

Aplicação de cadastros JavaScript

Introdução:

Neste artigo aprenderemos a criar uma tela de cadastros de clientes utilizando a linguagem JavaScript.

O que é Javascript?

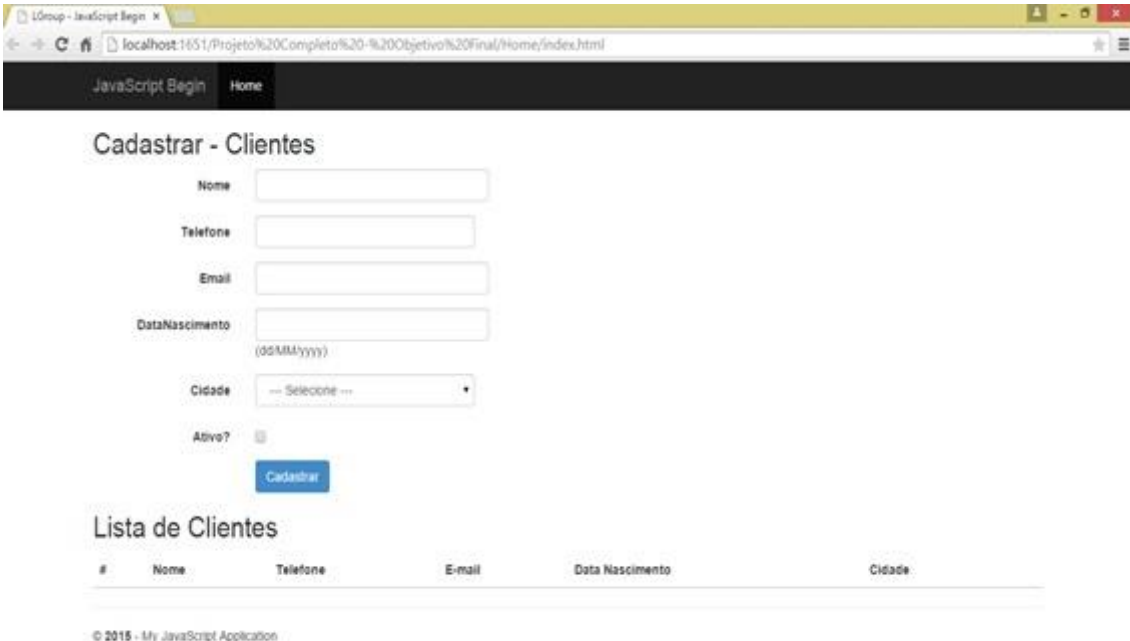
JavaScript é uma linguagem de programação baseada em scripts e padronizada pela ECMA International (associação especializada na padronização de sistemas de informação).

Foi criada por Brendan Eich (Netscape) e surgiu em 1995 como linguagem de script client-side de páginas web. É uma linguagem dinâmica, orientada a objetos e criada com sintaxe similar à linguagem C.

Apesar de conter Java no nome, a linguagem JavaScript é distinta da linguagem Java e apresenta recursos não disponibilizados em Java ou C++. Os scripts desenvolvidos em JavaScript são muito populares e amplamente integrados em páginas web devido à facilidade de interação com o Document Object Model (DOM) da página.

Uma linguagem de script tem a funcionalidade de ser executada no interior de programas ou de outra linguagem de programação. Outras linguagens usadas como script são: ActionScript, PHP, Python, VBScript, entre outras.

Sistema de exemplo



The screenshot shows a web browser window with the URL `localhost:1651/Projeto%20Completo%20-%20Objetivo%20Final/Home/index.html`. The page has a dark navigation bar with "JavaScript Begin" and "Home" links. The main content area is titled "Cadastrar - Clientes" and contains a registration form with the following fields:

- Nome:
- Telefone:
- Email:
- Data Nascimento: (dd/MM/yyyy)
- Cidade:
- Ativo?:

Below the form is a blue "Cadastrar" button. Underneath is a section titled "Lista de Clientes" with a table header:

