

# ENVIANDO DADOS COM PHP: `$_GET` , `$_POST` E `$_REQUEST`

## Como capturar e manipular dados enviados por formulários HTML utilizando PHP?

### Com as variáveis SuperGlobais

As variáveis superglobais são variáveis que estão disponíveis em todos os escopos do script, ou seja, elas podem ser acessadas em qualquer lugar do código, independentemente do escopo em que estão sendo usadas. Elas são "superglobais" porque estão disponíveis em todos os escopos, incluindo funções, classes e outros blocos de código.

As variáveis pré-definidas são variáveis que são definidas automaticamente pelo PHP e estão disponíveis para uso em qualquer lugar do script. Elas são "pré-definidas" porque são definidas antes que o script seja executado.

No PHP, as variáveis superglobais e pré-definidas são sinônimos e se referem às mesmas variáveis.

Você pode verificar a lista de variáveis pré-definidas na documentação oficial do PHP:

<https://www.php.net/manual/en/reserved.variables.php>

No momento, vamos aprender as três principais variáveis globais para manipulação de dados:

- `$_GET` : envia dados via URL.
- `$_POST` : envia dados ocultos no corpo da requisição.
- `$_REQUEST` : combina dados de `$_GET` , `$_POST` e `$_COOKIE` em uma única variável.

### `$_GET`

e oficial: <https://www.php.net/manual/en/reserved.variables.get.php>

Essa variável superglobal é usada para capturar dados enviados por meio de URLs, como parâmetros de consulta.

Por exemplo, se você tiver um formulário com um campo de texto chamado "nome" e um botão de envio, e o formulário for enviado via método GET, a URL resultante poderá ser algo como `http://example.com/formulario.php?nome=Joao` .

Nesse caso, você pode acessar o valor do campo "nome" usando `$_GET[ 'nome' ]` .

Exemplo de uso de `$_GET` :

```
if ($_SERVER['REQUEST_METHOD'] == 'GET') {  
    $nome = $_GET['nome'];  
    echo "Olá, $nome!";  
}
```

## **\$\_POST**

<https://www.php.net/manual/en/reserved.variables.post.php>

Essa variável superglobal é usada para capturar dados enviados por meio de formulários HTML que utilizam o método POST.

Por exemplo, se você tiver um formulário com um campo de texto chamado "nome" e um botão de envio, e o formulário for enviado via método POST, os dados serão enviados no corpo da requisição HTTP e você pode acessá-los usando `$_POST['nome']`.

Exemplo de uso de `$_POST`:

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
    $nome = $_POST['nome'];  
    echo "Olá, $nome!";  
}
```

## **\$\_REQUEST**

<https://www.php.net/manual/en/reserved.variables.request.php>

Essa variável superglobal contém os dados de `$_GET`, `$_POST` e `$_COOKIE`. Ela é útil quando você não se importa com a origem dos dados (se vieram de um formulário, de uma URL ou de um cookie). No entanto, é recomendado evitar o uso de `$_REQUEST` sempre que possível, pois pode levar a problemas de segurança, pois pode incluir dados indesejados, como cookies.

Exemplo de uso de `$_REQUEST`:

```
$nome = $_REQUEST['nome'];  
echo "Olá, $nome!";
```

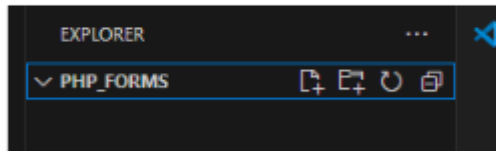
## **Atividade prática**

### **Passo Zero: Criar o projeto PHP**

1. Abrir o Xampp e iniciar o servidor Apache e MySQL.
2. Abrir o VsCode.
3. Abrir a pasta htdocs:

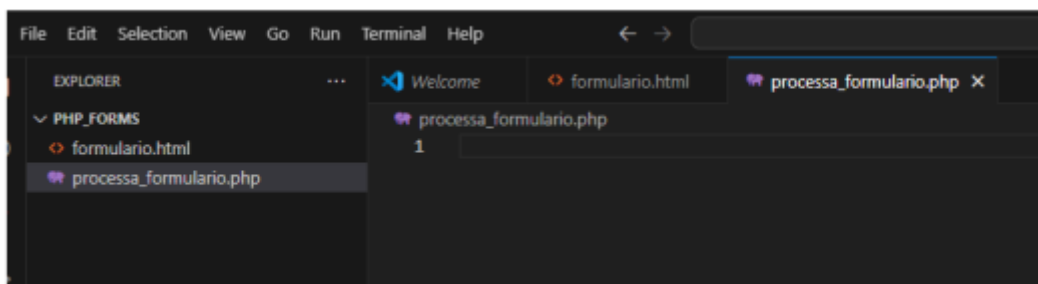
- File -> Open Folder -> C:/xampp/htdocs -> Nova pasta -> FormulárioPHP -> Selecionar Pasta

## File > Open Folder > Selecione a Pasta



### 4. Crie os arquivos:

- formulario.html
- processa\_formulario.php



## Etapa HTML: Criar um Formulário

Abaixo segue o código HTML completo para a criação de um formulário básico. Crie um arquivo HTML chamado "index\_form.html" e copie o código. Na sequência, o código é explicado em mais detalhes.

