Ministério da Educação
Universidade Federal de Itajubá – *Itajubá*
XDES03 – Programação Web
Exercícios —JavaScript DOM
Prof.: Phyllipe Lima

Instruções Gerais:
- Atividade Deverá ser Entregue Individualmente.
- Entregar a solução completa em formato ZIP pela tarefa do SIGAA.
- Horário de Entrega definido na Tarefa do SIGAA
- Formato de Entrega: Coloque os arquivos referentes as tarefas em suas respectivas pastas. Isto é, separe por Tarefa1, Tarefa2 etc. Em seguida coloque essas pastas em outra chamada "Exec", compacte em formato ".zip" e faça a entrega no SIGAA.
- A lista possui 2 (duas) Tarefas.
- O HTML é fornecido

Objetivos: Familiarizar com a manipulação do DOM com JavaScript:

- Selecionar Elementos
- Criar Elementos
- Tratar Eventos

Tarefa 1: Gerador de Cores Aleatórias.

Nessa tarefa você deverá criar um botão que responde ao evento "click". Toda vez que clicar no botão, uma cor aleatória deverá ser gerada.
Adicionalmente coloque um cabeçalho H1 acima do botão que irá informar, em formato rgb ou hexa, qual a cor atual.

Tanto o botão quanto o cabeçalho devem ser centralizados e agrupados por uma seção. Toda vez que o botão for clicado, a cor de fundo da seção deverá mudar. Isto é, não é para modificar o fundo de toda a página (body), apenas da seção com o botão e o cabeçalho.

As Figuras 1 e 2 mostram um exemplo da Tarefa1

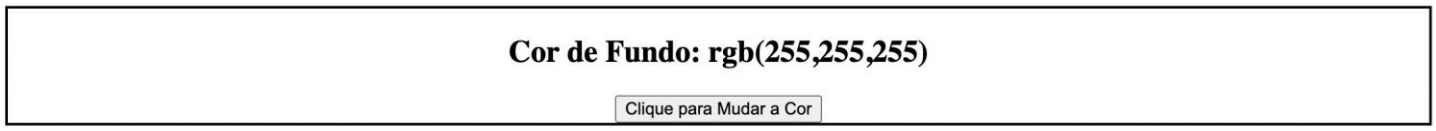

Figura 1

Figura 2

Tarefa 2: Lista de Compras

Nessa tarefa você deverá criar uma lista de compras.
Terá um formulário com duas entradas, uma para a quantidade e outro para o nome do produto. E para fechar um formulário teremos um botão “Adicionar” que irá colocar produtos no carrinho.

O Carrinho será representado por uma lista não-numerada. Toda vez que um produto for adicionado (clicar no botão “Adicionar” , um novo item será criado e colocado nessa lista. Esse item adicionado terá um botão que irá permitir se excluir da lista.

Exemplo de HTML que deverá ser gerado dinamicamente para um item, simulando que “1 Playstation 5” foi adicionado no carrinho.

```
<li>1: Playstation 5 <button>X</button></li>
```

Na página temos três seções. Uma de apresentação do carrinho, a segunda com o formulário para adicionar itens e a terceira com a lista dos itens adicionados.

Na terceira seção temos um parágrafo que informa: “Seu pedido está vazio. Adicione produtos no carrinho!”. Ao adicionar elementos, essa mensagem fica oculta.

A Figura 3 apresenta o carrinho vazio, a Figura 4 mostra o carrinho com 4 itens adicionados e a Figura 5 apresenta o carrinho com 2 itens removidos.