#	Nome	Telefone	E-mail	Data Nascimento	Cidade
---	------	----------	--------	-----------------	--------

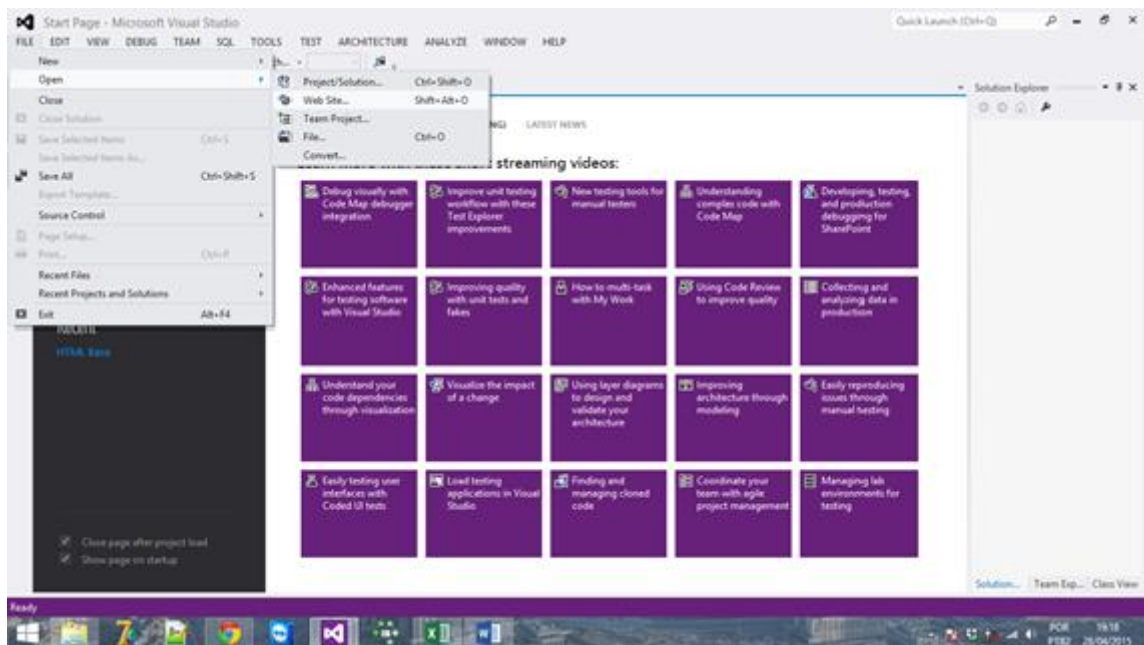
At the bottom left, there is a copyright notice: "© 2015 - My JavaScript Application".

