<!DOCTYPE html>

<**html**>

<**head**>

<**meta** charset="utf-8" />

<**style** type="text/css">

\*{**font**: 25px calibri;}

**h1**{**font**: bold 40px calibri;}

</**style**>

</**head**>

<**body**>

<**h1**>Testando formatação de valores monetários e máscaras em campos de formulários</**h1**>

<**form** id="formulario">

<**label** for="ivalor">Valor SEM formatação (enquanto digita):</**label**><**br**/>

<**input** id="ivalor" type="text"

onkeypress="mMoeda(this)"/>

<**br**/> <**br**/>

<**label** for="iCPF">CPF:</**label**><**br**/>

<**input** id="iCPF" type="text" placeholder="999.999.999-99"

onkeyup="mCpf(this)" maxlength="14"

onblur="validarCPF(this)"/>

<**br**/> <**br**/>

<**label** for="iCEP">CEP:</**label**><**br**/>

<**input** id="iCEP" type="text" placeholder="99.999-999"

onkeypress="mCEP(this)" maxlength="10"/>

<**br**/> <**br**/>

<**label** for="iCelular">Celular:</**label**><**br**/>

<**input** id="iCelular" type="text" placeholder="(99)99999-9999"

onkeyup="mascara('(##)#####-####',this,event)"/>

<**br**/> <**br**/>

<**label** for="tNumero3">Telefone:</**label**><**br**/>

<**input** type="text" id="tNumero3" placeholder="(99)9999-9999"

onkeypress="mTel(this)" maxlength="13"/>

<**br**/><**br**/>

<**label** for="tNumero6">CNPJ:</**label**><**br**/>

<**input** type="text" id="tNumero6" placeholder="99.999.999/9999-99"

onkeyup="mascara('##.###.###/####-##',this,event)" />

</**form**>

<**script** src="MascarasValidaCPF.js"></**script**>

</**body**>

</**html**>

**function** mMoeda () {//extrai os 3 primeiros caracteres relativos ao 'R$ '

**var** v = (event.target.value).**substring**(3);

v = v.**replace**(/\D/**g**, "");

v = v.**replace**(/^0+/**g**, "");

v = v.**replace**(/(\d{1})(\d{13})$/, "$1.$2");

v = v.**replace**(/(\d{1})(\d{10})$/, "$1.$2");

v = v.**replace**(/(\d{1})(\d{7})$/, "$1.$2");

v = v.**replace**(/(\d{1})(\d{4})$/, "$1.$2");

v = v.**replace**(/(\d{1})(\d{1,1})$/, "$1,$2");// Para retornar os valores que

estão sendo digitados com a formatação ao elemento que chamou a função

event.target.value = "R$ " + v;

}

**function** mCpf() { // Esta função trata somente da formatação

**var** cpf = event.target.value;

cpf = cpf.**replace**(/\D/**g**, "")

cpf = cpf.**replace**(/(\d{3})(\d)/, "$1.$2")

cpf = cpf.**replace**(/(\d{3})(\d)/, "$1.$2")

cpf = cpf.**replace**(/(\d{3})(\d{1,2})$/, "$1-$2")

event.target.value = cpf;

}

**function** mTel () {

**var** tel = event.target.value;

tel = tel.**replace**(/\D/**g**, "")

tel = tel.**replace**(/^(\d)/, "($1")

tel = tel.**replace**(/(.{3})(\d)/, "$1)$2")

**if** (tel.**length** == 9) {

tel = tel.**replace**(/(.{1})$/, "-$1")

} **else** **if** (tel.**length** == 10) {

tel = tel.**replace**(/(.{2})$/, "-$1")

} **else** **if** (tel.**length** == 11) {

tel = tel.**replace**(/(.{3})$/, "-$1")

} **else** **if** (tel.**length** == 12) {

tel = tel.**replace**(/(.{4})$/, "-$1")

} **else** **if** (tel.**length** > 12) {

tel = tel.**replace**(/(.{4})$/, "-$1")

}

event.target.value = tel;

}

**function** mCEP () {

**var** cep = event.target.value;

cep = cep.**replace**(/\D/**g**, "")

cep = cep.**replace**(/^(\d{2})(\d)/, "$1.$2")

cep = cep.**replace**(/.(\d{3})(\d)/, ".$1-$2")

event.target.value = cep;

}

///////////////////////////////////////////////////////////////////////////////////////////////

