



Atividade - Manipulação DOM

Objetivo

O objetivo deste exercício é praticar a manipulação do DOM e o tratamento de eventos em JavaScript, criando dinamicamente uma tabela HTML baseada em um conjunto de dados de montanhas e permitindo a adição de novas linhas via interação do usuário.

Descrição do Exercício

Você deve construir uma tabela HTML que liste informações sobre várias montanhas. Os dados serão fornecidos como um array de objetos em JavaScript, onde cada objeto representa uma montanha com as seguintes propriedades: `nome`, `altura` e `local`.

A tabela deve ser gerada dinamicamente a partir desses dados, com uma coluna para cada propriedade e uma linha para cada objeto. A primeira linha da tabela deve conter cabeçalhos (`<th>`) com os nomes das propriedades.

Além disso, você deve criar um formulário com campos para `nome`, `altura` e `local`, e um botão que, ao ser clicado, adiciona uma nova linha à tabela com os valores inseridos pelo usuário. O evento de clique do botão deve ser tratado usando `addEventListener`.

Requisitos

- Gere a tabela dinamicamente a partir do array de objetos fornecido.
- As colunas da tabela devem ser derivadas automaticamente das propriedades do primeiro objeto no array.
- Exiba a tabela resultante no elemento com o atributo `id` igual a `mountains`.
- Crie um formulário com três campos de entrada (`<input>`) para `nome`, `altura` e `local`, e um botão para adicionar uma nova linha.
- Use `addEventListener` para tratar o evento de clique do botão, adicionando uma nova linha à tabela com os valores dos campos de entrada.
- Valide que o campo `altura` contém um valor numérico antes de adicionar a linha, exibindo uma mensagem HTML acusando o erro.
- Após criar a tabela, estilize-a utilizando CSS. Certifique-se de:
 - Alinhar à direita as células que contêm valores numéricos.
 - Aplicar estilos de borda, preenchimento (`padding`) e cores de fundo para melhorar a aparência da tabela.
 - Aplicar uma cor de fundo diferenciada para as linhas pares, criando um efeito zebraado.

Estrutura Inicial

Aqui está a estrutura básica do HTML onde você deve implementar o exercício:

```
1 <h1>Montanhas</h1>
2
3 <div id="mountains"></div>
4
5 <form id="mountainForm">
6   <label for="nome">Nome:</label>
7   <input type="text" id="nome" name="nome" required>
8   <label for="altura">Altura (m):</label>
9   <input type="number" id="altura" name="altura" required>
10  <label for="local">Local:</label>
11  <input type="text" id="local" name="local" required>
12  <button type="submit">Adicionar Montanha</button>
13</form>
14
15 <div id="errorMessage"></div>
16
17 <script>
18   const MONTANHAS = [
19     {nome: "Kilimanjaro", altura: 5895, local: "Tanzania"},
20     {nome: "Everest", altura: 8848, local: "Nepal"},
21     {nome: "Monte Fuji", altura: 3776, local: "Japan"},
22     {nome: "Vaalserberg", altura: 323, local: "Netherlands"},
23     {nome: "Denali", altura: 6168, local: "USA"},
24     {nome: "Popocatepetl", altura: 5465, local: "Mexico"},
25     {nome: "Mont Blanc", altura: 4808, local: "Italy/France"}
26   ];
27
28 </script>
```

Estilização da Tabela

Utilize CSS para estilizar a tabela e o formulário. Assegure-se de incluir as seguintes regras:

- Alinhe o texto das células numéricas à direita.
- Adicione bordas às células da tabela.
- Defina o preenchimento (*padding*) das células para melhorar a legibilidade.
- Aplique uma cor de fundo diferenciada para as linhas pares, criando um efeito zebrado.
- Estilize o formulário para alinhar os campos de entrada e o botão de forma clara e acessível.
- Exiba mensagens de erro (se houver) em vermelho abaixo do formulário.

Dicas

- Utilize o método `Object.keys()` para obter as propriedades do primeiro objeto e gerar dinamicamente os cabeçalhos da tabela.
- O método `forEach()` pode ser útil para iterar sobre os arrays.
- Para criar um efeito zebrado na tabela, use a pseudoclasse `:nth-child(even)` em conjunto com `tr`.

- Para estilizar as células numéricas, você pode verificar o tipo de dado utilizando `typeof`.
- Use `addEventListener('click', ...)` ou `addEventListener('submit', ...)` no formulário ou botão para capturar a ação de adicionar uma nova montanha.
- Valide o campo `altura` com `isNaN()` para garantir que seja um número.
- Use `event.preventDefault()` no evento de `submit` do formulário para evitar o comportamento padrão de envio de formulário, que recarrega a página. Isso é necessário porque o formulário HTML, por padrão, tenta enviar os dados para um servidor, o que causaria a perda do estado atual da página e impediria a adição dinâmica de linhas na tabela. Ao chamar `preventDefault()`, você garante que a página permaneça intacta e que o JavaScript possa manipular o DOM para adicionar a nova linha.