

Por Escola Dnc

Introdução

Neste ebook, vamos explorar o processo de criação de uma lista de tarefas interativa usando JavaScript, HTML e CSS. Abordaremos desde a estruturação básica da página até a implementação de funcionalidades como renderização dinâmica de itens, manipulação do DOM e estilização para uma melhor experiência do usuário.

Estrutura Básica e Renderização Inicial

Criando a Estrutura HTML

Para começar, precisamos criar a estrutura básica da nossa página HTML:

```
<section class="todoListSection"> <ul id="todoList"></ul>
</section>
```

Esta estrutura serve como base para nossa lista de tarefas. A `<section>` com a classe `todoListSection` contém uma lista não ordenada (``) com o ID `todoList` , onde renderizaremos nossas tarefas.

Configurando o JavaScript

Criamos um arquivo `index.js` e o importamos no HTML:

```
<script src="javascript/index.js"></script>
```

Criando o Array de Tarefas

No arquivo JavaScript, definimos um array com objetos representando nossas tarefas:

```
const tasks = [ { id: 1, description: "Comprar pão", checked:
false }, { id: 2, description: "Passear com o cachorro", checked:
false }, { id: 3, description: "Fazer o almoço", checked: false
}];
```

Cada tarefa possui um `id` , uma `description` e um estado `checked` .

Função para Criar Checkbox

```
function getCheckboxInput(id, description) {  const checkbox =
document.createElement("input");  const label =
document.createElement("label");  const wrapper =
document.createElement("div");  checkbox.type = "checkbox";
checkbox.id = `checkbox-${id}`;  checkbox.checked = task.checked;
label.textContent = description;  label.htmlFor = checkbox.id;
wrapper.className = "checkbox-label-container";
wrapper.appendChild(checkbox);  wrapper.appendChild(label);  return
wrapper;}
```

Modificamos nossa função de renderização para incluir os checkboxes:

```
tasks.forEach(task => { const todo = document.createElement("li");
const checkbox = getCheckboxInput(task.id, task.description);
todo.appendChild(checkbox); todoList.appendChild(todo);});
```

Estilização da Lista de Tarefas

Para melhorar a aparência da nossa lista, adicionamos alguns estilos CSS:

```
.todoListSection { margin: 0.8rem 0 1.6rem; min-height: 5rem;}.todoListSection ul { padding-inline-start: 0; min-height: 230px; overflow: auto;}.todoListSection li { list-style-type: none; padding: 0.8rem 1.6rem; font-size: 1.6rem; background-color: #fefefe; display: flex; justify-content: space-between; align-items: center; margin-bottom: 0.8rem;}.todoListSection li:last-child { margin-bottom: 0;}.checkbox-label-container { display: flex; align-items: center; justify-content: center;}.checkbox-label-container label { cursor: pointer;}.checkbox-label-container input { margin-right: 8px; cursor: pointer;}
```

Estes estilos garantem um layout agradável e responsivo para nossa lista de tarefas.

Adicionando um Botão de Exclusão

Embora não implementado nesta fase, preparamos o estilo para um futuro botão de exclusão:

```
.todoListSection li button { width: 2.4rem; height: 2.4rem;
background-color: #f64; color: white; border: none; border-
radius: 4px; cursor: pointer;}.todoListSection li button:hover {
background-color: #da5c4d;}
```

Este estilo será aplicado quando adicionarmos a funcionalidade de exclusão de tarefas.

Conclusão

Neste ebook, exploramos os fundamentos da criação de uma lista de tarefas interativa usando JavaScript, HTML e CSS. Abordamos:

- Estruturação básica do HTML
- Implementação da lógica JavaScript para renderização dinâmica
- Criação de checkboxes interativos
- Estilização para melhorar a aparência e usabilidade

Nos próximos passos, poderíamos expandir este projeto adicionando funcionalidades como:

- Criação de um formulário para adicionar novas tarefas
- Implementação da funcionalidade de exclusão de tarefas
- Persistência de dados usando armazenamento local
- Melhorias de acessibilidade e responsividade

Este projeto serve como uma excelente base para explorar conceitos mais avançados de desenvolvimento web front-end e manipulação do DOM com JavaScript.