***Iremos disponibilizar o index.html (só a tela sem as funções) e iremos fazer todas as funções em JavaScript.**

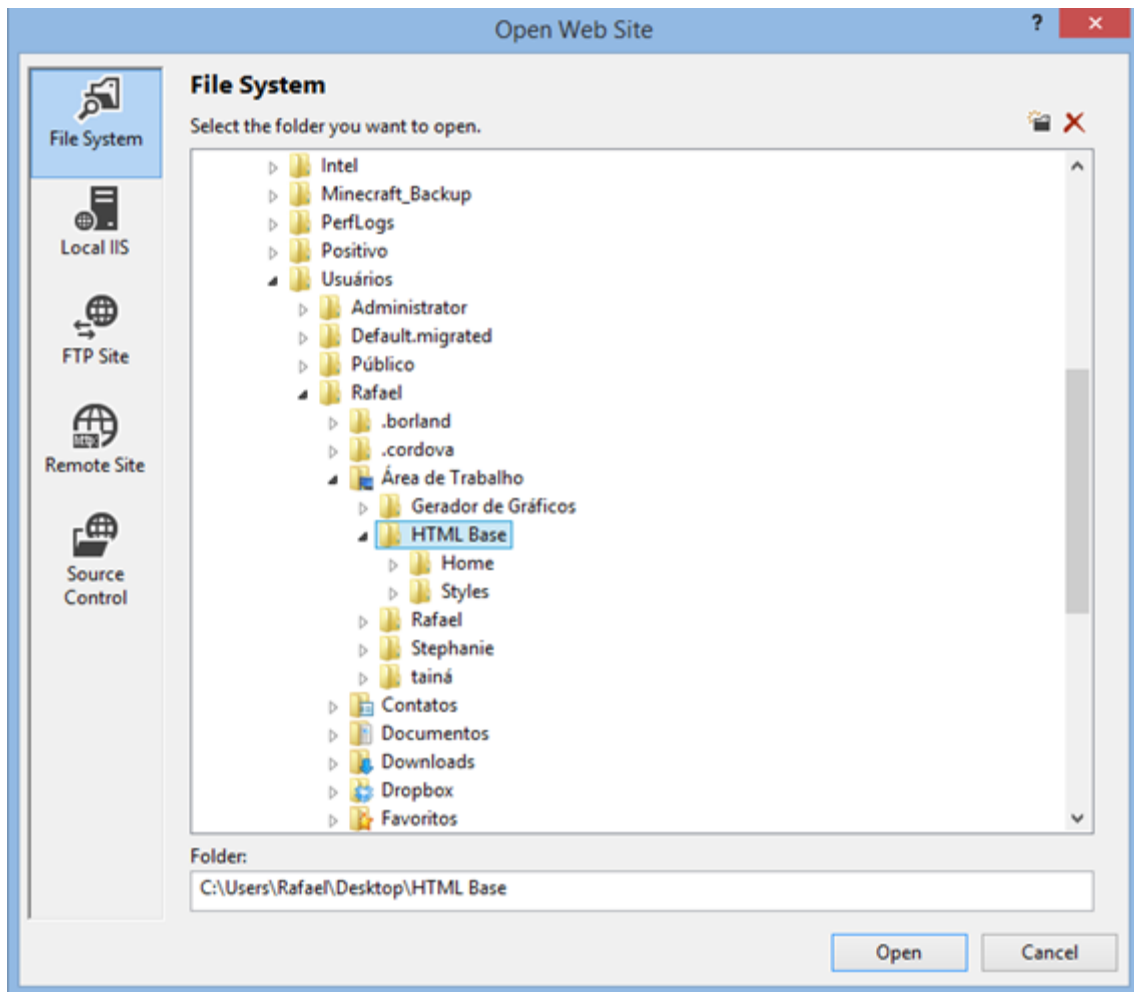
Nesta tela iremos ter os itens (Nome, Telefone, Email, Cidade, “Ativo?”) e ao clicar no botão Cadastar, será listado os dados no Grid Listar Clientes conforme print acima.

Para Criar este exemplo iremos utilizar a ferramenta Visual Studio 2012

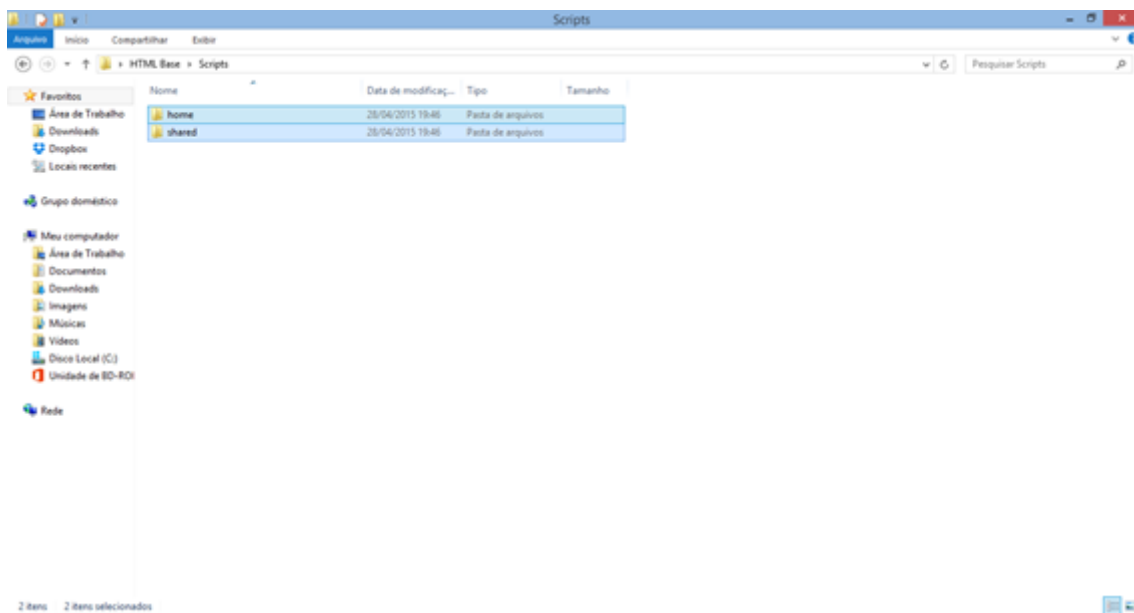
Clique no Menu File / Open/ Web Site.



