

Desenvolvimento de Sistemas para a Web I

Capítulo 11 - Validação de Formulários com JavaScript

Validação de Formulário

- É o processo de verificar do dados de entrada do usuário em um formulário web *antes* de enviar para o servidor, a fim de garantir que estes atendam ao formato, tipo ou intervalo esperados
- Pode ser feito via atributos HTML ou código JavaScript
- Vantagens
 - Melhora a experiência com o usuário (o feedback é imediato)
 - Diminui o tráfego da rede

Validação de Formulário

- Desvantagens
 - Não funciona se o cliente web não suportar versões do recentes do padrão HTML ou a linguagem JavaScript
 - Pode ser desabilitado do lado cliente

Observação: a validação do lado cliente **NÃO** exclui de fazer a validação **também** do lado servidor

Validação com HTML vs Validação com JavaScript

- A validação com HTML é limitada a tamanhos e expressões regulares
- A validação com JavaScript pode fazer tudo que a HTML faz e muito mais
 - Cálculos
 - Datas
 - Etc.

Validação de Usuário e E-mail com HTML

```
<form>
  <div>
    <label for="nome">Nome do usuário: </label>
    <input type="text" id="nome" name="nome" required
      pattern="[a-z]{4,8}" title="4 a 8 letras minúsculas">
    <p>Nomes de usuário devem ser em minúsculas e ter 4-8 caracteres.</p>
    <label for="nome">E-mail: </label>
    <input type="text" id="email" name="email" required
      pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$">
    <p>E-mails devem ter o seguinte formato: fulano@dominio.com.</p>
  </div>
  <div>
    <button>Enviar</button>
  </div>
</form>
```

Validação de Usuário e E-mail com JavaScript

```
<form name="meuForm" onsubmit="return validaForm();">
  <div>
    <label for="nome">Nome do usuário: </label>
    <input type="text" id="nome" name="nome">
    <p>Nomes de usuário devem ser em minúsculas e ter 4-8 caracteres.</p>
    <label for="email">E-mail: </label>
    <input type="text" id="email" name="email">
    <p id="mensagem">Preencha o formulário e clique em Enviar</p>
  </div>
  <div>
    <button>Enviar</button>
  </div>
</form>
```

Validação de Usuário e E-mail com JavaScript

```
function validaForm() {  
    let msg = document.getElementById("mensagem");  
    const nome = document.forms["meuForm"]["nome"].value;  
    const email = document.forms["meuForm"]["email"].value;  
    try {  
        if (nome == null || !nome.match(/[a-z]{4,8}/)) {  
            throw "o nome do usuário deve ter de 4-8 caracteres";  
        }  
        if (email == null || !email.match(/[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$/)) {  
            throw "o e-mail deve ter o seguinte formato: fulano@dominio.com";  
        }  
        return true;  
    } catch(err) {  
        msg.style.color = "#ff0000";  
        msg.innerHTML = "Erro: " + err;  
        return false;  
    }  
}
```

Observações

- Não utilizar a expressão regular para verificar e-mails dos exemplos anteriores. Este tópico é bastante complexo:
<https://stackoverflow.com/questions/201323/how-can-i-validate-an-email-address-using-a-regular-expression>
- `document.forms` representa uma coleção de formulários
- `document.forms["nome-do-formulario"]["nome-do-input"]` representa o input cujo atributo `name` é `nome-do-input` e
`document.forms["nome-do-formulario"]` representa o formulário cujo `name` é `nome-do-formulario`

Referências

- LinkedIn. "What are the advantages and disadvantages of client-side and server-side validation?",
<https://www.linkedin.com/advice/1/what-advantages-disadvantages-client-side-server-side>, acessado em 19/06/2023
- MDN Web Docs. "HTML attribute: pattern",
<https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/pattern>, acessado em 19/06/2023
- W3Schools. "HTML `<input>` pattern Attribute",
https://www.w3schools.com/tags/att_input_pattern.asp, acessado em 19/06/2023