

Validar Formulários

Rodrigo Attique
Desenvolvedor de Sistemas



Formulários e HTML5

- **O HTML5 nos permitiu inserir validações a nível de HTML**
- **Mas ainda há tarefas que precisam usar programação.**
- **Como validações a nível de cliente.**

```
<body>
  <form>
    <h3>Conversor de moeda</h3>
    <label>Real</label>
    <input type="number" step="0.1" name="real" id="r">
    <p>Para</p>
    <label>Dolar</label>
    <input type="number" step="0.1" name="dolar" id="d">
    <br><br>
    <button type="button" onclick="converter();">Conv
  </form>
```

Como funciona?



- Basicamente todas as informações preenchidas no formulário são validadas com JavaScript.
- Exemplo, digamos que temos um número de CPF.
- Sabemos que o melhor tipo para armazenar CPF é string ou campos textuais.
- Entretanto o CPF possui uma máscara e uma matemática que o valida.
- Antes de enviar esse dado para ser validado no back-end devemos validá-lo no front-end.

Exemplo de um conversor de moeda

Construa o form

- Perceba o evento no botão conerter

```
9      <form>
10          <h3>Conversor de moeda</h3>
11          <label>Real</label>
12          <input type="number" step="0.1" name="real" id="real" />
13          <p>Para</p>
14          <label>Dolar</label>
15          <input type="number" step="0.1" name="dolar" id="dolar" disabled />
16          <br><br>
17          <button type="button" onclick="converter();">Converter</button>
18      </form>
19
```

Como funciona?

- O usuário digita o valor em Real
- **Clica no botão converter**
- **O valor em dólar é exibido no input dólar**
- **Mas é claro que nada vai acontecer pois não programamos o evento em uma função.**



Implemente a função

```
20  <script>
21      function converter(){
22          // primeiro sempre inicializar variáveis
23          // Usamos value para pegar o valor do input
24          // Caso contrário acessamos toda a tag input
25          let real = document.getElementById("real").value;
26          let dolar = document.getElementById("dolar");
27          let aliquota = 5.82; //aliquota de conversão
28
29          //lógica de conversão
30          dolar.value = real * aliquota; //altera valor do input
31          //alert(dolar); //exibe o objeto em um alert
32      }
33  </script>
```


Resultado

Conversor de moeda

Real

Para

Dolar

Converter



Tipos de INPUT

- Antes de prosseguirmos é importante consultar a documentação do HTML5 para saber quais os tipos de input e validações que eles possuem.
- <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/input>

Exercícios

- **Crie um formulário que receba.**
- **Um email e uma senha do usuário.**
- **Se email: 'admin@admin.com' e senha: 'admin', exiba em uma tag h1 'usuário logado!' se não:**
- **Erros no formulário.**



Implemente o formulário

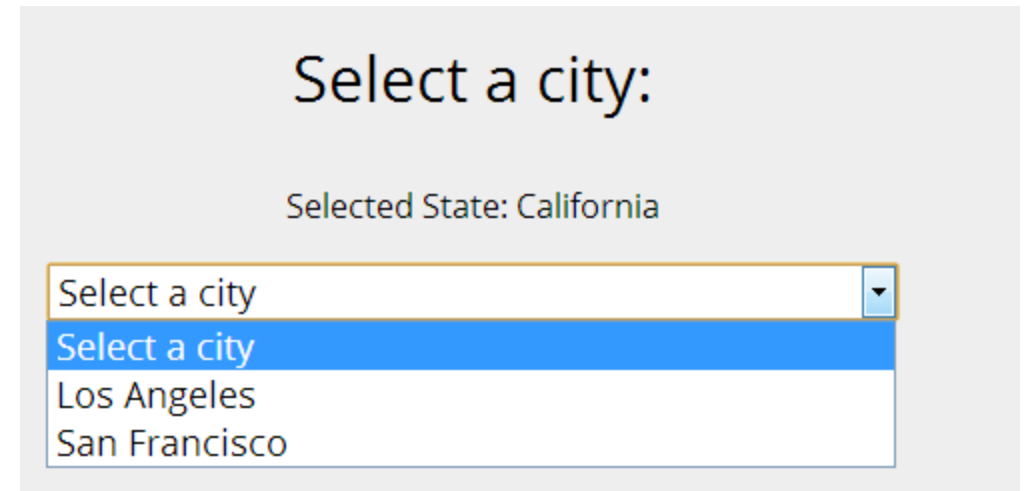
```
<form>
  <h3>Login</h3>
  <h1 id="msg"></h1>
  <label>Usuário</label>
  <input type="email" name="usuario" id="usuario" required />
  <p></p>
  <label>Senha</label>
  <input type="password" name="senha" id="senha" required/>
  <br><br>
  <button type="button" onclick="login();">Login</button>
</form>
```

Implemente o Script abaixo do form

```
21  <script>
22  function login(){
24      // primeiro sempre inicializar variáveis
25      // Vamos acessa toda a tag input
26      let usuario = document.getElementById("usuario");
27      let senha = document.getElementById("senha");
28      let msg = document.getElementById("msg");
29      //lógica de login
30      if (usuario=='admin@admin.com' && senha=='admin'){
31          msg.innerHTML = 'Usuário Logado!';
32          alert('Usuário Logado!');
33      } else{
34          msg.innerHTML = 'Erros no formulário!';
35          alert('Erros no formulário!');
36      }
37  }
38  </script>
```

Exemplo de Selectbox

- Selectbox é um tipo de lista suspensa.
- Normalmente usamos esse campo quando temos uma lista finita de opções que o usuário possui.
- Por exemplo em campos onde ele precisa escolher o estado onde mora.



