Resumo de PHP
Prof Vinícius Alves Hax
IFSul – Campus Jaguarão

Na linguagem PHP os nossos programas sempre começam com a tag de abertura <?php e encerram com a tag de fechamento ?>

Isso é necessário pois o PHP foi criado como uma linguagem para a web e portanto foi pensada desde o início para ser usado em conjunto com o HTML. As tags são necessárias para saber onde começa e onde termina o código.

Tudo que estiver antes e depois das tags será reproduzido no arquivo de saída. Embora seja possível configurar de maneira diferente na maioria das vezes para os servidores web interpretarem os programas PHP corretamente eles devem ser salvos com a extensão .php

Por exemplo o código abaixo

```
<html>
<body>
<p>
<?php
echo “Olá”;
</p>
</body>
</html>
```

Geraria o código HTML abaixo

```
<html>
<body>
<p>Olá</p>
</body>
</html>
```

Variáveis em PHP

Diferente de outras linguagens o PHP usa um símbolo antes do nome de cada variável. Esse símbolo é o $
Além disso não precisamos declarar o tipo de uma variável. Esse tipo é definido, durante a execução, pelo próprio interpretador.

Veja o exemplo abaixo

```
<?php
$numero_inteiro = 1;
$numero_com_virgula = 3.14;
$palavra = “Olá”;
?>
```

Comentários

Os comentários em PHP podem ser feitos começando uma linha com duas barras. Veja o código abaixo

```
<?php

// Este linha é um comentário
echo “Esta linha não é um comentário”;
?>
```

Concatenação

A concatenação é a operação de juntar elementos de string. Se um dos objetos a ser concatenado não for uma string ele será convertido em uma. Usamos o símbolo . (ponto) para concatenar strings

```
<?php
echo “Este ” . “trecho ” . “ é” . “ um” . “ conjunto ” . “de palavras concatenadas”;

// Concatenando strings fixas e variáveis
$mensagem = “numeros”;
echo “Podemos juntar strings e ” . $mensagem;

// Concatenando strings e variáveis numéricas
$numero = 15;
$mensagemfinal = $mensagem . $numero;

// Mostra “numeros15”
echo $mensagemfinal;

?>
```

Como a saída é interpretada pelo browser, se quisermos que apareça visualmente uma quebra de linha devemos usar tags html, como por exemplo br

```
<?php

echo “Linha1<br />Linha2”;

?>
```

Estrutura de seleção

O PHP tem uma estrutura de seleção semelhante à linguagem Java

```
<?php

$numero = 3;
if ($numero == 3)
	echo “Entrou no if”;
eles
	echo “Não entrou no if”;

?>
```

Estruturas de repetição

O PHP também tem as estruturas de repetição for e while. Veja os programas, usando for e while, que mostram os números de 1 até 10

```
<?php

for($i = 1; $i<11; $i++) {
	echo $i . “<br />”;
}

$contador = 1;
while($contador < 11)  {
	echo $contador . “<br />”;
	$i = $i + 1;
}

?>
```

Funções

Uma parte importante de qualquer linguagem são as funções. Elas permitem repetir código em PHP. Para usar funções em PHP usamos a palavra-chave function

```
<?php

function mostra_br() {
	echo “<br />”;

}

echo “Linha 1”;
mostra_br();
echo “Linha 2”;

?>
```

As funções também podem levar parâmetros que mudam o comportamento das mesmas. Por exemplo veja a função abaixo que mostra todos os números no intervalo indicado por início e fim

```
<?php

function mostra_numeros($inicio, $fim) {

	for($i=$inicio; $i<=$fim; $i++)
		echo “<br />” . $i;

}

?>
```

As funções também podem retornar resultados. A função abaixo, por exemplo, soma três números e retorna o resultado.

```
<?php

function soma_tres($primeiro, $segundo, $terceiro) {
	$resultado = $primeiro + $segundo + $terceiro;
	return $resultado;
}

// $numero passará a armazenar o valor 60
$numero = soma_tres(10, 20, 30);

?>
```

Arrays em PHP

Os arrays são interessantes para armazenar conjuntos de dados. Podemos declarar um array conforme abaixo

```
<?php

$arr = array(1, 2, 3, 6.89, “palavra”);

?>
```

Note que em PHP os arrays podem ter múltiplos tipos de dados ao mesmo tempo. O array acima tem números inteiros, números com vírgula e também uma string.

Podemos mostrar o valor de um array usando a função print_r

```
<?php

$arr = array(1, 2, 3, 6.89, “palavra”);

print_r($arr);

?>
```

Uma outra função útil relacionada com arrays é a função count que conta quantos elementos um array possui.

```
<?php

$arr = array(1, 2, 3, 6.89, “palavra”);

// $quantidade vai passar a valer 5 pois o array tem 5 elementos
$quantidade = count($arr);

?>
```

Podemos acessar um elemento específico de um array usando o seu índice entre colchetes

```
<?php

$arr = array(1, 2, 3, 6.89, “palavra”);


// Mostra 1
echo $arr[0];

// Mostra 3
echo $arr[2];

// Mostra “palavra”
echo $arr[4];


?>
```

Os índices (também chamados de chaves) de um array podem ser definidos depois da declaração do mesmo e não precisam ser números, e quando forem números eles não precisam estar ordenados

```
<?php

$arr = array(1, 2, 3, 6.89, “palavra”);

$arr[10] = “Novo elemento com índice 10”;

$arr[‘nome’] = “Aqui usamos uma chave não numérica para o array”;

// Irá mostrar
// Array ( [0] => 1 [1] => 2 [2] => 3 [3] => 6.89 [4] => “palavra” [10] => Novo elemento com
índice 10 [nome] => Aqui usamos uma chave não numérica para o array )
print_r($arr);

?>
```

Existe uma estrutura de repetição, na linguagem PHP, especial para percorrer todo os elementos de um array. Essa estrutura se chama foreach e pode ser usada da seguinte maneira:

```
<?php
$arr = array(1, 2, 3, 6.89, “palavra”);

foreach($arr as $valor)
	echo "<br />" . $valor;

?>
```

A saída será:

```
1
2
3
```

6.89
“palavra”

Podemos também declarar arrays sem usar índices numéricos. Uma maneira de fazer isso pode ser vista no exemplo abaixo.

```
<?php
$arr = array('fulano'=>10, 'beltrano' => 20, 'ciclano'=> 3);

foreach($arr as $valor)
	echo "<br />" . $valor;

?>
```

Se precisarmos usar o índice (também chamado de chave) de cada elemento existe uma modificação do foreach para isso. Veja o exemplo abaixo.

```
<?php
$arr = array('chave1'=>10, 'chave2' => 20, 'chave3'=> 3);

foreach($arr as $chave => $valor)
	echo "Chave: " . $chave . " , valor: ". $valor . "<br />" ;

?>
```

O resultado é:
Chave: chave1 , valor: 10
Chave: chave2 , valor: 20
Chave: chave3 , valor: 3

A linguagem PHP é muito usada e tem bem mais recursos do que foram mostrados acima. Para mais detalhes consulte, entre outros materiais:

https://www.php.net/manual/pt_BR/ - Documentação em português brasileiro

https://www.w3schools.com/php/ - Tutorial do ótimo site W3Schools

https://mundi.ifsul.edu.br/portal/linguagem-php.php – Curso em vídeo (com certificado) oferecido pelo IFSul