<!DOCTYPE html>

<**html**>

<**head**>

<**title**>Tabela dinâmica</**title**>

<**meta** charset="utf-8" />

<**style**>

\*{**font**: 20px calibri;}

.myTable {

**width**: 50%;

**border**: 1px solid #000;

}

.myTable **tr**,**td**,**th** {

**border**: 1px solid #000;

}

</**style**>

</**head**>

<**body**>

<**h1** style="**font**: 30px calibri;">Criando tabelas de forma dinâmica</**h1**>

<**hr**/>

<**form**>

<**input** type="text" id="ra" placeholder="Informe o RA"/>

<**input** type="text" id="name" placeholder="Informe o Nome"/>

<**input** type="text" id="course" placeholder="Informe o Curso"/>

<**button** type="button" onclick="adicionarItem2()">cadastrar</**button**>

</**form**>

<**hr**/>

<**div**>

<**table** class="myTable" id="tabela">

</**table**>

</**div**>

<**script** src="funcoes.js"></**script**>

</**body**>

</**html**>

// criar um vetor de objetos

**var** alunos = [

{id:1, nome:'Gustavo Tacaki',curso:'ADS'},

{id:2, nome:'Any Velasco',curso:'ADS'},

{id:3, nome:'Felipe Carro',curso:'BCC'},

{id:4, nome:'João Victor',curso:'ADS'},

]

**function** carregaTabela(dados){

//vincular a variável dados com o <table> do arquivo HTML

**var** tab = document.getElementById('tabela');

// criar o elemento para a linha com o cabeçalho

**var** trCabecalho = document.createElement('tr');

// criar os elementos para cada coluna do cabeçalho

**var** thId = document.createElement('th');

**var** thNome = document.createElement('th');

**var** thCurso = document.createElement('th');

// Colocar os textos nos elementos criados

thId.innerHTML = 'RA';

thNome.innerHTML = 'Nome do Aluno';

thCurso.innerHTML = 'Curso';

// adicionar as colunas no elemento que gerencia a linha do cabeçalho

trCabecalho.appendChild(thId);

trCabecalho.appendChild(thNome);

trCabecalho.appendChild(thCurso);

//colocar a linha do cabeçalho no elemento tabela

tab.appendChild(trCabecalho);

//criar as linhas para representar o corpo desta tabela e exibir os dados que estão cadastrados no vetor

**for**(**let** i=0; i<dados.**length**; i++){

**let** trCorpo = document.createElement('tr');

**let** tdId = document.createElement('td');

**let** tdNome = document.createElement('td');

**let** tdCurso = document.createElement('td');

//associar a um objeto aluno que estás dentro do vetor que foi recebido como parâmentro

tdId.innerHTML = dados[i].id;

tdNome.innerHTML = dados[i].nome;

tdCurso.innerHTML = dados[i].curso;

// adicionar os elementos de coluna na linha

trCorpo.appendChild(tdId);

trCorpo.appendChild(tdNome);

trCorpo.appendChild(tdCurso);

// colocar o elemento da linha na tabela

tab.appendChild(trCorpo);

}

}

**function** adicionarItem(){

**let** trCorpo = document.createElement('tr');

**let** tdId = document.createElement('td');

**let** tdNome = document.createElement('td');

**let** tdCurso = document.createElement('td');

// aqui vamos pegar os dados que o usuário digitou nos inputs e colocar na tabela do HTML

tdId.innerHTML = document.querySelector('#ra').value;

tdNome.innerHTML = document.querySelector('#name').value;

tdCurso.innerHTML = document.querySelector('#course').value;

// adicionar os elementos de coluna na linha

trCorpo.appendChild(tdId);

trCorpo.appendChild(tdNome);

trCorpo.appendChild(tdCurso);

**var** tab = document.getElementById('tabela');

tab.appendChild(trCorpo);

}

// Implementando as mesmas funções com uso de Literal String (ou Template Literal)

**function** carregaTabela2(dados){

// fazer o vínculo com o elemento HTML da tabela

**let** tab = document.querySelector('#tabela');

// montar a linha do cabeçalho da tabela usando Literal String, usando o caracter CRASE

**let** html = `<tr>

<th>RA</th> <th>Nome do Aluno</th> <th>Curso</th>

</tr>`

// usar um for do tipo for each, com um controle diferente, funciona bem com objetos em listas/vetores

