



Construção. web sites

Prof: José

ENTRADA DE DADOS COM FORMULÁRIO

Um formulário web é uma página (ou parte de uma página) que contém elementos usados para coletar informações fornecidas pelo usuário, como caixas de texto, listas de seleção de opções, checkboxes e botões de ação.

Um formulário é delimitado pelos marcadores `<form>` e `</form>`.

```
<form name="NOME" action="script" method="método_envio" >
```

elementos do formulário

```
</form>
```

- Onde:
- **name** é o nome do formulário, que pode ser usado para referenciá-lo em uma função, por exemplo.
- **action** indica o script que deverá efetuar o processamento dos dados inseridos no formulário. Em nosso caso, será um script em PHP.
- **method** é o método de envio dos dados para o servidor web onde está o script de processamento. Pode ser GET ou POST.

Métodos GET e POST

Há duas formas que o navegador pode usar para enviar os dados do formulário ao servidor: GET e POST. Ao enviar as informações, o navegador as codifica usando um esquema denominado Codificação de URL (URL Encoding). Nesse esquema, os dados a serem enviados são agrupados em pares contendo um nome e um valor, separados por um caractere **&**. Por exemplo, ao enviar dados de nome, sobrenome e idade de um usuário coletados de um formulário, esses dados seriam agrupados assim:

**nome=Jose&sobrenome=Alves+de+Sousa&i
dade=27**

Note que é usado um caractere "+" no lugar do espaço entre as palavras "Alves" e "Sousa"; outros caracteres especiais são substituídos por valores hexadecimais. Após essa codificação, a informação é então enviada ao servidor usando um dos métodos citados.

- O método de envio GET é limitado a 1024 caracteres no envio de dados, e o método POST não possui limite. Além disso, o método GET envia os dados anexados ao nome do script que foi passado para o parâmetro ACTION, sendo que esses dados (campos e valores) são posicionados após um caractere ? na barra de endereços do navegador, seguindo o nome do script. Claramente esse método não é muito seguro, pois pode expor dados dos usuários durante o envio - por exemplo, uma senha pode ficar claramente visível ao usar esse método. Já no método POST os dados são enviados cabeçalhos HTTP da mensagem e, portanto, é mais seguro e é o método recomendado.
- O PHP fornece um array associativo de nome **\$_POST** que permite acessar os dados enviados usando o método POST.

Controles de Formulários HTML

Seguindo a tag <form> indicamos os dados que o usuário deve fornecer e o tipo de cada informação. Os dados são inseridos em elementos do formulário (controles), com o uso da tag <input>. A sintaxe para uso da tag <input> é a seguinte:

```
<input type="tipo" name="nome" value="valor_inicial" size="tamanho"  
maxlength="caracteres" checked>
```

Os parâmetros da tag input são os seguintes:

- **name**: Identificação do campo.
- **value**: valor inicial do campo; é opcional, exceto para os tipos CHECKBOX e RADIO
- **size**: tamanho do controle; para os tipos TEXT e PASSWORD indica o n° de caracteres que aparecem no campo
- **maxlength**: número máximo de caracteres que podem ser digitados no campo; válido somente para TEXT e PASSWORD.
- **checked**: define se um campo deve estar marcado inicialmente, sendo válido para os tipos CHECKBOX e RADIO.
- **type**: O parâmetro type define o tipo de campo a ser utilizado para entrada de dados.

Os principais tipos de controles são descritos a seguir:

- **text**: controle de texto de linha única
- **password**: caixa de senha.
- **textarea**: controle de texto multilinhas (multiline)
- **checkbox**: campo que permite escolher opções clicando em caixas de verificação
- **radio**: campo que permite escolher uma entre várias opções clicando em um botão de rádio
- **hidden**: campo que não é visível para o usuário, e que é usado para o envio de dados para o servidor sem que o usuário interfira
- **submit**: Botão de envio do formulário
- **reset**: Botão para "zerar" o formulário, limpando todos os campos e restaurando seus valores iniciais
- **file**: Campo usado para o envio de arquivos.
- **select**: Caixa de listagem drop-down que permite listar várias opções, das quais o usuário pode selecionar uma ou mais.


```
<form action="" >
    Nome: <input type="text" name="personal" /><br
/>
    Senha: <input type="password" name="senha"
value="">
    <input type="submit"/>
</form>
<?php
if($_GET){
    echo("Nome " . $_GET["personal"] . " Senha:
" . $_GET["senha"]);
}
?>
```

IF e ELSE

- If, else e elseif em PHP são desvios condicionais para controlar o fluxo do seu código. Com eles você pode fazer com que uma ação seja executada dependendo do resultado de uma, ou várias condições.
- As condições sempre vão retornar um valor verdadeiro (true) ou falso (false) dependendo dos operadores que você utilizar. Dentre eles, temos os Operadores de comparação e os Operadores lógicos.
- Os operadores de comparação são capazes de comparar os valores (se são maiores, menores ou iguais) e os lógicos servem para verificar mais de uma condição.

```
if ( condição ) {  
    ação  
}
```

```
<?php  
// Variáveis numéricas  
$a = 1;  
$b = 2;  
  
// Verifica se $b é maior que $a  
if ( $b > $a ) {  
    echo '$b é maior que $a';  
}  
?>
```

If e else

- Pode acontecer casos em que você precise verificar mais de uma condição e executar uma ação para cada uma, neste caso, é possível utiliza elseif.
- Elseif não tem significado em inglês, mas seria a junção das palavras if (se) e else (se não), então, poderíamos fazer uma tradução livre para "mas se" (nada a ver, mas no código é assim que funciona). Ou seja, com elseif fazemos testes condicionais do tipo: "Se (if) isso for verdadeiro, faça isso; MAS SE (elseif) isso for verdadeiro, faça isso; se não (else) faça isso."
- Assim como else, elseif precisa de um if para existir; o else é opcional neste caso.

```
if ( primeira condição ) {  
  Primeira ação  
} elseif ( segunda condição ) {  
  Segunda Ação  
} elseif ( terceira condição ) {  
  Terceira Ação  
} elseif ( Quantas condições precisar ) {  
  Ação a ser executada  
} else {  
  Ação padrão  
}
```

Exercicio

1. Crie uma pagina de Logim que apresente a mensagem Bem vindo usuario se o usuario: user e a senha: senha
2. Crie uma pagina que peça dois números ao usuário e mostre qual o maior e qual o menor
3. Crie uma pagina que receba 3 numero e diga qual e o maior e qual e o menor

Exercicio

1. Escreva um programa em C que recebe um inteiro e diga se é par ou ímpar.
Use o operador matemático % (resto da divisão ou módulo) e o teste condicional if.
2. Para doar sangue é necessário ter entre 18 e 67 anos. Faça uma de cadastro que pergunte a idade, nome de uma pessoa e diga se ela pode doar sangue ou não. Use alguns dos operadores lógicos OU (||) e E (&&).

Exercicio

1. Crie uma pagina que pergunte o raio de uma circunferência, e em seguida mostre o diâmetro, comprimento e área da circunferência
2. Crie uma pagina que pessa uma nota entre 0 a 10 e cheque se você passou direto, ficou de recuperação ou foi reprovado na matéria.

A regra é a seguinte:

Nota 7 ou mais: passou direto

Entre 4 e 7: tem direito de fazer uma prova de recuperação

Abaixo de 4: reprovado direto