Correção de Quest:
JavaScript Intermediário

Aluno: Daniela Oliveira - Turma 19

https://github.com/DanielaOliveiraDsg/quest-js-intermediario-form-validation

Requisitos Obrigatórios:

- ~~A Validação deve ser feita com JS Vanilla.~~
- ~~O fundo do formulário deve ser feito usando a imagem em anexo na aula.~~
- ~~Ao clicar para enviar o formulário, se caso algum campo não estiver preenchido, a borda do input deve ficar vermelha e uma mensagem de "campo obrigatório" deve aparecer embaixo do campo que não foi preenchido, conforme o Figma.~~

Pontuações:

1. Em um formulário, tanto o **input** do tipo “**button**” quanto a **tag button** com o atributo **type** definido como “**button**” podem ser usados para criar botões em formulários HTML. No entanto, a **tag button** oferece mais flexibilidade em termos de personalização e permite incluir outros elementos dentro do botão. Já o **input** é mais adequado quando se precisa enviar um valor junto com o formulário.

   Neste formulário, como o botão “enviar” apenas envia os dados dos outros inputs para a validação, o mais

correto seria utilizar a **tag button** com o atributo **type** definido como "**submit**", exatamente como você fez.
Ex:

```
<p>*campos obrigatórios</p>
<button type="submit" id="btn">Enviar</button>
form>
```

Por fim, só bastaria colocar as suas **Classes** ou **ID's** que ficaria igual. Caso não ficasse, bastaria consertar alguns pequenos pontos com o CSS.

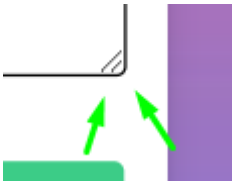
2. Aqui **foi ótimo ter feito** uso da **Tag TextArea**:

```
<p class="campo-obrigatorio">campo obrigatório</p>

<textarea id="mensagem" class="input-text message"
<p class="campo-obrigatorio">campo obrigatório</p>
<p>*campos obrigatórios</p>
```

Essa tag é especialmente recomendada para uma *área de texto* editável em uma página web. Ela permite que usuários possam escrever e manipular um texto mais longo, como em um **formulário de comentários** ou em uma **caixa de texto para a digitação de mensagens**.

3. Na **tag TextArea**, existe uma propriedade CSS chamada **Resize**, ela é responsável por esse pedacinho aqui da tag:

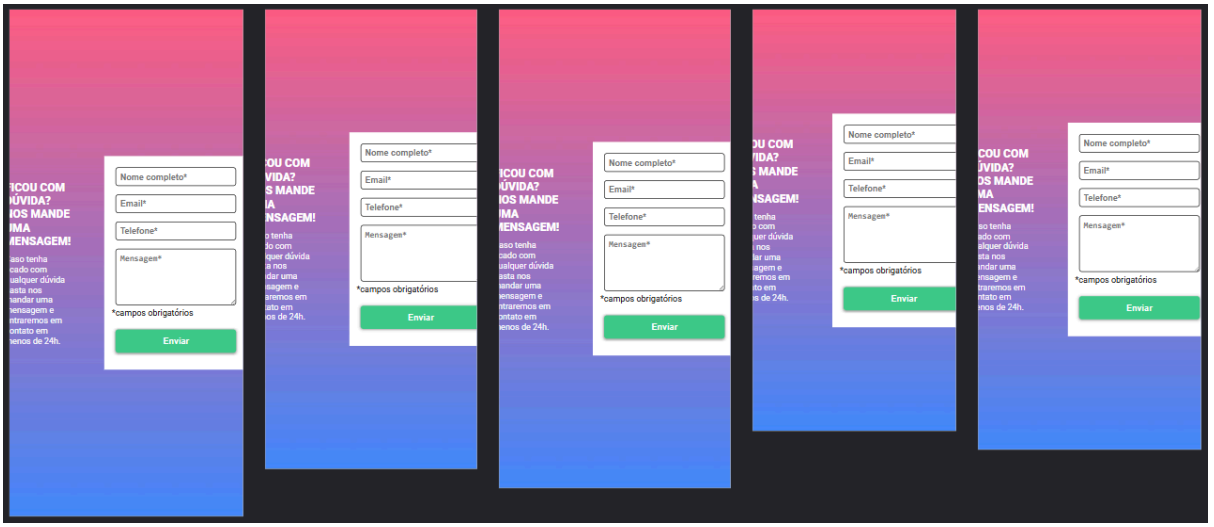


Avalia bem o contexto, se não for útil é bom que seja removida pois o usuário pode acabar manipulando de forma errada.

EX:

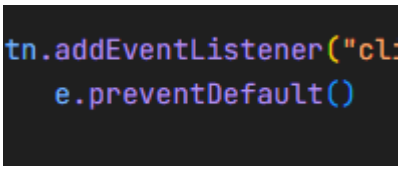
Para remover, basta declarar a propriedade **Resize** como **None**.

4. Percebi que seu projeto **não foi bem adaptado às diferentes telas**. Apesar de não ser obrigatório, garantir que seus projetos sejam responsivos é uma prática importante, pois muitos usuários acessam aplicativos em diferentes telas.



5. Muito bom ter usado o **preventDefault()** para interceptar o evento de “Submit” (envio de formulário)

e prevenir o comportamento padrão do navegador de enviar o formulário.



6. A lógica do seu código JS está **muito boa**. Organizou tudo muito bem e fez um ótimo uso do **forEach** - que foi ensinado nas aulas anteriores - para manipular os elementos. Essa abordagem é especialmente útil quando lidamos com validações em **múltiplos campos**. Mandou bem!

No fim, completou os desafios de JavaScript Intermediário, tá mandando bem!
Anota essas observações, se preferir, e vai treinando tudo isso. Usa essas mesmas observações nos próximos projetos que vão te ajudar bastante.
Como desafio final, tenta refatorar esse seu código usando essas dicas, com a prática você pega o jeito da coisa.

~ Boa sorte, Daniela! ☕