



Incluindo JavaScript na página HTML

```
<script type="text/javascript">  
//JS code goes here  
</script>
```

Chamando um documento externo no JavaScript

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

Incluindo comentários

```
//  
única linha de comentário.
```

```
/* deixe seu comentário */  
várias linhas de comentário.
```

Variables

var, const, let

var

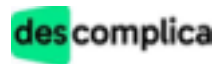
O variable statement declara uma variável, opcionalmente é possível atribuir a ela um valor em sua inicialização.

let

Declara uma variável local no escopo do bloco atual, opcionalmente iniciando-a com um valor.

const

Constantes possuem escopo de bloco, semelhantes às variáveis declaradas usando a palavra-chave let. O valor de uma constante não pode ser alterado por uma atribuição, e ela não pode ser declarada.



Data Types

```
var age = 23
```

Number (Númerico)

```
var x
```

Variável

```
var a = "init"
```

Texto (string)

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

Operação

```
var c = true
```

Declaração de verdadeiro ou falso, Boolean

```
const PI = 3.14
```

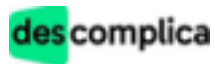
Constante numérica, Number

```
var name = {firstName:"John", lastName:"Doe"}
```

Objectos

Objects

```
var carro = {  
  marca:"Volkswagen",  
  nome: "Gol",  
  ano: 2008,  
  motor: "Flex"  
};
```



Arrays

```
var pessoas = ["Julio", "Luiza", "Clara"];
```

Métodos Array

concat()

juntar vários arrays em um só.

indexOf()

retorna a primeira posição em que um determinado elemento aparece no array.

join()

combina os elementos de um array em uma string e retorna a string.

lastIndexOf()

retorna a última posição em que um determinado elemento aparece no array.

reverse()

inverte a ordem dos elementos no array.

pop()

remove o último elemento do array.

push()

adiciona um novo elemento no final do array.

shift()

Remove o primeiro elemento de um array.

sort()

Ordena os elementos por ordem alfabética no array.

`slice()`

Retorna uma cópia de parte de um array a partir de um subarray criado entre as posições início e fim (fim não é necessário) de um array original. O Array original não é modificado.

`splice()`

Altera o conteúdo de uma lista, adicionando novos elementos enquanto remove elementos antigos.

`toString()`

Converte os elementos para string (texto).

`valueOf()`

Retorna o valor primitivo do objeto especificado.

`unshift()`

Adiciona um ou mais elementos no início de um array e retorna o número de elementos.

Operadores

Operadores básicos

- + Adição
- Subtração
- * Multiplicação
- / Divisão
- (..) Operador de agrupamento
- % Módulo (Resto da divisão)
- ++ Incrementa
- Decrementa

Operadores de comparação

- == Igual a
- === Igual ao valor e o tipo
- != Diferente de
- !== Diferente do valor e do tipo
- > Maior que

< Menor que
>= Maior ou igual
<= Menor ou igual
? Operador ternário

Operadores lógicos

&& Operador lógico de AND (E)
!! Operador lógico de OR (OU)
! Operador lógico de NOT (NÃO)

Funções

```
function nomeDaFunção(parametro1, parametro2, parametro3) {  
    // o que a função faz  
}
```

Saída de dados

`alert()`

Dados de saída em uma caixa de alerta na janela do navegador

`confirm()`

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

`console.log()`

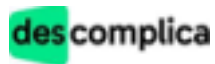
Grava informações no console do navegador, bom para fins de depuração

`document.write()`

Escreve diretamente no documento HTML

`prompt()`

Cria um diálogo para entrada do usuário



Global Functions

`decodeURI()`

Decodifica um Uniform Resource Identifier (URI) criado por `encodeURIComponent` ou similar

`decodeURIComponent()`

Decodifica um componente URI

`encodeURIComponent()`

Codifica um URI em UTF-8

`encodeURIComponent()`

Igual, mas para componentes URI

`eval()`

Computa um 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()`

É um objeto encapsulado que permite você trabalhar com valores numéricos.

