



PROGRAMAÇÃO PARA INTERNET

AULA:

Validação de Formulário

Agenda

- Botão submit e ação de submissão de formulário
 - Método GET
 - Método POST
- Propriedade required
- Campos HTML com validação
- Evento de submissão de formulário
- Chamada de função para validação
- Personalização de mensagens para campo required

Botão Submit

- ❑ O formulário, ao ser criado, é com a intenção de enviar os dados para algum processamento
- ❑ O tipo “**submit**” cria um botão para que os dados possam ser enviados para o destino

```
<form>
  <label>Nome: </label>
  <input type="text" id="nome"><br/>

  <label>E-mail: </label>
  <input type="email" id="email"><br/>

  <label>Senha: </label>
  <input type="password" id="senha"><br/>

  <label>Confirmar Senha: </label>
  <input type="password" id="confirmarsenha"><br/>
  <button type="submit">Enviar Dados</button>
</form>
```

Button Submit

□ Mas enviar para onde?

- Há um atributo **action** a ser inserido na tag <form> quando o formulário é criado que deve indicar para onde os dados serão submetidos. Ao ser pressionado o botão do tipo submit, os dados são enviados.

```
<form action="recebedados.html">
  <label>Nome: </label>
  <input type="text" id="nome"><br/>

  <label>E-mail: </label>
  <input type="email" id="email"><br/>

  <label>Senha: </label>
  <input type="password" id="senha"><br/>

  <label>Confirmar Senha: </label>
  <input type="password" id="confirmarsenha"><br/>
  <button type="submit">Enviar Dados</button>
</form>
```

Button Submit

- Mas enviar para onde?
 - Outro atributo que é importante para o envio de dados é o parâmetro **method**. Ele indica de que modo os dados do formulário serão empacotados.

```
<form action="recebedados.html" method="get" >  
  <label>Nome: </label>  
  <input type="text" id="nome"><br/>  
  
  <label>E-mail: </label>  
  <input type="email" id="email"><br/>  
  
  <label>Senha: </label>  
  <input type="password" id="senha"><br/>  
  
  <label>Confirmar Senha: </label>  
  <input type="password" id="confirmarsenha"><br/>  
  <button type="submit">Enviar Dados</button>  
</form>
```

Button Submit

- Os métodos existentes são GET e POST. Ambos cumprem a mesma tarefa, mas de maneiras diferentes
 - POST: empacota as variáveis e as envia de forma oculta para o servidor
 - GET: também empacota as variáveis, porém anexa os valores antes de enviá-las para o servidor

<http://www.teste.com.br/recebedados.php?nome=Silvio&email=silvio.viegas@qi.edu.br>

Button Reset

- O tipo “reset” faz com que o formulário volte ao estado inicial (vazio ou com o valor que for especificado no atributo **value**)

```
<form action="recebedados.html" method="get">
  <label>Nome: </label>
  <input type="text" id="nome"><br/>

  <label>E-mail: </label>
  <input type="email" id="email"><br/>

  <label>Senha: </label>
  <input type="password" id="senha"><br/>

  <label>Confirmar Senha: </label>
  <input type="password" id="confirmarsenha"><br/>
  <button type="submit">Enviar Dados</button>
  <button type="reset">Limpar campos</button>
</form>
```


