altura especificadas

## scrollBy()

Rola o documento pelo número especificado de pixels

## scrollTo()

Rola o documento para coordenadas específicas

## setInterval()

Chama uma função ou avalia uma expressão em intervalos especificados

## setTimeout()

Chama uma função ou avalia uma expressão após um intervalo especificado

## stop()

Impede que a janela seja carregada

# Propriedades da tela

## availHeight

Retorna a altura da tela (excluindo a barra de tarefas do Windows)

## availWidth

Retorna a largura da tela (excluindo a barra de tarefas do Windows)

## colorDepth

Retorna a profundidade de bits da paleta de cores para exibição de imagens

## height

A altura total da tela

## pixelDepth

A resolução de cores da tela em bits por pixel

#### width

A largura total da tela

# **Eventos**

## Mouse

#### onclick

O evento ocorre quando o usuário clica em um elemento

#### oncontextmenu

O usuário clica com o botão direito do mouse em um elemento para abrir um menu de contexto

#### ondblclick

O usuário clica duas vezes em um elemento

#### onmousedown

O usuário pressiona um botão do mouse sobre um elemento

#### onmouseenter

O ponteiro se move sobre um elemento

#### onmouseleave

O ponteiro sai de um elemento

#### onmousemove

O ponteiro está se movendo enquanto está sobre um elemento

#### onmouseover

Quando o ponteiro é movido para um elemento ou um de seus filhos

#### onmouseout

O usuário move o ponteiro do mouse para fora de um elemento ou um de seus filhos

## onmouseup

O usuário solta um botão do mouse enquanto estiver sobre um elemento

## Teclado

## onkeydown

Quando o usuário pressiona uma tecla para baixo

## onkeypress

O momento em que o usuário começa a pressionar uma tecla

# onkeyup

O usuário libera uma tecla

## Frame

#### onabort

O carregamento de uma mídia é abortado

#### onbeforeunload

O evento ocorre antes do documento ser descarregado

#### onerror

Ocorre um erro ao carregar um arquivo externo

# **Eventos**

## Mouse

#### onclick

O evento ocorre quando o usuário clica em um elemento

#### oncontextmenu

Usuário clica com o botão direito em um elemento para abrir um menu de contexto

#### ondblclick

O usuário clica duas vezes em um elemento

#### onmousedown

Usuário pressiona um botão do mouse sobre um elemento

## onmouseenter

O ponteiro se move para um elemento

#### onmouseleave

Ponteiro sai de um elemento

## onmousemove

O ponteiro está se movendo enquanto está sobre um elemento

#### onmouseover

Quando o ponteiro é movido para um elemento ou um de seus filhos

#### onmouseout

Usuário move o ponteiro do mouse para fora de um elemento ou um de seus filhos

#### onmouseup

O usuário solta um botão do mouse enquanto sobre um elemento

## Teclado

## onkeydown

Quando o usuário pressiona uma tecla para baixo

## onkeypress

O momento em que o usuário começa a pressionar uma tecla

## onkeyup

O usuário libera uma tecla

#### Frame

#### onabort

O carregamento de uma mídia é abortado

#### onbeforeunload

Evento ocorre antes do documento ser descarregado

#### onerror

Um erro ocorre ao carregar um arquivo externo

## onhashchange

Houve mudanças na parte do anchor de uma URL

#### onload

Quando um objeto é carregado

## onpagehide

O usuário navega para longe de uma página da web

## onpageshow

Quando o usuário navega para uma página da web

#### onresize

A visualização do documento é redimensionada

#### onscroll

A barra de rolagem de um elemento está sendo rolada

#### onunload

Evento ocorre quando uma página foi descarregada

## Formulário

#### onblur

Quando um elemento perde o foco

## onchange

O conteúdo de um elemento de formulário muda (para <input>, <select> e <textarea>)

#### onfocus

Um elemento obtém foco

## onfocusin

Quando um elemento está prestes a receber foco

#### onfocusout

O elemento está prestes a perder o foco

## oninput

Entrada do usuário em um elemento

#### oninvalid

Um elemento é inválido

#### onreset

Um formulário é resetado

#### onsearch

O usuário escreve algo em um campo de pesquisa (para <input="search">)

#### onselect

O usuário seleciona algum texto (para <input> e <textarea>)

