

PRO3151 – Laboratório de SI

L06 – PHP Hypertext Preprocessor

Terceira aula de PHP

Prof. Dr. Marcelo Schneck de Paula Pessôa

Prof. Dr. Marco Aurélio de Mesquita

Prof. Dr. Mauro de Mesquita Spinola

O que já vimos



- Arquivos php
- Integração php com HTML
- Entrada de dados – post e get
- Manipulação de strings
- Operadores aritméticos
- Operadores lógicos
- Controle de fluxo do programa

O que vamos ver



- Session()
- Arquivos

- A `session()` é um comando que permite armazenar dados voláteis em um escopo maior que um arquivo
- As variáveis normais possuem validade dentro de uma mesma página
- https://www.w3schools.com/php/php_sessions.asp

Session – criação da variável



- Sintaxe

```
session_start();  
// Recebe os dados do arquivo anterior  
  
$_SESSION['data'] = POST['variavel1'];
```

- A variavel1 HTML torna-se a variavel data na SESSION
- Na página pode ser usada como uma variável comum

session – criação da variável



- Sintaxe – alternativa de entrada

```
session_start();  
// Recebe os dados do arquivo anterior  
  
$_SESSION['produto'] = isset($_POST['produto'])?$_POST['produto']:'';  
$produto=$_SESSION['produto'];
```

- Uma boa prática é manter o mesmo nome HTML na variável PHP
- isset recebe o valor se digitado e um "" (sem caracter) se nada for digitado evitando erro
- Na linha de baixo a variável \$produto sobe para a session

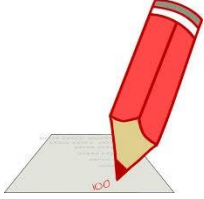
session – recepção

- Sintaxe – recepção dos dados
- Os dados podem ser usados diretamente ou atribuídos a variáveis locais
- Para usar os nomes originais com `$_SESSION` basta criar o `session_start()`;
- Para usar variáveis com nomes mais fáceis de visualizar passar os valores para estas

```
session_start();

$data=$_SESSION['data'];
$produto=$_SESSION['produto'];
$cliente=$_SESSION['cliente'];
$nsad=$_SESSION['nsad'];
$fi=$_SESSION['fi'];
$vmin=$_SESSION['vmin'];
$vmax=$_SESSION['vmax'];
$v=$_SESSION['v'];
$n=count($v);
$is=$_SESSION['i'];
$vfi=$_SESSION['vfi'];
$dmax=$_SESSION['dmax'];
$rd=$_SESSION['rd'];
$f=$_SESSION['f'];
$vd=$_SESSION['vd'];
$B=trim($_SESSION['B']);
$nucro=trim($_SESSION['nucsec']);
$gap=trim($_SESSION['gap']);
```





Exercício 1



- Fazer uma entrada de dados de 4 números
- Receber os dados em uma página
- Preparar uma terceira página apresentando cada um dos números, a sua média e a sua soma

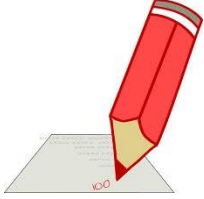
- *Salvar os três arquivos em uma pasta Ex1*

- Já vimos que há objetos persistentes e não persistentes
- Os objetos persistentes são gravados em arquivos ou armazenados em bancos de dados
- Hoje vamos estudar como trabalhar com esses arquivos

Objetos persistentes – em arquivos



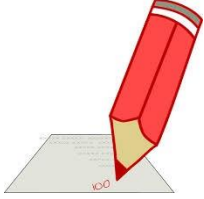
- Os arquivos são estruturas que a linguagem de programação e o sistema operacional oferecem para que sejam guardados objetos persistentes
- Os arquivos são estruturas de informação padronizadas para que as aplicações possam realizar a leitura
- Podem ser arquivos texto, arquivos com imagem, filme, entre outros
- No caso que será estudado, os arquivos do PHP são arquivos texto ou arquivos binários



Exercício 2



- Abra uma planilha
- Crie uma tabela com 3 colunas: nome, NUSP, curso
- Entre com 10 dados (10 nomes, números USP e cursos)
- Salvar o arquivo com o nome alunos_excel
- Salvar de novo o arquivo no formato CSV com o nome alunos_csv



Exercício 2



- Abrir o arquivo excel original no Notepad++
 - Observar o conteúdo
 - Abrir o arquivo no formato .csv
 - Observar o conteúdo
 - Qual é a diferença ???
-
- *Salvar os dois arquivos em uma pasta Ex2*

Arquivos planos



- Agora vamos ler o arquivo csv com o PHP

Objetos persistentes – em arquivos



- Nos próximos slides estão descritos os comandos de manipulação de arquivos do PHP para que se possa armazenar objetos persistentes sem o uso de bancos de dados

