

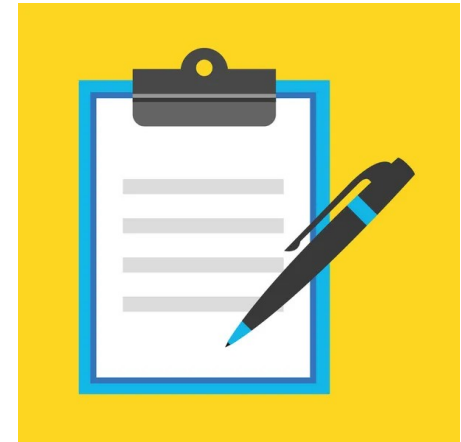
Linguagem PHP

Prof. Daniel Di Domenico

Validação de formulários



- O que vimos nas últimas aula:
 - Envio de informações do cliente para o servidor
 - Verbos/métodos HTTP:
 - GET
 - POST
 - Captura dos valores no PHP:
 - Superglobais:
 - \$_GET
 - \$_POST
 - Formulários
 - Persistência em arquivos JSON



{JSON}

PHP: validação de formulário

- **Objetivos da aula:**
 - Validar os dados de um formulário
 - Lado cliente (front-end)
 - Lado servidor (back-end)



PHP: validação front-end

- Para validar os dados no front-end (lado cliente), utiliza-se JavaScript
 - Pode-se fazer uso do evento **onsubmit** do formulário
 - Caso receba **false**, o formulário **não é submetido** ao servidor

```
<form method="POST" action="processa.php" onsubmit="return false;">  
  
  <!-- Campos input... -->  
  
  <button type="submit">Enviar</button>  
</form>
```

Pode-se chamar uma função JS para validar

Formulário não será submetido

PHP: validação front-end

- Exemplo de validação com JavaScript

```
<!-- processa.php -->
```

```
<form method="POST"  
  action="processa.php"  
  onsubmit="return validar();">
```

```
  <input type="text" id="nome"  
    name="nome" />
```

```
  <br><br>
```

```
  <button type="submit">  
    Enviar</button>
```

```
</form>
```

```
<script  
  src="validacao.js"></script>
```

O JavaScript acessa
os campos pelo ID

```
//validacao.js
```

```
function validarCampos() {
```

```
  //Valida o campo nome
```

```
  var nome =
```

```
    document.getElementById('nome');
```

```
  if(nome.value == '') {  
    alert("Informe o nome");
```

```
    return false;
```

```
    //form NÃO será submetido
```

```
  }
```

```
  return true; //form será submetido
```

```
}
```

PHP: validação back-end

- Para validar os dados no back-end (servidor), deve-se implementar as validações no código PHP
 - Antes de salvar os dados no banco, deve-se verificar se todos os campos foram preenchidos
 - **Efeito colateral:**
 - Ao submeter o formulário para o servidor, **a página é recarregada e os valores preenchidos nos campos são apagados**
 - É preciso realizar um tratamento para mantê-los preenchidos

PHP: validação back-end

- Exemplo de validação com PHP - HTML

```
<!-- processa.php - HTML -->
```

```
<form method="POST" action="processa.php">
```

```
<input type="text" name="nome" value="<?= $nome ?>" />
```

```
<br><br>
```

```
<input type="text" name="idade" value="<?= $idade ?>" />
```

```
<br><br>
```

```
<button type="submit">Enviar</button>
```

```
<input type="hidden" name="submetido" value="1" />
```

```
</form>
```

```
<div style="color: red;"><?= $msgErro; ?></div>
```

```
<div style="color: green;"><?= $msgSucesso; ?></div>
```

Retorna o valor do
nome para o campo

Retorna o valor do
idade para o campo

Exibe a mensagem
de erro

PHP: validação back-end

- Exemplo de validação com PHP – *Continuação...*

```
//processa.php - Continuação (top da página)
$msgErro = '';
$msgSucesso = '';

$nome = '';
$idade = '';
$submetido = isset($_POST['submetido']) ? $_POST['submetido'] : null;
if($submetido) {
    $nome = trim($_POST['nome']) ? trim($_POST['nome']) : null;
    $idade = trim($_POST['idade']) ? trim($_POST['idade']) : null;

    if(! $nome) //Se não preenchido
        $msgErro = 'Informe o nome!';

    else if(! $idade) //Se não preenchido
        $msgErro = 'Informe a idade!';

    else //Código após validação
        $msgSucesso = "Nome: " . $nome . " - Idade: " . $idade;
}
```

trim(): remove os espaços
do início e fim da **string**

PHP: diferenças entre validações

- **Front-end:**

- A submissão do formulário não é realizada
 - Não há perda das informações preenchidas, pois a página não é recarregada

- **Back-end:**

- A submissão do formulário é realizada, resultando no recarregamento da página
- É necessário setar as informações novamente para os campos

Exercícios

- **1-** Utilize o cadastro de Livros visto na última aula (campos título, gênero e quantidade de páginas), implementando as validações ao formulário, sendo:
 - 1.1- Validação front-end com JavaScript
 - 1.2- Validação back-end com PHP
- **2-** Complemente o exercício 1 acima, adicionando no back-end as seguintes validações:
 - 2.1- Quantidade de páginas maior que zero;
 - 2.2- Título do livro com no mínimo 3 e no máximo 50 caracteres (utilize a função `strlen($string)`);
 - 2.3- **DESAFIO:** não permitir cadastro de com título repetido
 - Validar com uma consulta ao arquivos JSON