

## JAVASCRIPT exercício de composição de objetos PARTE 2

**Segue abaixo o mesmo enunciado do homework anterior. Neste, você deve estender seu código para armazenar o carro, bem como as peças que estiverem montadas, na Storage local do browser** (ao final do doc segue o código usando em sala de aula).

Em adição (opcional), veja se consegue usar também a **Storage de session** (por exemplo, vc pode abrir duas abas no browser para a mesma página, e contar em cada uma quantos carros ou peças foram criadas durante a sessão)

**DICA:** escreva functions que manipulam o storage, e botões ‘gravar’ e ‘recuperar’ que executam estas functions.

Enunciado do homework:

Um carro é composto de peças. As peças são:

Quatro rodas - Um volante - Uma marcha  
Um motor - Acessórios - Freio de mão  
Acelerador - Embreagem

Para que o carro possa ser considerado como montado, é preciso que funcionem os seguintes comportamentos: pode ser ligado, pode ser desligado, pode trocar de marcha, pode acelerar e freiar, pode estacionar (caso em que o freio de mão deve ser puxado), e deve ter pelo menos um acessório (p.ex., um ar condicionado) que pode ser ligado ou desligado.

**Requerido:** Na interface, você deve fazer um botão “criar” para cada peça: sempre que um botão é clicado, uma peça daquele tipo é criada.

**Requerido:** Também deve haver um botão para cada “comportamento” do carro (p.ex. Ligar, desligar, passar marcha, etc. *Porém, esses botões só podem funcionar se o carro estiver com todas as peças montadas* (exibir também uma mensgem se um desses botões for clicado mas o carro não estiver inteiramente montado).

**DICA:** como sabemos, podemos nos valer da flexibilidade e dinamismo das listas (arrays) e de objetos, para criarmos estruturas tanto “fixas” quanto modificáveis. No exercício, podemos pensar que: talvez um carro seja composto de um array de peças, ou que: talvez um carro tenha “sub-peças” (componentes) que por sua vez são objetos.

**NOTA:** cuidado, que ao modelar esse sistema de componente, alguns comportamentos podem ser *dependentes* de outros (p.ex., talvez, ligar o carro dependa de ligar o motor). No exercício, as dependências não estão explicitadas, portanto você deve decidir quais são (não precisa ir

até o máximo detalhe, basta pensar em qual seria o “mínimo de coerencia” possível para um carro poder ser ligado, em relação aos objetos que constam no enunciado

**Código usado em sala de aula:**

```
class cadastro {
    nome = ""
    endereco = ""
    telefone = ""
    idade = 10
}

c = new cadastro()
c.nome = "dudu"
c.endereco = "Rua A, 25"
c.telefone = "99-99-99-99"

// Store (armazenamento Local)
function gravar() {
    //localStorage.setItem("nomeUsuario", c.nome);
    // teste
    // localStorage.setItem("usuario", c); // só grava chave - valor string
    localStorage.setItem("idadeUsuario", c.idade.toString())
}

// Recuperando um dado gravado com getItem
function ler() {
    valor = localStorage.getItem("idadeUsuario")
    document.getElementById("msg").innerHTML = valor
}

// O storage tem a capacidade máxima de 5Mbytes
//=====
```

=====

**Entregue a PASTA com o arquivo .html (e outros arquivos na pasta, se necessário)**

=====