

# PRO3151 – Laboratório de SI

## L04 – PHP Hypertext Preprocessor

### Segunda aula de PHP

Prof. Dr. Marcelo Schneck de Paula Pessôa

Prof. Dr. Marco Aurélio de Mesquita

Prof. Dr. Mauro de Mesquita Spinola

# PRO2511



- **PHP** – **PHP: Hypertext Preprocessor**
- Aula 2

# PHP – o que vimos?

- Sintaxe da linguagem
- Uso de variáveis para receber dados
- Integração com HTML

# PHP – o que falta ver?

- Escopo de variáveis
- Controle de fluxo
- Operadores aritméticos e lógicos
- Manipulação de strings
- Tratamento de arrays
- Funções
- Tratamento de arquivos planos

# Escopo de variáveis



# Escopo de variáveis

- Identificação de variáveis com \$
- Cada variável vale dentro da página
- Mudou de página é perdida a variável
  
- Significa que uma variável é sempre local dentro de uma única página

# Escopo de variáveis

- Passagem de valores de uma para outra página requer:
  - Uso de GET ou POST (*já visto*)
  - Uso de `<input type='hidden' ...`
  - Uso de `session()` (*será visto em funções*)

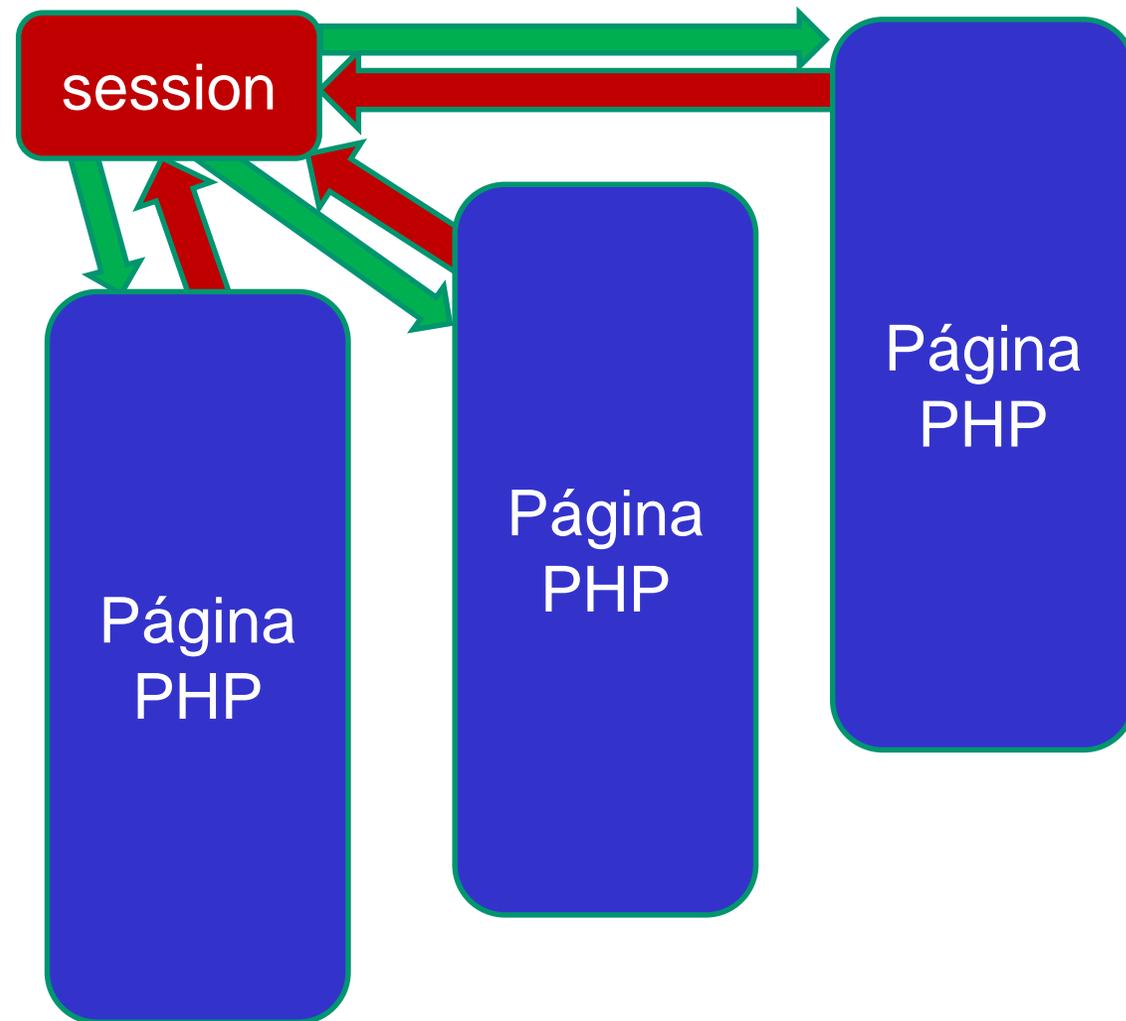
# Escopo de variáveis

- Passagem de valores de uma para outra página requer:
  - Uso de GET ou POST
  - Uso de `<input type='hidden' ...`
  - Uso de `session()`

# Escopo de variáveis

session()

- Trata-se de um truque para preservar as variáveis em uma área de memória separada das páginas



# Controle de Fluxo



# Controle de fluxo

O PHP como outras linguagens de programação possui controle do fluxo do programa:

- If
- Switch
- While
- For

# Controle de fluxo



## ■ If

```
<?php
    if ($condicao)
    {
        echo "Aqui se verdadeiro";
    }
Else
{
    echo "Aqui se falso";
}
?>
```

# Controle de fluxo

## ■ Switch

```
<?php
    switch($cor)
    {
        case "amarelo":
            echo "A cor é amarela";
            break;
        case "vermelho":
            echo "A cor é vermelha";
            break;
        default
            echo "Não sei que cor é";
            break;
    }
```

# Controle de fluxo

## ■ While

```
<?php
    while ($condicao)
    {
        executa o bloco;
    }
```

- Se \$condicao for TRUE o loop é infinito

# Controle de fluxo

## ■ For

```
<?php
    for($j=0; $j<6; $j++)
    {
        executa o bloco;
    }
```

# Exemplo if



```
<?php
$day=date(l);
/* date, parametro l: retorna dia da semana
