# **🎓 APRENDA HTML DO ZERO**

## **Criando um Sistema de Recepção Passo a Passo**

# **📚 INTRODUÇÃO - O QUE VAMOS APRENDER**

Neste tutorial, você vai aprender a criar páginas web do ZERO! Vamos construir juntos um sistema simples e depois evoluir para algo profissional.

## **O que é HTML?**

**HTML** = HyperText Markup Language (Linguagem de Marcação)

É a "estrutura" de qualquer site

Como o "esqueleto" de uma casa

## **O que é CSS?**

**CSS** = Cascading Style Sheets (Folhas de Estilo)

É a "aparência" do site

Como a "pintura e decoração" da casa

## **O que é JavaScript?**

**JavaScript** = Linguagem de programação

É a "inteligência" do site

Como a "eletricidade e automação" da casa

# **🛠️ PREPARANDO O AMBIENTE**

## **O que você precisa:**

**Um computador** (Windows, Mac ou Linux)

**Bloco de Notas** (já vem no Windows) ou **Notepad++** (melhor)

**Um navegador** (Chrome, Firefox, Edge)

**Vontade de aprender!** 💪

## **Criando sua pasta de trabalho:**

Na Área de Trabalho, crie uma pasta chamada **"Meu Primeiro Site"**

Todos os arquivos ficarão aqui

# **📝 LIÇÃO 1: SEU PRIMEIRO HTML**

## **Passo 1: Criando o arquivo**

Abra o **Bloco de Notas**

**Salve imediatamente** como:

Nome: index.html

Tipo: **Todos os arquivos**

Local: Pasta "Meu Primeiro Site"

## **Passo 2: Estrutura básica**

Digite exatamente isto:

<!DOCTYPE html>  
<html>  
<head>  
 <title>Meu Primeiro Site</title>  
</head>  
<body>  
 <h1>Olá, Mundo!</h1>  
 <p>Este é meu primeiro site!</p>  
</body>  
</html>

## **Explicando cada linha:**

<!DOCTYPE html>

↑ **Isso diz:** "Este documento é HTML5" (versão mais nova)

<html>

↑ **Início do HTML** - Tudo fica dentro desta tag

<head>

↑ **Cabeçalho** - Informações que não aparecem na página

<title>Meu Primeiro Site</title>

↑ **Título** - Aparece na aba do navegador

</head>

↑ **Fim do cabeçalho** - Toda tag que abre, deve fechar

<body>

↑ **Corpo** - Tudo que aparece na página fica aqui

<h1>Olá, Mundo!</h1>

↑ **Título grande** - H1 = Heading 1 (Título nível 1)

<p>Este é meu primeiro site!</p>

↑ **Parágrafo** - P = Paragraph

</body>  
</html>

↑ **Fechamento** - Fecha body e html

## **Testando:**

**Salve** o arquivo (Ctrl+S)

**Dê duplo clique** no arquivo index.html

**Veja no navegador!** 🎉

# **🎨 LIÇÃO 2: ADICIONANDO ESTILO (CSS)**

## **Passo 1: CSS interno**

Vamos deixar bonito! Adicione isto dentro do <head>:

<!DOCTYPE html>  
<html>  
<head>  
 <title>Meu Primeiro Site</title>  
 <style>  
 body {  
 background-color: #f0f0f0;  
 font-family: Arial;  
 margin: 20px;  
 }  
   
 h1 {  
 color: #2563eb;  
 text-align: center;  
 }  
   
 p {  
 color: #333;  
 font-size: 18px;  
 }  
 </style>  
</head>  
<body>  
 <h1>Olá, Mundo!</h1>  
 <p>Este é meu primeiro site!</p>  
</body>  
</html>

## **Explicando o CSS:**

body {  
 background-color: #f0f0f0;  
 font-family: Arial;  
 margin: 20px;  
}

↑ **Estilo do body:**

background-color = cor de fundo (cinza claro)

font-family = tipo de letra

margin = espaço ao redor

h1 {  
 color: #2563eb;  
 text-align: center;  
}

↑ **Estilo do h1:**

color = cor do texto (azul)

text-align = alinhamento (centro)

# **🏗️ LIÇÃO 3: CRIANDO UM FORMULÁRIO**

Vamos criar um formulário de cadastro simples:

<!DOCTYPE html>  
<html>  
<head>  
 <title>Sistema de Cadastro</title>  
 <style>  
 body {  
 background-color: #f0f0f0;  
 font-family: Arial;  
 margin: 20px;  
 }  
   
 .container {  
 background-color: white;  
 padding: 30px;  
 border-radius: 10px;  
 max-width: 500px;  
 margin: 0 auto;  
 box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
 }  
   
 h1 {  
 color: #2563eb;  
 text-align: center;  
 }  
   
 label {  
 display: block;  
 margin-top: 15px;  
 font-weight: bold;  
 }  
   
 input {  
 width: 100%;  
 padding: 10px;  
 margin-top: 5px;  
 border: 1px solid #ddd;  
 border-radius: 5px;  
 box-sizing: border-box;  
 }  
   
 button {  
 background-color: #2563eb;  
 color: white;  
 padding: 12px 30px;  
 border: none;  
 border-radius: 5px;  
 margin-top: 20px;  
 cursor: pointer;  
 font-size: 16px;  
 }  
   
 button:hover {  
 background-color: #1d4ed8;  
 }  
 </style>  
</head>  
<body>  
 <div class="container">  
 <h1>Cadastro de Pessoa</h1>  
   
