

Linguagem JavaScript (Validação Formulário)

Objeto form

Um dos objetos mais importantes do JavaScript é o que permite a manipulação de formulários HTML, muito utilizado para validar informações e realizar outros processamentos. O objeto `form` representa um formulário contido em uma página HTML e, por meio dele, é possível acessar todos os seus campos e valores e manipulá-los de acordo com a necessidade. Os principais recursos do `form` são:

Comando	Descrição
<code>form.action</code>	Representa o atributo <code>action</code> informado na marcação <code><form></code> .
<code>form.name</code>	Representa o nome do formulário.
<code>form.method</code>	Representa o atributo <code>method</code> informado na marcação <code><form></code> .
<code>form.submit()</code>	Envia o formulário e suas informações para o destino informado no atributo <code>action</code> .
<code>form.nomecampo</code>	Permite acessar as informações do campo cujo nome tenha sido informado no lugar da propriedade <code>nomecampo</code> .

Linguagem JavaScript (Validação Formulário)

Montando um formulário HTML

O formulário HTML que será utilizado nos próximos tópicos será um formulário simples de cadastro, com algumas características. Ele terá os atributos `method` e `action` da marcação `<form>`, terá o seu botão de enviar definido como um botão comum (`type=BUTTON`) e receberá o nome `meu_form` por meio do atributo `name` utilizado na marcação `<form>`. Desta forma o botão de envio do formulário, ao invés de enviar os dados para o servidor, chamará uma função JavaScript para validar o formulário, e, se tudo estiver correto, encaminhará o mesmo para o servidor processar.

Linguagem JavaScript (Validação Formulário)

```
formularioemJS.html
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4 <meta charset="utf-8">
5 <title>Formulário em JavaScript!</title>
6 </head>
7 <body>
8 <script>
9     function validarFormulario(){
10         /* Código de validação */
11     }
12 </script>
13 <form name="meu_form" method="get" action="?">
14     Nome completo:
15     <input name="campo_nome" type="text">
16     <br />
17     Sexo:
18     <input name="campo_sexo" type="radio" value="M">Masculino
19     <input name="campo_sexo" type="radio" value="F">Feminino
20     <br />
21     <input name="campo_maior" type="checkbox" value="maior">
22     Declaro ser maior de 18 anos
23     <br />
```

Linguagem JavaScript (Validação Formulário)

```
24 Como conheceu o site:
25 <select name="campo_como">
26   <option value="Nenhum" default>Selecione...</option>
27   <option value="Ind">Indicação de amigo</option>
28   <option value="Out">Outro site</option>
29   <option value="Goo">Google</option>
30 </select>
31 <br />
32 <input type="reset" value="limpar dados">
33 <input type="button" value="Enviar" onClick="validarFormulario();">
34 </form>
35 </body>
36 </html>
```

Nome completo:

Sexo: ☐ Masculino ☐ Feminino

☐ Declaro ser maior de 18 anos

Como conheceu o site:

Linguagem JavaScript (Validação Formulário)

Capturando dados de campos texto

Os campos de textos são os mais simples e intuitivos de serem manipulados por JavaScript, por permitirem que seus valores sejam acessados diretamente pela propriedade `value`. Mas antes, para que possamos acessar as propriedades de um campo de formulário, é necessário primeiro acessar o formulário em si via JavaScript.

Os formulários HTML são disponibilizados no JavaScript por meio do objeto `document`, que representa os elementos da página web, mais precisamente por um array de formulários contido na propriedade `forms` deste objeto. Por este motivo, é necessário acessar o elemento `document`, sua propriedade `forms`, seguido do nome do formulário entre colchetes e aspas, indicando seu índice de acesso, construído na marcação `<form>`. Como o nome do formulário é `meu_form`, é este o valor que deve ser informado entre colchetes. O código-fonte a seguir demonstra como fazer este acesso completo para capturar o valor de um campo texto, no caso o campo nome, e exibi-lo na tela:

Linguagem JavaScript (Validação Formulário)

```
function validarFormulario() {  
    conteudo_nome = document.forms['meu_form'].campo_nome.value;  
    alert('O nome informado no formulário é: ' + conteudo_nome);  
}
```

Observe a instrução `document.forms['meu_form'].campo_nome.value`. Ela inicia acessando a página web por meio de `document`. Em seguida o formulário por meio de `forms['meu_form']`, seguido do nome do campo `campo_nome` e, por último, da propriedade que acessa o valor contido em um campo de texto, que é a propriedade `value`. Este valor é atribuído para a variável `conteudo_nome`, sendo impresso na tela pelo comando `alert` apresentado na linha seguinte, quando o botão Enviar for clicado. Antes de validarmos esse campo, vamos dar uma olhada em como acessar os valores preenchidos e/ou selecionados nos outros tipos de campos, via JavaScript.

