

Aula 05 - Eventos do formulário

Manipulação de Eventos de Formulário em JavaScript

1 Introdução

Os formulários em HTML permitem a interação do usuário com a página, possibilitando o envio de informações. No entanto, para garantir que os dados sejam válidos antes do envio, usamos JavaScript para manipular eventos como `submit` e `change`.

Nesta aula, vamos aprender a:

- ✓ Lidar com os eventos `submit` e `change`.
- ✓ Criar validações dinâmicas para os campos.
- ✓ Usar `preventDefault()` para evitar envios incorretos.
- ✓ Criar mensagens de erro personalizadas.
- ✓ Adicionar os dados enviados a uma tabela.

2 Manipulando Eventos de Formulário

1. O evento `submit`

O evento `submit` é disparado quando um formulário é enviado. Podemos interceptá-lo para validar os campos antes que os dados sejam processados pelo servidor.

Exemplo: Capturando o evento `submit`

```
<form id="meuFormulario">
  <label for="nome">Nome:</label>
  <input type="text" id="nome" name="nome">

  <button type="submit">Enviar</button>
```

```
</form>

<script>
  document.getElementById("meuFormulario").addEventListener("submit",
function(event) {
  alert("Formulário enviado!");
});
</script>
```

i Neste exemplo, quando o usuário clica em "Enviar", um alerta aparece antes do envio do formulário.

2. O evento **change**

O evento **change** é disparado quando o usuário altera o valor de um campo e sai do foco dele.

Exemplo: Capturando **change para validar um campo de e-mail**

```
<label for="email">E-mail:</label>
<input type="email" id="email" name="email">

<script>
  document.getElementById("email").addEventListener("change", function
(event) {
    let email = event.target.value;
    if (!email.includes("@")) {
      alert("E-mail inválido!");
    }
  });
</script>
```

i Se o usuário digitar um e-mail sem "@" e sair do campo, um alerta será exibido.

3 Usando **preventDefault()** para Evitar Envio Incorreto

O método `preventDefault()` impede que o formulário seja enviado até que os dados sejam validados corretamente.

```
<form id="formPrevent">
  <label for="nome">Nome:</label>
  <input type="text" id="nomePrevent" name="nome">
  <button type="submit">Enviar</button>
</form>

<script>
  document.getElementById("formPrevent").addEventListener("submit", function(event) {
    let nome = document.getElementById("nomePrevent").value;

    if (nome === "") {
      event.preventDefault(); // Impede o envio
      alert("O campo Nome não pode estar vazio!");
    }
  });
</script>
```

i Se o usuário tentar enviar sem preencher o nome, o envio será bloqueado e um alerta será exibido.

4 Criando Mensagens de Erro Personalizadas

Podemos exibir mensagens de erro personalizadas abaixo dos campos ao invés de usar alertas.

```
<form id="formErro">
  <label for="emailErro">E-mail:</label>
  <input type="email" id="emailErro" name="email">
  <span id="erroEmail" style="color: red;"></span>

  <button type="submit">Enviar</button>
</form>

<script>
```

```

document.getElementById("formErro").addEventListener("submit", function(event) {
    let email = document.getElementById("emailErro").value;
    let erroEmail = document.getElementById("erroEmail");

    if (!email.includes("@")) {
        event.preventDefault();
        erroEmail.innerText = "E-mail inválido!";
    } else {
        erroEmail.innerText = ""; // Remove erro se estiver correto
    }
});
</script>

```

i Se o e-mail for inválido, a mensagem "E-mail inválido!" aparecerá abaixo do campo.

5 Adicionando os Dados Enviados em uma Tabela

Após a validação, podemos capturar os dados do formulário e adicioná-los dinamicamente a uma tabela na página.

```

<form id="formTabela">
  <label for="nomeTabela">Nome:</label>
  <input type="text" id="nomeTabela" name="nome">

  <label for="emailTabela">E-mail:</label>
  <input type="email" id="emailTabela" name="email">

  <button type="submit">Adicionar à Tabela</button>
</form>

<table border="1" id="tabela">
  <thead>
    <tr>
      <th>Nome</th>
      <th>E-mail</th>
    </tr>

```

```

</thead>
<tbody>
  <!-- Os dados serão adicionados aqui -->
</tbody>
</table>

<script>
  document.getElementById("formTabela").addEventListener("submit", function(event) {
    event.preventDefault(); // Impede o envio do formulário

    let nome = document.getElementById("nomeTabela").value;
    let email = document.getElementById("emailTabela").value;

    if (nome === "" || email === "") {
      alert("Preencha todos os campos!");
      return;
    }

    let tabela = document.getElementById("tabela").getElementsByTagName("tbody")[0];

    let novaLinha = tabela.insertRow();
    let celularNome = novaLinha.insertCell(0);
    let celularEmail = novaLinha.insertCell(1);

    celularNome.innerText = nome;
    celularEmail.innerText = email;

    // Limpar os campos após adicionar à tabela
    document.getElementById("formTabela").reset();
  });
</script>

```

i Quando o usuário preencher os campos e clicar no botão, os dados serão adicionados à tabela sem recarregar a página.

Exercícios para os Alunos

- 1 Crie um formulário de login com campos de usuário e senha e valide se ambos foram preenchidos antes de enviar.
 - 2 Crie um formulário com um campo de idade e exiba uma mensagem de erro se o usuário inserir um número menor que 18.
 - 3 Crie um formulário para cadastro de produtos com nome e preço. Valide se os campos foram preenchidos e adicione os produtos a uma tabela.
-

Conclusão

- ✓ O evento `submit` permite capturar o envio do formulário e validar os dados antes de serem enviados.
- ✓ O evento `change` permite validar os campos assim que o usuário modifica um valor.
- ✓ O método `preventDefault()` impede o envio do formulário até que os dados estejam corretos.
- ✓ Podemos criar mensagens de erro personalizadas para melhorar a experiência do usuário.
- ✓ Os dados podem ser adicionados dinamicamente a uma tabela para visualização imediata.