 <form>  
 <label for="nome">Nome Completo:</label>  
 <input type="text" id="nome" placeholder="Digite seu nome">  
   
 <label for="email">E-mail:</label>  
 <input type="email" id="email" [placeholder="seu@email.com](mailto:placeholder="seu@email.com)">  
   
 <label for="telefone">Telefone:</label>  
 <input type="tel" id="telefone" placeholder="(11) 99999-9999">  
   
 <button type="button" onclick="cadastrar()">Cadastrar</button>  
 </form>  
   
 <div id="mensagem"></div>  
 </div>  
</body>  
</html>

## **Novos conceitos:**

### **1. Classes CSS:**

.container {  
 /\* estilos \*/  
}

↑ O ponto (.) significa "classe" - pode usar em vários elementos

### **2. Formulário:**

<form>  
 <!-- campos aqui -->  
</form>

↑ Agrupa campos de entrada

### **3. Input (entrada):**

<input type="text" id="nome" placeholder="Digite seu nome">

↑ Campo onde usuário digita:

type="text" = texto normal

type="email" = e-mail

type="tel" = telefone

id = identificador único

placeholder = dica que aparece

### **4. Efeito hover:**

button:hover {  
 background-color: #1d4ed8;  
}

↑ Muda cor quando passa o mouse

# **💡 LIÇÃO 4: ADICIONANDO INTELIGÊNCIA (JavaScript)**

Agora vamos fazer o formulário funcionar! Adicione antes do </body>:

<script>  
 function cadastrar() {  
 // Pegar os valores digitados  
 var nome = document.getElementById('nome').value;  
 var email = document.getElementById('email').value;  
 var telefone = document.getElementById('telefone').value;  
   
 // Verificar se preencheu tudo  
 if (nome === '' || email === '' || telefone === '') {  
 alert('Por favor, preencha todos os campos!');  
 return;  
 }  
   
 // Mostrar mensagem de sucesso  
 document.getElementById('mensagem').innerHTML =   
 '<p style="color: green; text-align: center; margin-top: 20px;">' +  
 '✅ Cadastro realizado com sucesso!<br>' +  
 'Nome: ' + nome + '<br>' +  
 'E-mail: ' + email + '<br>' +  
 'Telefone: ' + telefone +  
 '</p>';  
   
 // Limpar os campos  
 document.getElementById('nome').value = '';  
 document.getElementById('email').value = '';  
 document.getElementById('telefone').value = '';  
 }  
</script>

## **Explicando o JavaScript:**

### **1. Função:**

function cadastrar() {  
 // código aqui  
}

↑ Cria uma ação chamada "cadastrar"

### **2. Pegar valores:**

var nome = document.getElementById('nome').value;

↑ Pega o que foi digitado no campo "nome"

### **3. Verificar (if):**

if (nome === '') {  
 alert('Preencha o nome!');  
}

↑ SE o nome estiver vazio, mostra aviso

### **4. Mostrar na tela:**

document.getElementById('mensagem').innerHTML = '<p>Texto</p>';

↑ Coloca HTML dentro do elemento "mensagem"

# **📊 LIÇÃO 5: SALVANDO DADOS (localStorage)**

Vamos fazer os dados persistirem! Substitua o script por:

<script>  
 // Array para guardar pessoas  
 var pessoas = [];  
   
 // Carregar dados salvos  
 window.onload = function() {  
 var pessoasSalvas = localStorage.getItem('pessoas');  
 if (pessoasSalvas) {  
 pessoas = JSON.parse(pessoasSalvas);  
 mostrarPessoas();  
 }  
 }  
   
 function cadastrar() {  
 // Pegar valores  
 var nome = document.getElementById('nome').value;  
 var email = document.getElementById('email').value;  
 var telefone = document.getElementById('telefone').value;  
   
 // Verificar  
 if (nome === '' || email === '' || telefone === '') {  
 alert('Por favor, preencha todos os campos!');  
 return;  
 }  
   
 // Criar objeto pessoa  
 var novaPessoa = {  
 id: Date.now(),  
 nome: nome,  
 email: email,  
 telefone: telefone  
 };  
   
 // Adicionar ao array  
 pessoas.push(novaPessoa);  
   
 // Salvar no localStorage  
 localStorage.setItem('pessoas', JSON.stringify(pessoas));  
   
 // Mostrar mensagem  
 alert('Cadastro realizado com sucesso!');  
   
 // Limpar campos  
 document.getElementById('nome').value = '';  
 document.getElementById('email').value = '';  
 document.getElementById('telefone').value = '';  
   
