

Programação para WEB

PHP Básico

Prof. Bruno Camargo Ribeiro

Tipos de Dados

Assuntos que serão abordados:

1	Tipos de dados suportados pelo PHP	2
2	Tipos de dados escalares	2
2.1	Boolean	2
2.2	Inteiro	2
2.3	Float	2
2.4	String	3
3	Tipos de dados compostos	3
3.1	Array	3
3.2	Objetos	3
4	Convertendo tipo de dados usando <i>Type Casting</i>	4
5	Adaptando tipos de dados com o Tipo Juggling	4
6	Variáveis e Declaração	5
6.1	Designação de Referência	5
7	Escopo da variável	6
8	Variáveis Supor Globais	7
9	Constantes	9
10	Expressões	9
10.1	Operandos	9
10.2	Operadores	10

1 Tipos de dados suportados pelo PHP

O PHP há muito tempo tem oferecido um conjunto bem amplo de dados, os tipos mais comuns incluem **boolean**, **integer**, **float**, **string** e **array**.

2 Tipos de dados escalares

São capazes de conter apenas um tipo de informação.

2.1 Boolean

Recebeu esse nome em homenagem ao matemático George Boole (1815-1864). Uma variável **boolean** suporta dois valores: **TRUE** e **FALSE**, como alternativa pode usar o número **zero** para representar o **FALSE** e qualquer **outro valor** para representar o **TRUE**.

```
$variavel = false    // é falso.  
$variavel = 1 // é verdadeiro.  
$variavel = -1 // é verdadeiro.  
$variavel = 7 // é verdadeiro.  
$variavel = 0 // é falso.  
$variavel = true    // é verdadeiro.
```

2.2 Inteiro

Representa qualquer número inteiro, na base decimal(10), octogonal(8) e hexadecimal(16).

```
65          // decimal  
101         // octogonal (65 decimal)  
41          // hexadecimal  
0xC4A      // hexadecimal
```

2.3 Float

Representa qualquer número com casa decimal.

```
3.5456  
5.0  
7.5e4  
1.2E+11
```

2.4 String

Representa uma sequência de caracteres tratados como um grupo (arrays), delimitados por aspas simples (') ou duplas (").

"Aula de PHP"

'testando o PHP'

'893720940125'

"8477*(&%&`#"

3 Tipos de dados compostos

Permitem múltiplos itens do mesmo tipo agregados em uma única entidade administrativa.

3.1 Array

É formalmente definida como uma coleção indexada de valores de dados.

```
$vetor[0] = "Aula";
```

```
$vetor[1] = "de";
```

```
$vetor[2] = "PHP";
```

A estrutura de **array** e sua manipulação, serão abordados posteriormente. Teremos um tópico exclusivo.

3.2 Objetos

Conceito central da programação orientada a objeto, o qual deve ser explicitamente declarado e instanciado na programação.

```
classe Aulaphp {    private
    $_variavel    function
    setValor($valor){
        $this->_variavel = $valor;
    }
}
...
$objeto = new Aulaphp;
```

4 Convertendo tipo de dados usando *Type Casting*

Ao indicarmos um operador de tipo antes da variável a ser impressa podemos modificar seu tipo inicial.

Operadores	Conversão
(array)	Array
(bool) ou (boolean)	Boolean
(int) ou (integer)	Integer
(int64)	Integer de 64 bits (PHP 6)
(object)	Object
(real) ou (double) ou (float)	Floar
(string)	String

Considere alguns exemplos:

\$valor = (double)13 // \$valor = 13.0

\$teste = (int) 17.53 // \$teste = 17

5 Adaptando tipos de dados com o Tipo Juggling

O PHP considera as variáveis automaticamente dependendo das circunstâncias nas quais estão referenciadas.

```

1 <?php
2     $valor = 10;           // inteiro
3     $expressao = "15";    // string
4     $total = $valor + $expressao;
5     echo $total;
6 ?>

```

Resposta: 25

```

1 <?php
2     $expressao = "10 laranjas"; // string
3     $valor = 3;                 // inteiro
4     $total = $valor + $expressao;
5     echo $total;
6 ?>

```

Resposta: 13

```
1 <?php
2     $valor = "1.0";
3     if ($valor) echo "Acho que entendeu verdadeiro";
4     ?>
```

Resposta: "Acho que entendeu verdadeiro"

6 Variáveis e Declaração

Uma variável começa com o símbolo de dólar (\$), que é então seguido pelo seu nome, o qual deverá começar com uma letra ou um *underscore* e pode ser constituída por letras, números e *underscore*. São validas:

```
1 <?php
2     // declarações corretas
3     $cor = "azul";
4     $cor_de_fundo = "amarelo";
5     $_cor = "vermelho";
6
7     // as variáveis abaixo não possuem nenhum relação uma com a outra
8     $fundo = "azul";
9     $Fundo = "amarelo";
10    $FUNDO = "vermelho";
11    echo $fundo."-".$Fundo."-".$FUNDO;
12    ?>
```