Linguagem JavaScript (Validação Formulário)

Capturando dados de campos de seleção (combo box)

Com a mesma facilidade de acesso aos conteúdos dos campos de texto, é possível acessar o valor selecionado dentro de um campo de seleção (**marcação <select>**), por meio da mesma propriedade **value**.

Existe ainda uma propriedade chamada **selectedIndex** para os campos deste tipo, que permite saber o número da opção que foi escolhida, seguindo a ordem de apresentação no campo HTML. Esta alternativa pode ser útil em casos como este, para saber se o usuário selecionou alguma opção ou se manteve o campo sem nenhuma seleção, pois geralmente a primeira opção dos campos de seleção é justamente o texto **Selecione...**, que ocupa o índice **0** dessa listagem.

O código-fonte a seguir exhibe as duas propriedades comentadas neste tópico: o valor selecionado pelo usuário e o índice numérico que este valor possui na listagem de opções do campo:

```
function validarFormulario() {  
    conteudo_como = document.forms['meu_form'].campo_como.value;  
    numero_como = document.forms['meu_form'].campo_como.selectedIndex;  
    alert('A opção escolhida é: ' + numero_como + ': ' + conteudo_como);  
}
```

Linguagem JavaScript (Validação Formulário)

```
}
```

Vale a pena citar que quando se fala em informar o valor da opção selecionada, fala-se em exibir o valor preenchido no atributo `value` da opção em questão. Por exemplo, ao selecionar a opção `Google` no `campo_como`, este exemplo exibirá na tela o texto `3: Goo`, pois é o valor `Goo` que foi atribuído para esta opção na marcação existente.

```
function validarFormulario(){  
    numero_como = document.forms['meu_form'].campo_como.selectedIndex;  
    descricao_como = document.forms['meu_form'].campo_como.options[numero_como].text;  
    alert('A descrição da opção escolhida é: ' + descricao_como);  
}
```

O primeiro passo para exibir a descrição de uma opção em um campo de seleção é capturar a posição que foi selecionada. Neste caso, a informação pode ser acessada por meio da propriedade `selectedIndex` vista anteriormente, a qual foi armazenada na variável `numero_como` neste exemplo.

Em seguida, sabendo a posição numérica da opção selecionada, deve ser realizado um acesso à propriedade `options` do campo `campo_como`, que apresenta um array com a lista de opções escritas por extenso, e,

Linguagem JavaScript (Validação Formulário)

nessa lista, a posição numérica da opção selecionada deve ser informada entre colchetes. Como a posição numérica está contida na variável `numero_como`, basta informar essa variável entre colchetes. Por último, é necessário acessar a propriedade `text` da opção em questão.

Como você pôde perceber, são vários acessos, o que pode parecer complicado neste primeiro momento, mas fica bastante simples quando se analisa cada acesso isoladamente. Para deixar ainda mais claro, observe a ordem em que os acessos ocorrem:

```
document
```

Acessa a página web que contém o formulário.

```
document.forms['meu_form']
```

Acessa o formulário `meu_form`, que contém o campo.

```
document.forms['meu_form'].campo_como
```

Acessa o `campo campo_como`, que contém as opções de seleção.

```
document.forms['meu_form'].campo_como.options[numero_como]
```

Linguagem JavaScript (Validação Formulário)

Acessa o elemento que representa a opção de seleção na posição `numero_como`.

```
document.forms['meu_form'].campo_como.options[numero_como].text
```

Acessa a descrição da opção de seleção em questão.

