

Objetos

Exemplo de um objeto

objeto: carro

↳ propriedades:

marca: fiat

modelo: polo

cor: branco

↳ métodos: ^{→ são ações que podem ser executadas em objetos.}

carro.dirigir()

carro.parar()

carro.acelerar()

↳ um método é uma função armazenada como uma propriedade.

é boa prática declarar um objeto com const, mas pode-se usar let.

```
let Pessoa = {  
  nome: 'ano',  
  idade: 18,  
  corOchro: 'azul',  
}
```

- console.log(Pessoa)

saída: nome: 'ano', idade: 18, corOchro: 'azul'

- console.log(Pessoa.nome)

saída: ano

Eventos

Eventos são "coisas" que acontecem com elementos, sendo algo que o navegador faz ou o usuário faz.

onchange = alterar

onclick = ao clicar

onmouseover = ao passar o mouse

onmouseout = ao tirar o mouse do elemento

onkeydown = ao pressionar alguma tecla

onload = ao carregar a página 100%

Métodos de Strings

- slice = extrai uma parte do string
- substring = mesmo coisa que slice
- substr = mesmo coisa slice
- replace = substitui um valor por outro
- replaceAll = substitui todos os termos
- toUpperCase = torna maiúsculo
- toLowerCase = torna minúsculo
- concat = concatena duas strings ou mais
- trim = remove espaços em branco em ambos lados
- trimStart = remove espaço em branco do come
- trimEnd = remove espaço em branco do final
- padStart = preenche uma string desde início
- padEnd = preenche a string a partir do final
- charAt = mostra o caractere do índice especificado
- charCodeAt = mesmo que o anterior, mas retorna o único
- split = converte a string para um array
- length = calcula o comprimento

String search

- index Of = retorna o índice (posição)
- last Index Of = retorna o índice da última ocorrência
- search = procura uma string e retorna a posição
- match = retorna uma match
- match All = retorna um iterador
- includes = retorna true se o termo passado estiver na string
- starts with = retorna true se uma string começa com o termo passa
- ends with = verifica se o final tem o termo especificado

modelos literais / cadeia de modelos / modelos de strings
ou sintaxe de back-ticks

modelos literais usam acentos graves ` ` em vez de aspas " " para definir uma string. Assim podemos usar aspas simples e duplas dentro de uma string:

["testando "aspas" na string"]
↳ isso NÃO pode, dá erro

['testando "aspas" dentro "da" string']
↳ isso é permitido

" com as duplas
ou simples não
consigo escrever
uma string assim"
↳ isso dá erro

` com modelo literal
podemos escrever uma
string com várias linhas`
↳ isso pode

também usamos o modelo literal para ser um valor sem a necessidade de concatenação

A média é igual a:

$\$ \{media\}$

Podemos realizar expressões matemáticas dentro do String



a variável é interpretada

métodos numéricos

- toString retorna um número como um string
- toExponential escreve o número em notação exponencial
- toFixed retorna um número com casas decimais
- toPrecision retorna com um comprimento especificado
- ValueOf retorna um número como um número

let x = 123 → number

x.toString() → mudou para String

let num = 9.870432

num.toExponential(2)

↳ parâmetro que indica o número de casas após a vírgula = 9.87e+0

num.toFixed(3)

↳ parâmetro que indica o número de casas decimais