

HTML

A estrutura HTML apresentada representa uma aplicação simples, mas funcional, de uma lista de tarefas, com uma organização clara e pensada tanto para a usabilidade quanto para a manutenção futura. Tudo começa com a declaração `<!DOCTYPE html>`, que informa ao navegador que o documento segue o padrão HTML5, garantindo compatibilidade com recursos modernos da web. O elemento `<html>` define que o idioma principal da página é o português brasileiro, o que é especialmente útil para leitores de tela, motores de busca e tradutores automáticos.

No `<head>`, temos o básico essencial: a codificação de caracteres é definida como UTF-8, o que permite que acentos e caracteres especiais usados no português apareçam corretamente. O elemento `<meta name="viewport">` é fundamental para que o layout se adapte corretamente em dispositivos móveis, possibilitando uma experiência responsiva. O título da aba do navegador aparece como "Lista de Tarefas", identificando claramente a funcionalidade da página. O arquivo CSS externo é importado com `<link>`, permitindo que a estilização fique separada do conteúdo e torne o código mais limpo e reaproveitável.

O corpo da página (`<body>`) começa com um `<header>` que contém um título principal (`<h1>`) — "Minha Lista de Tarefas" —, o qual funciona como o cabeçalho da aplicação e indica imediatamente ao usuário qual é o propósito do site.

O conteúdo principal está dentro da tag `<main>`, dividida em duas seções: uma para adicionar tarefas e outra para listá-las.

Na primeira seção, com o `id="adicionar-tarefa"`, temos um campo de entrada de texto (`<input type="text">`) com o `id="nova-tarefa"`, onde o usuário digita o que precisa ser feito. Ele vem com um `placeholder` que orienta: "Adicionar nova tarefa". Ao lado, um botão com `id="btn-adicionar"` serve para acionar a adição dessa nova tarefa à lista. Logo abaixo, há uma `<div>` com `id="erro-input"` e a classe "erro", que provavelmente será usada para exibir mensagens quando o usuário tentar adicionar uma tarefa vazia ou inválida.

A segunda seção (`id="lista-de-tarefas"`) contém uma lista desordenada (``) com o `id="tarefas"`, onde as tarefas serão adicionadas dinamicamente. Inicialmente ela está vazia, esperando o preenchimento via JavaScript.

O rodapé (`<footer>`) finaliza a página com um aviso de direitos autorais, indicando que o conteúdo é de 2025 e faz parte da "Lista de Tarefas Interativa", fechando a estrutura da página com uma identidade simples e funcional.

Por fim, o `<script src="script.js"></script>` carrega o JavaScript externo que deve conter toda a lógica interativa, como adicionar tarefas à lista, validar entradas, remover itens ou até marcar tarefas como concluídas.

Em resumo, essa estrutura HTML é objetiva, clara e funcional.

JS

Esse trecho de código JavaScript foi estruturado para transformar uma página HTML simples em uma aplicação interativa, onde o usuário pode adicionar, concluir e remover tarefas. Ele começa com um cuidado essencial: aguardar o carregamento completo do DOM antes de executar qualquer coisa. Isso garante que todos os elementos da página já estejam prontos e disponíveis para manipulação.

A primeira ação do script é selecionar e armazenar referências aos principais elementos da interface: o campo de texto onde a tarefa será digitada, o botão de adicionar, a lista que vai exibir as tarefas, e a div que mostrará mensagens de erro quando algo der errado.

Em seguida, é definida a função principal do código: `adicionarTarefa()`. Essa função é o coração da aplicação, pois cuida de todo o processo de criação de uma nova tarefa. Ela começa pegando o texto digitado pelo usuário e usando `trim()` para eliminar espaços em branco no início ou fim da entrada — uma forma de prevenir que uma "tarefa vazia" com apenas espaços seja adicionada.

Se, mesmo após esse tratamento, o campo estiver vazio, uma mensagem de erro clara é exibida na tela, avisando o usuário de que precisa digitar algo. A div de erro, que normalmente está oculta, passa a ficar visível, chamando a atenção do usuário.

Mas, se o usuário tiver digitado uma tarefa válida, o script prossegue criando um novo item de lista (``). Dentro desse item, ele insere o texto da tarefa e duas opções de ação: um botão para **concluir** (que irá riscar o texto) e outro para **remover** a tarefa completamente. Esses botões são incluídos dinamicamente com `innerHTML`, já formatados com classes específicas para facilitar a estilização e a identificação de seus comportamentos.

