

Validação de Formulários com JavaScript

Prof. Bruno Torres Marques

Objetivo da Aula

- Entender a importância da validação de formulários
- Aprender a validar dados com JavaScript
- Oferecer feedback em tempo real ao usuário
- Evitar envio de dados inválidos

Por que validar formulários?

- Garantir integridade dos dados
- Evitar envio de informações incompletas ou incorretas
- Melhorar a experiência do usuário
- Proteger o sistema contra erros ou abusos

Tipos de validação

- Validação nativa (HTML5)
- Validação com JavaScript (manual)
- Ambas podem ser usadas em conjunto

Exemplo: Validação HTML5

```
<form>  
  <input type="email" required />  
  <input type="submit" value="Enviar" />  
</form>
```

- `required`, `type="email"`, `minlength` etc.

Limitações do HTML5

- Personalização visual limitada
- Mensagens genéricas
- Não cobre todos os casos
- Por isso, usamos **JavaScript**

Capturando o evento de envio

```
const form = document.querySelector("form");  
  
form.addEventListener("submit", function(event) {  
  event.preventDefault(); // impede envio automático  
});
```

- Agora podemos validar manualmente!

Validando campos com JS

```
let nome = document.getElementById("nome");  
  
if (nome.value === "") {  
    alert("O campo nome é obrigatório!");  
}
```

- Podemos validar **qualquer** regra

Exemplo completo

```
<input type="text" id="nome" />  
<span id="erroNome" class="erro"></span>
```

```
if (nome.value.trim() === "") {  
    document.getElementById("erroNome").textContent = "Nome obrigatório";  
}
```

Melhorando com CSS

```
.erro {  
  color: red;  
  font-size: 0.9em;  
}  
input:invalid {  
  border-color: red;  
}
```

- Visual mais amigável para o usuário

Validando e-mail

```
let email = document.getElementById("email");  
  
if (!email.value.includes("@")) {  
    alert("E-mail inválido!");  
}
```

- Melhor ainda: usar expressões regulares

Expressão regular para e-mail

```
const regex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;  
  
if (!regex.test(email.value)) {  
  alert("E-mail inválido!");  
}
```

Validação de senha

Regras comuns:

- Mínimo 8 caracteres
- Pelo menos uma letra maiúscula
- Pelo menos um número

Exemplo de validação de senha

```
const senha = document.getElementById("senha");  
const regexSenha = /^(?=.*[A-Z])(?=.*\d){8,}$/;  
  
if (!regexSenha.test(senha.value)) {  
    alert("Senha fraca!");  
}
```

Feedback em tempo real

```
campo.addEventListener("input", function() {  
  if (campo.value === "") {  
    mostrarErro();  
  } else {  
    limparErro();  
  }  
});
```

- Dá retorno imediato ao usuário

Usando classes dinamicamente

```
campo.classList.add("invalido");  
campo.classList.remove("valido");
```

- Alterna estilos conforme a validação

Evitando envio com dados inválidos

```
form.addEventListener("submit", function(e) {  
    if (!formularioValido()) {  
        e.preventDefault();  
        alert("Corrija os erros antes de enviar.");  
    }  
});
```

Criando função `formularioValido()`

```
function formularioValido() {  
    return nomeValido() && emailValido() && senhaValida();  
}
```

- Cada campo pode ter sua função de verificação

Projeto prático

Criar um formulário com:

- Nome (obrigatório)
- E-mail (formato válido)
- Senha (mínimo 8 caracteres, 1 número e 1 maiúscula)
- Confirmação de senha (iguais)

Conclusão

- Validação é essencial para segurança e usabilidade
- JavaScript permite personalização e controle total
- Valide **antes** de enviar
- Dê feedback visual e mensagens claras ao usuário

Obrigado!

Dúvidas?

Vamos praticar!