

# Desenvolvimento Web com JavaScript: Criando uma Lista de Tarefas

Por Escola Dnc

Neste ebook, vamos explorar o processo de criação de uma lista de tarefas usando JavaScript, HTML e CSS. Abordaremos conceitos fundamentais do Document Object Model (DOM), a importância do JavaScript na criação de páginas web dinâmicas, e como utilizar ferramentas essenciais para otimizar o fluxo de trabalho no desenvolvimento front-end.

## Configuração do Ambiente de Desenvolvimento

### Instalando o Visual Studio Code e Live Server

Para iniciar nosso projeto, utilizaremos o Visual Studio Code (VS Code) como nosso editor de código. Uma extensão crucial para o desenvolvimento web é o Live Server.

#### Passos para instalar o Live Server:

1. Abra o VS Code
2. Acesse a seção de extensões (ícone de quatro quadrados ou CTRL+SHIFT+X)
3. Pesquise por "Live Server"
4. Instale a extensão com mais downloads (aproximadamente 50 milhões)

O Live Server permite criar um servidor local com hot reload, atualizando automaticamente o navegador quando salvamos alterações no código.

### Criando o Projeto

Após instalar o Live Server, siga estes passos:

1. Crie uma nova pasta chamada "To Do List"
2. Abra esta pasta no VS Code
3. Crie um novo arquivo HTML (index.html)
4. Insira a estrutura básica do HTML5

Para iniciar o servidor:

- Clique com o botão direito no arquivo HTML
- Selecione "Abrir com Live Server"

Isso iniciará um projeto na porta 5500, abrindo automaticamente no seu navegador padrão.

## Estruturando o HTML e CSS Básico

### Criando a Estrutura HTML

Vamos começar com uma estrutura HTML simples:

```
<!DOCTYPE html><html lang="pt-BR"><head>    <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">    <title>Lista de Tarefas - Projeto Final
JavaScript</title>    <link rel="stylesheet" href="styles.css">
</head><body>    <main>        <h1>Lista de Tarefas</h1>    </main>
</body></html>
```

### Estilizando com CSS

Crie um arquivo `styles.css` e adicione os seguintes estilos:

```
html {    height: 100%;    width: 100%;    box-sizing: border-box;
font-size: 10px;    font-family: sans-serif;}body {    height:
100%;    width: 100%;    margin: 0;    background: linear-
gradient(#41E5D9, #A4D8FE2);    display: flex;    justify-content:
center;    align-items: center;}h1 {    font-size: 3.2rem;
margin: 0 0 1.6rem;}main {    width: 100%;    max-width: 40rem;
background-color: #fff;    border-radius: 0.8rem;    padding:
2.4rem 3.2rem;}
```

Estes estilos criarão um layout centralizado com um gradiente de fundo e um container branco para nossa lista de tarefas.

## Implementando a Funcionalidade JavaScript

### Criando a Lista de Tarefas

Na próxima etapa, implementaremos a funcionalidade da lista de tarefas usando JavaScript. Vamos focar em:

- Renderizar uma lista no navegador
- Adicionar checkboxes para cada tarefa
- Implementar a interação com o usuário

### Conceitos Importantes

Ao desenvolver a funcionalidade, abordaremos conceitos cruciais:

- **Manipulação do DOM:** Como interagir e modificar elementos HTML usando JavaScript
- **Eventos:** Responder a ações do usuário, como clicar em um checkbox
- **Local Storage:** Armazenar tarefas localmente para persistência de dados

### Estrutura Básica do JavaScript

Crie um arquivo `script.js` e adicione-o ao seu HTML. Aqui está uma estrutura básica para começar:

```
document.addEventListener('DOMContentLoaded', () => {    // Código
para inicializar a lista de tarefas});function
adicionarTarefa(texto) {    // Lógica para adicionar uma nova
tarefa}function marcarTarefa(id) {    // Lógica para marcar uma
tarefa como concluída}function removerTarefa(id) {    // Lógica
para remover uma tarefa}function salvarTarefas() {    // Lógica
para salvar tarefas no Local Storage}function carregarTarefas() {
// Lógica para carregar tarefas do Local Storage}
```

## Aprimorando a Interface do Usuário

## Adicionando Interatividade

Para melhorar a experiência do usuário, podemos adicionar:

- Um campo de entrada para novas tarefas
- Botões para adicionar e remover tarefas
- Animações suaves ao adicionar/remover tarefas

## Exemplo de HTML Atualizado

```
<main>      <h1>Lista de Tarefas</h1>      <div class="nova-tarefa">
<input type="text" id="input-tarefa" placeholder="Nova tarefa">
<button id="btn-adicionar">Adicionar</button>      </div>      <ul
id="lista-tarefas"></ul></main>
```

## CSS Adicional

```
.nova-tarefa { display: flex; margin-bottom: 1.6rem;}#input-
tarefa { flex-grow: 1; padding: 0.8rem; font-size: 1.6rem;
border: 1px solid #ccc; border-radius: 0.4rem 0 0 0.4rem;}#btn-
adicionar { padding: 0.8rem 1.6rem; font-size: 1.6rem;
background-color: #4CAF50; color: white; border: none;
border-radius: 0 0.4rem 0.4rem 0; cursor: pointer;}#lista-
```

```
tarefas { list-style-type: none; padding: 0; }.tarefa {  
display: flex; align-items: center; padding: 0.8rem;  
border-bottom: 1px solid #eee; }.tarefa input[type="checkbox"] {  
margin-right: 1rem; }
```

## Conclusão

Neste ebook, exploramos o processo de criação de uma lista de tarefas usando JavaScript, HTML e CSS. Abordamos desde a configuração inicial do ambiente de desenvolvimento até a implementação de funcionalidades básicas e a estilização da interface do usuário.

Pontos-chave abordados:

- Configuração do VS Code com Live Server
- Estruturação básica do HTML e CSS
- Implementação de funcionalidades JavaScript
- Manipulação do DOM e eventos
- Uso do Local Storage para persistência de dados
- Aprimoramento da interface do usuário

Este projeto serve como uma excelente introdução ao desenvolvimento web front-end, demonstrando como diferentes tecnologias se integram para criar uma aplicação funcional e interativa. Continue explorando e expandindo este projeto para aprimorar suas habilidades em desenvolvimento web!



