

Curso de PHP

PHP

FATEC - Jundiaí

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Aspas Invertidas

Ao utilizarmos as aspas invertidas como delimitadores, estaremos usando uma função muito interessante que o PHP oferece, que é executar comandos do sistema operacional através de um programa PHP.

Podemos enviar ao Linux, Windows ou qualquer outro sistema operacional, comandos para serem executados por ele.

```
<?php
    echo `ls -l *.inc` ; // Unix, Linux...
?>
```

```
<?php
    echo `dir c:\*.inc` ; // Windows...
?>
```

Obs.: lembre-se que poderá apenas executar se tiver direitos de acesso aos diretórios. O comando é executado no servidor e o resultado é mostrado no navegador.

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Constantes

O PHP possui algumas constantes pré-definidas, indicando a versão do PHP, o Sistema Operacional do servidor, o arquivo em execução, e diversas outras informações. Para ter acesso a todas as constantes pré-definidas, pode-se utilizar a função `phpinfo()`, que exibe uma tabela contendo todas as constantes pré-definidas, assim como configurações da máquina, servidor http e versão do PHP instalada.

```
<?php
    phpinfo();
?>
```

São valores que são predefinidos no início do programa, e que não mudam ao longo de sua execução. Para definir novas constantes devemos utilizar o comando `define`.

Define (`<constante>`,`<valor>`);

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Constantes

```
<?php
    define ("anoatual",2004);
    define ("time","São Paulo");
    echo "Estamos em " . anoatual . "<br>";
    echo "E o " . time . " será o Campeão Brasileiro";
?>
```

Obs.: verifique no exemplo acima que referenciamos as constantes diretamente pelo nome, e não utilizamos na frente delas o símbolo `$`, pois esse símbolo é utilizado para representar variáveis.

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores Aritméticos

Os operadores aritméticos são utilizados para realizar operações sobre os elementos numéricos.

```
$int1 = 100; $num1 = 12.30;
```

```
$result = $num1 + $int1; // resultado será 112.30
```

```
$result = $int1 - $num1; // resultado será 87.70
```

```
$result = ($int1 * $num1) / 8; // resultado será 153.75
```

+ adição, - subtração, * multiplicação, / divisão

++ incrementa, -- decrementa

% módulo (resto da divisão)

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores de Atribuição

Os operadores de atribuição são utilizados para atribuir ou modificar valores das variáveis.

```
$var1 = 10;
```

```
$var1 += 5; // seria o mesmo que $var1 = $var1 + 5;
```

```
$var1 -= 5; // seria o mesmo que $var1 = $var1 - 5;
```

```
$var1 *= 5; // seria o mesmo que $var1 = $var1 * 5;
```

```
$var1 /= 5; // seria o mesmo que $var1 = $var1 / 5;
```

```
$var1 %=5; // seria o mesmo que $var1 = $var1 % 5;
```

```
$texto = "PHP";
```

```
$texto .= " uma linguagem rápida e poderosa.";
```

```
/*
```

```
seria o mesmo que
```

```
$texto = $texto + " uma linguagem rápida e poderosa.";
```

```
*/
```

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores de Atribuição

Exemplo:

```
<?php
  $soma = 0;
  $var1 = 10;
  $var2 = 20;
  $var3 = 30;
  $soma += $var1; // $soma fica com 10
  $soma += $var2; // $soma fica com 10+20 = 30
  $soma *= $var3; // $soma fica com 30*30 = 900
  $soma %= 100; // $soma fica com 900%100 = 0
  echo $soma;
?>
```

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores Unários

Operador	Descrição
-oper	Troca de sinal do operando.
++oper	Pré-incremento. Primeiro incrementa o valor do operando e depois realiza a operação.
--oper	Pré-decremento. Primeiro decrementa o valor do operando e depois realiza a operação.
oper++	Pós-incremento. Primeiro realiza a operação e depois faz o incremento do valor.
oper--	Pós-decremento. Primeiro realiza a operação e depois faz o decremento do valor.

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Exemplo de pré-incremento e pós-incremento

```
<?php
$a = 1;
$b = 3;
$c = 5;
$res1 = ++$b - $a;
$res2 = $c-- + $a;
$res3 = --$a + $c++;
echo "a = $a<br>b = $b<br>c = $c<br><br>";
echo "res1 = $res1 - res2 = $res2 - res3 = $res3";
?>
```

Resposta:

```
a = 0
b = 4
c = 5
res1 = 3 - res2 = 6 - res3 = 4
```

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores Lógicos

Os operadores lógicos implementam a lógica booleana, retornando valor (0) falso ou valor (1) verdadeiro.

```
$cond1 OR $cond2; // verdadeiro se $cond1 ou $cond2 for verdadeiro
$cond1 || $cond2; // verdadeiro se $cond1 ou $cond2 for verdadeiro
$cond1 AND $cond2; // verdadeiro se $cond1 e $cond2 forem verdadeiros
$cond1 && $cond2; // verdadeiro se $cond1 e $cond2 forem verdadeiros
$cond1 XOR $cond2; // verdadeiro se só $cond1 ou só $cond2 for verdadeiro
!$cond1 // verdadeiro se $cond1 for falso
```

```
<?php
if ($cond1 OR $cond2) { echo "um dos dois - faça"; }
if ($cond1 AND $cond2) { echo "sendo os dois - faça"; }
if (!$cond1) { echo "caso não seja - faça"; }
?>
```

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores Lógicos

Operador AND (E)

EXP1	EXP2	Resultado
V	V	V
V	F	F
F	V	F
F	F	F

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores Lógicos

Operador OR (OU)

EXP1	EXP2	Resultado
V	V	V
V	F	V
F	V	V
F	F	F

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores Lógicos

Operador XOR (OU exclusivo)

EXP1	EXP2	Resultado
V	V	F
V	F	V
F	V	V
F	F	F

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores Lógicos

Operador ! (NOT)

EXP1	Resultado
V	F
F	V

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores Ternário

É uma forma abreviada de usar o comando condicional if. Uma condição é avaliada, e, se ela for verdadeira, atribui-se um valor à variável, e se a condição for falsa atribui-se um outro valor.

