



INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Pelotas

PHP

Aula 2 - Funções para PostgreSQL

Professora Marla Cristina da Silva Sopeña



FORMULÁRIOS

O atributo **action** define qual o arquivo receberá os dados do formulário

O método **GET** ou **POST** define o modo como os dados são enviados para o servidor

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

Nome : `<input type="text" name="nome">`

Email : `<input type="text" name="email">`

CPF: `<input type="text" name="cpf">`

`<input type="reset" name="botao" value="Limpar">`

`<input type="submit" name="botao" value="Enviar">`

`</form>`

arquivo index.php
pasta exemplo1

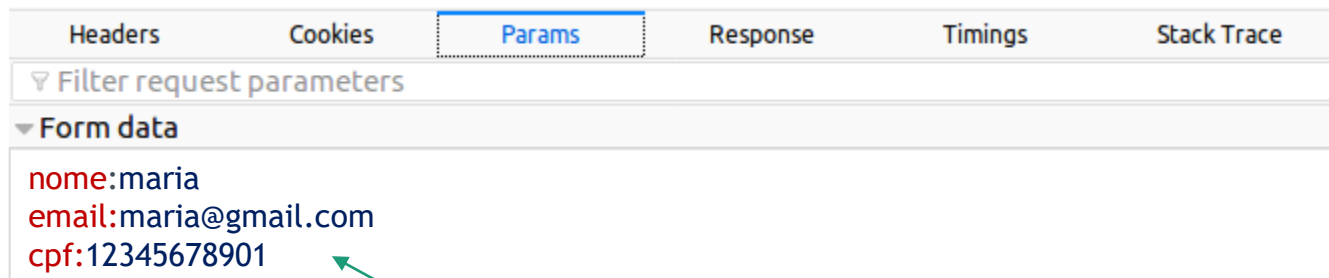
Botão submit serve para submeter os dados a um arquivo que será responsável pelo processamento. Esse arquivo deve ser definido no parâmetro **action**. Neste exemplo, o arquivo é o **cadastro_usuario.php**

MÉTODO POST

Neste método os dados são enviados no corpo da mensagem HTTP, enviada ao servidor. A URL, desta forma, não contém as informações enviadas pelo formulário, evitando expor mais facilmente os dados e regras de negócio do site. Outra vantagem do Método POST é enviar uma quantidade maior de dados, bem como, arquivos anexados, o que não é possível como o GET.

http://nomedosite/cadastro_usuario.php

← URL mais limpa e objetiva



os dados são passados no corpo da mensagem HTTP, inspecionando a requisição estes dados estão disponíveis

MÉTODO GET

Quando utilizamos o GET, os parâmetros são passados no cabeçalho da requisição. Por isso, podem ser vistos pela URL. Neste método, os dados são enviados juntamente com o nome da página (na URL) que irá processar os dados.

as variáveis **nome**, **email**, **cpf** (identificadas pelo atributo **name** do formulário) recebem os respectivos valores que correspondem aos conteúdos digitados nos campos do formulário

http://nomedosite/cadastro_usuario.php?nome=Maria&email=mariasilva@gmail.com&cpf=12345678901

o caracter ? representa o início da cadeia de informação

o caracter & identifica o início de uma nova variável

RECUPERANDO VALORES ENVIADOS PELO FORMULÁRIO

Para recuperar os valores dentro da página definida no atributo action do formulário, podem ser utilizados os vetores `$_GET` e `$_POST`, respectivamente, para cada método de envio. Estes vetores predefinidos pelo PHP, são variáveis globais e podem ser utilizadas em qualquer parte do código .

Podemos ver o resultado destas variáveis com as seguintes funções:

`var_dump($_GET)` ou `var_dump($_POST)`

`print_r($_GET)` ou `print_r($_POST)`

TRATANDO AS INFORMAÇÕES RECEBIDAS

```
<?php
    echo $_POST['nome'];
    echo $_POST['email'];
    echo $_POST['cpf'];
?>
```

O array \$_GET ou \$_POST é criado a partir da propriedade **method** do formulário

```
<?php
    echo $_GET['nome'];
    echo $_GET['email'];
    echo $_GET['cpf'];
?>
```

TRATANDO AS INFORMAÇÕES RECEBIDAS

```
<?php
```

```
$nome= $_POST['nome'];  
$email= $_POST['email'];  
$cpf= $_POST['cpf'];  
echo $nome;  
echo $email,  
echo $cpf;
```

```
?>
```

```
<?php
```

```
$nome= $_GET['nome'];  
$email= $_GET['email'];  
$cpf= $_GET['cpf'];  
echo $nome;  
echo $email,  
echo $cpf;
```

```
?>
```

Para facilitar a manipulação dos dados, pode-se criar variáveis comuns no início do programa para usá-las no restante do código.

BANCO DE DADOS

Em PHP existe uma sequência lógica para se trabalhar com banco de dados, que consiste em três passos:

1. Primeiro, criar variáveis informando o nome do banco, a porta, o usuário e a senha, e utilizar uma função própria, que dependerá do tipo do banco, para abrir uma nova conexão;
2. Segundo, utilizar as funções de manipulação para executar o código SQL;
3. Terceiro, fechar a conexão.

