# Roteiro de Apresentação: Lista de Tarefas (HTML, CSS, JS)

Este roteiro foi elaborado para guiar uma apresentação detalhada sobre a estrutura e o funcionamento do código da aplicação de Lista de Tarefas, utilizando os arquivos aula03.html, aula03.css e aula03.js.

## 1. Introdução (5 minutos)

| Slide | Título | Pontos de Discussão |
| --- | --- | --- |
| **1** | **Lista de Tarefas: Anatomia de um Projeto Web** | **Objetivo:** Apresentar o projeto e o escopo da análise. |
|  |  | \* O que é o projeto: Uma aplicação funcional de Lista de Tarefas (To-Do List). |
|  |  | \* Foco: Demonstrar a integração fundamental entre as três tecnologias web. |
| **2** | **Os Três Pilares** | **Arquivos:** Introduzir os arquivos e seus papéis. |
|  |  | \* **aula03.html:** A Estrutura (O Esqueleto). |
|  |  | \* **aula03.css:** O Estilo (A Aparência). |
|  |  | \* **aula03.js:** A Interatividade (O Comportamento). |

## 2. HTML: A Estrutura (aula03.html) (10 minutos)

| Slide | Título | Pontos de Discussão |
| --- | --- | --- |
| **3** | **Configuração Inicial e Metadados** | **Linhas 1-9:** O que o navegador precisa saber. |
|  |  | \* <!DOCTYPE html> e <meta charset="UTF-8">. |
|  |  | \* **Viewport:** Importância do <meta name="viewport"...> para responsividade. |
|  |  | \* **Conexão CSS:** Onde o aula03.css é linkado (<link rel="stylesheet"...>). |
| **4** | **O Cabeçalho Interativo** | **Linhas 12-16:** A área de input e ação. |
|  |  | \* **div class="header":** Contêiner principal. |
|  |  | \* **<input type="text" id="tarefa">:** O campo de entrada. Destacar o id="tarefa" como ponto de contato para o JavaScript. |
|  |  | \* **<span onclick="addElemento()" class="addBtn">:** O botão "Incluir". Destacar o evento onclick chamando a função JS. |
| **5** | **A Lista de Itens** | **Linhas 17-21:** Onde as tarefas residem. |
|  |  | \* **ul id="itemLista":** A lista não ordenada. Destacar o id para manipulação via JS. |
|  |  | \* **<li class="checked">:** Mostrar um item inicial e a classe checked aplicada, que é um estado que será manipulado pelo JS. |
| **6** | **Conexão JavaScript** | **Linha 22:** Onde a mágica da interatividade começa. |
|  |  | \* <script src="aula03.js"></script>: A importância de carregar o JS no final do <body> para garantir que todos os elementos HTML já estejam disponíveis (DOM pronto) antes da execução do script. |

## 3. CSS: O Estilo (aula03.css) (10 minutos)

| Slide | Título | Pontos de Discussão |
| --- | --- | --- |
| **7** | **Reset e Box Model** | **Linhas 1-8:** Configurações globais. |
|  |  | \* **box-sizing: border-box;:** Explicar brevemente como isso simplifica o layout. |
|  |  | \* min-width: 250px; no body: Garantia de usabilidade mínima. |
| **8** | **Estilizando a Lista** | **Linhas 15-35:** A aparência das tarefas. |
|  |  | \* **Efeito Zebrado:** Uso de ul li:nth-child(odd) para alternar cores de fundo. |
|  |  | \* **Interação Visual:** Uso de ul li:hover para feedback visual. |
|  |  | \* **Estado Concluído:** A classe ul li.checked (cor de fundo, cor da fonte e text-decoration: line-through). |
| **9** | **Layout do Cabeçalho e Botões** | **Linhas 49-87:** Organização e Ação. |
|  |  | \* **Layout com float:** Mostrar como input (75%) e .addBtn (25%) usam float: left para ficar lado a lado. |
|  |  | \* **Clearfix:** Explicar o uso do .header:after para conter os elementos flutuantes. |
|  |  | \* **Botão Fechar (.close):** Uso de position: absolute dentro do li (que tem position: relative) para posicionamento preciso. |

## 4. JavaScript: A Interatividade (aula03.js) (15 minutos)

| Slide | Título | Pontos de Discussão |
| --- | --- | --- |
| **10** | **Inicialização: Injetando o Botão de Fechar** | **Linhas 1-9:** Preparando os itens existentes. |
|  |  | \* document.getElementsByTagName("li"): Seleção inicial dos elementos. |
|  |  | \* O laço for: Criação do <span> com a classe close e o caractere 'x' (\u00D7). |
|  |  | \* **Conceito:** O JS está modificando o HTML após o carregamento. |
| **11** | **Delegação de Eventos: Marcar Tarefa** | **Linhas 20-26:** O método mais eficiente. |
|  |  | \* list.addEventListener('click', ...): O ouvinte está no ul, não em cada li. |
|  |  | \* if (ev.target.tagName === 'LI'): Verifica se o clique ocorreu em um li. |
|  |  | \* ev.target.classList.toggle('checked'): Alterna a classe, ativando o estilo CSS. |
| **12** | **Função addElemento(): Adicionando Novos Itens** | **Linhas 28-56:** O fluxo de criação de uma nova tarefa. |
|  |  | \* **Captura de Valor:** document.getElementById("tarefa").value. |
|  |  | \* **Criação de Elementos:** Criação do novo li e injeção do texto. |
|  |  | \* **Validação Simples:** O if (inputValue === '') { alert(...) }. |
| **13** | **O Ponto Crítico: Exclusão de Itens** | **Linhas 11-18 e 50-55:** A lógica de exclusão. |
|  |  | \* **Exclusão Inicial (Linhas 11-18):** Define o onclick para os itens *existentes*. |
|  |  | \* **Exclusão para Novos Itens (Linhas 50-55):** **Problema/Solução:** Mostrar que o laço é repetido dentro de addElemento(). |
|  |  | \* **Melhoria Sugerida:** Mencionar que seria mais eficiente usar Delegação de Eventos também para a exclusão, em vez de redefinir o onclick para todos os botões a cada nova tarefa. |

## 5. Conclusão e Próximos Passos (5 minutos)

| Slide | Título | Pontos de Discussão |
| --- | --- | --- |
| **14** | **Conclusão: O Ciclo Web** | **Revisão:** Como as partes se encaixam. |
|  |  | \* HTML define a **estrutura** (o que existe). |
|  |  | \* CSS define a **apresentação** (como se parece). |
|  |  | \* JavaScript define o **comportamento** (o que acontece). |
| **15** | **Sugestões de Evolução** | **Para onde ir a partir daqui.** |
|  |  | \* **Persistência de Dados:** Como salvar as tarefas (Web Storage/LocalStorage). |
|  |  | \* **Refatoração JS:** Implementar a delegação de eventos para o botão de fechar. |
|  |  | \* **Acessibilidade:** Adicionar atributos aria e melhorar a navegação por teclado. |

**Tempo Total Estimado:** 40 minutos (com margem para perguntas e demonstração).