

Exercício React elements JSX

início do Exercício

1. Criar o workspace

```
npx create-react-app task-manager  
cd task-manager
```

2. Instalar Dependências Adicionais Caso o projeto use bibliotecas extras, como prop-types:

```
npm install prop-types
```

3. Rodar o Projeto

```
npm start
```

Criando Elementos Container e Componentes Pai

1. No arquivo src/App.js, remova o conteúdo padrão e crie um layout com dois containers:

- Um cabeçalho para o título do projeto.
- Uma seção para exibir as tarefas.

2. Dentro do cabeçalho, insira o título do projeto usando um componente Header. Crie este componente em um novo arquivo src/components/Header.js.

3. Dentro da seção, use um componente TaskContainer que será responsável por exibir a lista de tarefas. Crie este componente em src/components/TaskContainer.jsx.

4. Estrutura esperada em JSX (no App.js):

```
import React from 'react';  
import Header from './components/Header';  
import TaskContainer from './components/TaskContainer';  
  
function App() {  
  return (  
    <div className="App">  
      <Header />  
      <TaskContainer />  
    </div>  
  );  
}
```

```
    </div>
  );
}

export default App;
```

2. Utilizando React Fragment e Elementos Aninhados

1. No arquivo TaskContainer.js, use React.Fragment para envolver os itens sem adicionar um nó extra no DOM.
2. Adicione uma lista de tarefas fictícias diretamente no componente como elementos JSX. Estruture-as como elementos div aninhados dentro de um React.Fragment.
3. Exemplo para referência:

```
import React from 'react';

function TaskContainer() {
  return (
    <React.Fragment>
      <div className="task">
        <h3>Tarefa 1</h3>
        <p>Descrição da Tarefa 1</p>
      </div>
      <div className="task">
        <h3>Tarefa 2</h3>
        <p>Descrição da Tarefa 2</p>
      </div>
    </React.Fragment>
  );
}

export default TaskContainer;
```

Criando e Renderizando um Arquivo JSX

1. Crie um novo arquivo chamado src/components/Task.js. Este componente deve receber props contendo o título e a descrição de uma tarefa e renderizá-los.

- Exemplo de código do Task.jsx:

```
import React from 'react';
import PropTypes from 'prop-types';

function Task({ title, description }) {
  return (
    <div className="task">
      <h3>{title}</h3>
```

```
        <p>{description}</p>
      </div>
    );
  }

Task.propTypes = {
  title: PropTypes.string.isRequired,
  description: PropTypes.string.isRequired,
};

export default Task;
```

2. Modifique o TaskContainer para usar o componente Task em vez de criar os elementos diretamente.

- Atualize TaskContainer.js:

```
import React from 'react';
import Task from './Task';

function TaskContainer() {
  const tasks = [
    { id: 1, title: 'Tarefa 1', description: 'Descrição da Tarefa 1' },
    { id: 2, title: 'Tarefa 2', description: 'Descrição da Tarefa 2' },
  ];

  return (
    <div>
      {tasks.map((task) => (
        <Task key={task.id} title={task.title} description={task.description} />
      ))}
    </div>
  );
}

export default TaskContainer;
```