   Sunday, Monday, Tuesday, Wednesday,
   Thursday, Friday, Saturday
*/
// echo $hora;

if ($day == "Saturday")
    echo "Dia de Festa!";
else
{
    echo "Dia de descansar ou estudar.";
    echo "<br/> lembrar das tarefas!";
}
?>
```

```
<?php
$day=date(l);
/* date, parametro l: retorna dia da semana
   Sunday, Monday, Tuesday, Wednesday,
   Thursday, Friday, Saturday
*/
// echo $hora;

if ($day == "Saturday")
    echo "Dia de Festa!";
elseif ($day == "Sunday")
    echo "Dia de descansar.";
else
    echo "Dia de estudar.";
?>
```

# Exemplo switch - case



```
<?php
$day=date(l);
/* date, parametro l: retorna dia da semana
   Sunday, Monday, Tuesday, Wednesday,
   Thursday, Friday, Saturday
*/

switch($day)
{
  case "Sunday":
    echo "Domingo";
    break;
  case "Saturday":
    echo "Sábado";
    break;
  default:
    echo "Dia de aula!";
    break;
}

?>
```

# Exemplo for - if



```
<?php
echo "<table width='100' align='center'>";
for($i=0; $i<=5; $i=$i+1)
{
    if($i % 2 == 0)
    {
        echo "<tr>";
        echo "<td style='background-color:red'>";
        echo $i;
        echo "</td>";
        echo "</tr>";
    }
    else
    {
        echo "<tr>";
        echo "<td style='background-color:green'>";
        echo $i;
        echo "</td>";
        echo "</tr>";
    }
}
echo "</table>";
?>
```

# Exemplo while - if



```
<?php
echo "<table width='100' align='center'>";
$i=0;
while ($i<=5)
{
    if($i == 3)
        break;
    if($i % 2 == 0)
    {
        echo "<tr>";
        echo "<td style='background-color:red'>";
        echo $i;
        echo "</td>";
        echo "</tr>";
    }
    else
    {
        echo "<tr>";
        echo "<td style='background-color:green'>";
        echo $i;
        echo "</td>";
        echo "</tr>";
    }
    $i=$i+1;
}
echo "</table>";
?>
```

# Exemplo for – if para estilo



```
<?php
echo "<table width='100' align='center'>";
for($i=0; $i<=5; $i=$i+1)
{
    if($i % 2 == 0)
    {
        echo "<tr>";
        echo "<td style='background-color:red'>";
        echo $i;
        echo "</td>";
        echo "</tr>";
    }
    else
    {
        echo "<tr>";
        echo "<td style='background-color:green'>";
        echo $i;
        echo "</td>";
        echo "</tr>";
    }
}
echo "</table>";
?>
```

