

MANIPULANDO FORMULÁRIOS

Para criar um formulário em HTML, você usa o `<form>`

form

```
1 <form action="/signup" method="post" id="signup">
2 </form>
```

O elemento

tem dois atributos importantes: `action` e `method`.

O `action` especifica uma URL que processará o envio do formulário. Neste exemplo, a ação é a `/signupURL`. O `method` especifica o método HTTP com o qual enviar o formulário.

Normalmente, o método é `post` ou `get`.

JavaScript usa o `HTMLFormElement` para representar um formulário. O `HTMLFormElement` tem as seguintes propriedades que correspondem aos atributos HTML:

`action`— é a URL que processa os dados do formulário. `method`— é o método HTTP que é equivalente ao `method` do elemento.

O `HTMLFormElement` também fornece os seguintes métodos úteis:

`submit()`— enviar o formulário. `reset()`— redefinir o formulário.

Referenciando o formulário

Para fazer referência ao elemento, você pode usar métodos de seleção do DOM, como `getElementById()`:

getElement

```
1 const form = document.getElementById('subscribe');
```

Um documento HTML pode ter vários formulários. A propriedade `document.forms` retorna uma coleção de formulários (`HTMLFormControlsCollection`) no documento:

forms

```
1 document.forms;  
2  
3 document.forms[0];
```

Enviando um formulário

Normalmente, um formulário tem um botão de envio. Quando você clica nele, o navegador envia os dados do formulário para o servidor. Para criar um botão de envio, você usa `<input>` ou `<button>` com o tipo submit:

submit

```
1 <input type="submit" value="Subscribe">  
2  
3 //ou  
4  
5 <button type="submit">Subscribe</button>
```

Quando você envia o formulário, o evento submit é acionado antes que a solicitação seja enviada ao servidor. Isso lhe dá a chance de validar os dados do formulário. Se os dados do formulário forem inválidos, você poderá interromper o envio do formulário.

Para anexar um ouvinte de evento ao submitevento, você usa o método `addEventListener()` do elemento de formulário da seguinte forma:

submit

```
1 const form = document.getElementById('signup');  
2  
3 form.addEventListener('submit', (event) => {  
4   // trabalha com os dados  
5 });
```

Para impedir que o formulário seja enviado, você chama o método `preventDefault()` dentro do manipulador de eventos assim:

prevent

```
1 form.addEventListener('submit', (event) => {  
2   // Cancela o envio
```

```
3     event.preventDefault();
4 });
```

Para enviar o formulário em JavaScript, você chama o método `submit()` do objeto de formulário:

```
submit

1 form.submit();
```

Observe que o `form.submit()` não aciona o evento `submit`. Portanto, você deve sempre validar o formulário antes de chamar esse método.

Acessando campos do formulário

Para acessar campos de formulário, você pode usar métodos DOM como `getElementsByName()`, `getElementById()`, `querySelector()`, etc.

Além disso, você pode usar a propriedade `elements` do objeto `form`. A propriedade `form.elements` armazena uma coleção de elementos do formulário.

O JavaScript permite acessar um elemento por índice, id ou nome. Suponha que você tenha o seguinte formulário de inscrição com dois elementos:

```
form

1 <form action="signup.html" method="post" id="signup">
2   <h1>Cadastro</h1>
3   <div class="field">
4     <label for="name">Nome:</label>
5     <input type="text" id="name" name="name" placeholder="Nome
6 completo" />
7   </div>
8   <div class="field">
9     <label for="email">Email:</label>
10    <input type="text" id="email" name="email" placeholder="Insira
11 seu e-mail" />
12  </div>
    <button type="submit">Cadastrar</button>
  </form>
```

Para acessar os elementos do formulário, você obtém primeiro o elemento do formulário:

```
const form = document.getElementById('signup');
```

E use index, id ou name para acessar o elemento.

acesso

```
1 // O código a seguir acessa o primeiro elemento de formulário:
2
3 form.elements[0]; // index
4 form.elements['name']; // name
5 form.elements['name']; // id (name & id são os mesmos neste caso)
6
7 // Segundo elemento
8
9 form.elements[1]; // index
10 form.elements['email']; // name
11 form.elements['email']; // id
```

Após acessar um campo de formulário, você pode utilizar a propriedade value para acessar seu valor, por exemplo:

value

```
1 const form = document.getElementById('signup');
2 const name = form.elements['name'];
3 const email = form.elements['email'];
4
5 // pegando o valor
6 let fullName = name.value;
7 let emailAddress = email.value;
```



Exercício

- Crie um formulário
- Crie um input text para item
- Crie um submit com o valor cadastrar
- Ao clicar em cadastrar no formulário o valor do item deve ser adicionado em um li abaixo do form
- Crie um link com o texto excluir na frente de cada item, ao clicar em excluir o item desaparece da tela