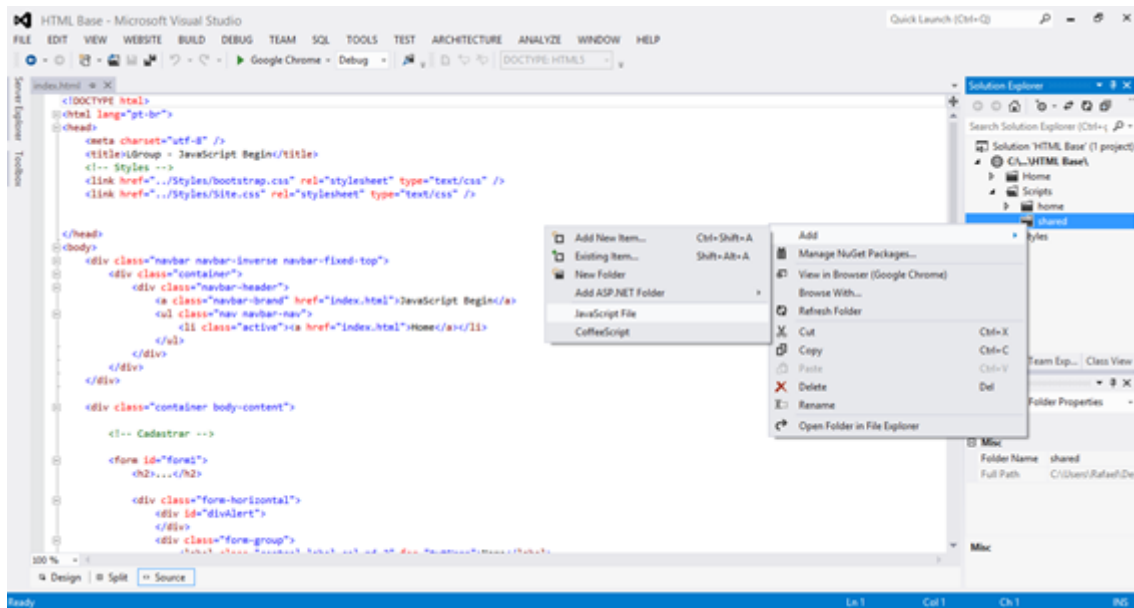
Selecione a sua pasta HTML BASE.



Em sua pasta crie a pasta Scripts e dentro dela crie duas pastas chamadas home e shared, estas pastas será onde ficará os scripts.

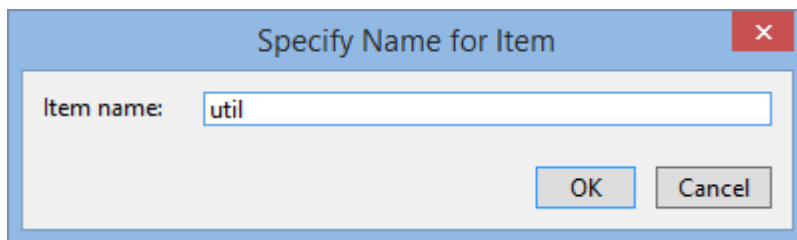


Para criar um JavaScript Selecione uma pasta.