 // Atualizar lista  
 mostrarPessoas();  
 }  
   
 function mostrarPessoas() {  
 var lista = '<h2>Pessoas Cadastradas:</h2>';  
   
 if (pessoas.length === 0) {  
 lista += '<p>Nenhuma pessoa cadastrada ainda.</p>';  
 } else {  
 lista += '<ul>';  
 for (var i = 0; i < pessoas.length; i++) {  
 lista += '<li>' +   
 pessoas[i].nome + ' - ' +   
 pessoas[i].email + ' - ' +   
 pessoas[i].telefone +  
 '</li>';  
 }  
 lista += '</ul>';  
 }  
   
 document.getElementById('mensagem').innerHTML = lista;  
 }  
</script>

## **Novos conceitos:**

### **1. Array (lista):**

var pessoas = [];

↑ Lista que guarda várias pessoas

### **2. Objeto:**

var novaPessoa = {  
 id: Date.now(),  
 nome: nome,  
 email: email  
};

↑ Agrupa informações relacionadas

### **3. localStorage:**

localStorage.setItem('pessoas', JSON.stringify(pessoas));

↑ Salva no navegador (permanente)

### **4. Loop (repetição):**

for (var i = 0; i < pessoas.length; i++) {  
 // repete para cada pessoa  
}

↑ Percorre toda a lista

# **🎯 LIÇÃO 6: CRIANDO ABAS**

Vamos criar um sistema com múltiplas páginas:

<!DOCTYPE html>  
<html>  
<head>  
 <title>Sistema Completo</title>  
 <style>  
 body {  
 background-color: #f0f0f0;  
 font-family: Arial;  
 margin: 0;  
 padding: 20px;  
 }  
   
 .tabs {  
 display: flex;  
 background-color: white;  
 border-radius: 10px 10px 0 0;  
 overflow: hidden;  
 }  
   
 .tab {  
 padding: 15px 30px;  
 background-color: #e0e0e0;  
 border: none;  
 cursor: pointer;  
 font-size: 16px;  
 transition: background-color 0.3s;  
 }  
   
 .tab:hover {  
 background-color: #d0d0d0;  
 }  
   
 .tab.active {  
 background-color: #2563eb;  
 color: white;  
 }  
   
 .content {  
 background-color: white;  
 padding: 30px;  
 border-radius: 0 0 10px 10px;  
 min-height: 400px;  
 }  
   
 .tab-panel {  
 display: none;  
 }  
   
 .tab-panel.active {  
 display: block;  
 }  
 </style>  
</head>  
<body>  
 <div class="tabs">  
 <button class="tab active" onclick="mostrarAba('inicio')">Início</button>  
 <button class="tab" onclick="mostrarAba('cadastro')">Cadastro</button>  
 <button class="tab" onclick="mostrarAba('lista')">Lista</button>  
 </div>  
   
 <div class="content">  
 <div id="inicio" class="tab-panel active">  
 <h1>Bem-vindo ao Sistema!</h1>  
 <p>Use as abas acima para navegar.</p>  
 </div>  
   
 <div id="cadastro" class="tab-panel">  
 <h1>Cadastrar Pessoa</h1>  
 <!-- Formulário aqui -->  
 </div>  
   
 <div id="lista" class="tab-panel">  
 <h1>Lista de Pessoas</h1>  
 <!-- Lista aqui -->  
 </div>  
 </div>  
   
 <script>  
 function mostrarAba(aba) {  
 // Esconder todas as abas  
 var paineis = document.getElementsByClassName('tab-panel');  
 for (var i = 0; i < paineis.length; i++) {  
 paineis[i].classList.remove('active');  
 }  
   
 // Remover active de todos os botões  
 var botoes = document.getElementsByClassName('tab');  
 for (var i = 0; i < botoes.length; i++) {  
 botoes[i].classList.remove('active');  
 }  
   
 // Mostrar aba selecionada  
 document.getElementById(aba).classList.add('active');  
   
 // Marcar botão como ativo  
 event.target.classList.add('active');  
 }  
 </script>  
</body>  
</html>

# **🚀 LIÇÃO 7: PROJETO FINAL - MINI SISTEMA**

Agora vamos juntar tudo em um sistema funcional:

<!DOCTYPE html>  
<html lang="pt-BR">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Sistema de Recepção - Versão Simples</title>  
 <style>  
 /\* Reset básico \*/  
 \* {  
 margin: 0;  
 padding: 0;  
 box-sizing: border-box;  
 }  
   
 /\* Corpo da página \*/  
 body {  
 font-family: Arial, sans-serif;  
 background-color: #f5f5f5;  
 color: #333;  
 line-height: 1.6;  
 }  
   
 /\* Cabeçalho \*/  
 header {  
 background-color: #2563eb;  
 color: white;  
 padding: 20px;  
 text-align: center;  
 box-shadow: 0 2px 5px rgba(0,0,0,0.1);  
 }  
   
 /\* Container principal \*/  
 .container {  
 max-width: 1200px;  
 margin: 20px auto;  
 padding: 0 20px;  
 }  
   
