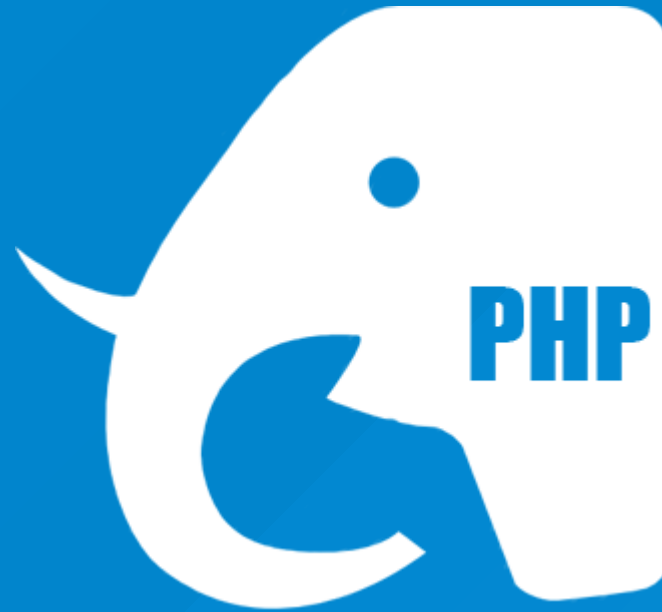


Validação de Formulário



Bacharelado em Sistemas de Informação - IFCE Campus Cedro

Professor Zé Olinda ([@joseolinda](#))

jose.olinda@ifce.edu.br

BSI

Bacharelado em Sistemas de Informação

IFCE Campus Cedro

S5 - PWEBI

Programação para a Web I

Professor Zé Olinda

Conteúdos

- Preparação do Formulário
- Atributo de **<FORM>**
- Métodos GET e POST
- Include e Require
- Acessando superglobal **\$_POST**
- Validações básicas do formulário (exercício)
- Exibindo informações (exercício)

Preparação do Formulário

Preparação do Formulário

Antes de receber as informações *via* **PHP**, é necessário preparar seu formulário HTML. Baixe o seguinte código para usar como exemplo.

<https://github.com/joseolinda/RevisaoForm2>

Preparação do Formulário

1. Defina o nomes e valores dos campos

Para a correta "captura" dos dados usando PHP, é necessário informar o atributo **name** em todos os campos do formulário que se deseja capturar os dados. Para o PHP, é enviado, como um elemento de *array*, um índice com o nome definido em **name** e o valor definido em **value**.

Exemplo

```
<input type="radio" value="Arroz" name="acompanhamento" />
```

Preparação do Formulário

1.1 Name para campo *text* e *password*

Use nomes significativos, que facilite a identificação do campo. Não use espaço ou caracteres especiais.

Exemplo

```
<input type="text" id="onenumber" name="telefone_contato" />  
<input type="password" id="pass" name="senha_usuario" />
```


Preparação do Formulário

1.2 Name para campo *radio*

Campos do tipo *radio* são usados para que o usuário escolha um dentre várias opções. Você deve definir o mesmo **name** para **todos os campos radio** que representam a mesma informação. Para diferir cada opções, você deve atribuir **values** diferentes.

Exemplo

```
<input type="radio" value="Arroz" name="acompanhamento" />Arroz branco  
<input type="radio" value="Macarrao" name="acompanhamento" />Macarrão
```

Preparação do Formulário

1.3 Name para campo *checkbox*

Campos do tipo *checkbox* são usados para que o usuário escolha uma **ou mais** dentre várias opções. Você pode definir diferentes **name** para **cada campo checkbox**. Entretanto, se um conjunto de checkbox representa um único grupo de dados, pode-se definir o mesmo nome usando `[]` ao final. Deste modo, o PHP entenderá como um vetor de dados.

