LINGUAGEM DE PROGRAMAÇÃO IV - INTERNET

ILP540

- Histórico
- A intenção inicial e acerto final
- Os tipos de dados de campos de form
- TAGS e Atributos de TAGS
- HANDS ON TECH
- Montando um form com tipos de dados HTML 4
- Completando com tipos de dados HTML 5
- 'tabelando' os campos de entrada de dados
- Montando um programa que lê e mostra os dados dos campos do form

- Uma forma desenvolvida para que o usuário pudesse ter alguma interação com o conteúdo de um site.
- A intenção inicial não era ter uma forma de entrada de dados para sistemas de informação. Mas, acabou sendo essa a aplicação mais frequente.
- Bom... Então temos um modo para entrada de dados (lá no lado do cliente).
- Os campos são dispostos em alguns formatos ‘padrões’ (podem ser melhorados com CSS).
- Mas como ligar páginas de forms (é a forma abreviada de formulários) com outros arquivos de páginas? ➔ Atributos de TAGs. Mas quais TAGs?
- Como enviar os valores digitados para outros programas? (Os servidores de páginas são bons em entender linguagens de marcação – ML. Mas são bons só nisso! Não tratam os dados... ➔ Atributos de TAGs...
-

- A TAG que trata de formulários é a TAG <form>...</form>.
- Mas... O que é um formulário?
- Formulário é uma forma que a HTML ‘arrumou’ para o leitor (usuário) de uma página ter alguma interação com a mesma. O envio de uma mensagem, a indicação de uma escolha são formas de interação usuário/página.
- Em HTML4 (mesmo com CSS) o formulário (form) dependia de alguns segmentos de códigos na camada CGI (ou em javascript – executando do lado do cliente). Para melhorar o desempenho do navegador, do lado do cliente, em HTML 5 foram introduzidos mais tipos de campos de formulário (podendo retirar parte da carga de JS do lado cliente).
- Mas de outro lado as linguagens da camada CGI começaram a melhorar e dar mais recursos para tratar os dados que vinham de um formulário.
- Resumo: em HTML podemos usar diferentes tipos de campos de formulário para entrada de dados.
- Em um quadro apresentamos os tipos (HTML 4 & 5), a seguir...
-

- HTML 4 & 5... Tipos de campos de formulários:
- Todos os tipos que valem para a versão 4 valem para a versão 5.
- Parece que o W3C melhorou o tratamento da HTML para desenvolvimento de sistemas de informação.
- Para referência completa para todos os elementos de forms acesse: http://www.w3schools.com
- **Importante**: a HTML5, os navegadores e os servidores de páginas **NÃO TÊM** atualizações sincronizadas.

  Por isso, ainda existem navegadores que não operam o esperado quando trabalhamos com alguns tipos de dados da HTML5.

  Por isso vai a dica:

  SEMPRE TESTE O SEU SITE EM VÁRIOS NAVEGADORES.

  E mesmo depois de hospedados: TESTEM seu site!!

HTML 4	HTML 5
text	color
radio	date
checkbox	datetime-local
button	time
reset	week
submit	month
hidden	email
password	file
	image
<textarea>...</textarea>	number
	range
<select>	search
<option>...</option>	tel
</select>	url

- A TAG <form> recebe atributos, porém dois deles são obrigatórios:
- **ACTION** – Declara o nome (e localização) do arquivo de programa que vai ler e tratar dos valores que são digitados nos campos do form.
- Os servidores de páginas não tem a função de tratar os dados digitados nos campos de formulários. Por isso na declaração do formulário devemos indicar qual o programa que vai receber os dados digitados. Isso se faz no atributo ACTION da declaração: <form action="nomedoprograma"...>...</form>
- No atributo action escrevem-se URL, na forma:
  - Serviço : Domínio / Sub-Diretórios / Arquivo → estrutura do nome da URL
  - Podem ser escritos de dois modos básicos:
    Caminhos Absolutos:

    http://localhost/ilp/teste/form.html → exemplo de URL (não existem os espaços).
    www.w3schools.com/ (index.html), o arquivo quando for index.html ou index.php pode ser omitido

    Caminhos Relativos (usando . ou .. para localizar um arquivo ou sub-diretório relativamente localizado em relação ao arquivo onde o formulário está escrito. Por exemplo:
    ../sub1/arquivo - arquivo está em sub1 que é um sub-dir ao lado do diretório onde o arquivo do formulário está.
    ./arquivo - arquivo está no mesmo diretório onde o arquivo do formulário está.
    ./sub2/arquivo - arquivo está no sub-diretório sub2 onde o arquivo do formulário está.