 /\* Sistema de abas \*/  
 .tabs {  
 display: flex;  
 background-color: white;  
 border-radius: 10px 10px 0 0;  
 box-shadow: 0 2px 5px rgba(0,0,0,0.1);  
 }  
   
 .tab {  
 flex: 1;  
 padding: 15px;  
 background-color: #e0e0e0;  
 border: none;  
 cursor: pointer;  
 font-size: 16px;  
 font-weight: bold;  
 transition: all 0.3s;  
 }  
   
 .tab:hover {  
 background-color: #d0d0d0;  
 }  
   
 .tab.active {  
 background-color: white;  
 color: #2563eb;  
 }  
   
 /\* Conteúdo das abas \*/  
 .content {  
 background-color: white;  
 padding: 30px;  
 border-radius: 0 0 10px 10px;  
 box-shadow: 0 2px 5px rgba(0,0,0,0.1);  
 min-height: 500px;  
 }  
   
 .tab-panel {  
 display: none;  
 animation: fadeIn 0.5s;  
 }  
   
 .tab-panel.active {  
 display: block;  
 }  
   
 @keyframes fadeIn {  
 from { opacity: 0; }  
 to { opacity: 1; }  
 }  
   
 /\* Formulários \*/  
 .form-group {  
 margin-bottom: 20px;  
 }  
   
 label {  
 display: block;  
 margin-bottom: 5px;  
 font-weight: bold;  
 color: #555;  
 }  
   
 input, select, textarea {  
 width: 100%;  
 padding: 10px;  
 border: 2px solid #ddd;  
 border-radius: 5px;  
 font-size: 16px;  
 transition: border-color 0.3s;  
 }  
   
 input:focus, select:focus, textarea:focus {  
 outline: none;  
 border-color: #2563eb;  
 }  
   
 /\* Botões \*/  
 .btn {  
 padding: 12px 30px;  
 border: none;  
 border-radius: 5px;  
 font-size: 16px;  
 font-weight: bold;  
 cursor: pointer;  
 transition: all 0.3s;  
 margin-right: 10px;  
 }  
   
 .btn-primary {  
 background-color: #2563eb;  
 color: white;  
 }  
   
 .btn-primary:hover {  
 background-color: #1d4ed8;  
 transform: translateY(-2px);  
 box-shadow: 0 5px 15px rgba(37, 99, 235, 0.3);  
 }  
   
 .btn-danger {  
 background-color: #ef4444;  
 color: white;  
 }  
   
 .btn-danger:hover {  
 background-color: #dc2626;  
 }  
   
 /\* Cartões \*/  
 .card {  
 background-color: #f9f9f9;  
 border: 1px solid #eee;  
 border-radius: 10px;  
 padding: 20px;  
 margin-bottom: 20px;  
 transition: all 0.3s;  
 }  
   
 .card:hover {  
 box-shadow: 0 5px 15px rgba(0,0,0,0.1);  
 transform: translateY(-2px);  
 }  
   
 /\* Lista de pessoas \*/  
 .person-list {  
 display: grid;  
 grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));  
 gap: 20px;  
 margin-top: 20px;  
 }  
   
 .person-card {  
 background-color: white;  
 border: 2px solid #e0e0e0;  
 border-radius: 10px;  
 padding: 20px;  
 transition: all 0.3s;  
 }  
   
 .person-card:hover {  
 border-color: #2563eb;  
 box-shadow: 0 5px 15px rgba(37, 99, 235, 0.2);  
 }  
   
 /\* Mensagens \*/  
 .alert {  
 padding: 15px;  
 border-radius: 5px;  
 margin-bottom: 20px;  
 display: none;  
 }  
   
 .alert.success {  
 background-color: #d4edda;  
 color: #155724;  
 border: 1px solid #c3e6cb;  
 }  
   
 .alert.error {  
 background-color: #f8d7da;  
 color: #721c24;  
 border: 1px solid #f5c6cb;  
 }  
   
 /\* Estatísticas \*/  
 .stats {  
 display: grid;  
 grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));  
 gap: 20px;  
 margin-bottom: 30px;  
 }  
   
 .stat-box {  
 background: linear-gradient(135deg, #2563eb 0%, #1d4ed8 100%);  
 color: white;  
 padding: 20px;  
 border-radius: 10px;  
 text-align: center;  
 box-shadow: 0 5px 15px rgba(37, 99, 235, 0.3);  
 }  
   
 .stat-box h3 {  
 font-size: 36px;  
 margin-bottom: 10px;  
 }  
   
 .stat-box p {  
 font-size: 14px;  
 opacity: 0.9;  
 }  
   
 /\* Rodapé \*/  
 footer {  
 text-align: center;  
 padding: 20px;  
 margin-top: 50px;  
 color: #666;  
 }  
 </style>  
</head>  
<body>  
 <!-- Cabeçalho -->  
 <header>  
 <h1>🏢 Sistema de Recepção</h1>  
 <p>Versão Simplificada para Aprendizado</p>  
 </header>  
   