Sintaxe: condição ? expressão1 : expressão2

```
<?php
    $nota = ($frequencia >= 0.75) ? ($nota+2) : ($nota-2);
?>
```

O equivalente seria:

```
<?php
    if ($frequencia >= 0.75) {
        $nota = $nota + 2;
    }
    else {
        $nota = $nota - 2;
    }
?>
```

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Operadores de Comparação

Os operadores de comparação executam testes entre os conteúdos de uma variável.

```
$cond1 == $cond2    // igual a
$cond1 != $cond2    // diferente de
$cond1 <> $cond2    // diferente de
$cond1 > $cond2     // maior que
$cond1 >= $cond2    // maior ou igual a
$cond1 < $cond2     // menor que
$cond1 <= $cond2    // menor ou igual a
```

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Precedência de Operadores

Qual é a ordem que o PHP utiliza para tratar os operadores.

- ! ++ --	Negativo, não lógico, incremento e decremento
* / %	Multiplicação, divisão e resto da divisão
+ - .	Adição, subtração e concatenação
> < >= <=	Maior que, menor que, maior ou igual, menor ou igual
== != <>	Igual e diferente
&&	AND lógico
	OR lógico
?:	Operador ternário
= += -= *= /= %= .=	Operadores de atribuição
AND	AND lógico (de menor prioridade)
XOR	XOR lógico (de menor prioridade)
OR	OR lógico (de menor prioridade)

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Precedência de Operadores

Exemplo:

```
<?php
$num = 5;
$resultado = 8 + 3 * 2 + ++$num;
echo "$num<br>";
echo $resultado;
?>
```

O que será que irá ser apresentado na tela do terminal?

6
20

Obs.: maior prioridade será para o incremento, depois para a multiplicação e por último a adição -> 8 + 6 + 6

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Precedência de Operadores

Exemplo:

```
<?php
$num = 7;
$resultado = 8 * $num % 2;
echo $resultado;
?>
```

O que será que irá ser apresentado na tela do terminal?

0

Obs.: como vimos na tabela anterior os operadores de multiplicação(*) e resto da divisão tem a mesma precedência. Primeiro é realizado a multiplicação $8 * 7 = 56$ e depois calculado o resto da divisão de 56 por 2 = 0

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Precedência de Operadores

Exemplo:

```
<?php
$num = 7;
$resultado = 8 * ($num % 2);
echo $resultado;
?>
```

O que será que irá ser apresentado na tela do terminal?

8

Obs.: neste caso então será calculado primeiramente o valor do resto da divisão de 7 por 2, que é um e depois multiplicado por 8, resultando em 8

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Escopo das Variáveis

As variáveis existem somente no contexto em que são definidas. São variáveis com o mesmo nome sendo utilizadas em pontos diferentes dentro do programa, assumindo valores diferentes. Um exemplo disso seria:

```
<?php
    $num = 500;
    function novo_escopo()
    {
        $num +=5;
        echo $num . "dentro da função <br>";
    }
    echo $num . "fora da função <br>";
    novo_escopo();
?>
```

Obs.: o resultado seria, primeira linha com o número 500 e a segunda linha com o número 5.

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Escopo das Variáveis

Para que uma mesma variável seja alterada em qualquer parte do programa, devemos declará-la como global. O exemplo anterior ficaria assim:

```
<?php
    $num = 500;
    function novo_escopo()
    {
        global $num;
        $num +=5;
        echo $num . "dentro da função <br>";
    }
    echo $num . "fora da função <br>";
    novo_escopo();
?>
```

Obs.: o resultado seria, primeira linha com o número 500 e a segunda linha com o número 505.

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Escopo das Variáveis

Outra maneira de definir um escopo global é usando o array pré-definido \$GLOBALS:

```
<?php
$a = 1; // escopo global

Function Test () {
    echo $GLOBALS["a"];
    $GLOBALS["a"] = 4;
}

Test ();
echo "<br>$a"; // irá mostrar a variável global
?>
```

Obs.: o resultado seria, primeira linha com o número 1 e a segunda linha com o número 4.

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Interpolação de Variáveis

A interpolação consiste em escrever o valor de uma ou mais variáveis dentro da string que será mostrada na tela ou atribuída a outra variável.

```
<?php
$f = "futebol"
$t = "penta";
echo "No $f" . " o Brasil é $tcampeão";
?>
```

Obs.: cuidado aqui... Veja que o resultado será: No futebol o Brasil é

```
<?php
$f = "futebol"
$t = "penta";
echo "No $f" . " o Brasil é ${t}campeão"; // ou "... é" . $t . "campeão";
?>
```

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Variáveis criadas em tempo de execução

Podemos criar variáveis dinamicamente, ou seja, durante a execução do programa. Utilizamos o valor de uma variável como identificador para outra que será criada. Para isso usamos duas vezes o símbolo \$. Exemplo:

```
<?php
    $time = "Seleção brasileira de futebol";

    $identificador = "teste";

    $$identificador = $time;

    echo $teste . " está fora das olimpíadas! <br>";
?>
```

Prof. Cláudio Farias Rossoni

PHP – Aula 3

Termino da aula

Prof. Cláudio Farias Rossoni