6.1 Designação de Referência

Introduzido na versão 4 do PHP, permite que uma variável criada por referência tenha essencialmente o conteúdo de outra e vice-versa.

```
1 <?php
2     $texto1 = "Aula";
3     $texto2 =& $texto1;
4     echo "texto01 => ".$texto1;
5     echo "<br>";
6     $texto2 = "PHP";
7     echo "texto01 => ".$texto1;
8     ?>
```

7 Escopo da variável

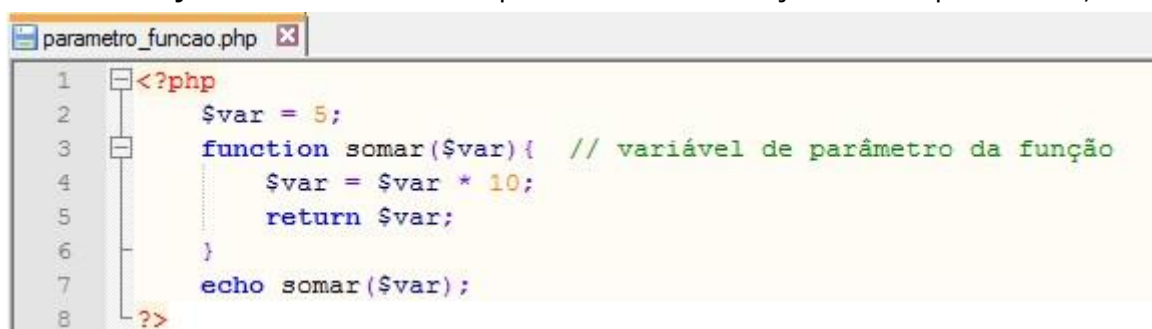
Podemos declarar as variáveis em qualquer lugar de um script PHP, mas a sua localização influencia no domínio em que pode ser acessado, esse domínio é conhecido como escopo.

Variáveis Locais – são aquelas declaradas dentro do *script* ou das funções;



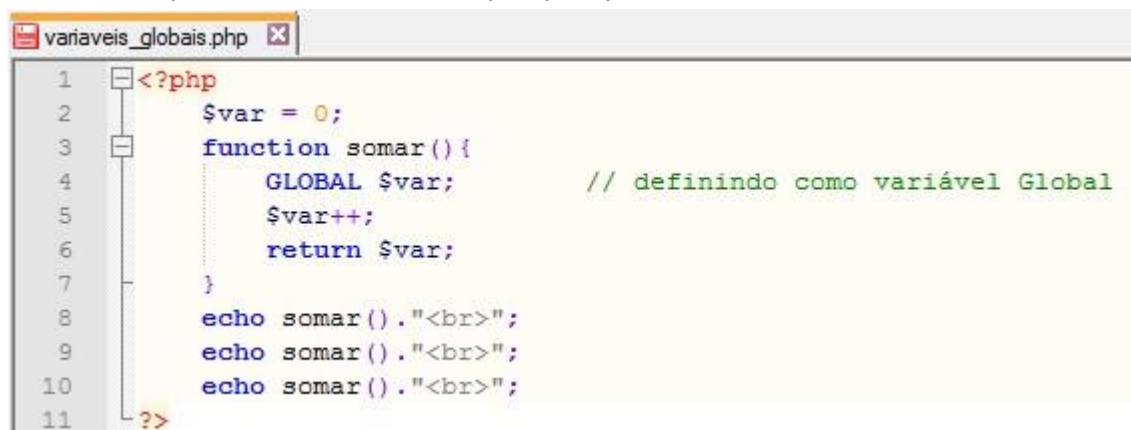
```
1 <?php
2     $var = 10;                // variável local no script
3     function somar(){
4         $var = 5;            // variável local para a função
5         echo "valor de dentro: ".$var;
6     }
7     somar();
8     echo "<br>";
9     echo "valor de fora: ".$var
10  ?>
```

Parâmetros de Função – são declaradas depois no nome da função e entre parênteses;



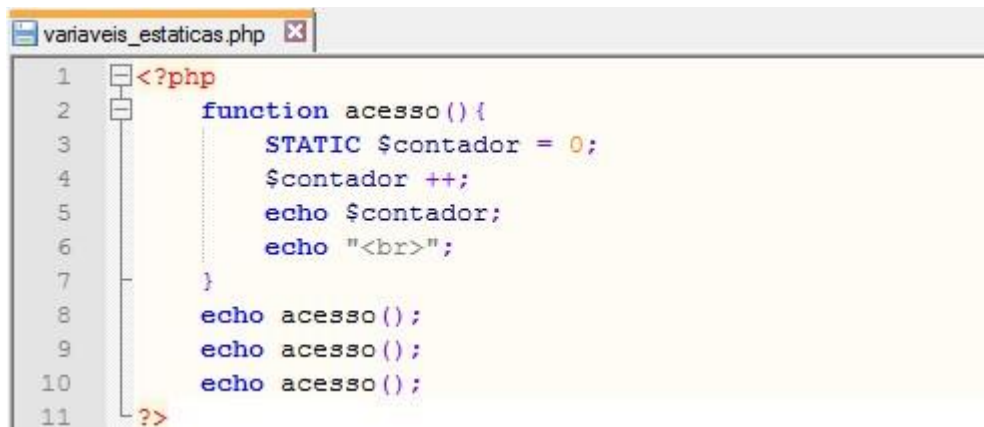
```
1 <?php
2     $var = 5;
3     function somar($var){    // variável de parâmetro da função
4         $var = $var * 10;
5         return $var;
6     }
7     echo somar($var);
8     ?>
```