 <!-- Container principal -->  
 <div class="container">  
 <!-- Sistema de abas -->  
 <div class="tabs">  
 <button class="tab active" onclick="mostrarAba('dashboard')">  
 📊 Dashboard  
 </button>  
 <button class="tab" onclick="mostrarAba('cadastro')">  
 👤 Cadastrar  
 </button>  
 <button class="tab" onclick="mostrarAba('agendamento')">  
 📅 Agendar  
 </button>  
 <button class="tab" onclick="mostrarAba('lista')">  
 📋 Lista  
 </button>  
 </div>  
   
 <!-- Conteúdo -->  
 <div class="content">  
 <!-- Mensagem de alerta -->  
 <div id="alerta" class="alert"></div>  
   
 <!-- Dashboard -->  
 <div id="dashboard" class="tab-panel active">  
 <h2>Dashboard - Visão Geral</h2>  
   
 <div class="stats">  
 <div class="stat-box">  
 <h3 id="total-pessoas">0</h3>  
 <p>Pessoas Cadastradas</p>  
 </div>  
 <div class="stat-box" style="background: linear-gradient(135deg, #22c55e 0%, #16a34a 100%);">  
 <h3 id="total-agendamentos">0</h3>  
 <p>Agendamentos</p>  
 </div>  
 <div class="stat-box" style="background: linear-gradient(135deg, #f59e0b 0%, #d97706 100%);">  
 <h3 id="agendamentos-hoje">0</h3>  
 <p>Agendamentos Hoje</p>  
 </div>  
 </div>  
   
 <div class="card">  
 <h3>📅 Próximos Agendamentos</h3>  
 <div id="proximos-agendamentos">  
 <p>Nenhum agendamento próximo.</p>  
 </div>  
 </div>  
 </div>  
   
 <!-- Cadastro -->  
 <div id="cadastro" class="tab-panel">  
 <h2>Cadastrar Pessoa</h2>  
   
 <form id="form-cadastro" onsubmit="cadastrarPessoa(event)">  
 <div class="form-group">  
 <label for="nome">Nome Completo: \*</label>  
 <input type="text" id="nome" required>  
 </div>  
   
 <div class="form-group">  
 <label for="email">E-mail: \*</label>  
 <input type="email" id="email" required>  
 </div>  
   
 <div class="form-group">  
 <label for="telefone">Telefone: \*</label>  
 <input type="tel" id="telefone" placeholder="(11) 99999-9999" required>  
 </div>  
   
 <div class="form-group">  
 <label for="tipo">Tipo:</label>  
 <select id="tipo">  
 <option value="visitante">Visitante</option>  
 <option value="colaborador">Colaborador</option>  
 </select>  
 </div>  
   
 <button type="submit" class="btn btn-primary">  
 ✅ Cadastrar  
 </button>  
 </form>  
 </div>  
   
 <!-- Agendamento -->  
 <div id="agendamento" class="tab-panel">  
 <h2>Criar Agendamento</h2>  
   
 <form id="form-agendamento" onsubmit="criarAgendamento(event)">  
 <div class="form-group">  
 <label for="pessoa-agendamento">Pessoa: \*</label>  
 <select id="pessoa-agendamento" required>  
 <option value="">Selecione uma pessoa</option>  
 </select>  
 </div>  
   
 <div class="form-group">  
 <label for="data">Data: \*</label>  
 <input type="date" id="data" required>  
 </div>  
   
 <div class="form-group">  
 <label for="horario">Horário: \*</label>  
 <input type="time" id="horario" required>  
 </div>  
   
 <div class="form-group">  
 <label for="sala">Sala: \*</label>  
 <select id="sala" required>  
 <option value="">Selecione uma sala</option>  
 <option value="sala-azul">🔵 Sala Azul (11 pessoas)</option>  
 <option value="sala-vermelha">🔴 Sala Vermelha (5 pessoas)</option>  
 <option value="sala-verde">🟢 Sala Verde (5 pessoas)</option>  
 </select>  
 </div>  
   
 <div class="form-group">  
 <label for="observacoes">Observações:</label>  
 <textarea id="observacoes" rows="3"></textarea>  
 </div>  
   
 <button type="submit" class="btn btn-primary">  
 📅 Agendar  
 </button>  
 </form>  
 </div>  
   
 <!-- Lista -->  
 <div id="lista" class="tab-panel">  
 <h2>Lista de Pessoas</h2>  
   
 <div class="form-group">  
 <input type="text" id="busca" placeholder="🔍 Buscar por nome ou email..."   
 onkeyup="buscarPessoas()">  
 </div>  
   
 <div id="lista-pessoas" class="person-list">  
 <!-- Lista será preenchida via JavaScript -->  
 </div>  
 </div>  
 </div>  
 </div>  
   
 <!-- Rodapé -->  
 <footer>  
 <p>Sistema de Recepção - Criado para Aprendizado</p>  
 <p>© 2025 - Tutorial HTML/CSS/JavaScript</p>  
 </footer>  
   
