



Prof. Taciano Balardin
taciano@ulbra.edu.br

E-MAIL:

taciano@ulbra.edu.br



SITE DA DISCIPLINA:

<http://www.taciano.pro.br/>

SENHA:

[@lpweb](#)



Formulários em PHP
Métodos GET e POST

LINGUAGEM DE PROGRAMAÇÃO WEB

AULA 09

Formulários PHP

- Para definir o arquivo que receberá informações, utilizamos o parâmetro **action** da tag **form**

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

- Existem dois métodos de envio:
 - GET
 - POST

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

Método GET

- Método padrão de envio de dados:
 - Se não for especificado o método na tag action, GET é assumido pelo PHP;
 - Os dados são enviados juntamente com o *link* da página (URL).

Método GET

```
<form action="recebe_dados.php">  
  <p>Nome: <input type="text" name="nome" size="30"></p>  
  <p>Idade: <input type="text" name="idade" size="3"></p>  
  <p><input type="submit" value="Enviar" name="enviar"></p>  
</form>
```

http://www.site.com.br/recebe_dados.php?nome=Maria&idade=20

- **?** → representa o início da cadeia de variáveis;
- **=** → separa as variáveis dos seus respectivos valores;
- **&** → identifica o início de uma nova variável.

Método GET

- Desvantagens:
 - Limite de caracteres é de 2048;
 - Os dados enviados são visíveis na barra de endereço do navegador.



Método GET

- **Vantagem:**
- **Passagem de parâmetros por link:** Uma loja virtual com um link para cada produto em que o identificador é passado como parâmetro. Podem ser passados mais de um parâmetro, tais como:

`http://www.site.com.br/produto.php?id_produto=14&sub_cat=7`

- Recebidos os parâmetros, o PHP processa os dados carregando as informações do banco de dados, por exemplo.



Método POST

- Basta usar o método **POST** na tag **form**:

```
<form action="recebe_dados.php" method="POST">  
  <p>Nome: <input type="text" name="nome" size="30"></p>  
  <p>Idade: <input type="text" name="idade" size="3"></p>  
  <p><input type="submit" value="Enviar" name="enviar"></p>  
</form>
```

- Diferente do GET, o POST envia os dados por meio do corpo da mensagem encaminhada ao servidor.

Método **POST**



- **Vantagens:**

- Variáveis não são visíveis:

`http://www.site.com.br/recebe_dados.php`

- Limite de tamanho dos dados é configurado no servidor, sendo mais usado para formulários com grande quantidade de informações
- Envia outros tipos de dados, não aceitos pelo GET, como imagens ou outros arquivos.

Método **POST**

- **Desvantagem:**
 - Não é possível a passagem de parâmetros por url.



Recebendo Informações

Existem duas maneiras de acessar os dados recebidos:

1. Tratar como variáveis adicionando **\$** ao nome dos campo especificados no formulário. O campo nome ficaria **\$nome** e o campo idade ficaria **\$idade**.
 - É necessário que a opção **register_globals** seja ativada
 - Cuidado para não criar outras variáveis com o mesmo nome, pois o valores serão sobrescritos.

Recebendo Informações

Existem duas maneiras de acessar os dados recebidos:

2. Usar os arrays superglobais predefinidos pelo php:

`$_GET` e **`$_POST`**

- Os nome dos campos são usado como chaves associativas **`$_GET`**`["nome"]` ou **`$_POST`**`["idade"]`
- Por questões de segurança é recomendado o uso dos arrays **`$_GET`** e **`$_POST`**.

GET x POST

- **Visibilidade** – A grande diferença entre os métodos GET e POST é a visibilidade. Uma requisição GET é enviada como string anexada a URL, enquanto que a requisição POST é encapsulada junto ao corpo da requisição HTTP e não pode ser vista.
- **Tamanho** – Como a requisição GET é feita via URL, obviamente há uma limitação no tamanho da mensagem enviada. A string não pode conter mais que 2048 caracteres. Já na requisição POST, enviada no corpo da requisição HTTP, o limite pode ser parametrizado via configuração do servidor.

GET x POST

- **Performance** – A requisição GET é relativamente mais rápida, já que na requisição POST há uma perda de tempo no encapsulamento da mensagem.
- **Tipos** – Como a requisição GET é enviada via URL, ela só transporta textos. A requisição POST não tem restrições, pode transportar tanto texto, como dados binários.
- **Cache e Favoritos** – Por se tratar apenas de uma URL, a requisição GET pode ser armazenada em cache ou a URL salva como favorita. A mesma coisa não é possível para requisições POST.

Exercício \$_POST

A biblioteca de uma universidade deseja fazer um formulário que solicite:

- O código e nome do livro que será emprestado;
- O tipo de usuário (professor ou aluno);
- O código e nome do usuário.

Ao enviar o formulário é apresentado um recibo com a estrutura abaixo, considere que o professor tem 10 dias para devolver o livro e o aluno 3.

Nome do Livro (código)

Tipo de Usuário - Nome do aluno/professor (código)

Data para Devolução: 22/09/2014.