**Fórmulários HTML, PHP e banco de dados MySQL**

**Formulários HTML**

Os formulários HTML possuem alguns elementos cuja função é obter informações do visitante da página. Em geral, tais informações são armazenadas em um banco de dados instalado no servidor, mas também podem ser eniadas por e-mail via código.

Os principais elementos de um formulário HTML são:

* Caixa de texto de 1 linha (textbox);
* Caixa de texto de várias linhas (textarea);
* Caixa de checagem (checkbox);
* Botão de seleção (radio);
* Caixa de listagem (combo box);
* Botão;

Cada elemento desses visto na tela do navegador do visitante possui um código HTML correspondente. Os códigos HTML correspondentes são:

* **<input type="text" name="textfield" maxlenght="80">** - Caixa de texto de 1 linha (textbox);
* **<textarea name="textarea" cols="30" rows="3"></textarea>** - Caixa de texto de várias linhas (textarea);
* **<input type="checkbox" name="checkbox">** - Caixa de checagem (checkbox);
* **<input type="radio" name="radio" value="C">** - Botão de seleção (radio);
* **<select name="select"> <option value="A">1</option> </select>** - Caixa de listagem (combo box);
* **<input type="submit" name="button" value="Submit">** - Botão;

O parâmetro mais importante para se coletar e armazenar dados a partir de um formulário é o parâmetro **name="nome do elemento"**, os demais configuram a aparência do elemento (tamanho, cor, máximo de caracteres digitados, etc):

**<input type="tipo"** - Com exceção da caixa de texto de múltiplas linhas e da caixa de seleção, informa o tipo de elemento de formulário que será mostrado. Os tipos básicos são:

- text: caixa de texto;

- checkbox: caixa de checagem;

- radio: botão de seleção;

- submit / reset: para botões - submit envia os dados e reset limpa o formulário;

**<textarea** - Informa que será mostrada uma caixa de texto multilinhas

**<select** - Informa que será mostrada uma caixa de listagem

**name="nome"** - Este item é importante. É aqui que você dará um nome único para seu elemento de formulário. Porque único? Por que mais tarde você vai usar este nome para "capturar" a informação que o visitante do seu site forneceu.

**value=** - Especifica, em alguns elementos, o valor da informação fornecida pelo visitante a ser "capturada". Note que na caixa de listagem, o valor "A" será capturado quando o visitante selecionar o valor "1". Já no botão "radio" o valor "C" é o que será capturado.Pode ser usado também pra recuperar dados anteriormente armazenados por outro formulário. Em botões é utilizado para informar o que o botão irá realizar

**cols="30" rows="3"** - Para caixas de texto de multiplas linhas. Informa que que teremos 30 colunas por 3 linhas de texto exibidas de tamanho.

**maxlenght="80"** - Em caixas de texto de 1 linha, informa o máximo de caracteres que o visitante poderá digitar (no caso 80 caracteres). Muito útil para validar tamanho de dados de entrada.

**Capturando dados de Formulários HTML com PHP**

Com nosso formulário pronto, necessitamos coletar as informações que os visitantes deixam. A coleta e a manipulação de informações deve ser feita com uma linguagem de programação server side, como [PHP](http://pt.wikibooks.org/wiki/PHP) ou ASP, já que queremos, no final, armazenar tais informações em um banco de dados e um servidor.

PHP é acrônimo recursivo para Hypertext Preprocessor. É uma linguagem server side, quer dizer, roda no servidor de hospedagem. Porém, você pode instalá-lo em seu micro para testar suas páginas antes de carregá-las para o servidor. Neste caso, você precisará também de um servidor, como o Apache. Sugiro você baixar o [wampserver](http://www.wampserver.com), mais prático pois já vem com as três ferramentas juntas, Apache-PHP-Mysql

Não vamos nos ater aos detalhes da linguagem PHP nem à sua instalação. O que nos interessa no momento é a "captura" das informações de um formulário. Para isso, vamos criar um formulário simples em um arquivo chamado form.htm (HTML), constando de uma caixa de texto simples. Depois, vamos criar um aqruivo PHP (grava.php) que vai coletar a informação e, também, gravá-la no banco de dados.