# Operadores Aritméticos e Lógicos



# operadores aritméticos e operadores lógicos



- Aritméticos
- Atribuição
- Comparação
- Lógicos

# Operadores aritméticos



Operador	Descrição	Exemplo	Resultado
+	Adição	$x=2$ $x+2$	4
-	Subtração	$x=2$ $5-x$	3
*	Multiplicação	$x=4$ $x*5$	20
/	Divisão	$15/5$	3
%	Resto de divisão	$7\%2$	1
++	Incrementa	$x=5$ $x++$	6
--	Decrementa	$x=5$ $x--$	4

# Operadores aritméticos



Operador	Exemplo	Equivalente a
=	$x=y$	atribuição
+=	$x+=y$	$x=x+y$
-=	$x-=y$	$x=x-y$
*=	$x*=y$	$x=x*y$
/=	$x/=y$	$x=x/y$
.=	$x.=y$	$x=x.y$ (concatena)
%=	$x%=y$	$x=x\%y$

# Operadores comparação



Operador	Descrição	Exemplo
==	Igual	5==8 false
!=	Diferente	5!=8 true
<>	Diferente	5<> true
>	Maior que	5>0 true
<	Menor que	5<8 true
>=	Maior ou igual	5>=8 true
<=	Menor ou igual	5<=8 true

# Operadores lógicos



Operador	Descrição	Exemplo
&&	and	<code>\$x=6</code> <code>\$y=3</code> <code>(\$x&lt;10 &amp;&amp; \$y&gt;1)</code> true
	or	<code>\$x=6</code> <code>\$y=3</code> <code>(\$x==5 &amp;&amp; \$y==51)</code> false
!	not	<code>\$x=6</code> <code>\$y=3</code> <code>!(\$x==\$y)</code> true

# Manipulação de Strings



# Manipulação de strings

- Há uma infinidade de funções para manipular strings
- Antes de criar funções, procurar se já não existe
- [http://www.w3schools.com/php/php\\_ref\\_string.asp](http://www.w3schools.com/php/php_ref_string.asp)
- [https://secure.php.net/manual/pt\\_BR/index.php](https://secure.php.net/manual/pt_BR/index.php)

# Manipulação de strings

- Função `strlen()` – retorna o tamanho do string

```
<?php
    echo strlen("Hello World");
?>
```

- Função `strpos()` – retorna a posição do string ou FALSE

```
<?php
    echo strpos("Hello World", "World");
?>
```

# Tratamento de Arrays



# Arrays numéricos



```
<?php
$contato_nome[0]="Maria";
$contato_nome[1]="Joaquim";
$contato_nome[2]="Antonio";

// $contato_nome=array("Maria","Joaquim","Antonio");

echo "Primeiro nome de contato: ".$contato_nome[0];
echo "<br/>";
echo "Segundo nome de contato: ".$contato_nome[1];
echo "<br/>";
echo "Terceiro nome de contato: ".$contato_nome[2];

?>
```

# Arrays associativos



```
<?php
/*
$contato_email["Maria"]="maria@usp.br";
$contato_email["Joaquim"]="joaquim@gmail.com";
$contato_email["Antonio"]="antonio@hotmail.com";
*/

$contato_email=array
(
"Maria"=>"maria@usp.br",
"Joaquim"=>"joaquim@gmail.com",
"Antonio"=>"antonio@hotmail.com"
);

echo "E-mail de Maria: ".$contato_email["Maria"];
echo "<br/>";
echo "E-mail de Joaquim: ".$contato_email["Joaquim"];
echo "<br/>";
echo "E-mail de Antonio: ".$contato_email["Antonio"];

?>
```

# Funções



# Definição de function



```
<?php
function soma($p1,$p2)
{
    $total=$p1+$p2;
    return $total;
}

echo "25 + 16 = ".soma(25,16);

?>
```

# Exemplo function 1



```
<?php
$a=1; //escopo global
function teste()
{
    $a=5; //escopo local
    echo "Valor da variável a dentro da função : ".$a; // dentro da função
}
echo "Valor da variável a fora da função : ".$a; // fora da função
?>
```