Capturando dados em campos de marcação (checkbox)

Os campos de marcação também possuem características específicas para terem suas informações acessadas. Se o código JavaScript tentar acessar o seu valor utilizando a propriedade `value`, a mesma irá trazer o valor definido na caixa de marcação por meio do atributo `value`, independente se a mesma tiver sido marcada ou não pelo usuário.

Para saber se uma caixa de marcação foi assinalada, é necessário acessar a propriedade `checked` do campo em questão. Observe no código-fonte a seguir as duas situações comentadas neste tópico: o acesso ao valor definido pelo atributo `value` de um campo de marcação e, em seguida, o acesso à propriedade que indica se o mesmo está marcado ou não:

```
function validarFormulario() {  
    conteudo_maior = document.forms['meu_form'].campo_maior.value;
```

Linguagem JavaScript (Validação Formulário)

```
alert('O valor da caixa de marcação é: ' + conteudo_maior);  
marcacao_maior = document.forms['meu_form'].campo_maior.checked;  
alert('A caixa de marcação foi marcada? ' + marcacao_maior);  
}
```

Neste exemplo, note que o primeiro comando `alert` sempre apresentará o valor `maior` que foi definido no formulário HTML, independente se a caixa de marcação `Declaro ser maior de 18 anos` estiver marcada ou não. Para saber se está marcada, o segundo comando `alert` captura a propriedade `checked`, que retorna `true` caso a mesma esteja marcada, ou `false` caso contrário.

Capturando dados de campos de rádio (radio button)

De todos os tipos de campos existentes no HTML, o tipo rádio é o mais complexo de ser manipulado via JavaScript, pois as diferentes opções rádio para um mesmo campo devem ser definidas com o mesmo nome de campo.

Parece confuso, não é mesmo? Mas observe o exemplo apresentado, onde o campo de nome `campo_sexo` deve apresentar as opções `Masculino` e `Feminino` utilizando botões do tipo rádio. Notou que ambas apresentam o atributo `name="campo_sexo"`? Isso deve ser feito assim para que o navegador web possa

Linguagem JavaScript (Validação Formulário)

desmarcar uma opção quando outra for selecionada. Experimente marcar este campo clicando na bolinha que representa a opção **Feminino** e depois na bolinha que representa a opção **Masculino**. Você vai perceber que, automaticamente, a opção **Feminino** será desmarcada quando você clicar em **Masculino**.

A ideia envolvida na busca pela opção de rádio selecionada para um determinado campo inclui uma varredura em todas as opções de rádio existentes no formulário, referente ao campo em questão, verificando cada uma até encontrar qual foi selecionada. O exemplo a seguir demonstra esta funcionalidade, com alguns comentários na sequência:

```
function validarFormulario()
{
    total_opcoes = document.forms['meu_form'].campo_sexo.length;
    opcao_escolhida = null;
    for(i=0; i<total_opcoes; i++)
    {
        if(document.forms['meu_form'].campo_sexo[i].checked == true){
            opcao_escolhida = document.forms['meu_form'].campo_sexo[i].value;
        }
    }
}
```

Linguagem JavaScript (Validação Formulário)

```
}  
    alert('O sexo marcado no formulário é: ' + opcao_escolhida);  
}
```

