

Explicação dos console.log

1. No início da função carregarTarefas():

javascript

```
console.log('Carregando tarefas...');
```

- **Explicação:** Indica que a função para carregar as tarefas está sendo chamada. Isso ajuda a confirmar quando a atualização da lista de tarefas está em andamento.

2. Dentro do loop forEach ao criar uma nova tarefa:

javascript

```
console.log('Criando tarefa:', tarefa);
```

- **Explicação:** Mostra as informações da tarefa que está sendo criada e adicionada ao contêiner. Útil para depuração para verificar quais tarefas estão sendo processadas.

3. No evento de clique do botão de excluir:

javascript

```
console.log('Excluir tarefa com ID:', id);
```

```
console.log('Tarefa excluída:', id);
```

- **Explicação:** Mostra o ID da tarefa que será excluída e confirma que a tarefa foi removida com sucesso.

4. No evento de clique do botão de editar:

javascript

```
console.log('Editar tarefa com ID:', id);
```

```
console.log('Tarefa atualizada:', tarefa);
```

- **Explicação:** Indica qual tarefa está sendo editada e mostra a tarefa atualizada após a edição.

5. Quando a página é carregada:

javascript

```
console.log('Página carregada');
```

- **Explicação:** Confirma que o carregamento da página foi concluído e que a função para carregar as tarefas foi chamada.

6. Quando uma nova tarefa é adicionada via formulário:

javascript

```
console.log('Nova tarefa adicionada:', novaTarefa);
```

- **Explicação:** Mostra as informações da nova tarefa que foi adicionada ao array e exibida na lista.

Esses console.log ajudam a rastrear a execução do código e a entender o fluxo de dados e as interações do usuário com a página.

Vamos detalhar cada parte do código:

javascript

```
const index = tarefas.findIndex(t => t.id === id);
```

- **Explicação:** findIndex é um método de array que retorna o índice do primeiro elemento que satisfaz a condição fornecida.
 - `t => t.id === id` é uma função de flecha que verifica se o id da tarefa atual (t.id) é igual ao id que você deseja excluir.
 - `const index` armazena o índice encontrado da tarefa que corresponde ao id especificado. Se não encontrar, `index` será -1.

javascript

```
if (index > -1) {
```

- **Explicação:** Verifica se a tarefa foi encontrada. findIndex retorna -1 se nenhum elemento atender à condição.
 - Se `index` for maior que -1, isso significa que a tarefa foi encontrada no array.

javascript

```
tarefas.splice(index, 1); // Remove a tarefa do array
```

- **Explicação:** splice é um método de array que modifica o array original.
 - `index` é o ponto de partida onde a remoção começa.

- 1 é o número de elementos a serem removidos a partir do índice especificado.
- Assim, splice(index, 1) remove exatamente um elemento da posição index do array tarefas.

javascript

carregarTarefas(); // Atualiza a lista de tarefas

- **Explicação:** Após a remoção da tarefa, você chama carregarTarefas() para atualizar a exibição das tarefas na página.
 - Isso garante que a lista de tarefas no DOM (Document Object Model) reflita o estado atual do array tarefas após a exclusão.
-

Trecho de código que cria um objeto novaTarefa, que será adicionado ao array tarefas. Vamos dividir a explicação:

1. Criando o Objeto novaTarefa

javascript

```
const novaTarefa = {
  id: tarefas.length ? tarefas[tarefas.length - 1].id + 1 : 1,
  descricao: descricao
};
```

Aqui, um novo objeto novaTarefa é criado com dois campos:

- id: Um identificador único para a nova tarefa.
- descricao: A descrição da nova tarefa, que vem do input fornecido pelo usuário.

2. Definição do Campo id

javascript

```
id: tarefas.length ? tarefas[tarefas.length - 1].id + 1 : 1
```

O valor de id é definido de acordo com a seguinte lógica:

- **tarefas.length:** Verifica se o array tarefas já contém alguma tarefa.
 - Se tarefas.length for maior que 0 (ou seja, o array tem elementos), o código executa a parte antes do : (verdadeira).

- Se `tarefas.length` for 0 (ou seja, o array está vazio), o código executa a parte depois do `:` (falsa).

- **Se o array tarefas não estiver vazio:**

javascript

`tarefas[tarefas.length - 1].id + 1`

- `tarefas[tarefas.length - 1]` acessa a última tarefa no array.
- `.id` pega o id da última tarefa.
- `+ 1` adiciona 1 ao id da última tarefa para criar um novo id único.
- Exemplo: Se o último id for 3, o novo id será 4.

- **Se o array tarefas estiver vazio:**

javascript

1

- Se não houver tarefas no array (ou seja, `tarefas.length` é 0), o id da nova tarefa será 1.

3. Definição do Campo descricao

javascript

`descricao: descricao`

- A `descricao` da nova tarefa é atribuída a partir da variável `descricao`, que é preenchida pelo valor digitado pelo usuário no campo de input.

Resumo

- **Objetivo:** O trecho cria um novo objeto `novaTarefa` com um id único e uma `descricao` fornecida pelo usuário.
- **Lógica:** Se o array `tarefas` já tem elementos, o id da nova tarefa é o id da última tarefa + 1. Se o array estiver vazio, o id começa em 1.