# Exemplo function 2



```
<?php
$day=date(l);
/* date, parametro l: retorna dia da semana
   Sunday, Monday, Tuesday, Wednesday,
   Thursday, Friday, Saturday
*/
// echo $hora;

if ($day == "Saturday")
    echo "Dia de Festa!";
else
    {
    echo "Dia de descansar ou estudar.";
    echo "<br/> lembrar das tarefas!";
    }
?>
```

```
<?php
$day=date(l);
/* date, parametro l: retorna dia da semana
   Sunday, Monday, Tuesday, Wednesday,
   Thursday, Friday, Saturday
*/
// echo $hora;

if ($day == "Saturday")
    echo "Dia de Festa!";
elseif ($day == "Sunday")
    echo "Dia de descansar.";
else
    echo "Dia de estudar.";
?>
```

# Exemplo function 3



```
<?php
$day=date(l);
/* date, parametro l: retorna dia da semana
   Sunday, Monday, Tuesday, Wednesday,
   Thursday, Friday, Saturday
*/

switch($day)
{
  case "Sunday":
    echo "Domingo";
    break;
  case "Saturday":
    echo "Sábado";
    break;
  default:
    echo "Dia de aula!";
    break;
}

?>
```

# Tratamento de Arquivos



- Na parte teórica está em andamento o estudo da análise orientada a objetos.
- Nesta teoria há dois tipos de objetos:
  - Objetos persistentes
  - Objetos não persistentes ou voláteis

# Objetos não persistentes ou voláteis



- Os objetos voláteis são aqueles que são criados e existem em tempo de execução.
- Em outras palavras, são objetos que são criados durante a execução do programa e ficam armazenados na memória do computador
- Quando o programa se encerra, esses objetos deixam de existir.
- São as variáveis criadas dentro do programa, contadores e outros objetos que não há interesse em guardá-los quando o programa se encerra.

# Objetos persistentes



- Os objetos persistentes são aqueles que, por alguma razão, se deseja preservá-los quando o programa termina sua execução.
- Para tanto, é necessário criar alguma forma de guardar esses objetos.
- Há, para isso, três formas de guardar esses objetos:
  - No próprio programa fonte
  - Bancos de dados
  - Arquivos

# Objetos persistentes

## – no código



- Isso **NUNCA** deve ser feito!
- As boas práticas de programação recomendam que todos os objetos – informações dependentes da aplicação em si- devem ser armazenadas em local separado da programação.
- O exemplo visto atrás de Arrays é um caso que não deve ser utilizado em programas pois, para alterar o email da Maria, seria necessário um programador PHP!

# Objetos persistentes – em banco de dados



- Os bancos de dados são sistemas gerenciadores preparados para armazenamento de um grande número de informações estruturadas.
- Os bancos de dados são denominados SGBD-sistema gerenciador de banco de dados
- Os bancos de dados são, na verdade, um complexo de arquivos de dados e linguagem de programação capaz de manipular esses dados
- Na disciplina será estudado o banco de dados MySQL

# 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

# 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 1



- Vá até o htdocs, apague tudo que tem por lá
- Preparar um programa para:
  - 1) Entrada de dois strings
  - 2) Contar o número de caracteres do primeiro string
  - 3) Determinar se o segundo string está contido no primeiro. Caso positivo dizer a posição e caso negativo declarar que não está contido

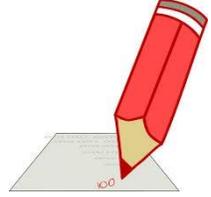


## Exercício 2



1. Perguntar quantos números serão entrados
2. Entrada desses valores numéricos
3. Imprimir na tela a soma e a média desses números
4. Mostrar esses números com uma casa depois da vírgula

*dica: procurar as funções de formatação no W3schools*

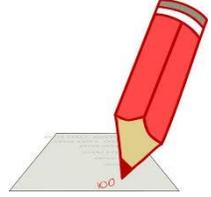


## Exercício 3



- Digitar uma data no formato dia/mês/ano
- Enviar uma mensagem dizendo:
- O dia (data) é uma (dia da semana) do mês de (nome por extenso) de (ano).

*dica: procurar as funções de data no PHP (PHP: date – Manual)*



# Entrega



- Fazer os exercícios em arquivos separados e entregá-los (funcionando) no *e-disciplinas* fazendo upload de um arquivo zipado contendo todos os três códigos produzidos

# PRO 2511 – L05

## Sistemas de Informação

### PHP



## Segunda aula de PHP

Ver. abril 2018

Rev. abril 2023