**for** (**let** alu of dados){

html += `<tr>

<td>${alu.id}</td>

<td>${alu.nome}</td>

<td>${alu.curso}</td>

</tr>`

}

//alert(html);

tab.innerHTML=html;

}

**function** adicionarItem2(){

// aqui vamos pegar os dados que o usuário digitou nos inputs e colocar em variáveis

**let** vRA = document.querySelector('#ra').value;

**let** vNome = document.querySelector('#name').value;

**let** vCurso = document.querySelector('#course').value;

//Somente para atualizar o vetor de objetos alunos

alunos.push({id:vRA, nome:vNome, curso:vCurso});

**let** htmlLinha = `<tr>

<td>\* ${vRA}</td>

<td>\* ${vNome}</td>

<td>\* ${vCurso}</td>

</tr>`

**let** tab = document.querySelector('#tabela');

// vamos adicionar mais uma linha na tabela

tab.innerHTML+=htmlLinha; // ou tab.innerHTML = tab.innerHTML + htmlLinha;

}

carregaTabela2(alunos);

<**html**>

<**head**>

<**style**>

\* { **font-family**: calibri;

zoom: 110%;

}

.btnExcluir {

**cursor**: pointer;

}

.tab{

**border**: 1px solid black;

**width**: 60%;

}

.coluna{

**border**: 1px solid black;

}

</**style**>

<**script** src="script.js"></**script**>

</**head**>

<**body**>

<**form** id="f\_Lista">

<**input** type="text" id="txt-texto" />

<**input** type="button" id="btn-add" value="+" />

<**br**>

<**br**>

<**br**>

<**div**>

<**table** id="grid" class="tab">

<**thead**>

<**tr**>

<**th** class="coluna">

<**input** type="checkbox" id="ckTodos" onchange=""></**th**>

<**th** class="coluna">Itens</**th**>

<**th** class="coluna">Ação</**th**>

</**tr**>

</**thead**>

<**tbody** id="tb-body">

</**tbody**>

</**table**>

</**div**>

<**br**><**br**>

<**div**>

<**input** type="button" value="excluir selecionados" id="btnExcluirSelecionados">

</**div**>

</**form**>

</**body**>

</**html**>

**var** dados = [

{id: 1, nome:'Felipe Carro'},

{id: 2, nome:'João Victor'},

{id: 3, nome:'Gustavo Tacaki'},

{id: 4, nome:'Any Velasco'},

{id: 5, nome:'Ana Carolina'},

]

**function** montarTabela(){

**let** tbody = document.querySelector('#tb-body');

**let** html = '';

/\* DATA ATTRIBUTES (ou atributos de dados) - a sintaxe é simples.

Os data-\*atributos são usados ​​para armazenar dados personalizados privados da página ou aplicativo.

Os data-\*atributos nos dão a capacidade de incorporar atributos de dados personalizados em todos os elementos HTML.

Os dados armazenados (personalizados) podem então ser usados ​​no JavaScript da página para criar uma experiência de usuário mais envolvente (sem quaisquer chamadas Ajax ou consultas de banco de dados do lado do servidor). \*/

**for** (**let** **item** of dados){

html+=`<tr>

<td><input type="checkbox" data-id="${**item**.id}"></td>

<td>${**item**.nome}</td>

<td><a class="btnExcluir" onclick="excluirItem(${**item**.id})">&#9746;</a></td>

</tr>`;

}

// associar a string html com as linhas da tabela no objeto tbody da tabela

tbody.innerHTML = html;

}

**function** adicionarItem(){

**let** nomeInput = document.querySelector('#txt-texto');

**let** novoObj = {

id: **new** Date().**getTime**() //vai pegar a hora em milesegundos e representar o id

,nome:nomeInput.value}

// para adicionar o objeto no vetor

dados.push(novoObj);

montarTabela();

nomeInput.value='';

nomeInput.focus(); // coloca o foco neste elemento

}

**function** excluirItem(idDel){

**let** vetAux = []; // vamos usar esta lista para colocar os objetos que permanecerão no vetor

**for** (**let** i=0; i<dados.**length**;i++){

**if** (dados[i].id != idDel)

vetAux.push(dados[i]);

}

//atualizar o vetor de objetos sem o elemento que foi selecionado para exclusão e montar a tabela novamente para atualizar a tela

dados = vetAux;

montarTabela();