 <!-- JavaScript -->  
 <script>  
 // ===== VARIÁVEIS GLOBAIS =====  
 // Arrays para armazenar dados  
 let pessoas = [];  
 let agendamentos = [];  
   
 // ===== INICIALIZAÇÃO =====  
 // Quando a página carregar  
 window.onload = function() {  
 // Carregar dados salvos  
 carregarDados();  
   
 // Atualizar interface  
 atualizarDashboard();  
 atualizarListaPessoas();  
 atualizarSelectPessoas();  
   
 // Configurar data mínima  
 document.getElementById('data').min = new Date().toISOString().split('T')[0];  
 };  
   
 // ===== FUNÇÕES DE DADOS =====  
 // Carregar dados do localStorage  
 function carregarDados() {  
 // Tentar carregar pessoas  
 const pessoasSalvas = localStorage.getItem('pessoas');  
 if (pessoasSalvas) {  
 pessoas = JSON.parse(pessoasSalvas);  
 }  
   
 // Tentar carregar agendamentos  
 const agendamentosSalvos = localStorage.getItem('agendamentos');  
 if (agendamentosSalvos) {  
 agendamentos = JSON.parse(agendamentosSalvos);  
 }  
 }  
   
 // Salvar dados no localStorage  
 function salvarDados() {  
 localStorage.setItem('pessoas', JSON.stringify(pessoas));  
 localStorage.setItem('agendamentos', JSON.stringify(agendamentos));  
 }  
   
 // ===== FUNÇÕES DE INTERFACE =====  
 // Mostrar aba selecionada  
 function mostrarAba(abaId) {  
 // Esconder todas as abas  
 const paineis = document.querySelectorAll('.tab-panel');  
 paineis.forEach(painel => {  
 painel.classList.remove('active');  
 });  
   
 // Remover classe active de todos os botões  
 const botoes = document.querySelectorAll('.tab');  
 botoes.forEach(botao => {  
 botao.classList.remove('active');  
 });  
   
 // Mostrar aba selecionada  
 document.getElementById(abaId).classList.add('active');  
   
 // Marcar botão como ativo  
 event.target.classList.add('active');  
   
 // Atualizar conteúdo se necessário  
 if (abaId === 'dashboard') atualizarDashboard();  
 if (abaId === 'lista') atualizarListaPessoas();  
 if (abaId === 'agendamento') atualizarSelectPessoas();  
 }  
   
 // Mostrar mensagem de alerta  
 function mostrarAlerta(mensagem, tipo) {  
 const alerta = document.getElementById('alerta');  
 alerta.textContent = mensagem;  
 alerta.className = 'alert ' + tipo;  
 alerta.style.display = 'block';  
   
 // Esconder após 5 segundos  
 setTimeout(() => {  
 alerta.style.display = 'none';  
 }, 5000);  
 }  
   
 // ===== FUNÇÕES DE CADASTRO =====  
 // Cadastrar nova pessoa  
 function cadastrarPessoa(event) {  
 // Prevenir envio do formulário  
 event.preventDefault();  
   
 // Pegar valores dos campos  
 const nome = document.getElementById('nome').value.trim();  
 const email = document.getElementById('email').value.trim();  
 const telefone = document.getElementById('telefone').value.trim();  
 const tipo = document.getElementById('tipo').value;  
   
 // Verificar se email já existe  
 if (pessoas.find(p => p.email === email)) {  
 mostrarAlerta('Este e-mail já está cadastrado!', 'error');  
 return;  
 }  
   
 // Criar objeto pessoa  
 const novaPessoa = {  
 id: Date.now(), // ID único baseado no timestamp  
 nome: nome,  
 email: email,  
 telefone: telefone,  
 tipo: tipo,  
 dataCadastro: new Date().toISOString()  
 };  
   
 // Adicionar ao array  
 pessoas.push(novaPessoa);  
   
 // Salvar  
 salvarDados();  
   
 // Limpar formulário  
 document.getElementById('form-cadastro').reset();  
   
 // Mostrar mensagem de sucesso  
 mostrarAlerta('Pessoa cadastrada com sucesso!', 'success');  
   
 // Atualizar interface  
 atualizarDashboard();  
 atualizarListaPessoas();  
 }  
   
 // ===== FUNÇÕES DE AGENDAMENTO =====  
 // Criar novo agendamento  
 function criarAgendamento(event) {  
 event.preventDefault();  
   
 const pessoaId = document.getElementById('pessoa-agendamento').value;  
 const data = document.getElementById('data').value;  
 const horario = document.getElementById('horario').value;  
 const sala = document.getElementById('sala').value;  
 const observacoes = document.getElementById('observacoes').value;  
   
 // Encontrar pessoa selecionada  
 const pessoa = pessoas.find(p => p.id == pessoaId);  
   
 // Criar objeto agendamento  
 const novoAgendamento = {  
 id: Date.now(),  
 pessoaId: parseInt(pessoaId),  
 pessoaNome: pessoa.nome,  
 data: data,  
 horario: horario,  
 sala: sala,  
 observacoes: observacoes,  
 dataCriacao: new Date().toISOString()  
 };  
   