Descrição detalhada do sistema

1º - Crie o util.js



Primeiro Js, que ficará na pasta shared, terá a função de deixar a label “dataNascimento” com o formato DATE e também criará um Array que guardará as Cidades.

Código:

```
// 1ª JS

// Essa função vai ser executar a função SetYearFooter e a função PreencheDDLCidade.
(function () {
    // Função quer será executada assim que a página for carregada!
    window.onload = ExecutarTodasFuncoes;
})();

// Função Nomeada (SetYearFooter)
function SetYearFooter() {
    // debugger: uma palavra reservada do JavaScript para debugar o código. Quando o
    Navegador estiver com o Console (f12) dele aberto, vai parar a interpretação
    // do código quando encontrar esse palavra
    debugger;
    var lblFooter = document.getElementById('lbltimeNow');
    // new Date() = Estamos instanciando a função Date() do javascript, e chamando o
    getFullYear() para retornar o ano
```

```

lblFooter.textContent = new Date().getFullYear().toString();
}

function ExecutarTodasFuncoes() {
    SetYearFooter();
    PreencheDDLcidade();
}

// Variável Global com o uma lista de objetos
var Cidades = [
    { texto: "São Paulo", valor: "SP" },
    { texto: "Rio de Janeiro", valor: "RJ" },
    { texto: "Bahia", valor: "BA" },
    { texto: "Minas Gerais", valor: "MG" },
    { texto: "Distrito Federal", valor: "DF" }
];

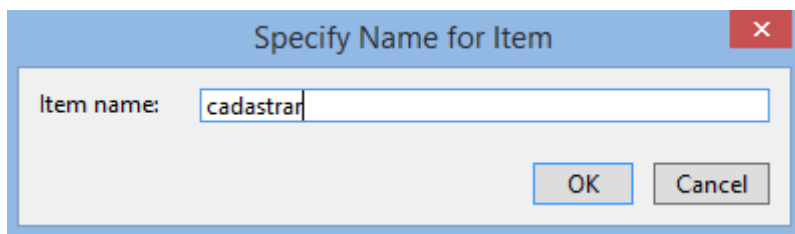
// Função para preencher o DropDownList
function PreencheDDLcidade() {
    var ddlCidade = document.getElementById('ddlCidade');

    var options = '<option value="0">--- Selecione ---</option>';
    var index = 0;
    do {
        options += '<option value="';
        options += Cidades[index].valor;
        options += '>';
        options += Cidades[index].texto;
        options += '</option>';
        index++;
    } while (Cidades.length > index);

    ddlCidade.innerHTML = options;
}

```

2º - Crie o cadastrar.js



2º Js, ficará na pasta home, terá a função:

*Validar os dados;

*Cadastrar;

*Limpar as mensagens de erro que ocorre quando não é válido os dados.