The screenshot shows a web form with a light gray background. At the top, the text "Select a city:" is displayed in a dark blue font. Below it, the text "Selected State: California" is shown in a smaller, gray font. The main element is a select box with a white background and a thin orange border. The box contains the text "Select a city" and a small blue downward arrow on the right. A blue highlight is visible over the "Select a city" text, indicating it is the selected option. Below the select box, the options "Los Angeles" and "San Francisco" are listed in a gray font.

Escolha um estado e carregue cidades

- Nesse exemplo vamos criar um selectbox que ao selecionar um estado carrega as cidades em outro selectbox.
- Vamos precisar de uma lista de estados e cidades que você deverá terminar de preencher.
- Também vamos aprender a usar eventos de carregamento de página onload.

Adicione o evento ao body.

<https://www.w3schools.com/jsref/event onload.asp>

Quando a página carregar irá disparar um evento que carrega os estados no selectbox

```
6      <title>Cidades</title>
7  </head>
8  <body  onload="load_estados();">
9  <form>
10     <h3>Escolha sua cidade</h3>
```


Implemente o form

```
9  <form>
10    <h3>Escolha sua cidade</h3>
11    <label>Estado</label>
12    <select id="estado" name="estado" onchange="load_cidades();">
13      <option value="">Escolha um estado</option>
14    </select>
15    <p></p>
16    <label>Cidade</label>
17    <select id="cidade" name="cidade">
18      <option value="">Escolha um estado</option>
19    </select>
20  </form>
21
22  <script>
```

Selectbox estados

- O primeiro selectbox de estados irá ser carregado através do evento definido no body.
- O segundo selectbox será carregado quando o usuário alterar o selectbox de estados.
- Por isso definimos um evento onchange nele.



Vamos implementar o script em etapas

- Primeiro vamos criar um objeto para armazenar as cidades.
- Lembrando que você deverá completa-lo com as cidades que desejar.

```
22      <script>
23          var cidades = {
24              "Barretos": "SP"
25              , "Tupaciguara": "MG"
26              , "Catalão": "GO"
27              , "Itumbiara": "GO"
28              , "Diadema": "SP"
29              , "Cabo Frio": "RJ"
30          }
```

Evento de estados

- Perceba que essa função cria um elemento dinamicamente através do createElement.
- Dessa forma evitamos fazer Options manualmente.

```
29         , "Cabo Frio": "RJ"
30     }
31     function load_estados(){
32         let estados = ["SP", "RJ", "GO", "MG"];
33         let box = document.getElementById("estado");
34
35         for (e of estados){
36             let op = document.createElement("option");
37             op.innerHTML = e;
38             box.appendChild(op);
39         } //endfor
40     } //endload
```

Por fim o evento que preenche as cidades do estado.

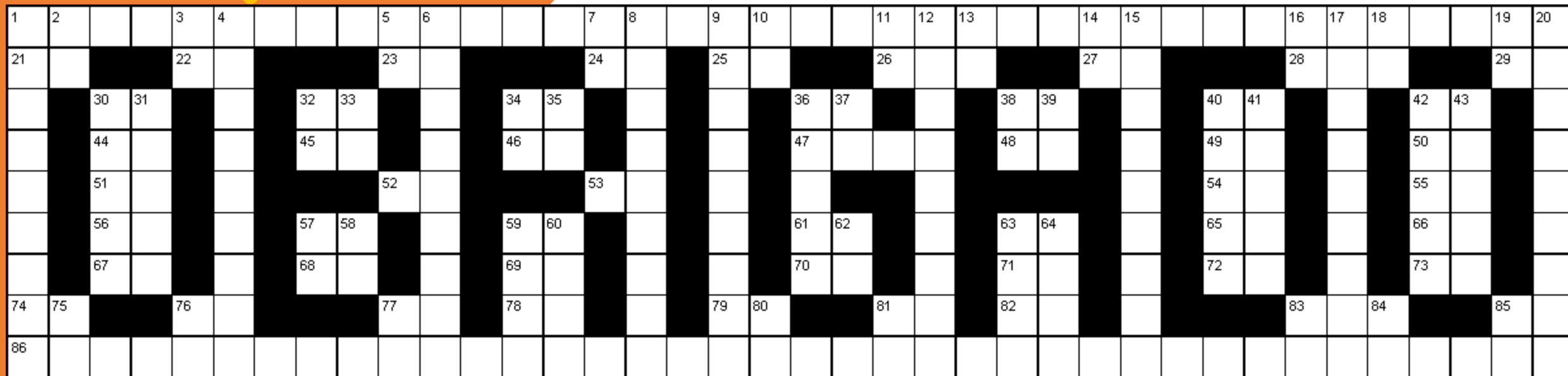
```
40      }//endload
41
42  ✓    function load_cidades(){
43      let estado = document.getElementById("estado").value;
44      let box = document.getElementById("cidade");
45      box.innerHTML=null;//limpa estados antigos
46
47  ✓    for (c in cidades){
48  ✓        if (cidades[c]==estado){
49            let op = document.createElement("option");
50            op.innerHTML = c;
51            box.appendChild(op);
52        }//endif
53    }//endfor
54    }//end load_cidades
55  </script>
```

Pronto!

Ainda há muito o que aprender sobre JavaScript, para nos aperfeiçoar é fundamental a prática de exercícios e situações que nos permitam evoluir.

Documentação de referência

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements>



[Esta Foto](#) de Autor Desconhecido está licenciado em [CC BY-NC](#)