

```

<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100..900;1,100..900&family=Special+Gothic&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="styles.css">
  <title>To-Do List</title>
</head>

<body>
  <div class="all-container">
    <h1>📋 Minha Lista de Tarefas</h1>
    <form id="all-form">
      <input type="text" id="all-input" placeholder="Digite uma tarefa"
required>
      <button type="submit">Adicionar</button>
    </form>
    <div class="filters">
      <button data-filter="all" class="filter active">Todas</button>
      <button data-filter="active" class="filter">Pendentes</button>
      <button data-filter="completed" class="filter">Concluídas</button>
    </div>
    <ul id="all-list"></ul>
  </div>
</body>
<script src="./scripts.js"></script>

</html>

```

Esse código é a estrutura de uma **aplicação web de lista de tarefas** (To-Do List) feita com HTML, CSS e JavaScript.

◇ 1. Estrutura básica do HTML

```
<!DOCTYPE html>  
<html lang="pt-br">
```

- Define que é um documento HTML5.
- O idioma da página é **português do Brasil** (pt-br).

◇ 2. <head> - Cabeçalho da página

```
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-  
scale=1.0">
```

- charset="UTF-8" → Permite acentuação e caracteres especiais.
- viewport → Garante responsividade em dispositivos móveis.

Fontes do Google

```
<link rel="preconnect" href="https://fonts.googleapis.com">  
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
```

- Otimiza o carregamento das fontes com **conexões antecipadas**.

```
<link  
  
  href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100..900;1,100..900&family=Special+Gothic&display=swap"  
  rel="stylesheet">
```

- Importa as fontes **Roboto** e **Special Gothic** (mas neste HTML só usamos Roboto, segundo o CSS anterior).

```
<link rel="stylesheet" href="styles.css">
```

- Importa o CSS externo com o estilo da página.

```
<title>To-Do List</title>
```

- Define o título que aparece na aba do navegador.

◇ 3. <body> - Corpo da página

```
<div class="all-container">
```

- div principal da aplicação, com classe all-container. Esse nome vem do CSS e serve para estilizar o bloco central da lista.

Título da página

```
<h1> 📝 Minha Lista de Tarefas</h1>
```

- Um título com um emoji de prancheta.

Formulário de entrada

```
<form id="all-form">  
  <input type="text" id="all-input" placeholder="Digite uma  
tarefa" required>  
  <button type="submit">Adicionar</button>  
</form>
```

- Formulário para adicionar uma nova tarefa:
 - input → Campo de texto onde o usuário escreve.
 - required → Torna obrigatório digitar algo.
 - button → Botão para adicionar a tarefa.

O formulário tem id="all-form" para ser **referenciado pelo JavaScript**.

Filtros de visualização

```
<div class="filters">
  <button data-filter="all" class="filter active">Todas</button>
  <button data-filter="active" class="filter">Pendentes</button>
  <button data-filter="completed"
class="filter">Concluídas</button>
</div>
```

- Três botões que permitem filtrar a lista:
 - Todas → Exibe todas as tarefas.
 - Pendentes → Exibe somente as que ainda não foram feitas.
 - Concluídas → Exibe apenas as finalizadas.

Cada botão tem um atributo `data-filter` para ajudar o JavaScript a saber o tipo de filtro.

Lista de tarefas

```
<ul id="all-list"></ul>
```

- Lista onde as tarefas serão inseridas dinamicamente com JavaScript.
- Cada tarefa será adicionada como um `` dentro dessa ``.

◇ 4. Script

```
<script src="./scripts.js"></script>
```

- Conecta um arquivo `scripts.js` externo que provavelmente:
 - Adiciona tarefas à lista.
 - Permite marcar como concluídas.
 - Aplica filtros.
 - Pode permitir remover ou editar tarefas também.

Resumo

Esse HTML cria a **interface visual** de uma lista de tarefas com os seguintes recursos:

Recurso	Função
Campo de texto	Onde o usuário escreve uma nova tarefa
Botão "Adicionar"	Envia o formulário e insere a tarefa na lista
Botões de filtro	Permitem ver todas, só pendentes ou só concluídas
Lista ()	Onde as tarefas são exibidas
Arquivo JS	Faz toda a lógica interativa (adicionar, filtrar, marcar, etc)
Arquivo CSS	Aplica o estilo visual limpo, centralizado, com sombra e cores suaves