

Aula 10

Desenvolvimento JavaScript III

HTML DOM

Utilizando Formulários

Quando uma página da web é carregada pelo navegador, é criado um Document Object Model (DOM). Esse documento pode sofrer transformações e alterações, mas quando a página for recarregada, estará de volta ao estado inicial, pois o DOM é uma representação do código da página e suas alterações não alteram o código em si.

O DOM pode ser acessado e manipulado utilizando a linguagem de programação JavaScript. Com essa funcionalidade, podemos “pegar” informações da página e transformá-las ou utilizá-las como desejarmos.

Os métodos DOM mais utilizados são: `getElementById`, usado para encontrar um elemento que possui uma ‘id’ específica; e `innerHTML`, utilizado para obter ou substituir conteúdo de um elemento.

Validando Formulários

Vamos utilizar o formulário de cadastro que criamos na Locadora 2000. Para que tudo funcione corretamente, vamos adicionar nome ao <form> e a cada um dos elementos <input>.

```
<div class="container mb-5">
  <form name="cadastrar"> <!-- Adicionar nome ao form-->

  <input type="text" id="codFilme" class="form-control"
  placeholder="00" name="codigo"> <!-- Adicionar nome ao input
```

Precisamos criar um documento JavaScript para colocar nosso código que vai validar as respostas do formulário. Não esqueça de colocar o script com ‘src’ para anexar ao arquivo externo.

```
<script type="text/javascript" src="javascript.js"></script>
```

Funções e Condicionais

Na programação muitas vezes precisamos utilizar um mesmo código várias vezes. Então, a fim de evitar repetições em nosso programa, criamos uma função. Uma função nada mais é a encapsulação de um pedaço de código, esse código não será executado ao iniciar a execução do programa, somente quando for invocado/chamado por outro trecho do código.

Para criar a função a seguir, vamos escrevê-la em nosso arquivo JavaScript externo, seguindo a seguinte sintaxe:

```
function validarForm() {  
  
}
```

O texto em verde, é o nome da função, os parênteses servem para incluir dados quando necessário e as chaves delimitam o bloco de código.

Vamos criar dentro da função uma variável chamada validaCod, ela servirá para validar se o valor do código digitado no formulário está correto. O valor atribuído à variável é definido por:

`document.forms["nomeDoForm"]["nomeDoInput"].value`

Esse comando é utilizado para captar o valor que foi preenchido no input dentro do formulário, por isso utiliza-se o nome dos mesmos entre colchetes.

```
function validarForm() {  
  
    let validaCod = document.forms["cadastrar"]["codigo"].value;  
  
}
```

Agora que o valor de preenchimento está dentro da variável, podemos testar se ele foi realmente preenchido. Se o valor estiver vazio, quer dizer que não foi preenchido e devemos alertar o usuário de preenchê-lo corretamente.

Para isso, vamos adicionar um bloco condicional dentro de nossa função.

Uma Condicional é uma estrutura que executa uma afirmação, se determinada condição for verdadeira o conteúdo do bloco será executado, se for falsa, não será.

Adicione um bloco condicional "If", seguindo a seguinte sintaxe:

```
if (true) {  
  
}
```

O que fica contido dentro dos parênteses é a informação que deve ser verdadeira, para que o conteúdo dentro das chaves seja executado.

Vamos adicionar dentro dos parênteses a condição que testará se o valor da variável está vazio ou não:

```
if (validaCod == "") {  
  
}
```

Se o valor estiver de fato vazio, precisamos alertar o usuário. Então vamos adicionar uma caixa de alerta e definir que a função voltará falsa, desta maneira o formulário não será enviado, pois há um campo em branco.

```
if (validaCod == "") {  
    alert("O código deve ser preenchido");  
    return false;  
}
```

Ao final, nossa função estará assim:

```
function validarForm() {  
    let validaCod = document.forms["cadastrar"]["codigo"].value;  
  
    if (validaCod == "") {  
        alert("O código deve ser preenchido");  
        return false;  
    }  
}
```

Repita esse mesmo processo para cada um dos inputs que temos no nosso formulário. Crie uma variável para cada um deles, obtenha o valor de preenchimento utilizando `document.forms[].value` e crie um bloco condicional, para testar se o preenchimento está vazio ou não.