`parseFloat()`

Analisa um argumento e retorna um número de ponto flutuante

`parseInt()`

Analisa seu argumento e retorna um inteiro

Loops

```
for (antes do loop; condição para o loop; executa após o loop)
{
    // o que fazer durante o loop
}
```

for

A maneira mais comum de criar um loop em Javascript

while

Configura as condições sob as quais um loop é executado

do while

Semelhante ao loop while, no entanto, ele é executado pelo menos uma vez e realiza uma verificação no final para verificar se a condição foi atendida para executar novamente

break

Usado para parar e sair do ciclo em determinadas condições

continue

Pule partes do ciclo se certas condições forem atendidas



Declarações de If - Else (Se, senão)

```
if (condição) {  
    // o que fazer se a condição for atendida  
} else {  
    // o que fazer se a condição não for atendida  
}
```

Strings

```
var professor = "Henrique Cruz";
```

Caracteres de escape

```
\' - Aspas única  
\" - Aspas duplas  
\\ - Barra invertida
```

String Methods

charAt()

Retorna um caractere em uma posição especificada dentro de uma string

charCodeAt()

Retorna o unicode do caractere nessa posição

concat()

Concatena (junta) 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 em uma string

`lastIndexOf()`

O mesmo que `indexOf()`, mas com a última ocorrência, pesquisando para trás

`match()`

Retorna as correspondências de uma string em um padrão de pesquisa

`replace()`

Localizar e substituir 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 string em uma matriz de strings em uma posição especificada

`substr()`

Semelhante a `slice()`, mas extrai uma substring que depende de um número especificado de caracteres

`substring()`

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

`toLowerCase()`

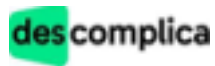
Converter strings para minúsculas

`toUpperCase()`

Converter strings para maiúsculas

`valueOf()`

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



Números e Math (Cálculos)

Propriedades de números

`MAX_VALUE`

O valor numérico máximo representável em JavaScript

`MIN_VALUE`

Menor valor numérico positivo representável em JavaScript

`NaN`

O valor “Não-Número” do inglês “Not-a-Number”.

`NEGATIVE_INFINITY`

O valor infinito negativo

`POSITIVE_INFINITY`

Valor infinito positivo

Métodos de números

`toExponential()`

Retorna uma string com um número arredondado escrito como notação exponencial

`toFixed()`

Retorna a string de um número com um número especificado de decimais

toFixed()

String de um número escrito com um comprimento especificado

toString()

Retorna um número como uma string

valueOf()

Retorna um número como um número

Math Methods

abs(x)

Retorna o valor absoluto (positivo) de x

acos(x)

O arco seno de x, em radianos

asin(x)

Arco Seno de x, em radianos

atan(x)

O arco tangente 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

cos(x)

O cosseno de x (x está em radianos)

exp(x)

Retorna e^x , onde x é o argumento, e e é a Constante de Euler, a base dos logaritmos naturais

floor(x)

O valor de x arredondado para o número inteiro mais próximo

log(x)

O logaritmo natural (base E) de x

max(x,y,z,...,n)

Retorna o número com o maior valor

min(x,y,z,...,n)

O mesmo para o número com o menor valor

pow(x,y)

X elevado a y

random()

Retorna um número aleatório entre 0 e 1

round(x)

O valor de x arredondado para o número inteiro mais próximo

sin(x)

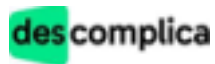
O seno de x (x está em radianos)

sqrt(x)

Raiz quadrada de x

tan(x)

A tangente de um ângulo



Lidando com datas

Definir datas

Date()

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

Date(2017, 5, 21, 3, 23, 10, 0)

Crie um objeto de data personalizado. Os números representam ano, mês, dia, hora, minutos, segundos, milissegundos. Você pode omitir o que quiser, exceto ano e mês.

