Módulo 2 - PHP | Exercício 6 – AVILA Crew Website

Arrays em PHP

Temas abordados

- Funções
- Arrays
- Sessões
- Estruturas de repetição
- Listagens dinâmicas

Versão resolvida em:

- http://labmm.clients.ua.pt/deca_16L4/deca_16L4_p/avila06/

Neste exercício pretende-se disponibilizar um sistema de registo no Website da AVILA Crew.

Para este exercício não é necessário descarregar ficheiros adicionais.

PARTE 1. NAVEGAÇÃO

Nesta parte pretende-se alterar a barra de navegação do Website da AVILA Crew. Para tal:

1.1. Faça uma cópia da página “index.php” e atribua-lhe o nome “sessoes_inscricao.php”;

1.2. Crie um componente designado por “session_register.php”;

1.3. Altera a página “sessoes_inscricao.php” para incluir o componente criado no ponto anterior;

1.4. Abra o ficheiro “components/navigation.php” e adicione uma ligação no menu com a designação “Inscrição” a página “sessoes_inscricao.php”;

Publique e verifique as alterações!

PARTE 2. FORMULÁRIO DE INSCRIÇÃO NAS SESSÕES AVILA Crew

Nesta parte vamos criar o formulário de registo numa sessão, para tal:

2.1. No componente “session_register.php” adicione um formulário idêntico ao que encontra no exemplo fornecido. Não esqueça os textos de instrução que devem ser introduzidos como placeholder dos campos;

2.2. O elemento de tipo “select” do formulário deve ser construído dinamicamente. Ou seja, as opções fornecidas devem ser criadas a partir dos dados do array onde armazenou os nomes das sessões;

2.3. Os dados do formulário devem ser enviados via “POST” para o componente “sessions_register_control.php”.

2.4. Crie o componente “sessions_register_control.php”;

2.5. Coloque um var_dump() da super-global $_POST neste último componente e garanta que recebe a informação esperada.

Publique e verifique as alterações!

PARTE 3. VALIDAÇÃO DE UM CAMPO DO FORMULÁRIO DE INSCRIÇÃO

No componente “sessions_register_control.php” adicione a lógica necessária para validar os dados preenchidos. Para começar sugere-se que se teste todo o procedimento apenas para o campo correspondente ao “Nome”.

3.1. Crie uma função “valida_campo()” que tem dois parâmetros de entrada: o valor introduzido no campo do formulário e o tipo de informação válida para esse campo;

3.2. Caso o tipo de informação seja **“nome”** a função deve verificar se o valor introduzido no campo está de acordo com as seguintes regras:
- não ser nulo
- não ser numérico
- ter no mínimo 3 caracteres
- ter no máximo 20 caracteres

3.3. Se alguma das regras anteriores não for verificada a função deve retornar o valor booleano *false*. Caso contrário deve retornar o valor booleno *true*.

3.4. Invoque a função neste componente e verifique se retorna o valor esperado, testando os diferentes cenários possíveis para o valor introduzido no campo “Nome”.

Publique e verifique!

Para implementar a lógica de feedback de erros no formulário vamos utilizar a querystring. Ou seja, o componente de controlo deve devolver informação na querystring que permita na página do formulário saber se a inscrição teve sucesso ou se falhou. Caso tenha falhado será necessário saber em que campos ocorreram erros.

3.5. No componente “sessions_register_control.php”, caso a validação do campo “Nome” não tenha erro, o componente deve redirecionar para a página do formulário de inscrição, retornando na querystring o parâmetro “registo” com o valo “ok”;

3.6. Caso contrário, deve redirecionar para a mesma página, mas na querystring deve retornar um parâmetro com o nome do campo a ser validado e o valor “erro”;

3.7. No componente “session_register.php”:

3.7.1. Verifique se foi passado o parâmetro que sinaliza sucesso na submissão do formulário e, nesse caso, apresente uma mensagem de “Registo efetuado com sucesso”;

3.7.2. Para o campo de texto do “nome”, caso tenha sido passado o respetivo parâmetro de erro, introduza no texto do formulário um “*”, utilizando a classe “text-danger” do bootstrap.

Publique e verifique!

PARTE 4. VALIDAÇÃO DE TODOS OS CAMPOS DO FORMULÁRIO DE INSCRIÇÃO

No componente “sessions_register_control.php” adicione a lógica necessária para validar todos os dados preenchidos.

4.1. Não deve criar novas funções. A função deve estar preparada para fazer as validações dependendo do valor passado no parâmetro “tipo”, nomeadamente:

4.1.1. O tipo “numero” deve validar se o valor introduzido não é nulo e se é um valor numérico;

4.1.2. O tipo “email” deve validar se o valor introduzido contem a string “@ua.pt”.

4.2. Implemente no componente do formulário as validações necessárias para mostrar o sinal de erro em qualquer um dos campos de texto do formulário.

Publique e verifique!

PARTE 5. AVANÇADO

Caso a validação tenha dado erro o utilizador não devia ser obrigado a reintroduzir todos os valores introduzidos anteriormente. Para essa situação, implemente uma solução que permita recuperar os valores introduzidos pelo utilizador antes da submissão.

PARTE 6. TPC

O sistema de validação desenvolvido podia ser mais flexível, permitindo passar parâmetros opcionais relativos a cada tipo de dados. Por exemplo, ao validar um campo de tipo “nome” podia ser possível passar os limites mínimo e máximo dos caracteres. Outro exemplo, num campo do tipo “número” podia também ser possível passar esses limites.
Implemente um sistema que permita esse tipo de flexibilidade.