 // Adicionar ao array  
 agendamentos.push(novoAgendamento);  
   
 // Salvar  
 salvarDados();  
   
 // Limpar formulário  
 document.getElementById('form-agendamento').reset();  
   
 // Mostrar mensagem  
 mostrarAlerta('Agendamento criado com sucesso!', 'success');  
   
 // Atualizar dashboard  
 atualizarDashboard();  
 }  
   
 // ===== FUNÇÕES DE ATUALIZAÇÃO =====  
 // Atualizar dashboard  
 function atualizarDashboard() {  
 // Total de pessoas  
 document.getElementById('total-pessoas').textContent = pessoas.length;  
   
 // Total de agendamentos  
 document.getElementById('total-agendamentos').textContent = agendamentos.length;  
   
 // Agendamentos de hoje  
 const hoje = new Date().toISOString().split('T')[0];  
 const agendamentosHoje = agendamentos.filter(a => a.data === hoje);  
 document.getElementById('agendamentos-hoje').textContent = agendamentosHoje.length;  
   
 // Próximos agendamentos  
 const proximosDiv = document.getElementById('proximos-agendamentos');  
 const proximosAgendamentos = agendamentos  
 .filter(a => a.data >= hoje)  
 .sort((a, b) => new Date(a.data + ' ' + a.horario) - new Date(b.data + ' ' + b.horario))  
 .slice(0, 5);  
   
 if (proximosAgendamentos.length === 0) {  
 proximosDiv.innerHTML = '<p>Nenhum agendamento próximo.</p>';  
 } else {  
 let html = '<ul>';  
 proximosAgendamentos.forEach(a => {  
 const nomeSala = {  
 'sala-azul': '🔵 Sala Azul',  
 'sala-vermelha': '🔴 Sala Vermelha',  
 'sala-verde': '🟢 Sala Verde'  
 }[a.sala];  
   
 html += `<li>  
 <strong>${a.pessoaNome}</strong> -   
 ${new Date(a.data).toLocaleDateString('pt-BR')} às ${a.horario} -   
 ${nomeSala}  
 </li>`;  
 });  
 html += '</ul>';  
 proximosDiv.innerHTML = html;  
 }  
 }  
   
 // Atualizar lista de pessoas  
 function atualizarListaPessoas() {  
 const lista = document.getElementById('lista-pessoas');  
   
 if (pessoas.length === 0) {  
 lista.innerHTML = '<p style="text-align: center;">Nenhuma pessoa cadastrada ainda.</p>';  
 return;  
 }  
   
 lista.innerHTML = pessoas.map(pessoa => `  
 <div class="person-card">  
 <h3>${pessoa.nome}</h3>  
 <p><strong>E-mail:</strong> ${pessoa.email}</p>  
 <p><strong>Telefone:</strong> ${pessoa.telefone}</p>  
 <p><strong>Tipo:</strong> ${pessoa.tipo}</p>  
 <p><small>Cadastrado em: ${new Date(pessoa.dataCadastro).toLocaleDateString('pt-BR')}</small></p>  
 <button class="btn btn-danger" onclick="excluirPessoa(${pessoa.id})">  
 🗑️ Excluir  
 </button>  
 </div>  
 `).join('');  
 }  
   
 // Atualizar select de pessoas  
 function atualizarSelectPessoas() {  
 const select = document.getElementById('pessoa-agendamento');  
 select.innerHTML = '<option value="">Selecione uma pessoa</option>';  
   
 pessoas.forEach(pessoa => {  
 select.innerHTML += `<option value="${pessoa.id}">${pessoa.nome} (${pessoa.tipo})</option>`;  
 });  
 }  
   
 // ===== FUNÇÕES DE BUSCA E EXCLUSÃO =====  
 // Buscar pessoas  
 function buscarPessoas() {  
 const termo = document.getElementById('busca').value.toLowerCase();  
 const pessoasFiltradas = pessoas.filter(p =>   
 p.nome.toLowerCase().includes(termo) ||   
 p.email.toLowerCase().includes(termo)  
 );  
   
 const lista = document.getElementById('lista-pessoas');  
   
 if (pessoasFiltradas.length === 0) {  
 lista.innerHTML = '<p style="text-align: center;">Nenhuma pessoa encontrada.</p>';  
 return;  
 }  
   
 lista.innerHTML = pessoasFiltradas.map(pessoa => `  
 <div class="person-card">  
 <h3>${pessoa.nome}</h3>  
 <p><strong>E-mail:</strong> ${pessoa.email}</p>  
 <p><strong>Telefone:</strong> ${pessoa.telefone}</p>  
 <p><strong>Tipo:</strong> ${pessoa.tipo}</p>  
 <button class="btn btn-danger" onclick="excluirPessoa(${pessoa.id})">  
 🗑️ Excluir  
 </button>  
 </div>  
 `).join('');  
 }  
   
