

Curso de PHP

PHP

FATEC - Jundiaí

Prof. Cláudio Farias Rossoni

PHP – Aula 2

TIPOS DE VARIÁVEIS

- Inteiros
- Ponto Flutuante
- String
- Array
- Objetos
- Booleanos

PHP utiliza checagem de tipos dinâmica, ou seja, uma variável pode conter valores de diferentes tipos em diferentes momentos da execução do script. Por este motivo não é necessário declarar o tipo de uma variável para usá-la. O interpretador PHP decidirá qual o tipo daquela variável, verificando o conteúdo em tempo de execução.

Ainda assim, é permitido converter os valores de um tipo para outro desejado, utilizando o typecasting ou a função settype.

Prof. Cláudio Farias Rossoni

PHP – Aula 2

INTEIROS

Uma variável pode conter um valor inteiro com atribuições que sigam as seguintes sintaxes:

```
$idade = 1234; // inteiro positivo na base decimal  
$idade = -234; // inteiro negativo na base decimal  
$idade = 0234; // inteiro na base octal-simbolizado pelo 0  
                // equivale a 156 decimal  
$idade = 0x34; // inteiro na base hexadecimal (simbolizado  
                // pelo 0x) – equivale a 52 decimal.
```

A diferença entre inteiros simples e long está no número de bytes utilizados para armazenar a variável. Como a escolha é feita pelo interpretador PHP de maneira transparente para o usuário, podemos afirmar que os tipos são iguais.

Prof. Cláudio Farias Rossoni

PHP – Aula 2

PONTO FLUTUANTE

Uma variável pode ter um valor em ponto flutuante com atribuições que sigam as seguintes sintaxes:

```
$fração = 1.234;  
$fração = 23e4; // equivale a 230.000
```

Prof. Cláudio Farias Rossoni

PHP – Aula 2

STRING

Strings podem ser atribuídas de duas maneiras:

1-) utilizando aspas simples (') – Desta maneira, o valor da variável será exatamente o texto contido entre as aspas (com exceção de \\ e \')

2-) utilizando aspas duplas (") – Desta maneira, qualquer variável ou caractere de escape será expandido antes de ser atribuído.

Exemplo:

```
<?
$lugar = "Fatec";
$teste = 'Estamos na $lugar \n';
echo "$teste";
?>
```

A saída desse script será "Estamos na \$lugar \n".

Prof. Cláudio Farias Rossoni

PHP – Aula 2

STRING

```
<?
$lugar = "Fatec";
$teste = "Estamos na $lugar\n";
echo "$teste";
?>
```

A saída desse script será "Estamos na Fatec" (com uma quebra de linha no final).

Prof. Cláudio Farias Rossoni

PHP – Aula 2

CARACTERES DE ESCAPE

Importância especial deve ser dada ao caractere de escape ("\""). Quando o PHP encontra este símbolo em uma string, o caractere que vem a seguir é analisado, e dependendo dele, um tratamento especial será realizado.

A tabela seguinte lista:

•\n	Nova linha
•\r	Retorno de carro (semelhante a \n)
•\t	Tabulação horizontal
•\\	A própria barra (\)
•\\$	O símbolo \$
•\'	Aspa simples
•\"	Aspa dupla

Prof. Cláudio Farias Rossoni

PHP – Aula 2

CARACTERES DE ESCAPE

Por exemplo, se quisermos exibir uma string que tenha aspas:

O presidente falou: "Vamos trabalhar gente!"

Teria que usar o seguinte código:

```
<?php
    echo "O presidente falou: \"Vamos trabalhar gente!\"";
?>
```

ou também poderíamos usar esse código:

```
<?php
    echo 'O presidente falou: "Vamos trabalhar gente!" ';
?>
```

Prof. Cláudio Farias Rossoni

PHP – Aula 2

CONVERSÃO DE STRING

Quando uma string é avaliada em uma conversão para número, algumas regras são seguidas:

1- se a string tiver algum dos caracteres ".", "e" ou "E", ele será considerado do tipo ponto flutuante, senão será inteiro.

2- o valor é definido pela sua parte inicial. Se começar com dados numéricos válidos, essa parte é que será usada. Senão, o valor convertido da string será zero.

Vamos ver alguns exemplos de conversões de string para número:

```
$var = 1 + "11.7"; // O valor será ponto flutuante (12.7)
```

```
$var = 1 + "-1.3e3"; // O valor será ponto flutuante (-1299)
```

```
$var = 1 + "bagunça-1.3e3"; // O valor será inteiro (1)
```

```
$var = 1 + "12 cabras"; // O valor será inteiro (13)
```

Prof. Cláudio Farias Rossoni

PHP – Aula 2

ARRAY

Arrays em PHP podem ser observados como mapeamentos ou como vetores indexados. Mais precisamente, um valor do tipo array é um dicionário onde os índices são as chaves de acesso. Vale ressaltar que os índices podem ser valores de qualquer tipo e não somente inteiros. Inclusive, se os índices forem todos inteiros, estes não precisam formar um intervalo contínuo.