Código:

```
// 2º JS
```

```
// Aqui no JS cadastrar, vamos manipular o DOM para validar o cadastrado do cliente. E só mostrar a mensagem de sucesso se estiver tudo
```

```
// devidamente preenchido.
```

```
function Validar() {  
    var txtNome = document.getElementById('txtNome');  
    var txtTelefone = document.getElementById('txtTelefone');  
    var txtEmail = document.getElementById('txtEmail');  
    var txtDataNascimento = document.getElementById('txtDataNascimento');  
    var ddlCidade = document.getElementById('ddlCidade');  
  
    var isValidado = true;  
  
    // === : Igualdade de tipo e valor. Essa comparação não faz conversão de valores para saber o valor. Boa pratica utiliza-lo!  
    // == : Igualdade de valor, somente. Ele converte um dos valores com base no outro, e assim faz a verificação dos valores.  
    // http://stackoverflow.com/questions/359494/does-it-matter-which-equals-operator-vs-i-use-in-javascript-comparisons  
  
    if (txtNome.value === '') {  
        isValidado = false;  
        txtNome.classList.add('errorInput');  
    }  
  
    if (txtTelefone.value === '') {  
        isValidado = false;  
        txtTelefone.classList.add('errorInput');  
    }  
  
    if (txtEmail.value === '') {  
        isValidado = false;  
        txtEmail.classList.add('errorInput');  
    }  
  
    if (txtDataNascimento.value === '') {  
        isValidado = false;  
        txtDataNascimento.classList.add('errorInput');  
    }  
  
    if (ddlCidade.value === '0') {  
        isValidado = false;  
        ddlCidade.classList.add('errorInput');  
    }  
  
    if (isValidado === false) {  
        var divAlerta = document.getElementById('divAlert');  
        var mensagem = '<strong>Atenção: </strong>Preencha os campos em destaque abaixo.'
```

```

    divAlerta.innerHTML = mensagem;
    divAlerta.classList.add('alert');
    divAlerta.classList.add('alert-danger');
}

return isValidado;
}

function LimparMensagensErros() {
    var listClassErrorInput = document.getElementsByClassName('errorInput');

    while (listClassErrorInput.length > 0) {
        var index = (listClassErrorInput.length - 1);
        listClassErrorInput[index].classList.remove('errorInput');
    }

    var divAlerta = document.getElementById('divAlert');
    if (divAlerta.classList.contains('alert') || divAlerta.classList.contains('alert-danger')) {
        divAlerta.classList.remove('alert');
        divAlerta.classList.remove('alert-danger');
        divAlerta.innerHTML = '';
    }
}

function onCadastrarClick() {
    LimparMensagensErros();

    if (Validar()) {
        var txtNome = document.getElementById('txtNome');
        var txtTelefone = document.getElementById('txtTelefone');
        var txtEmail = document.getElementById('txtEmail');
        var txtDataNascimento = document.getElementById('txtDataNascimento');
        var ddlCidade = document.getElementById('ddlCidade');
        var ckbAtivo = document.getElementById('ckbAtivo');

        // Declarando um objeto
        var Cliente = {}; // === new Object();

        Cliente.nome = txtNome.value.trim();
        Cliente.telefone = txtTelefone.value.trim();
        Cliente.email = txtEmail.value.trim();
        Cliente.dataNascimento = txtDataNascimento.value.trim();
        Cliente.cidade = ddlCidade.value;
        Cliente.isAtivo = ckbAtivo.checked;

        // *** Após criar o 3º Js (index.js) vamos adicionar a linha abaixo:
        // Adicionando Cliente na lista de Cliente, só que... esse JS esta acima do index.js. Então,
        // vamos organizar a ordem...
        // ** Falar sobre a forma de leitura do Navegador e a ordem interfere um com o outro...
        AdicionarCliente(Cliente);

        MensagemSucesso();
    }
}

```

```

    LimparCampos();

    // Depois de criar o 4º JS:
    ListarCliente();
    return true;
}

return false;
}

function MensagemSucesso() {
    var divAlerta = document.getElementById('divAlert');
    var mensagem = '<strong>Sucesso: </strong>Cliente cadastrado com sucesso!'
    divAlerta.innerHTML = mensagem;

    // Atribuindo a lista de classes da div, a nova classes alert
    divAlerta.classList.add('alert');
    divAlerta.classList.add('alert-success');
    DesaparecerMensagemSucesso();
}

// Função para LimparMensagem da Div Alerta
function LimparMensagem() {
    var divAlerta = document.getElementById('divAlert');
    divAlerta.innerHTML = "";

    // Remove da lista de classes da div, as classes abaixo
    divAlerta.classList.remove('alert');
    divAlerta.classList.remove('alert-success');
}

// Função para LimparCampos
function LimparCampos() {
    // Cada HTML tem sua particularidade para manipular os dados dele
    document.getElementById('txtNome').value = "";
    document.getElementById('txtTelefone').value = "";
    document.getElementById('txtEmail').value = "";
    document.getElementById('txtDataNascimento').value = "";
    document.getElementById('ddlCidade').selectedIndex = 0;
    document.getElementById('ckbAtivo').checked = false;
    document.getElementById('lblValorDDLcidade').textContent = "";
    document.getElementById('lblValorKeyPressTelefone').textContent = "";
}

function DesaparecerMensagemSucesso() {
    // setTimeout é uma função JavaScript que executa uma função após alguns minutos...
    window.setTimeout(
        // Criando uma function anônima
        // Essa função determinará o tempo em que a mensagem desaparecerá...
        function () {
            LimparMensagem();
        }, 5000);
}