Atributo required

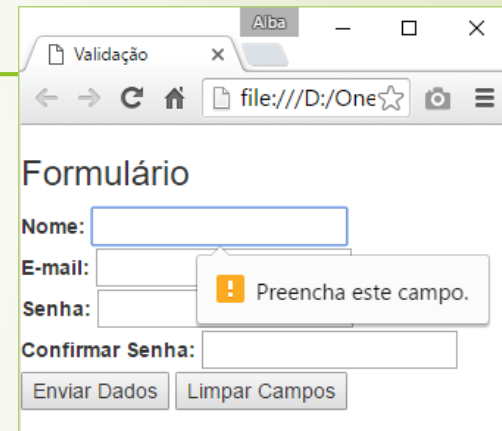
- Define que um campo que deve ser obrigatoriamente preenchido antes de realizar submit do form. Submeter com campo vazio gera mensagem de erro.

```
<form action="recebedados.html" method="get">
  <label>Nome: </label>
  <input type="text" id="nome" required ><br/>

  <label>E-mail: </label>
  <input type="email" id="email"><br/>

  <label>Senha: </label>
  <input type="password" id="senha"><br/>

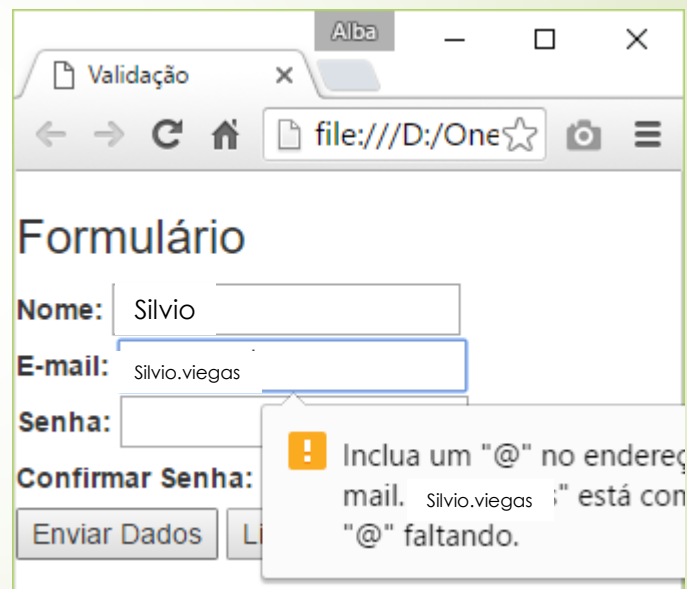
  <label>Senha: </label>
  <input type="password" id="confirmarsenha"><br/>
  <button type="submit">Enviar Dados</button>
  <button type="reset">Limpar campos</button>
</form>
```



A screenshot of a web browser window titled 'Alba' with a tab labeled 'Validação'. The address bar shows 'file:///D:/One'. The page content is titled 'Formulário' and contains four input fields: 'Nome:', 'E-mail:', 'Senha:', and 'Confirmar Senha:'. The 'Nome:' field is empty and has a red error message box over it that says 'Preencha este campo.' Below the fields are two buttons: 'Enviar Dados' and 'Limpar Campos'.

Campos HTML com validação

- HTML 5 trouxe inovações como validação automática de campos de formulário
- Inclui novos tipos de campo (type)
 - email
 - url
 - date, month, week, time, datetime
 - number
 - range
 - color
- Atributos:
 - max, min, pattern



Campos HTML com validação

- Atributo: min

- Define que um valor mínimo para um campo numérico

Idade: `<input type="number" name="idade" min="0">
`



Campos HTML com validação

- Atributo: **max**

- Define que um valor máximo para um campo numérico

Idade: `<input type="number" name="idade" min="0" max="150">
`



Campos HTML com validação

□ Atributo: **placeholder**

- Informa um texto inicial de auxílio de preenchimento de um campo

Nome: `<input type="text" name="nome" placeholder="Digite seu nome" required>
`

Nome:



Validação de Formulário

- ☐ Para o formulário criado, além da validação dos campos Nome estar preenchido, o campo E-mail conter um e-mail na formatação correta, ainda há outra regra
- ☐ os campos senha e confirmar senha precisam ter os valores iguais.
- ☐ Nesse caso, é preciso criar uma função Javascript que recupere o valor dos
- campos e efetue essa comparação.

```
function verificarSenha() {  
    var s1 = document.getElementById("senha").value;  
    var s2 =  
    document.getElementById("confirmarsenha").value;  
    if (s1 != s2) {  
        var msg = document.getElementById("mensagem");  
        msg.innerHTML = "As senhas não  
        conferem"; msg.className = "alert  
        alert-danger";  
    }  
}
```

Validação de Formulário

- A função criada deve ser chamada no evento onsubmit do formulário.

```
<div id="mensagem"></div>
<form action="recebedados.html" method="get" onsubmit="verificarSenha()" >
  <label>Nome: </label>
  <input type="text" id="nome"><br/>

  <label>E-mail: </label>
  <input type="email" id="email"><br/>

  <label>Senha: </label>
  <input type="password" id="senha"><br/>

  <label>Confirmar Senha: </label>
  <input type="password" id="confirmarsenha"><br/>
  <button type="submit">Enviar Dados</button>
</form>
```