Como a checagem de tipos em PHP é dinâmica, valores de tipos diferentes podem ser usados como índices de array, assim como os valores mapeados também podem ser de diversos tipos.

Prof. Cláudio Farias Rossoni

PHP – Aula 2

ARRAY

Exemplo:

```
<?
$cor[1] = "vermelho";
$cor[2] = "verde";
$cor[3] = "azul";
$cor["teste"] = 1;
?>
```

Equivalentemente, pode-se escrever:

```
<?
$cor = array(1=>"vermelho", 2=>"verde", 3=>"azul", "teste"=>1);
?>
```

Prof. Cláudio Farias Rossoni

PHP – Aula 2

LISTAS

As listas são utilizadas em PHP para realizar atribuições múltiplas. Através de listas é possível atribuir valores que estão num array para variáveis.

Exemplo:

```
list($a, $b, $c) = array("a", "b", "c");
```

O comando acima atribui valores às três variáveis simultaneamente. É bom notar que só são atribuídos às variáveis da lista os elementos do array que possuem índices inteiros e não negativos. No exemplo acima as três atribuições foram bem sucedidas porque ao inicializar um array sem especificar os índices eles passam a ser inteiros, a partir do zero. Um fator importante é que cada variável da lista possui um índice inteiro e ordinal, iniciando com zero, que serve para determinar qual valor será atribuído.

Prof. Cláudio Farias Rossoni

PHP – Aula 2

LISTAS

No exemplo anterior temos \$a com índice 0, \$b com índice 1 e \$c com índice 2. Outro exemplo:

```
$arr = array(1=>"um",3=>"três","a"=>"letraA",2=>"dois");  
list($a,$b,$c,$d) = $arr;
```

Após a execução do código acima temos os seguintes valores:

```
$a == null  
$b == "um"  
$c == "dois"  
$d == "três"
```

Devemos observar que à variável \$a não foi atribuído valor, pois no array não existe elemento com índice 0 (zero). Outro detalhe importante é que o valor "três" foi atribuído à variável \$d, e não a \$b, pois seu índice é 3, o mesmo que \$d na lista. Por fim, vemos que o valor "letraA" não foi atribuído a elemento algum da lista pois seu índice não é inteiro.

Prof. Cláudio Farias Rossoni

PHP – Aula 2

LISTAS

Os índices da lista servem apenas como referência ao interpretador PHP para realizar as atribuições, não podendo ser acessados de maneira alguma pelo programador. De maneira diferente do array, uma lista não pode ser atribuída a uma variável, servindo apenas para fazer múltiplas atribuições através de um array.

Prof. Cláudio Farias Rossoni

PHP – Aula 2

OBJETOS

Um objeto pode ser inicializado utilizando o comando `class` para instanciar uma classe para uma variável.

Exemplo:

```
class teste {  
    function nada() {  
        echo "nada";  
    }  
}  
$teste = new teste;  
$teste -> nada();
```

A utilização de objetos será mais detalhada mais à frente.

Prof. Cláudio Farias Rossoni

PHP – Aula 2

BOOLEANOS

PHP possui um tipo booleano (desde a sua versão 4.0), sendo capaz de avaliar expressões e retornar *true* ou *false*, através do tipo integer: é usado o valor 0 (zero) para representar o estado *false*, e qualquer valor diferente de zero (geralmente 1) para representar o estado *true*.

```
<?php  
$a = True; // assimila o valor TRUE para $a  
if ($a)  
{ echo "Verdadeiro"; } // se valor <> 0  
else  
{ echo "Falso"; } // se valor = 0  
?>
```

Prof. Cláudio Farias Rossoni

PHP – Aula 2

BOOLEANOS

Outro exemplo de uso de booleanos com comandos mesclados PHP e HTML

```
<?php if ($expressão) { ?>
    <b>Isso é verdadeiro.</b> // comando em HTML
<?php } else { ?>
    <b>Isso é falso.</b> // comando em HTML
<?php } ?>
```

Prof. Cláudio Farias Rossoni

PHP – Aula 2

CONVERTENDO BOOLEANOS

Para converter explicitamente um valor para booleano, utilize-se dos modificadores (bool) ou (boolean). Entretanto, na maioria dos casos, você não precisa utilizar o modificador, desde que qualquer valor será convertido automaticamente se um operador, função ou estrutura de controle requerer um argumento booleano

Quando convertendo para booleano, os seguintes valores são considerados **FALSE**:

- o booleano **FALSE**
- o inteiro 0 (zero)
- o ponto flutuante 0.0 (zero)
- uma string vazia e a string "0"
- um array sem elementos
- um objeto sem elementos
- o tipo especial NULL (incluindo variáveis não definidas)

Prof. Cláudio Farias Rossoni

PHP – Aula 2

PHP

Termino da aula

Prof. Cláudio Farias Rossoni