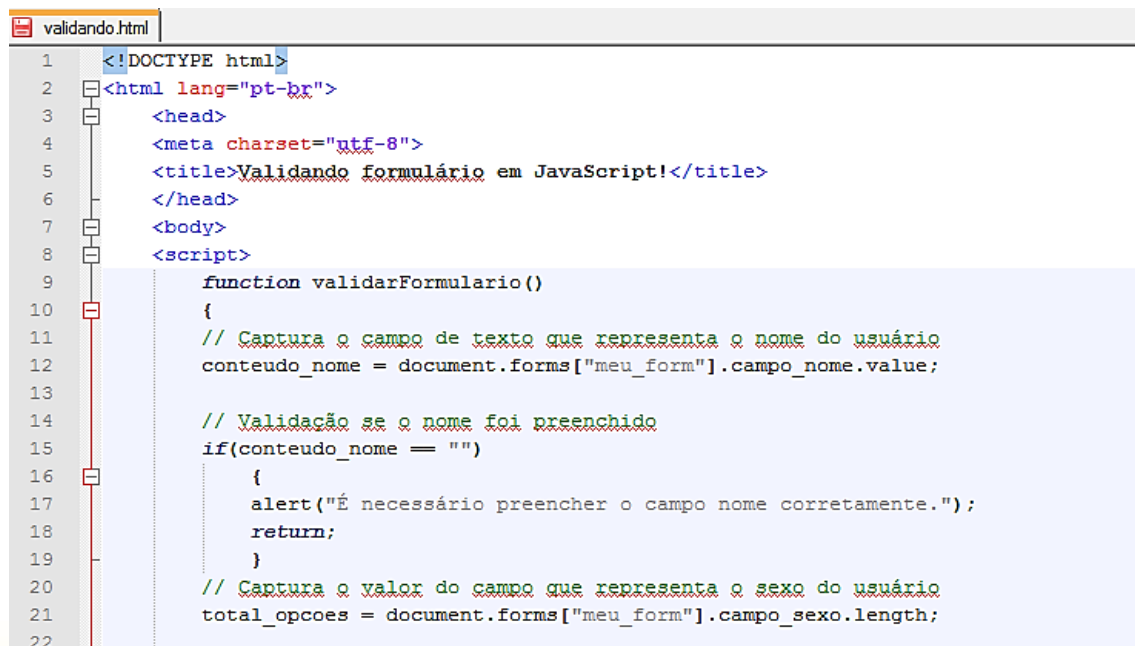
O primeiro passo do exemplo apresentado foi acessar a propriedade `length` do campo `campo_sexo`, que retorna quantas opções de rádio existem no formulário com este nome de campo. Com esta informação em mãos e armazenada na variável `total_opcoes`, é realizado um laço de repetição `for` que irá começar definindo a variável `i` com o valor `0` e irá repetir até que `i` atinja o valor `total_opcoes`, ou seja, estamos preparando um laço de repetição que irá navegar por todos os campos rádio de nome `campo_sexo`.

Dentro do laço de repetição `for` é acessada a propriedade `checked` do campo rádio em questão. Observe que `document.forms['meu_form'].campo_sexo` representa um array com as opções de rádio apresentadas para este campo. Por este motivo, ao acessar `campo_sexo[i]` você estará acessando cada uma dessas opções, verificando se a mesma foi selecionada, por meio da propriedade `checked`. O elemento da lista que retornar `true` para este acesso é justamente o elemento selecionado. A partir daí fica fácil: basta acessar a propriedade `value` do elemento `campo_sexo[i]` para saber o valor do campo selecionado.

Linguagem JavaScript (Validação Formulário)

Validando dados de um formulário e enviando-o ao servidor

Agora que você viu uma coleção de propriedades de acesso aos campos de formulários HTML em JavaScript, fica mais fácil pensar em um código que valide os mesmos, como faz o código-fonte :



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3   <head>
4     <meta charset="utf-8">
5     <title>Validando formulário em JavaScript!</title>
6   </head>
7   <body>
8     <script>
9       function validarFormulario()
10      {
11        // Captura o campo de texto que representa o nome do usuário
12        conteudo_nome = document.forms["meu_form"].campo_nome.value;
13
14        // Validação se o nome foi preenchido
15        if(conteudo_nome == "")
16        {
17          alert("É necessário preencher o campo nome corretamente.");
18          return;
19        }
20        // Captura o valor do campo que representa o sexo do usuário
21        total_opcoes = document.forms["meu_form"].campo_sexo.length;
22      }
```

Linguagem JavaScript (Validação Formulário)

```
23 // Validação se o sexo foi selecionado
24 opcao_escolhida = null;
25
26 for(i=0; i<total_opcoes; i++)
27 {
28     if(document.forms["meu_form"].campo_sexo[i].checked == true)
29     {
30         opcao_escolhida = document.forms["meu_form"].campo_sexo[i].value;
31     }
32 }
33
34 if(opcao_escolhida == null)
35 {
36     alert("É necessário selecionar o campo sexo corretamente.");
37     return;
38 }
39
40 // Captura se foi marcado a caixa de maior que 18 anos
41 // Mas neste exemplo nenhuma validação se faz necessária.
42 marcacao_maior = document.forms["meu_form"].campo_maior.checked;
43
```

Linguagem JavaScript (Validação Formulário)