Validação Automática

Utilizando propriedades do HTML, podemos verificar de maneira automática se um campo foi preenchido ou não, sem a necessidade de criar uma função em JavaScript. Para isto, basta adicionar o atributo 'required' à tag.

```
<input type="text" id="codFilme" class="form-control" placeholder="00"
name="codigo" required> <!-- Adicionar required ao input-->
```

Inserindo na tabela

O primeiro passo para inserir as informações registradas no formulário para a tabela, é preparar onde será inserido. Por isso, crie uma nova <tr> abaixo da última <tr> de sua tabela. Cuide para que esta tag esteja **antes** do fechamento da <tbody> e da <table>.

É importante colocar uma id para a tag, vamos precisar utilizá-la nos próximos passos.

```
<tr id="inserirTabela">

</tr>
```

Com o lugar de inserção pronto, precisamos configurar o formulário para que ele não recarregue a página. Os formulários HTML tem por padrão o comportamento de recarregar a página ao submeter suas informações para envio. Como vamos utilizar o DOM, a página não pode ser recarregada, ou nossas informações irão se perder.

Adicione o atributo "onsubmit", ele define o que acontecerá quando o evento de submeter for disparado. Como valor desse atributo, digite a função event.preventDefault(). Desta maneira, o evento que dispara a submissão vai ser parado antes de ocorrer, fazendo com que a página não seja recarregada.

```
<form name="cadastrar" onsubmit="event.preventDefault()">
```

Vamos para nosso arquivo JavaScript: crie uma nova função chamada `insere()`.

```
function insere(){  
  
}
```

Assim como fizemos na outra função, precisamos definir as variáveis para cada campo input do formulário e pegar seu valor de preenchimento.

```
let cod      = document.forms["cadastrar"]["codigo"].value;  
let tit      = document.forms["cadastrar"]["titulo"].value;  
let dir      = document.forms["cadastrar"]["diretor"].value;  
let data     = document.forms["cadastrar"]["data"].value;  
let rad      = document.forms["cadastrar"]["radio"].value;  
let ator     = document.forms["cadastrar"]["ator"].value;  
let imdb     = document.forms["cadastrar"]["imdb"].value;
```

Criamos uma nova variável chamada 'inserir'. Seu valor será definido pelo conteúdo contido dentro da tag `<tr>` cujo id é "inserirTabela", a qual definimos nos passos anteriores.

Agora podemos utilizar o comando `innerHTML` para determinar o conteúdo que será adicionado dentro da tag `<tr>`.

Note que o conteúdo inserido não é delimitado por aspas simples, mas sim por crase. Essa técnica se chama Template Strings (ou Template Literals), que consiste em Strings (variáveis de texto) que permitem expressões embutidas e utilização de placeholder. Uma espécie de concatenação do texto.

Desta forma, o texto é envolvido com acento grave (crase) e para mostrar o valor de uma variável criada anteriormente é necessário utilizar cifrão seguido de chaves:

`$\${exemploNomeVariavel}$`

```

let inserir = window.document.getElementById("inserirTabela");
inserir.innerHTML = `<th scope="row">${cod}</th>`;
inserir.innerHTML += `<td>${tit}</td>`;
inserir.innerHTML += `<td>${dir}</td>`;
inserir.innerHTML += `<td>${data}</td>`;
inserir.innerHTML += `<td>${rad}</td>`;
inserir.innerHTML += `<td>${ator}</td>`;
inserir.innerHTML += `<td>${imdb}</td>`;

```

Note também que a partir da terceira linha do código, não está usando apenas o símbolo de atribuição '=' sozinho, mas com o sinal '+' junto dele. Essa junção indica uma **incrementação**, isso significa que as linhas seguintes serão adicionadas abaixo sem apagar/substituir as anteriores.

Por fim, vamos voltar para a página HTML e inserir o atributo onclick no botão de envio do formulário, o valor do atributo será "insere()" que é o nome da função que acabamos de criar. Desta maneira, ao clicar no botão a programação contida em nossa função será executada e as informações digitadas no formulário serão inseridas na tabela.

```

<!-- Botão Enviar -->
<div class="col-auto">
  <button class="btn btn-primary mb-3" onclick="insere()">
    Cadastrar</button>
</div>

```

Tarefa de Casa

Para treinar o conhecimento adquirido nesta aula, crie um formulário de cadastro sobre o tema de sua escolha (livros, séries, produtos, receitas, etc), com no mínimo 3 itens de cadastramento. E a cada novo cadastramento, exiba o conteúdo em uma lista.

Exemplo:

Vou criar um formulário de cadastramento de cachorros para consultar em uma PetShop. O formulário precisará ter os seguintes campos:

- Nome do Cachorro
- Nome do Dono
- Telefone para contato

- Dia do agendamento