Exemplo

```
<input type="checkbox" value="Arroz" name="acompanhamento[]" />Arroz branco  
<input type="checkbox" value="Macarrao" name="acompanhamento[]" />Macarrão
```

Preparação do Formulário

1.4 Name para campo *select*

Campos do tipo *select* são usados para que o usuário escolha um **dentre uma lista suspensa** de opções. Você pode definir diferentes **name** para a tag **select**. Para cada tag **option**, defina o **value** correspondente.

Exemplo

```
<select id="prato" name='prato_principal' >  
  <option value="">Escolha uma opção</option>  
  <option value="bife-parmegiana" >Bife à parmegiana</option>  
  <option value="frango-parmegiana" >Frango à parmegiana</option>  
</select>
```

Atributos de <FORM>

Atributos de <FORM>

2.1 **action**

Define a URL para onde as informações do formulário serão enviadas.

Exemplo

```
<form action='pedidos.php' id="delivery-form" />
```

Atributos de <FORM>

2.2 `method`

Define o método HTTP será usado para enviar as informações do formulário para o endereço definido em `action`.

Exemplo

```
<form method='get' action='pedidos.php' id="delivery-form" />
```

GET vs POST

GET vs POST

GET

É usado para enviar dados de um formulário para um recurso/página específica. É o método HTTP mais comum. Suas principais características:

- Pode ser armazenado no cache
- Fica salvo no histórico do navegador
- Pode ser salvo nos favoritos
- Não deve ser usado para enviar senha e dados mais sensíveis

GET (continuação)

- Não aceita dados/texto muito longos
- É aconselhável usá-lo apenas para requisitar/solicitar dados. Nunca para salvar dados
- Mostra os dados enviados na barra de endereço do navegador

Exemplo:

```
/url/de/exemplo/pagina.php?usuario=joseolinda&senha=vcnaoviunada
```

GET vs POST

POST

É usado para enviar dados de um formulário para um recurso/página/servidor específico. Este método é o mais recomendado para criar, salvar e editar dados no servidor. Características:

- As requisições nunca* são armazenadas no cache
- As requisições não aparecem no histórico do navegador
- Não pode ser salvo nos favoritos
- Não há restrição de tamanho dos dados enviados

Mais métodos HTTP, leia:

<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Methods>

https://www.w3schools.com/tags/ref_httpmethods.asp

Referência oficial:

[RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1](#)

Include e Require

Include e Require

A funções `include` e `require` são usada para incluir um arquivo PHP. Isso permite separar, organizar e reutilizar código PHP.

```
<?php require "my_variables.php"; ?>
<?php include "my_functions.php"; ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title><?php displayTitle($home_page); ?></title>
</head>
<body>
<?php include "header.php"; ?>
<?php include "menu.php"; ?>
    <h1>Welcome to Our Website!</h1>
    <p>Here you will find lots of useful information.</p>
<?php include "footer.php"; ?>
```

Superglobais

Superglobais PHP

Várias variáveis pré-definidas no PHP são "superglobais", que significa que elas estão disponíveis em todos escopos para todo o script.

Superglobais são variáveis nativas que estão sempre disponíveis em todos escopos

Estas variáveis superglobais são:

- **`$GLOBALS`** // Lista de variáveis globais
- **`$_SERVER`** // Informações sobre o servidor
- **`$_GET`** // Array de dados enviados via GET
- **`$_POST`** // Array de dados enviados via POST
- **`$_FILES`** // Array de arquivos enviados via formulário
- **`$_COOKIE`** // Variáveis enviadas através de cookie
- **`$_SESSION`** // Variáveis enviadas através de sessão
- **`$_REQUEST`** // Variáveis enviadas através de requisição HTTP
- **`$_ENV`** // Variáveis de ambiente do PHP

Acessando superglobal \$_POST

Na página HTML, enviar informações para o POST:

```
<form action="registration.php" method="post">  
Name: <input type="text" name="name">  
Email: <input type="text" name="email">  
<input type="submit">  
</form>
```

registration.php

```
Bem-vindo <?php echo $_POST["name"]; ?>!  
Seu endereço de e-mail é <?php echo $_POST["email"]; ?>
```