Depois que os botões são inseridos, o código adiciona dois ouvintes de eventos: um para o botão de concluir e outro para o de remover. O botão de concluir alterna a classe CSS chamada `"concluida"` no texto da tarefa — uma estratégia comum que permite riscar o texto via CSS, visualmente indicando que aquela tarefa já foi feita. Já o botão de remover é mais direto: ele simplesmente remove o item da lista quando clicado, eliminando a tarefa da tela.

Finalizada a criação e configuração do novo item, ele é adicionado à lista com `appendChild()`, o campo de texto é limpo para que o usuário possa digitar outra tarefa sem precisar apagar manualmente, e a mensagem de erro, se existia, é removida e a div de erro volta a ficar oculta.

Por fim, dois ouvintes de eventos garantem que a função `adicionarTarefa()` seja chamada tanto quando o usuário clica no botão quanto quando pressiona a tecla **Enter** — algo que melhora bastante a experiência de uso, já que permite adicionar tarefas de forma rápida apenas com o teclado.

CSS

O estilo do `body` estabelece uma base sólida para o restante da página. Ao utilizar a fonte Arial com fallback para sans-serif, garante-se uma leitura fluida em qualquer sistema. O fundo em tom cinza claro proporciona suavidade visual e contrasta bem com os elementos centrais. A aplicação de `display: flex` com direção em coluna e alinhamento central horizontal confere uma organização vertical equilibrada, fazendo com que todos os elementos fiquem centralizados e distribuídos de forma harmoniosa, seja em uma tela de computador ou em um dispositivo móvel. O `padding` aplicado impede que os elementos fiquem colados nas bordas, mantendo uma área de respiro confortável.

O cabeçalho, posicionado no topo da página, adota um fundo verde vibrante com texto branco centralizado, criando uma identidade visual clara para o título da aplicação. O uso de `max-width` evita que o cabeçalho fique exageradamente largo em telas grandes, e os cantos superiores arredondados suavizam sua presença visual, contribuindo para um design mais amigável.

A área principal, representada pela tag `main`, funciona como um contêiner destacado, semelhante a um cartão flutuante. Seu fundo branco, bordas arredondadas e sombra leve criam uma separação clara em relação ao fundo da página, facilitando a leitura e a concentração do usuário nos elementos centrais. Esse contêiner também é limitado a uma largura máxima, evitando que o conteúdo se espalhe demais, o que mantém a legibilidade ideal.

Na seção onde o usuário insere novas tarefas, chamada `#adicionar-tarefa`, o uso de Flexbox permite o alinhamento eficiente entre o campo de texto e o botão de adicionar. O campo de entrada se expande para ocupar todo o espaço disponível, enquanto o botão mantém um tamanho consistente, ambos com bordas arredondadas e estilos simples, reforçando a coerência visual da aplicação. O botão, com cor de fundo verde, passa a uma tonalidade mais escura ao ser sobrevoado, proporcionando um efeito visual dinâmico e intuitivo.

A lista de tarefas, organizada em uma lista não ordenada, foi estilizada para eliminar marcadores e espaçamentos padrão, o que permite um controle total sobre sua aparência. Cada item da lista é exibido como um bloco flexível, com o texto da tarefa de um lado e os botões de ação do outro. A separação entre tarefas é feita por uma linha sutil, e o espaçamento vertical garante que cada item respire visualmente.

Ao interagir com os botões "Concluir" e "Remover", o usuário recebe respostas visuais claras. Quando a tarefa é marcada como concluída, seu texto é riscado e perde a intensidade da cor, sinalizando de forma objetiva seu novo estado. Os botões, dispostos lado a lado com um pequeno espaço entre si, utilizam cores distintas para sugerir ações diferentes: amarelo para concluir e vermelho para remover. Ambos possuem transições de cor suaves ao serem sobrevoados, criando uma experiência mais fluida e refinada.

A aplicação também está preparada para informar o usuário quando há erro na inserção de tarefas, como quando o campo é deixado vazio. Nesse caso, uma mensagem de erro

aparece em vermelho logo abaixo do formulário, destacando-se imediatamente sem ser agressiva visualmente. Essa mensagem, inicialmente oculta, é exibida dinamicamente conforme necessário.

O rodapé da página é discreto, centralizado e utiliza um tom de cinza mais claro, ocupando pouco espaço visual, mas encerrando a aplicação com uma sensação de acabamento. Ele não interfere na navegação, mas mantém uma consistência com o restante do design.