```



```

}

// Função executada no evento Blur do Text da Data de Nascimento
function onValidaDataNascimentoBlur(domHtml) {
  if (domHtml.value !== "") {
    // Validando se o formato da Data esta correta
    var arrayDiaMesAno = domHtml.value.split('/');
    if (arrayDiaMesAno.length == 3) {
      var dia = parseInt(arrayDiaMesAno[0]);
      var mes = parseInt(arrayDiaMesAno[1]);
      var ano = parseInt(arrayDiaMesAno[2]);

      // Validar se o que foi digitado só tem números
      if (isNaN(dia) || isNaN(mes) || isNaN(ano)) {
        alert('Por favor, digite uma data válida. ');
        domHtml.value = "";
      }

      // Validar se a data esta correta...
      var mesString = (mes.toString().length === 1 ? "0" + mes.toString() : mes.toString());
      var diaString = (dia.toString().length === 1 ? "0" + dia.toString() : dia.toString());
      var dateString = diaString + "/" + mesString + "/" + ano;
      var data = new Date(ano + "-" + mesString + "-" + diaString + "T03:24:00");

      if (dateString !== data.toLocaleDateString()) {
        alert('Por favor, digite uma data válida. ');
        domHtml.value = "";
      }
    }
  }
  else {
    alert('Por favor, adicionar uma data. ');
    domHtml.value = "";
  }
}

```

```

function onValidaEmailBlur(domHtml) {
  if (domHtml.value !== "") {
    // indexOf é uma função JavaScript disponível em valores do tipo string. Ele busca na
    string o char indicado no parâmetro de entrada.
    var resultadoIndexOf = domHtml.value.indexOf('@');
    if (resultadoIndexOf === -1) {
      alert('Por favor, digite um e-mail válido. ');
      domHtml.value = "";
    }
  }
}

```

```

// Função executada no evento Change do DropDownList (select)
function onShowValorChange(domHtml) {

```

```

var lblValorDDLCidade = document.getElementById('lblValorDDLCidade');
var textDDL = "";
if (domHtml.selectedIndex > 0) {
    textDDL = domHtml.options[domHtml.selectedIndex].text;
}
lblValorDDLCidade.textContent = textDDL;
}

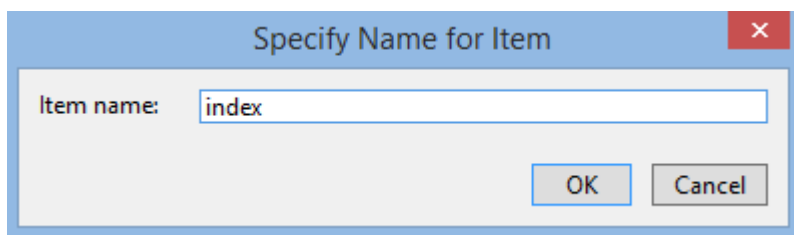
// Função que é executado no evento KeyDown
function onValidarTelefoneKeyDown(evt) {
    // Em todos os eventos, por tras dos panos, é setado na variável "event" o evento
    assionado...

    // If Ternário para verificar no objeto event que o JavaScript retorna, tem o código da tecla
    na propriedade keyCode ou na charCode
    var tecla = (evt.keyCode ? evt.keyCode : evt.charCode);
    var lblValorKeyPressTelefone = document.getElementById('lblValorKeyPressTelefone');

    // Teclar com o código entre 48 e 59 ou entre 96 e 105, são numéricos
    if ((tecla >= 48 && tecla <= 59) || (tecla >= 96 && tecla <= 105)) {
        lblValorKeyPressTelefone.textContent = 'Numérico';
    }
    else {
        lblValorKeyPressTelefone.textContent = "";
    }
}
}

```

3º - Crie o index.js



3º Js, também ficará na pasta home, este JavaScript apenas terá a função de deixar todos os clientes cadastrados guardados em um Array.