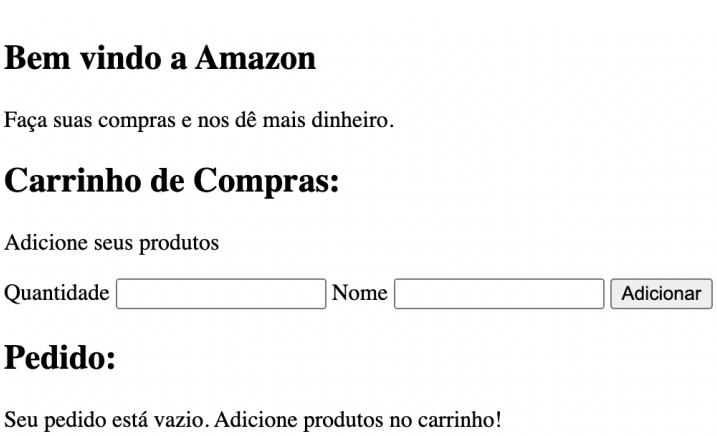

Figura 3

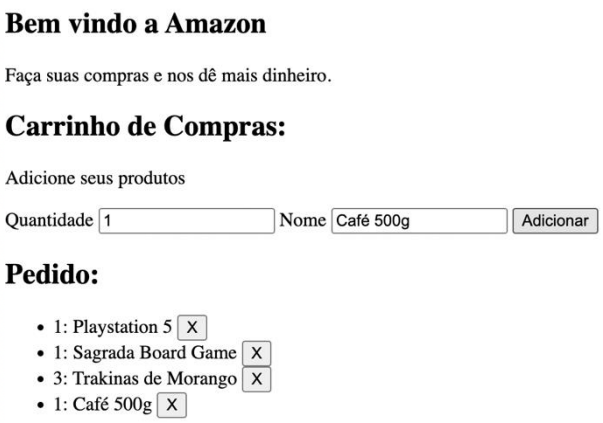

Figura 4

Figura 5

Dicas:

- Registre eventos nos elementos HTML com a função “addEventListener”. Esse recebe dois parâmetros, sendo o primeiro o tipo de evento e o segundo um *callback* para uma função que irá tratar o evento.
- Para essa lista estamos interessados nos eventos ‘click’ para botões e ‘submit’ para o formulário. O evento do formulário deverá ser adicionado no formulário e não no botão.

Exemplo1 : Registrando eventos de clique em um botão usando *arrow func*

```
const btn = document.querySelector('button');

btn.addEventListener('click', () => {
    console.log("Fui clicado");
});
```

Nesse exemplo ao invés de passar o ***callback***, fizemos a própria função dentro do campo do segundo parâmetro da função *addEventListener*. Foi utilizada a notação funcional, isto é, com “ *arrow function”*.

Exemplo 2 : Registrando eventos de submissão de um formulário com função callback e impedindo troca de página através do evento “e” recebido.

```
const form = document.querySelector('form');

form.addEventListener('submit', funcao);

const funcao = (e) => {
    //Essa primeira linha é necessária
    //para impedir o formulario de trocar de página
    //Para isso pegamos uma referência para o evento
    //disparado nesse clique, salvo como parâmetro "e"
    // (mas podemos dar qualquer nome)
    e.preventDefault();
    console.log("Fomulário submetido");
}
```

Para o evento de submissão de formulário podemos fazer da seguinte forma.

Exemplo 3 : Encontrando o nó pai de um elemento HTML e o removendo

```
const removerItem = (e) => {
    //Basta passar essa função como callback para o addEventListener
    //Acessando o evento, a variável “target” nos devolve o elemento que foi acionado
    //no evento
    e.target.parentNode.remove();
    const carrinho = document.querySelectorAll('ul li');
}
```

Dica final: Use console.log para que você entenda o que está ocorrendo e saindo na tela. Fique atento as exceções.

Exemplo 4 : Buscando valor de <input> e registrando “click”

Você também tratar o evento diretamente no botão. Nesse caso o tipo do evento é “click” e o botão requer que no HTML o type seja “button”. Para pegar o valor do <input> basta acessar a propriedade “value”

HTML

```
<form action="">
    <label for="username">Username</label>
    <input type="text" name="username" id="username">
    <button type="button">Clique Aqui</button>
</form>
```

JavaScript

```
const btn = document.querySelector('button');
const input = document.querySelector('#username');

btn.addEventListener('click', () => {
    let valorInput = input.value;
    console.log(valorInput); //Irá imprimir "Rick Roll"
});
```