JAVASCRIPT exercicio de composição de objetos PARTE 4

Segue abaixo o mesmo enunciado do homework anterior. *Neste (que já foi modificado na parte 3 para remoção dos botões)****, você deve agora armazenar o estado do carro como um todo no localStorage, mesmo se o usuário não escolheu todas as peças, e recuperar essa info no próximo load,*** *deixando a página como ela ficou na última visita****. Se o carro armazenado for incompleto, você deve criar botões somente para as peças que faltam. Além disso, este código que carrega as infos deve ser executado a partir de funções anônimas autoexecutáveis.***

Ao final do doc, segue o código que foi usado para mostrar a evolução de uma function normal, para uma function anônima auto-executável.

Requerido: uma ou mais funções anônimas
Requerido: seu código deve estar em arquivos .js. Não faça código na página html

DICA: faça uma div ou uma p para ser “o pai” dos botões

= = = = =

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 decdir 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

= = = = =

Codigo usado em sala de aula:

```
// FUNÇÕES ANÔNIMAS

// um objeto como já nos acostumamos a fazer
function objeto1() {
    this.nome = ""
    // aqui, temos uma função nomeada
    this.mediaNotas = function mediaNotas(nota1, nota2, nota3) {
        return this.media = (nota1 + nota2 + nota3)/3
    }

    // aqui, temos uma função anônima
    this.mediaNotas2 = function (nota1, nota2, nota3) {
        return this.media = (nota1 + nota2 + nota3)/3
    }
}

// teste
var obj1 = new objeto1()
r = obj1.mediaNotas(5, 7.3, 10).toFixed(2)  // duas casas decimais
r2 = obj1.mediaNotas2(5, 7.3, 10).toFixed(2)  // duas casas decimais

// fora de um objeto, posso criar uma função anônima
// desde que haja uma referência para ela
var media = function (n1,n2,n3) {
    return (n1 + n2 + n3)/3
```

```
}
r3 = media(8,9,5)

// eu posso reescrever a função acima como:
var media2 = (
    function (n1,n2,n3) {
    return (n1 + n2 + n3)/3
    });
r4 =  media2(3,2,5);
// a função inteira ficou dentro de parentesis, se transformando em
// uma EXPRESSÃO.

// colocando dentro de parentesis, a função
// é interpretada e descartada. Note que não há uma referência para
// ela, como na função anterior
// NOTA: ao fazer isso, sempre use ; no final
(function defineA() {
    var a = "dudu"
    console.log("defineA rodou")
});

// agora vou transformar a função acima
// EM AUTOEXECUTÁVEL. Adicionei () no final
(function defineB() {
    var a = "dudu"
    console.log("defineB rodou")
})();
// NOTA: ao fazer isso, sempre use ';' no final
// Esta função roda automáticamente no load da página
// Esta técnica é usada para funções que só precisam
// executar uma vez ("run once")
// Investigue na console aceso a defineA, e à variável 'a'
```

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