Código:

```

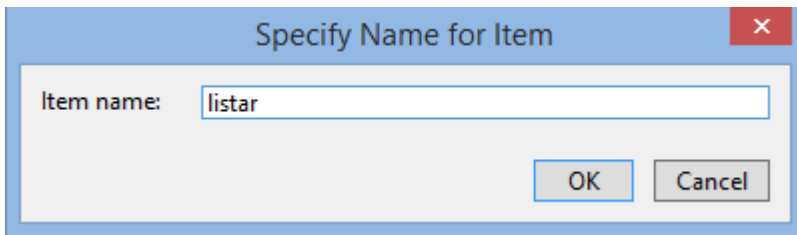
// 3º JS

// Nesse JS vamos deixar todos os Clientes que cadastrados de um Array de Clientes (Lista)
var listClientes = [];

function AdicionarCliente(cliente) {
    listClientes.push(cliente);
}

```

4º - Crie o listar.js



4º e último, ficará na pasta home, terá a função de listar os clientes cadastrados e também terá uma função de excluir os clientes cadastrados.

Código:

```
// 4º JS
```

```
// Nesse arquivo JS vamos listar todos os clientes que estão no Objeto listClientes, que esta no arquivo index.js.
```

```
function ListarCliente() {  
    // Fazendo um IF Ternário: (Se o listClientes !== undefined ? Então retorna listClientes,  
    senão :) retorna um array vazio)  
    var clientes = (listClientes !== undefined ? listClientes : []);  
  
    var tagsTbody = "";  
  
    for (var i = 0; i < clientes.length; i++) {  
        tagsTbody += '<tr onMouseOver="javascript: onClienteAtivoMouseOver(this, ' +  
        clientes[i].isAtivo + ');">  
  
            // Index  
            tagsTbody += '<td>  
            tagsTbody += (i + 1).toString();  
            tagsTbody += '</td>'  
  
            // Nome  
            tagsTbody += '<td>  
            tagsTbody += clientes[i].nome;  
            tagsTbody += '</td>'  
  
            // Telefone  
            tagsTbody += '<td>  
            tagsTbody += clientes[i].telefone;  
            tagsTbody += '</td>'  
  
            // Email  
            tagsTbody += '<td>  
            tagsTbody += clientes[i].email;  
            tagsTbody += '</td>'
```

```

// Data Nascimento
tagsTbody += '<td>'
tagsTbody += clientes[i].dataNascimento;
tagsTbody += '</td>'

// Cidade
tagsTbody += '<td>'
tagsTbody += clientes[i].cidade;
tagsTbody += '</td>'

// Ações que poderemos fazer
tagsTbody += '<td>'
tagsTbody += "<a href='javascript:void(0)' onclick=\"javascript: ExcluirCliente(" +
i.toString() + ");\">Excluir</a>";
tagsTbody += '</td>'
tagsTbody += '</tr>'
}

var tbodyClientes = document.getElementById('tbodyListaCliente');
tbodyClientes.innerHTML = tagsTbody;
}

// Função para Excluir o cliente dentro a listClientes (variável Global dentro do arquivo index.js
function ExcluirCliente(index) {
    if (listClientes !== undefined && listClientes.length > index) {
        // confirm : É uma função do JavaScript, parecida com o alert mas, é retorna um
        booleano! Se o usuário clicar em "Sim", retorna um true
        // e se clicar em "Não" retorna false.
        if (confirm("Tem realmente certeza que seja excluir esse cliente?")) {
            // splice é uma função no JavaScript, disponível no Objeto Array. O primeiro parâmetro
            é o index inicial do array, e o segundo parâmetro
            // é quantidade que será removido do Array.
            listClientes.splice(index, 1);
        }
    }
}

// Listamos os cliente novamente
ListarCliente();
}

// Função do MouseOver
function onClienteAtivoMouseOver(domHtml, isAtivo) {
    if (isAtivo === false) {
        domHtml.style.border = 'red 1px dashed';
    }
}
}

```

Por Hoje é só. Até mais pessoal.