}

**function** excluirSelecionados(){

// pegar todos os checkbox da tela e colocar eles num vetor

**let** vetCheckbox = document.querySelectorAll('[data-id]'); // usou o data attributes que foi definido no momento da criação de cada checkbox

//verificar se tem elementos selecionados

**if** (vetCheckbox.**length**>0){

// percorrer o vetor e apagar os que tem o checkbox selecionado

**for**(**let** ck of vetCheckbox){

**if** (ck.checked == true)

excluirItem(ck.dataset.id);

}

}

**else** **alert**('Não tem itens selecionados para serem excluídos!!');

}

**function** selecionarTodos(){

// pegar todos os checkbox da tela e colocar eles num vetor

**let** vetCheckbox = document.querySelectorAll('[data-id]'); // usou o data attributes

**let** ckPai = document.querySelector('#ckTodos');

**for**(**let** ck of vetCheckbox){

ck.checked = ckPai.checked; // vai colocar todos os checkbox do corpo da tabela do mesmo jeito que está o checkbox do cabeçalho da tabela

}

}

//Aqui será a parte que funcionará como o programa principal

document.addEventListener('DOMContentLoaded',

**function**(){

montarTabela();

**let** btnAdd = document.querySelector('#btn-add');

btnAdd.addEventListener('click',adicionarItem);

**let** btnExcluirSelec = document.querySelector('#btnExcluirSelecionados');

btnExcluirSelec.addEventListener('click',excluirSelecionados);

**let** ckPai = document.querySelector('#ckTodos');

ckPai.addEventListener('click',selecionarTodos);

},false);

**Informação Resumida**

<!DOCTYPE html>

<**html**>

<**head**>

<**title**>Tabela Dinâmica</**title**>

<**meta** charset="utf-8" />

<**style**>

\* { **font-family**: calibri; **font-size**: 20px; }

.myTable, .tab {

**border**: 1px solid #000;

**width**: 60%;

}

.myTable **tr**, .myTable **td**, .myTable **th**, .coluna {

**border**: 1px solid #000;

}

.btnExcluir {

**cursor**: pointer;

}

</**style**>

<**script** src="funcoes.js"></**script**>

</**head**>

<**body**>

<**h1**>Criando tabelas de forma dinâmica</**h1**>

<**hr**/>

<**form**>

<**input** type="text" id="ra" placeholder="Informe o RA"/>

<**input** type="text" id="name" placeholder="Informe o Nome"/>

<**input** type="text" id="course" placeholder="Informe o Curso"/>

<**button** type="button" onclick="adicionarItem()">Cadastrar</**button**>

</**form**>

<**hr**/>

<**div**>

<**table** class="myTable" id="tabela">

<!-- Conteúdo da tabela será adicionado aqui -->

</**table**>

</**div**>

</**body**>

</**html**>

// Dados dos alunos

**var** alunos = [

{id:1, nome:'Gustavo Tacaki',curso:'ADS'},

{id:2, nome:'Any Velasco',curso:'ADS'},

{id:3, nome:'Felipe Carro',curso:'BCC'},

{id:4, nome:'João Victor',curso:'ADS'},

];

// Função para montar a tabela (adicionar e excluir)

**function** montarTabela(){

**let** tab = document.querySelector('#tabela');

**let** html = `<tr><th>RA</th><th>Nome do Aluno</th><th>Curso</th></tr>`;

alunos.**forEach**(alu => {

html += `<tr>

<td>${alu.id}</td>

<td>${alu.nome}</td>

<td>${alu.curso}</td>

<td><a class="btnExcluir" onclick="excluirItem(${alu.id})">&#9746;</a></td>

</tr>`;

});

tab.innerHTML = html;

}

// Função para adicionar item

**function** adicionarItem(){

**let** vRA = document.querySelector('#ra').value;

**let** vNome = document.querySelector('#name').value;

**let** vCurso = document.querySelector('#course').value;

alunos.push({id:vRA, nome:vNome, curso:vCurso});

montarTabela();

}

// Função para excluir item

**function** excluirItem(idDel){

alunos = alunos.**filter**(aluno => aluno.id != idDel);

montarTabela();

}

// Event listeners para botões e carregamento inicial da tabela

document.addEventListener('DOMContentLoaded', () => {

montarTabela();

document.querySelector('#f\_Lista button').addEventListener('click', adicionarItem);

});