```
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
  />
  <title>Formulário</title>
</head>
<body>
  <header>
    <h1>Formulário de Contato</h1>
    <p>Preencha o formulário abaixo para entrar em contato conosco.</p>
  </header>

  <main>
    <form action="/submit" method="post" name="formulario_contato">
      <!-- Campo de nome -->
      <label for="nome">Nome:</label>
      <input
        type="text"
```

```

        id="nome"
        name="nome"
        required
        placeholder="Digite seu nome"
        maxlength="100"
    />
<br /><br />

<!-- Campo de email -->
<label for="email">Email:</label>
<input
    type="email"
    id="email"
    name="email"
    required
    placeholder="Digite seu email"
    pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$"
/>
<br /><br />
<footer>
    <p>CC-BY-SA 4.0
</footer>
</body>
</html>

```

## Estrutura do Formulário

- `<main>`
  - Elemento que representa o conteúdo principal da página.
- `<form action="/submit" method="post" name="formulario_contato">`
  - Elemento que define um formulário. O atributo `action` especifica a URL que receberá os dados do formulário, `method` especifica o método de envio (neste caso, POST) e `name` é o nome do formulário.

## Campo de Nome

- `<label for="nome">Nome:</label>`
  - Elemento que define uma etiqueta para o campo de nome.
- `<input type="text" id="nome" name="nome" required placeholder="Digite seu nome" maxlength="100">`
  - Elemento que define um campo de texto para o nome. O atributo `type` especifica o tipo de campo, `id` é o identificador do campo, `name` é o nome do campo, `required` indica que o campo é obrigatório, `placeholder` é o texto que aparece no campo antes de o usuário digitar algo e `maxlength` é o número máximo de caracteres permitidos.

## Campo de Email

- `<label for="email">Email:</label>`
  - Elemento que define uma etiqueta para o campo de email.
- `<input type="email" id="email" name="email" required placeholder="Digite seu email" pattern="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$">`
  - Elemento que define um campo de email. O atributo `type` especifica o tipo de campo, `id` é o identificador do campo, `name` é o nome do campo, `required` indica que o campo é obrigatório, `placeholder` é o texto que aparece no campo antes de o usuário digitar algo e `pattern` é um padrão de validação para o email.

## Form action = endereço do PHP

Repare que no HTML, a linha de código que abre o formulário, `<form>` determina o atributo `action` como `/submit`.

Esse valor deve ser substituído pelo nome e endereço do arquivo PHP que iremos criar a seguir.

## Etapa PHP: script para processar os dados

No arquivo "processa\_formulario.php", copie o seguinte código:

```
<?php
// Captura os dados do formulário
$nome = $_POST['nome'];
$email = $_POST['email'];

// Exibe os dados enviados
echo "<h1>Dados Recebidos</h1>";
echo "<p><strong>Nome:</strong> $nome</p>";
echo "<p><strong>E-mail:</strong> $email </p>";

// Exemplo com $_REQUEST
if (isset($_REQUEST['nome']) && isset($_REQUEST['email'])) {
    echo "<p><em>(Capturado via \$_REQUEST)</em></p>";
}
?>
```

Esse script pega os valores de `nome` e `idade` a partir dos dados enviados via `POST` e os exibe no navegador. Se as variáveis também estiverem definidas no array `$_REQUEST`, ele exibe uma mensagem adicional informando que os dados foram capturados por essa superglobal.

## Entendendo o código

```
<?php
```

- Início do código

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

- Captura o valor do campo "nome" do formulário enviado via método POST e armazena em uma variável chamada `$nome`.

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

- Captura o valor do campo "idade" do formulário enviado via método POST e armazena em uma variável chamada `$idade`.

```
echo "<h1>Dados Recebidos</h1>";
```

- Exibe um título "Dados Recebidos".

```
echo "<p><strong>Nome:</strong> $nome</p>";
```

- Exibe o valor da variável `$nome` (nome do usuário) dentro de uma tag `<p>` com um `<strong>` para destacar a palavra "Nome:".

```
echo "<p><strong>E-mail:</strong> $email </p>";
```

- Exibe o valor da variável `$email` dentro de uma tag `<p>` com um `<strong>` para destacar a palavra "E-mail:".

```
if (isset($_REQUEST['nome']) && isset($_REQUEST['email'])) {
```

- Verifica se as variáveis `nome` e `email` estão definidas e não são nulas dentro da superglobal `$_REQUEST`. Se ambas estiverem definidas, o código dentro do bloco IF será executado.

```
echo "<p><em>(Capturado via \$_REQUEST)</em></p>";
```

- Exibe uma mensagem dentro de uma tag `<p>` com um `<em>` para destacar a mensagem, indicando que os dados foram capturados via `$_REQUEST`.

```
}
```

- Fim do bloco IF.

```
?>
```

- Fim do código PHP.

## Testar o código

Certifique-se que o servidor Apache esteja rodando no Xampp, depois acesse seu navegador e digite o seguinte endereço:

- <http://localhost/FormularioPHP/formulario.html>

**ATENÇÃO!** O endereço pode mudar se você tiver dados diferentes nomes para a pasta e para o formulário.

Preencha os campos do formulário e verifique se os dados foram corretamente enviados para "processa\_formulario.php" e exibidos.

## Referencia

Linguagem de Programação Web, Escola Técnica Cristo Redentor, Porto Alegre, 2025.