Javascript : Módulo - Imutabilidade

Desafio 1: Lista de Contatos

Requisitos:

Crie uma aplicação onde você pode:

1. Adicionar um novo contato com nome e telefone.
2. Atualizar o telefone de um contato existente.
3. Remover um contato pelo nome.

O sistema deve garantir que as operações sejam feitas mantendo a imutabilidade dos dados.

Exemplos de Entrada e Saída:

1. **Adicionar Contato**
   - Entrada: `addContact([{ name: 'Ana', phone: '1234-5678' }], 'Pedro', '8765-4321')`
   - Saída: `[{ name: 'Ana', phone: '1234-5678' }, { name: 'Pedro', phone: '8765-4321' }]`

2. **Atualizar Contato**
   - Entrada: `updateContact([{ name: 'Ana', phone: '1234-5678' }], 'Ana', '0000-0000')`
   - Saída: `[{ name: 'Ana', phone: '0000-0000' }]`

3. **Remover Contato**
   - Entrada: `removeContact([{ name: 'Ana', phone: '1234-5678' }], 'Ana')`
   - Saída: `[]`

Desafio 2: Biblioteca de Livros

Requisitos:

Crie uma aplicação onde você pode:

1. Adicionar um novo livro com título, autor e status "disponível" por padrão.
2. Atualizar o status de um livro existente (ex.: de "disponível" para "emprestado").
3. Remover um livro pelo título.

O sistema deve garantir que as operações sejam feitas mantendo a imutabilidade dos dados.

Exemplos de Entrada e Saída:

1. **Adicionar Livro**
   - Entrada: `addBook([{ title: 'JavaScript Avançado', author: 'Fulano', status: 'disponível' }], 'Imutabilidade em JS', 'Beltrano')`
   - Saída: `[{ title: 'JavaScript Avançado', author: 'Fulano', status: 'disponível' }, { title: 'Imutabilidade em JS', author: 'Beltrano', status: 'disponível' }]`

2. **Atualizar Status**
   - Entrada: `updateBookStatus([{ title: 'JavaScript Avançado', author: 'Fulano', status: 'disponível' }], 'JavaScript Avançado', 'emprestado')`
   - Saída: `[{ title: 'JavaScript Avançado', author: 'Fulano', status: 'emprestado' }]`

3. **Remover Livro**
   - Entrada: `removeBook([{ title: 'JavaScript Avançado', author: 'Fulano', status: 'emprestado' }], 'JavaScript Avançado')`
   - Saída: `[]`