Para que o arquivo grava.php possa fazer seu trabalho, temos de chamá-lo a partir do arquivo form.htm. Isso é feito colocando-se o código dos elementos de formulário entre as tags **<form></form>**, com alguns parâmetros:

arquivo **form.htm** - arquivo que contém o formulário

<Form action="grava.php" method="post">

<input type="text" name="txt\_nome" maxlenght="80" size="40">

</Form>

Vamos às explicações:

action="grava.php"

indica o arquivo que receberá as informações;

method="post"

indica a forma como será trabalahada, no caso, armazenada na página grava.php. Um outro método seria GET. Nesse caso seria feita uma armazenagem temporária no software servidor para posterior processamento. GET é útil quando queremos passar variáveis junto a links. Sua desvantagem é os dados passados ficam expotos ao visitante do site, não sendo recomendável, portanto, usar GET quando estivermos passando senhas ou logins;

<input type="text" name="txt\_nome" maxlenght="80" size="40">

é o código de uma caixa de texto simples, tamanho de 40 caracteres (size) e que permite digitar até 80 caracteres (maxlenght);

arquivo **grava.php**

arquivo que processará o formulário

O importante para ligar o elemento de um formulário a uma variável PHP (marcada com $) é o parâmetro **name=""**, veja abaixo:

<?php

$nome=$\_POST['txt\_nome'];

?>

O que fizemos? Utilizamos a função $\_POST[''] do PHP para coletar a informação contida no elemento de formulário de nome **txt\_nome** e armazenamos na variável $nome.

Agora estamos prontos para manipular esta informação dentro do nosso código PHP ou armazená-la em um banco de dados.

**Armazenando dados de Formulários HTML com PHP em banco MYSQL**

Vamos mostrar como armazenar dados em um banco de dados MYSQL com PHP a partir de um formulário HTML.

Nas seções anteriores, produzimos um arquivo HTML (form.htm) com um formulário simples (coleta somente o nome) e já iniciamos a coleta da informação digitada pelo visitante do site em um arquivo php (grava.php)

arquivo **form.htm** - arquivo que contém o formulário

<Form action="grava.php" method="post">

<input type="text" name="txt\_nome" maxlenght="80" size="40">

</Form>

Até o momento, nosso arquivo possui somente o código para pegar o dados a partir do elemento de caixa de texto simples do nosso arquivo HTML. Agora, continuaremos escrevendo no arquivo grava.php. Acrescentaremos código para:

- Abrir um banco de dados MYSQL em um servidor fictício - Incluir a informação - Fechar o banco de dados

arquivo **grava.php** - arquivo que processa o formulário

<?php

$nome=$\_POST['txt\_nome'];

?>

Após armazenarmos o valor que o internauta digitou na variável $nome (PHP), iremos realizar, no mesmo arquivo, as seguintes operações:

1 - Abrir um banco da dados existente (com PHP) 2 - Construir uma string SQL para inserir o dado e inserí-lo 3 - Fechar o banco de dados

1 - Abrir um banco da dados existente (com PHP)

Vamos lá. Utilizaremos 4 variáveis PHP para abrir o banco de dados:

$Str\_ServerMysql="endereço do server mysql"; $Str\_LoginMysql="login do banco de dados"; $Str\_SenhaMysql="senha do banco de dados"; $mdb="nome do banco de dados";

Feito isso, definiremos uma variável de conexão com este banco ($Conn) com umcomando PHP chamado mysql\_connect("server","login","senha") :

$Conn=mysql\_connect($Str\_ServerMysql,$Str\_LoginMysql,$Str\_SenhaMysql) or die("abertura falhou");

e, utilizando a variável $Conn, faremos a conexão com o banco de dados utilizando o comando mysql\_select\_db("nome do banco de dados","conexão"):

mysql\_select\_db($mdb,$Conn) or die("conexão falhou");

2 - Construir uma string SQL para inserir o dado e inserí-lo

Feita a conexão, agora vamos inserir nosso dado no banco. O dado que nós colhemos no formulário (vide acima) foi armazenado na variável $nome. Criamos mais uma variável para armazenar a string de inserção mysql ($SQL\_Instr), veja:

$SQL\_Instr="INSERT INTO nome da tabela (Nome) values('".$nome."')";

Com a string construída, damos a ordem de armazenamento, com o comando mysql\_query("instrução SQL", "conexão"):

$query=mysql\_query($SQL\_Instr,$Conn);

3 - Fechar o banco de dados

Se tudo deu certo, agora vamos fechar o banco com o comando mysql\_close:

$fecha=mysql\_close($Conn);