- A TAG <form> recebe atributos, porém dois deles são obrigatórios:
- METHOD – Declara qual é o método de transmissão de valores dos campos entre o servidor de página e o programa (ou o ambiente de controle de execução) que vai receber os campos.
-
- A String de Transmissão de Dados (STD) é formada por segmentos de atribuição campo-valor separados pelo sinal '&'.
-
- Então... Se um form tem dois campos C1 e C2 e o usuário digita: ABCD em C1 1234 em C2, a STD fica assim: C1="ABCD"&C2="1234"
-
- Hã... Ok! E o que tem a ver a **STD** com o método? O Método pode ser:
- **GET** – A **STD** é limitada ao comprimento total de 1kb até 2kb (dependendo do servidor de página). Isso pode limitar a quantidade de campos em um form.
- **POST** – O Servidor de página usa uma variável para guardar um endereço de memória (no servidor) onde a **STD** fica armazenada.
-
- O GET é mais rápido e limitado, o POST é mais lento e quase 'ilimitado' em tamanho (conteúdo dos dados).
-
- Quanto à segurança... A **STD** funciona entre Servidor de Páginas e programa que trata os campos (camada CGI). Entre Cliente-Servidor os dois métodos são vulneráveis (mesmo com POST e HTTTPS).

- Desenvolvendo um PA-PHP que recebe os dados de um formulário e exibe os dados em uma página simples em HTML.
- Note: este programa poderia exibir estes dados dentro de um outro form e pedir ao usuário para confirmar seus valores adotando então a gravação dos mesmo em um banco de dados.
- Mas agora vamos só montar o 'mostrador' dos dados.
- Onde gravar o arquivo .html e o .php?
- Em qualquer sub-dir sob seu localhost! Para facilitar, escreva PA-PHP e o montador do form no mesmo diretório e no ACTION use './' para localizar o .php.
- Lembra da STD: a string que transmite os dados no formato "$campo_1$=$valor_1$&$campo_2$=$valor_2$&...&$campo_n$=$valor_n$"?
- Lembrando:
  você tem que "quebrar" a STD e definir para cada atribuição... Xi!
- Calma... O PHP tem três modos de facilitar a quebra da STD.

- Calma... O PHP tem três modos de facilitar a quebra da STD.
- Quando o form transmite os dados o PHP cria vetores para receber estes dados onde os índices do vetor podem ser os nomes dos campos do form e os seus valores estarão nas posições respectivas do vetor.
- Os vetores dependem do METHOD do form.
- Para o método GET são gerados o vetor $_GET[] (versão≤4.7) $_REQUEST[] (versão > 4.7).
- Para o método POST são gerados o vetor $_POST[] (versão≤4.7) $_REQUEST[] (versão > 4.7).
- Portanto para versões do PHP ≥ 5.0 usamos o $_REQUEST[]
- Mas como?

- Veja o código desta página… [Este é o Slide REFERÊNCIA]

```
<!DOCTYPE html>
<html>
<head><meta charset='UTF-8'><title>FORM</title><link rel='stylesheet' type='text/css' href='./estilos.css'></head>
<body>
Primeiro Formulário<br>
<form action='./recebe.php' method='POST'>
<input type='hidden' name='bloco' value='1'>
<table border=1>
<tr><td>Nome:</td>    <td><input type='text' name='nome' size='60' maxlength='150'></td></tr>
<tr><td valign='top'>Resenha:</td><td><textarea name='resenha' rows='5' cols='60'>Resenha das atividades profissionais</textarea></td></tr>
<tr><td>Tipo<br>Sanguíneo:</td><td>(<input type='radio' name='tiposangue' value='1'>)-A |
                                   (<input type='radio' name='tiposangue' value='2'>)-B |
                                   (<input type='radio' name='tiposangue' value='3'>)-AB |
                                   (<input type='radio' name='tiposangue' value='4'>)-O</td></tr>
<tr><td>Profissão:</td><td><select name='profs'>
                              <option value='01'>Administrador</option>
                              <option value='02'>Agropecuarista</option>
                              ...
                              <option value='11'>Engenheiro civil</option>
                            </select></td></tr>
<tr><td>Hobbies:</td><td><input type='checkbox' name='hobby[]' value='1'>-Ler |
                         <input type='checkbox' name='hobby[]' value='2'>-Cantar |
                         <input type='checkbox' name='hobby[]' value='3'>-Andar |
                         <input type='checkbox' name='hobby[]' value='4'>-Ver filmes&séries</td></tr>
<tr><td>Acidentes<br>Sofridos:</td><td><select name='acids[]' Multiple>
                                        <option value='01'>Atropelamento</option>
                                        <option value='02'>Fratura</option>
                                        ...
                                        <option value='11'>Cortes & Facadas</option>
                                       </select> Ctrl+Clique para marcar mais de um item</td></tr>
<tr><td></td><td><input type='submit' name='bt' value='Enviar os dados'>
                 <button type='submit' name='bt' value='sub'>Enviar os dados</button>
                 <button type='reset'>Limpar form</button></td></tr>
<tr><td></td><td></td></tr>
</table> </form> </body></html>
```

- A página codificada no slide referência gera a visualização...