#### onsubmit

Um formulário é submetido

#### Arrastar

## ondrag

Um elemento está sendo arrastado

## ondragend

O usuário terminou de arrastar o elemento

## ondragenter

O elemento arrastado entra em um alvo de soltar

## ondragleave

Um elemento arrastado deixa o alvo de soltar

## ondragover

O elemento arrastado está em cima do alvo de soltar

## ondragstart

Usuário começa a arrastar um elemento

## ondrop

O elemento arrastado é solto no alvo de soltar

## Clipboard

#### oncopy

Usuário copia o conteúdo de um elemento

## oncut

O usuário corta o conteúdo de um elemento

## onpaste

Um usuário cola conteúdo em um elemento

## Media

#### onabort

Carregamento de mídia é abortado

## oncanplay

O navegador pode começar a reproduzir a mídia (por exemplo, um arquivo tem buffer suficiente)

## oncanplaythrough

Quando o navegador pode reproduzir a mídia sem interrupções

## ondurationchange

A duração da mídia é alterada

#### onended

A mídia chegou ao fim

#### onerror

Ocorre quando um erro ocorre ao carregar um arquivo externo

#### onloadeddata

Os dados da mídia são carregados

#### onloadedmetadata

Meta - Os metadados (como dimensões e duração) são carregados

#### onloadstart

O navegador começa a procurar pela mídia especificada

## onpause

A mídia é pausada pelo usuário ou automaticamente

## onplay

A mídia foi iniciada ou não está mais pausada

## onplaying

A mídia está sendo reproduzida após ter sido pausada ou parada para bufferizar

#### onprogress

O navegador está no processo de baixar a mídia

## onratechange

A velocidade de reprodução da mídia é alterada

#### onseeked

O usuário terminou de mover/pular para uma nova posição na mídia

#### onseeking

O usuário começa a mover/pular

#### onstalled

O navegador está tentando carregar a mídia, mas ela não está disponível

## onsuspend

O navegador não está carregando a mídia intencionalmente

## ontimeupdate

A posição de reprodução foi alterada (por exemplo, devido a um avanço rápido)

## onvolumechange

O volume da mídia foi alterado (incluindo mudo)

#### onwaiting

A mídia está pausada, mas espera-se que retome (por exemplo, buffering)

# Animação

#### animationend

Uma animação CSS é concluída

#### animationiteration

A animação CSS é repetida

#### animationstart

A animação CSS é iniciada

#### **Outros**

## transitionend

Disparado quando uma transição CSS é concluída

## onmessage

Uma mensagem é recebida através da fonte do evento

#### onoffline

O navegador começa a trabalhar offline

#### ononline

O navegador começa a trabalhar online

## onpopstate

Quando o histórico da janela é alterado

#### onshow

Um elemento <menu> é mostrado como menu de contexto

## onstorage

Uma área de armazenamento da web é atualizada

## ontoggle

O usuário abre ou fecha o elemento <details>

#### onwheel

A roda do mouse rola para cima ou para baixo sobre um elemento

#### ontouchcancel

O toque na tela é interrompido

## ontouchend

O dedo do usuário é removido de uma tela sensível ao toque

#### ontouchmove

Um dedo é arrastado pela tela

#### ontouchstart

O dedo é colocado na tela sensível ao toque

#### **Erros**

## try

Permite definir um bloco de código para testar erros

#### catch

Configura um bloco de código para executar em caso de erro

#### throw

Cria mensagens de erro personalizadas em vez dos erros padrão do JavaScript

## finally

Permite executar código, após try e catch, independentemente do resultado

```
function dividirNumeros(a, b) {
   try {
     if (b === 0) {
        throw "Não é possível dividir por zero!";
     }
     return a / b;
} catch (erro) {
     console.log("Erro: " + erro);
} finally {
     console.log("Fim da operação de divisão.");
}

dividirNumeros(10, 2); // retorna 5
dividirNumeros(10, 0); // exibe "Erro: Não é possível dividir por zero!" e
"Fim da operação de divisão."
```

## Valores de Nome de Erro

#### name

Define ou retorna o nome do erro

## message

Define ou retorna uma mensagem de erro em formato de string

#### **EvalError**

Ocorreu um erro na função eval()

#### RangeError

Um número está "fora do intervalo"

# ReferenceError

Uma referência ilegal ocorreu

# SyntaxError

Ocorreu um erro de sintaxe

# TypeError

Ocorreu um erro de tipo

# URIError

Ocorreu um erro em encodeURI()