```
44 //Captura o número e o valor da opção selecionada em COMO CONHECEU O SITE
45 conteudo_como = document.forms["meu_form"].campo_como.value;
46 numero_como = document.forms["meu_form"].campo_como.selectedIndex;
47
48 // Validação se o campo COMO CONHECEU O SITE foi selecionado
49 // diferente da opção "Selecione..."
50 if(numero_como == 0)
51 {
52     alert("É necessário selecionar como conheceu o site.");
53     return;
54 }
55
56 // Se não tiver realizado um return até este momento
57 // envia os dados para o servidor
58 document.forms["meu_form"].submit();
59 }
60 </script>
61
```


Linguagem JavaScript (Validação Formulário)

```
62 <form name="meu_form" method="get" action="?">
63
64     Nome completo:
65     <input name="campo_nome" type="text">
66     <<br />
67
68     Sexo:
69     <input name="campo_sexo" type="radio" value="M">Masculino
70     <input name="campo_sexo" type="radio" value="F">Feminino
71     <br />
72
73     <input name="campo_maior" type="checkbox" value="maior">
74     Declaro ser maior de 18 anos
75     <br />
76
77     Como conhecen o site:
78     <select name="campo_como">
79         <option value="Nenhum" default>Selecione...</option>
80         <option value="Ind">Indicação de amigo</option>
81         <option value="Out">Outro site</option>
82         <option value="Goo">Google</option>
83     </select>
84     <br />
85
```

Linguagem JavaScript (Validação Formulário)

```
86      <input type="reset" value="Limpar dados">
87      <input type="button" value="Enviar" onClick="validarFormulario();">
88
89      </form>
90  </body>
91 </html>
```

A validação dos campos do formulário começa basicamente capturando o valor preenchido no campo que representa o nome do usuário e verificando por meio de um comando **if**, se o valor contido no campo é uma string vazia. Se for, uma caixa de diálogo **alert** exibe uma mensagem para o usuário, informando que ele precisa preencher o campo corretamente e uma instrução **return** existente interrompe a execução do restante da função **validarFormulario**. Seguindo estes mesmos princípios, os demais campos do formulário também são validados.

Supondo que o campo que representa o nome do usuário esteja com um texto preenchido, a instrução **return** não será executada, permitindo a sequência da função de validação. A função continua validando o campo que representa o sexo do usuário.

Uma variável auxiliar chamada **opcao_escolhida** é criada e inicializada com o valor **null**. Se, após o laço de repetição que procura pela opção marcada, a variável **opcao_escolhida** continuar com o valor **null**,

Linguagem JavaScript (Validação Formulário)

isso indicará que nenhuma opção de sexo foi marcada pelo usuário. Neste caso, o comando `if` apresentado irá executar seu bloco de código, apresentando uma mensagem para o usuário selecionar o campo sexo corretamente.

Para o campo de checagem `campo_maior` nenhuma validação foi realizada, pois neste exemplo fictício é permitido que o usuário envie o formulário sendo menor de 18 anos (sem marcar o campo) ou sendo maior de 18 anos (marcando o campo). Se fosse necessário validar este campo, como campos de checagem que verificam se o usuário leu e está de acordo com os termos de uso do site, por exemplo, o seguinte comando `if` poderia ser utilizado (assumindo que o campo de checagem se chamasse `campo_de_checagem`):

```
if(document.forms["meu_form"].campo_de_checagem.checked != true) {  
    alert("É necessário concordar com os termos para prosseguir.");  
    return;  
}
```

Por último, é validado se o usuário selecionou alguma opção para o campo que representa como o usuário conheceu o site. Se nenhuma opção tiver sido selecionada, a variável `numero_como` conterá o valor `0`, capturado anteriormente. Se os campos tiverem sido preenchidos de acordo com as validações abordadas

Linguagem JavaScript (Validação Formulário)

neste tópico, nenhuma instrução `return` interromperá a execução da função `validarFormulario`, fazendo com que o código chegue e execute o envio do formulário, conforme instrução existente. Claro que neste exemplo, ao chegar nesta linha e enviar o formulário para o servidor, faltam programar ainda as instruções do lado servidor para poder gravar o usuário em um banco de dados, por exemplo.