Primeiro Formulário

Nome:	Alberto Pinheiro
Resenha:	Estagiário, engenheiro, professor
Tipo Sanguíneo:	( )-A | ( )-B | ( )-AB | ( )-O
Profissão:	Designer
Hobbies:	-Ler | -Cantar | -Andar | -Ver filmes&séries
Acidentes Sofridos:	Fratura Intoxicação Afogamento Incêncio Ctrl+Clique para marcar mais de um item
	Enviar os dados | Enviar os dados | Limpar form

- E ESTE PA-PHP recebe os dados do form (action na TAG <form>) apresentado no slide referência.

```php
<?php
# Este é o programa recebe.php.
# Serve como exemplo para captura de dados digitados em campos de formulários.
# exibindo os valores escolhidos no vetor $_REQUEST[]
printf("<HTML><head><meta charset='UTF-8'><title>Leitor-FORM</title>
                <link rel='stylesheet' type='text/css' href='./estilos.css'>
            </head>
        <BODY class='fC3C3C3'>");
printf("Mostrando todos os valores do vetor de uma vez:\n");
printf("<pre>\n");
print_r($_REQUEST);
printf("</pre>\n");
printf("</BODY></HTML>");
?>
```

- E ao processar a exibição dos dados digitados gera a tela:

Mostrando todos os valores do vetor de uma vez:

```
Array
(
    [bloco] => 1
    [nome] => Alberto Pinheiro
    [resenha] => Estagiário, engenheiro, professor
    [tiposangue] => 1
    [profs] => 04
    [hobby] => Array
        (
            [0] => 1
            [1] => 2
        )

    [acids] => Array
        (
            [0] => 02
            [1] => 05
            [2] => 06
        )

    [bt] => sub
)
```

- No segmento de código do PA-PHP foram apresentadas duas funções do PHP:
- Lendo os dados enviados do servidor de páginas:
- Os Vetores $_REQUEST[] e $_POST[] já fazem isso. Depois é só lidar com os índices nomeados (índices com 'identificadores' iguais aos nomes dos campos dos forms.
- Emitindo uma página (do PHP para o servidor de páginas):
- Todo o resto do código-fonte: *inicia uma página* e *emite TAGs HTML*, de formas diferentes, usando o ***printf()*** com valores do vetor 'dentro' e 'fora' da *string* do comando). Além disso se usa o comando ***print_r()*** que é a função do PHP para exibir os valores nas posições de um vetor (e no exemplo, também de um sub-vetor).
- Usou-se o '#' para declarar o que está à direita do símbolo como um comentário (poderia ser usado /*...*/ do mesmo modo como em "C".
- Agora ... hands on tech...

- Hands-on tech - ***EXERCÍCIO***
- Você leu no Slide 10 toda a especificação de uma página que monta quase todos os tipos de campos de formulário da HTML (<4.0). Então...
- Propomos um exercício:

- Crie um diretório com nome tar6 sob o diretório //localhost/ilp540/SigleDoSeuNomeCompleto/htmlcss/
- Edite um arquivo .HTML (nome: **`form.html`**) com uma página que contenha uma tabela com DUAS colunas, com borda de UM pixel colapsada, acomodando um formulário com todos os tipos de campos da HTML 4 e mais os seguintes tipos de dados para os campos do formulário da HTML 5: date, time, month, number, tel, url e range.
- Além desses tipos de dados para campos da HTML 5, escolha mais 2 tipos de dados [HTML 5] para seu formulário. Na última linha da tabela, a coluna da esquerda deve aparecer sem conteúdo e na coluna da direita devem aparecer TRÊS botões como exemplificado no Slide Referência. Além destes três botões construa um botão VOLTAR, implementado com <button> e o atributo onclick executando history.go(-1)
- Para os campos deste formulário use o atributo *placeholder*. Este atributo não foi mencionado em aula e nem neste material de suporte.
- Complemente seu exercício de desenvolvimento de formulário editando um PA-PHP que receba e exiba os valores dos campos digitados. Este arquivo deve ser gravado no diretório TAR6 e ser executado ao submeter os dados dos campos do formulário. O nome do seu PA-PHP dever ser tar6.php.
- Para entrega, siga ao mesmos procedimentos das tarefas anteriores. A Tarefa TEAMS JÁ disponibilizada.

LINGUAGEM DE PROGRAMAÇÃO IV - INTERNET

ILP540