Persistência de dados

1. Utilizando o formulário **registro.php,** fazer com que todos os campos preenchidos pelo usuário persistam depois do envio dos dados. Ou seja, retornar a página registro.php com os campos preenchidos com os dados enviados.

   Ex: <input type="text" name="nome" value="**<?php echo $_POST['nome']; ?>**" />

JSON

1. Criar uma variável $a contendo um array ('a'=>1, 'b'=>2, 'c'=>'Eu <3 JSON'):
   a. Fazer echo da variável $a.
   b. Utilizando json_encode, transformar o array em um json e atribuí-lo a $a.
   c. Fazer var_dump da variável $a.
   d. Criar uma nova variável $b contendo o json_decode da variável $a.
   e. Fazer var_dump de $b.
   f. Imprimir a frase "Eu <3 JSON | 1 | 2 |" utilizando os dados da variável $b.

ARQUIVOS

2. Criar um arquivo novo chamado **arquivos.php**.
   a. Criar uma função que verifique se existe um arquivo chamado **texto.txt** no mesmo diretório de **arquivos.php**. Se o arquivo existir, deve ser aberto com direitos de leitura e escrita, para que seja possível adicionar informações. Se ele não existir, deve ser criado com direitos de leitura e escrita.
   b. A frase "Olá mundo! testando." deve ser escrita 100 vezes no arquivo, 1 vez por linha. Depois disso, o arquivo deve ser fechado.
   c. Mostrar o conteúdo do arquivo **texto.txt**, lendo todo o arquivo de uma vez.
   d. Mostrar o conteúdo do arquivo **texto.txt**, lendo e imprimindo linha por linha.

ARQUIVOS JSON

3. Criar um arquivo **categorias.json** com os seguintes dados:

```
{"categorias" : [
        {"id": 1, "nome": "Esportes"},
        {"id": 2, "nome": "História"},
        {"id": 3, "nome": "Entretenimento"},
        {"id": 4, "nome": "Geografia"},
        {"id": 5, "nome": "Arte"}
        ]}
```

   a. Ler o arquivo e **imprimir** uma checkbox para cada categoria, que possam ser enviadas como array em um formulário. O value de cada checkbox será o id, e a label mostrada à direita será o nome proveniente do json, como vemos no array.

4. Criar um novo arquivo PHP.
   a. Criar uma função que defina uma variável do tipo string e faça echo dessa variável.
   b. Adicionar à função um echo do resultado de criptografar a variável com a função md5.
   c. Adicionar à função um echo do resultado de criptografar a variável com a função sha1.
   d. Adicionar à função um echo do resultado de criptografar a variável com password_hash, utilizando como algoritmo: PASSWORD_DEFAULT.
   e. Adicionar à função um echo do resultado de criptografar a variável com password_hash, utilizando como algoritmo: PASSWORD_BCRYPT.

Exercícios Complementares

5. Modificar **registro.php** (pode ser o arquivo utilizado nas aulas anteriores ou o arquivo utilizado no trabalho integrador) para que:
   a. Valide os dados do formulário.
   b. Esclareça os erros, se houver.
   c. Em caso de erro, preencha os campos que o usuário já tinha preenchido.
   d. Salve os dados do usuário em um array (a senha deve estar criptografada!).
   e. Transforme o array em JSON.
   f. Salve o usuário em um arquivo de texto.
   g. Em caso de sucesso, redirecione para uma página de sucesso.

Observação: A inscrição é um processo executado muitas vezes, por isso, cada usuário novo deve ser adicionado ao final do array de usuários. É recomendável que o arquivo contenha uma estrutura do tipo: {“usuarios”: [{...},{...},{...},{...},{...}]} em que {...} é o json_encode do array com os dados do usuário específico inscrito no momento de salvar.

6. Criar um arquivo chamado **login.php** (pode ser o arquivo utilizado nas aulas anteriores ou o arquivo utilizado no trabalho integrador) para que, ao enviar o formulário, verifique se o usuário existe ou não no arquivo de texto gerado no ponto anterior. Se o usuário existir, deve receber as boas-vindas. Caso contrário, o arquivo deve indicar que ele não existe.

7. Modificar **registro.php** para que, ao fazer a inscrição do usuário, valide no arquivo de texto que não existe outro usuário igual (é possível validar por nome de usuário, e-mail ou RG, dependendo do formulário).