Validação de Formulário

- ❑ Veja que, apesar de incluir a função no evento, o formulário continua sendo submetido.
- ❑ Isso porque a função precisa sinalizar ao evento que a operação de verificação falhou.
- ❑ Essa comunicação é feita através do valor de retorno da função:

```
function verificarSenha() {  
    var s1 = document.getElementById("senha").value;  
    var s2 = document.getElementById("confirmarsenha").value;  
    if (s1 != s2) {  
        var msg = document.getElementById("mensagem");  
        msg.innerHTML = "As senhas não conferem";  
        msg.className = "alert alert-danger";  
  
        return false;  
    }  
    return true;  
}
```


Validação de Formulário

- Além disso, devemos informar na chamada do evento onsubmit que o evento deve responder de acordo com o valor de retorno da função.

```
<form action="recebedados.html" method="get" onsubmit="return verificarSenha()">
  <label>Nome: </label>
  <input type="text" id="nome"><br/>

  <label>E-mail: </label>
  <input type="email" id="email"><br/>

  <label>Senha: </label>
  <input type="password" id="senha"><br/>

  <label>Confirmar Senha: </label>
  <input type="password" id="confirmarsenha"><br/>
  <button type="submit">Enviar Dados</button>
</form>
```

Validação de Formulário

- ❑ O return escrito faz com que a função seja interrompida no ponto que foi encontrado e passe o fluxo de execução para onde a função foi chamada, passando o valor false.
- ❑ A chamada da função com return verificarSenha(), no evento onsubmit recebe o valor false e interrompe a submissão do formulário.
- ❑ O formulário será submetido apenas se nenhum return false for disparado. Ou ainda, se algum return true for disparado.

Propriedade Validity

- ❑ Outra forma de realizar a validação do campo é através da alteração da propriedade **validity** do campo
- ❑ Nesse caso, não é necessário o monitoramento do evento `onsubmit`
- ❑ Essa propriedade é automaticamente verificada quando o formulário é enviado.
- ❑ É possível também definir a mensagem de validação através do método **setCustomValidity**

```
function validar() {  
    var campo1 = document.getElementById("senha");  
    var campo2 = document.getElementById("confirmarsenha");  
    if (campo1.value != campo2.value) {  
        campo2.setCustomValidity("As senhas não conferem");  
        campo2.validity = false;  
    } else {  
        campo2.setCustomValidity("");  
        campo2.validity = true;  
    }  
}
```

Propriedade Validity

- A função deve ser chamada quando houver modificação no campo afetado. Nesse caso, dois campos devem ser monitorados:

```
<form action="recebedados.html">
  <label>Nome: </label>
  <input type="text" id="nome"><br/>

  <label>E-mail: </label>
  <input type="email" id="email"><br/>

  <label>Senha: </label>
  <input type="text" id="senha" onchange="validar()"><br/>

  <label>Confirmar Senha: </label>
  <input type="text" id="confirmarsenha" onchange="validar()"><br/>
  <button type="submit">Enviar Dados</button>
  <button type="reset">Limpar Campos</button>
</form>
```

Exercício

- ☐ Crie um questionário com 3 questões para o usuário responder. Todas as questões são obrigatórias. Ao clicar no botão submit, verifique se todas as respostas estão corretas. Caso sim, submeta a página. Caso contrário, apresente uma borda vermelha nos campos que possuírem respostas incorretas.

REFERÊNCIAS

- [1] W3C School. JavaScript Tutorial. Disponível em:
http://www.w3schools.com/js/js_validation_api.asp
- http://www.w3schools.com/js/js_html_dom_elements.asp
- [2] MORISSON, Michael. Java Script - Use a Cabeça. Ed. 2. Rio de Janeiro: Altabooks
- [3] Manzano, José; Toledo, Suely. Guia de Orientação e Desenvolvimento de Sites - HTML, XHTML, CSS e JavaScript / Jscript. 2a. Edição
- [4] Windows Dev Center. setCustomValidity method. Disponível em:
<https://msdn.microsoft.com/en-us/library/windows/apps/hh441292.aspx>

The background of the slide is decorated with a geometric pattern of red triangles in various shades (dark red, medium red, and bright red). These triangles are arranged to form larger, abstract shapes, including a large triangle on the left and another on the right, with smaller triangles scattered throughout the white background.

Escolas e Faculdades

QI

qi.edu.br