Date("2017-06-23")

Declaração de data com uma string

Set Part of a Date

setDate()

Defina o dia como um número (1-31)

setFullYear()

Define o ano (opcionalmente mês e dia)

setHours()

Defina a hora (0-23)

setMilliseconds()

Definir milissegundos (0-999)

setMinutes()

Define os minutos (0-59)

setMonth()

Defina o mês (0-11)

setSeconds()

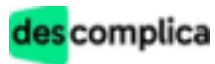
Define os segundos (0-59)

setTime()

Defina a hora (milissegundos desde 1º de janeiro de 1970)

setUTCDate()

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



DOM Node

Propriedades da interface Node

attributes

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

baseURI

Fornece a URL base absoluta de um elemento HTML

childNodes

Fornece uma coleção de nodes filhos (child nodes) de um elemento

firstChild

Retorna o primeiro node filho (child node) de um elemento

lastChild

O último node filho (child node) de um elemento

nextSibling

Dá-lhe o próximo node no mesmo nível da árvore de nodes

nodeName

Retorna o nome de um node

nodeType

Retorna o tipo de um node

nodeValue

Define ou retorna o valor de um node

ownerDocument

O objeto de documento de nível superior para este node

parentNode

Retorna o node pai de um elemento

previousSibling

Retorna o node imediatamente anterior ao atual

textContent

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

Métodos Node

appendChild()

Adiciona um novo node filho a um elemento como o último node filho

cloneNode()

Clona um elemento HTML

compareDocumentPosition()

Compara a posição do documento de dois elementos

getFeature()

Retorna um objeto que implementa as APIs de um recurso especificado

hasAttributes()

Retorna true se um elemento tiver algum atributo, caso contrário, false

hasChildNodes()

Retorna true se um elemento tiver node filhos, caso contrário, false

insertBefore()

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

isDefaultNamespace()

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

isEqualNode()

Verifica se dois elementos são iguais

isSameNode()

Verifica se dois elementos são o mesmo node

isSupported()

Retorna true se um recurso especificado for compatível com o elemento

lookupNamespaceURI()

Retorna o namespaceURI associado a um determinado node

lookupPrefix()

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

normalize()

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

removeChild()

Remove um node filho de um elemento

replaceChild()

Substitui um node filho em um elemento

Métodos de um Elemento

getAttribute()

Retorna o valor de atributo especificado de um node de elemento

getAttributeNS()

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

getAttributeNode()

Obtém node de atributo especificado

getAttributeNodeNS()

Retorna node de atributo para o atributo com o namespace e o nome fornecidos

getElementsByTagName()

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

getElementsByTagNameNS()

Retorna um HTMLCollection ativo de elementos com um determinado nome de tag pertencente ao dado namespace

hasAttribute()

Retorna true se um elemento tiver algum atributo, caso contrário, false

hasAttributeNS()

Fornece um valor verdadeiro/falso 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()`

Retira o node de atributo especificado e retorna o node removido

`setAttribute()`

Define ou altera o atributo especificado para um valor especificado

`setAttributeNS()`

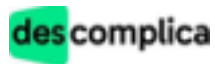
Adiciona um novo atributo ou altera o valor de um atributo com o namespace e o nome fornecidos

`setAttributeNode()`

Define ou altera o node de atributo especificado

`setAttributeNodeNS()`

Adiciona um novo node de atributo com namespace a um elemento



Trabalhando com o navegador

Propriedades da janela (Window)

closed

Verifica se uma janela foi fechada ou não e retorna verdadeiro ou falso

defaultStatus

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

document

Retorna o objeto de documento para a janela atual

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 Navigator para a janela

opener

Retorna uma referência à janela que criou a janela

outerHeight

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

outerWidth

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

pageXOffset

Número de pixels em que o documento atual foi rolado horizontalmente

pageYOffset

Número de pixels em que o documento foi rolado verticalmente

parent

A janela pai da janela atual

screen

Retorna o objeto Screen para a janela

screenLeft

A coordenada horizontal da janela (relativa à tela)

screenTop

A coordenada vertical da janela

screenX

O mesmo que `screenLeft`, mas necessário para alguns navegadores

screenY

Igual ao `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 superior do navegador