Lendo um arquivo, linha por linha



```
<?php
$file=fopen("HELP.txt","r") or exit("Arquivo não encontrado.");

while(!feof($file))
{
    echo "<font face='Arial'><i>".fgets($file)."</i></font><br/>";
}
fclose($file);
?>
```

Lendo um arquivo, caracter por caracter



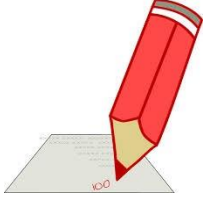
```
<?php
$file=fopen("HELP.txt","r") or exit("Arquivo não encontrado.");

while(!feof($file))
{
    echo "<font face='Arial'><i>".fgetc($file)."</i></font><br/>";
}
fclose($file);
?>
```


Modos de abertura de arquivo



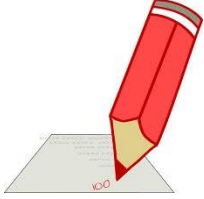
Modes	Description
r	Read only. Starts at the beginning of the file
r+	Read/Write. Starts at the beginning of the file
w	Write only. Opens and clears the contents of file; or creates a new file if it doesn't exist
w+	Read/Write. Opens and clears the contents of file; or creates a new file if it doesn't exist
a	Append. Opens and writes to the end of the file or creates a new file if it doesn't exist
a+	Read/Append. Preserves file content by writing to the end of the file
x	Write only. Creates a new file. Returns FALSE and an error if file already exists
x+	Read/Write. Creates a new file. Returns FALSE and an error if file already exists



Exercício 3



- Agora crie um programa em PHP que leia o arquivo csv do exercício anterior e mostre os dados na tela
- Use os comandos apresentados na tabela
- *Salvar o arquivo em uma pasta Ex3*



Exercício 3b



- Se o que foi apresentado não está por célula do arquivo original, separe cada célula em uma variável diferente

Gravando um arquivo

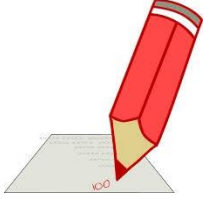
(em uma pasta *upload* no servidor)



```
<?php
$file=fopen("upload/teste1.txt","w+");
if($file)
{
    if(!fwrite($file,"Linha 1 do arquivo teste1.txt"))
        die("ERRO");
    echo "OK!";
    fclose($file);
}
?>
```

Como resultado do trecho acima é criado o arquivo “*teste1.txt*” e no seu conteúdo é armazenado “Linha 1....”

Observar que no diretório “*htdocs*” é necessário ter uma pasta “*upload*”.



Exercício 4

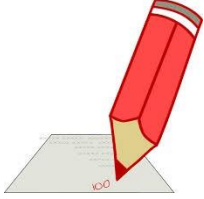


- Agora prepare um form para incluir dados no arquivo anterior (nome, NUSP, curso)
- Recebidos os dados gravar no arquivo plano separado por vírgula para manter a formatação

Observação – no caso não precisa fazer upload pois o arquivo já está no localhost e o acesso é direto

O upload é um comando HTML e não PHP

- *Salvar o arquivo em uma pasta Ex4*



Exercício 5



- O arquivo com mais dados preenchidos pelo PHP pode ser lido na planilha excel
- Faça isso e verifique se é verdade

Upload de arquivo



Filename: no file selected

```
<html>
<body>

<form action="upload_file2.php" method="post" enctype="multipart/form-data">
Filename: <input type="file" name="file" id="file" />
<br />
<input type="submit" name="submit" value="Submit" />
</form>

</body>
</html>
```

□ Upload-
form.htm

2
3

```
<?php
if ($_FILES["file"]["error"] > 0)
{
    echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
else
{
    echo "Upload: " . $_FILES["file"]["name"] . "<br />";
    echo "Type: " . $_FILES["file"]["type"] . "<br />";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
    echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
?>
```

□ upload_file.php


```
<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
    if ($_FILES["file"]["error"] > 0)
    {
        echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
    }
    else
    {
        echo "Upload: " . $_FILES["file"]["name"] . "<br />";
        echo "Type: " . $_FILES["file"]["type"] . "<br />";
        echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
        echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";

        if (file_exists("upload/" . $_FILES["file"]["name"]))
        {
            echo $_FILES["file"]["name"] . " already exists. ";
        }
        else
        {
            move_uploaded_file($_FILES["file"]["tmp_name"],
            "upload/" . $_FILES["file"]["name"]);
            echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
        }
    }
}
else
{
    echo "Invalid file";
}
?>
```

upload_file2.ph

p

2
5

Resumo comandos arquivos

VAR=fopen(nome do arquivo, operação) – abre arquivo

fclose (VAR) – fecha arquivo (sempre ao final)

feof (VAR) – fim arquivo (true ou false)

fgets (VAR) – le um string – até fim da linha

fgetc (VAR) – lê um caracter

fwrite (VAR, conteúdo) – grava o conteúdo

file_exists(nome do arquivo) – verifica se o arquivo existe

file_size(nome do arquivo) – lê o tamanho do arquivo