Cada banco de dados possui um grupo de funções diferentes para a manipulação dos dados

PHP e POSTGRES

Função de conexão com o banco: `pg_connect`

`<?php`

```
$servidor = "localhost";  
$porta = 5432;  
$bd = "aula";  
$usuario = "postgres";  
$senha = "senha5";  
$conexao = pg_connect("host=$servidor port=$porta dbname=$bd user=$usuario  
password=$senha");  
if(!$conexao)  
{  
    die("Não foi possível se conectar ao banco de dados.");  
}
```

`?>`

PHP e POSTGRES

Para executar comandos em um banco de dados PostgreSQL, utilizamos a função

Função de execução de comandos: `pg_query`

`pg_query(objeto de conexão , query)`

- O retorno da função `pg_query` pode ser um `pg_result` ou `false`
- Quando o resultado é `false` ocorreu um erro na execução da query
- Quando a query é executada sem problemas retorna um ponteiro contendo as informações da query, é necessário então aplicar funções para extrair as informações necessárias.

FUNÇÕES PHP PARA O POSTGRES

Exemplo de listagem de registros de uma tabela do banco de dados

```
<?php
```

```
include "conecta.php"; // inclui o arquivo conecta.php
```

```
$sql = "select * from pessoa"; // prepara a variável com o comando sql
```

```
$resultado = pg_query($conexao,$sql) or die("Erro"); // executa o comando sql
```

```
while($dados=pg_fetch_array($resultado)){
```

```
echo $dados['codpessoa'];
```

```
echo $dados['nome'];
```

```
echo $dados['email'];
```

```
echo $dados['cpf'].'<br>;
```

```
}
```

```
?>
```

codpessoa	nome	email	cpf
1	Maria da Silva	maria@gmail.com	12345678910
2	José da Silva	jose@gmail.com	09876543210
3	João da Silva	joao@gmail.com	08556683210

Representação do comando select que seleciona todas as colunas e linhas da tabela pessoa

Neste exemplo o while fará a iteração pelo **pg_result** armazenado na variável **\$resultado**, criando, através da função **pg_fetch_array**, um vetor, neste exemplo, chamado **\$dados[]** que armazena temporariamente, a cada iteração, uma linha do banco de dados . Os índices de **\$dados[]**, são os nomes das colunas do banco de dados, neste exemplo **\$dados[codpessoa]**; **\$dados[nome]**; **\$dados[email]**; **\$dados[cpf]**

FUNÇÕES PHP PARA O POSTGRES

Exemplo de inserção de um registro em uma tabela de um banco de dados

```
<?php
```

```
include "conecta.php";  
$nome=$_POST['nome'];  
$email=$_POST['email'];  
$cpf=$_POST['cpf'];
```

preparação da variável **\$sql** que armazena o comando sql para inserir um registro da tabela do banco de dados

```
$sql="insert into pessoa (nome,email,cpf) values ('$nome','$email','$cpf')";
```

```
$resultado=pg_query($conexao,$sql);  
if($resultado)  
{  
    echo "Cadastro Efetuado com sucesso";  
}  
else  
{  
    echo "Erro ao inserir".'<br>';  
}  
?>
```

execução da query utilizando a função **pg_query** que recebe como parâmetros a variável de conexão e a variável contendo a query (\$sql)
a função **retorna** para a variável **\$resultado**:
true (se ocorreu a inserção) ou false (se ocorreu algum erro)

verifica se o conteúdo de **\$resultado** é **true** escrevendo a mensagem de sucesso, caso contrário escreve o código e a mensagem de erro do banco de dados

FUNÇÕES PHP PARA O POSTGRES

Exemplo de alteração de um registro em uma tabela de um banco de dados

```
<?php
```

```
include "conecta.php";
```

```
$codpessoa=$_POST['codpessoa'];
```

```
$nome=$_POST['nome'];
```

```
$email=$_POST['email'];
```

```
$cpf=$_POST['cpf'];
```

preparação da variável **\$sql** que armazena o comando sql para alterar um registro da tabela do banco de dados

```
$sql="update pessoa set nome='$nome', email='$email', cpf='$cpf' where codpessoa = '$codpessoa'";
```

```
$resultado =pg_query($conexao,$sql);
```

execução da query utilizando a função **pg_query** que recebe como parâmetros a variável de conexão e a variável contendo a query (\$sql)
a função **retorna** para a variável **\$resultado**:
true (se ocorreu a alteração) ou false (se ocorreu algum erro)

```
if($resultado){
```

```
    echo "Registro Alterado com sucesso";
```

```
}
```

```
else{
```

```
    echo 'Erro ao alterar';
```

```
}
```

```
?>
```

verifica se o conteúdo de **\$resultado** é **true**
escrevendo a mensagem de sucesso, caso contrário escreve o código e a mensagem de erro do banco de dados

FUNÇÕES PHP PARA O POSTGRES

Exemplo de exclusão de um registro em uma tabela de um banco de dados

```
<?php
```

```
include "conecta.php";
```

```
$codpessoa=$_POST['codpessoa'];
```

```
$sql=delete from pessoa where codpessoa = '$codpessoa';
```

```
$resultado=pg_query($conexao,$sql);
```

```
if($resultado){
```

```
    echo "Cadastro Excluido com sucesso";
```

```
}
```

```
else
```

```
{
```

```
    echo 'Nenhum registro excluído';
```

```
}
```

```
?>
```

preparação da variável **\$sql** que armazena o comando sql para deletar um registro da tabela do banco de dados

execução da query utilizando a função **pg_query** que recebe como parâmetros a variável de conexão e a variável contendo a query (**\$sql**)
a função **retorna** para a variável **\$resultado**:
true (se ocorreu a exclusão) ou false (se ocorreu algum erro)

verifica se o conteúdo de **\$resultado** é **true**
escrevendo a mensagem de sucesso, caso contrário escreve o código e a mensagem de erro do banco de dados