 // Excluir pessoa  
 function excluirPessoa(id) {  
 if (!confirm('Tem certeza que deseja excluir esta pessoa?')) {  
 return;  
 }  
   
 // Remover pessoa  
 pessoas = pessoas.filter(p => p.id !== id);  
   
 // Remover agendamentos da pessoa  
 agendamentos = agendamentos.filter(a => a.pessoaId !== id);  
   
 // Salvar  
 salvarDados();  
   
 // Mostrar mensagem  
 mostrarAlerta('Pessoa excluída com sucesso!', 'success');  
   
 // Atualizar interface  
 atualizarDashboard();  
 atualizarListaPessoas();  
 }  
   
 // ===== FUNÇÕES UTILITÁRIAS =====  
 // Formatar data para exibição  
 function formatarData(data) {  
 return new Date(data).toLocaleDateString('pt-BR');  
 }  
   
 // Formatar telefone enquanto digita  
 document.getElementById('telefone').addEventListener('input', function(e) {  
 let value = e.target.value.replace(/\D/g, '');  
 if (value.length > 0) {  
 if (value.length <= 11) {  
 if (value.length > 6) {  
 value = `(${value.slice(0,2)}) ${value.slice(2,7)}-${value.slice(7)}`;  
 } else if (value.length > 2) {  
 value = `(${value.slice(0,2)}) ${value.slice(2)}`;  
 }  
 }  
 e.target.value = value;  
 }  
 });  
 </script>  
</body>  
</html>

# **📚 CONCEITOS IMPORTANTES EXPLICADOS**

## **1. Estrutura HTML**

<tag atributo="valor">Conteúdo</tag>

**Tag**: Define o tipo de elemento (h1, p, div, etc.)

**Atributo**: Propriedade da tag (id, class, style, etc.)

**Conteúdo**: O que aparece na tela

## **2. Seletores CSS**

tag { } /\* Seleciona todas as tags \*/  
.classe { } /\* Seleciona por classe \*/  
#id { } /\* Seleciona por ID \*/

## **3. Eventos JavaScript**

onclick="funcao()" // Quando clica  
onkeyup="funcao()" // Quando solta tecla  
onsubmit="funcao()" // Quando envia form

## **4. Manipulação DOM**

document.getElementById('id') // Pega elemento  
elemento.value // Pega valor  
elemento.innerHTML = 'HTML' // Muda conteúdo  
elemento.style.display = 'none' // Esconde

## **5. Armazenamento Local**

localStorage.setItem('chave', 'valor') // Salva  
localStorage.getItem('chave') // Recupera  
localStorage.removeItem('chave') // Remove

# **🎯 EXERCÍCIOS PRÁTICOS**

## **Exercício 1: Adicione um campo "Empresa"**

No formulário de cadastro, adicione um input para empresa

Salve essa informação junto com a pessoa

Mostre na lista

## **Exercício 2: Crie um botão "Editar"**

Adicione um botão editar em cada pessoa

Ao clicar, preencha o formulário com os dados

Atualize ao invés de criar novo

## **Exercício 3: Adicione validação**

Verifique se o email tem @

Verifique se o telefone tem 11 dígitos

Mostre mensagem de erro específica

## **Exercício 4: Crie filtro por tipo**

Adicione botões: Todos | Visitantes | Colaboradores

Filtre a lista conforme clicado

Destaque o botão ativo

## **Exercício 5: Adicione contadores**

Mostre quantos visitantes

Mostre quantos colaboradores

Atualize em tempo real

# **🚀 PRÓXIMOS PASSOS**

## **1. Aprenda mais HTML**

Tabelas (<table>)

Formulários avançados

Elementos semânticos (header, nav, main, footer)

## **2. Aprenda mais CSS**

Flexbox e Grid

Animações

Media queries (responsivo)

Variáveis CSS

## **3. Aprenda mais JavaScript**

Arrays e objetos

Promises e async/await

Fetch API (buscar dados)

Módulos

## **4. Ferramentas profissionais**

Visual Studio Code (editor)

Git (controle de versão)

Chrome DevTools (debug)

Node.js (JavaScript no servidor)

## **5. Frameworks (depois de dominar o básico)**

React

Vue.js

Angular

# **💪 DICAS FINAIS**

**Pratique todos os dias** - 30 minutos por dia

**Faça projetos próprios** - Lista de tarefas, calculadora

**Leia código dos outros** - GitHub é seu amigo

**Não tenha medo de errar** - Erro é aprendizado

**Pergunte** - Comunidades online ajudam muito

**Documente seu código** - Comentários ajudam

**Seja paciente** - Programação leva tempo

# **🎓 PARABÉNS!**

Você completou o tutorial! Agora você sabe:

✅ Criar estrutura HTML

✅ Estilizar com CSS

✅ Adicionar interatividade com JavaScript

✅ Salvar dados no navegador

✅ Criar um sistema funcional

**Continue praticando e logo estará criando sistemas incríveis!**

*Tutorial criado com ❤️ para iniciantes em programação*

*Lembre-se: Todo expert já foi iniciante!*