Variáveis Globais – pode ser acessada em qualquer parte do sistema;



```
1 <?php
2     $var = 0;
3     function somar(){
4         GLOBAL $var;        // definindo como variável Global
5         $var++;
6         return $var;
7     }
8     echo somar(). "<br>";
9     echo somar(). "<br>";
10    echo somar(). "<br>";
11    ?>
```

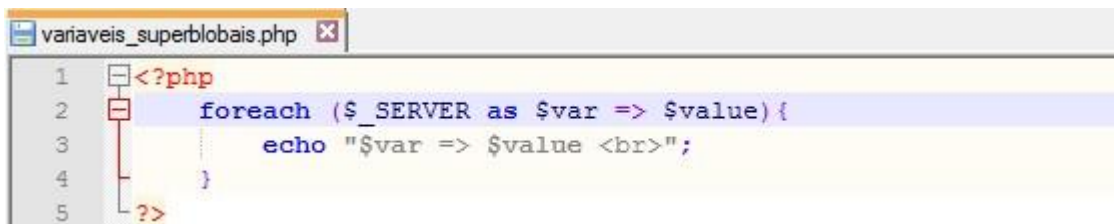
Variáveis Estáticas – supondo uma variável declarada dentro de uma função, quando esta função é fechada a variável não perde o seu valor;



```
1 <?php
2 function acesso(){
3     STATIC $contador = 0;
4     $contador ++;
5     echo $contador;
6     echo "<br>";
7 }
8 echo acesso();
9 echo acesso();
10 echo acesso();
11 ?>
```

8 Variáveis Supor Globais

O PHP oferece um número de variáveis pré-definidas que são acessíveis a partir de qualquer lugar dentro do script. Execute o script:



```
1 <?php
2 foreach ($_SERVER as $var => $value){
3     echo "$var => $value <br>";
4 }
5 ?>
```

Terá como retorno um uma lista de variáveis, algumas destas serão tratadas posteriormente.

MIBDIRS => C:/xampp/php/extras/mibs

MYSQL_HOME => \xampp\mysql\bin

OPENSSL_CONF => C:/xampp/apache/bin/openssl.cnf

PHP_PEAR_SYSCONF_DIR => \xampp\php

PHPRC => \xampp\php

TMP => \xampp\tmp

HTTP_HOST => localhost

HTTP_USER_AGENT => Mozilla/5.0 (Windows NT 6.1; WOW64; rv:41.0)

Gecko/20100101 Firefox/41.0

HTTP_ACCEPT => text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

HTTP_ACCEPT_LANGUAGE => pt-BR,pt;q=0.8,en-US;q=0.5,en;q=0.3

HTTP_ACCEPT_ENCODING => gzip, deflate

HTTP_REFERER => http://localhost/CPS%20-%20PHP/aula01/

HTTP_CONNECTION => keep-alive

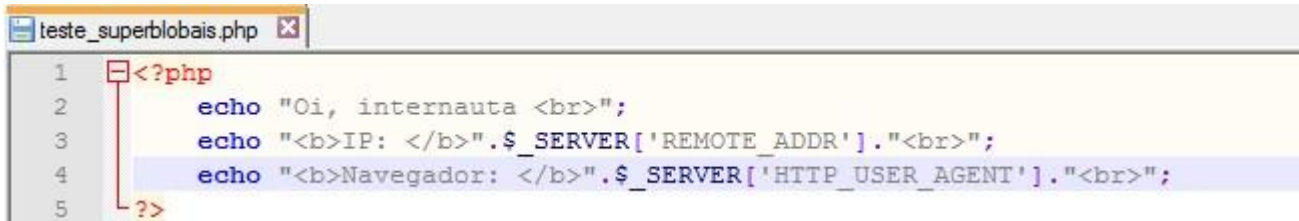
HTTP_CACHE_CONTROL => max-age=0

Etec "Sales Gomes" – 101 – Tatuí

```
PATH => C:\Program Files\Common Files\Microsoft Shared\Windows Live;C:\Program
Files (x86)\Common Files\Microsoft Shared\Windows Live;C:\Program Files
(x86)\Java\jdk1.6.0_19\bin;C:\Program Files (x86)\Windows Live\Shared;;C:\Program
Files (x86)\Microsoft SQL Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\100\Tools\Binn\;C:\Program Files\Microsoft SQL
Server\100\DTS\Binn\;C:\Program Files\MATLAB\R2009a\bin;C:\Program
Files\MATLAB\R2009a\bin\win64;C:\Windows\system32\config\systemprofile\.dnx\bin;
C:\Program Files\Microsoft DNX\Dnvm\
SystemRoot => C:\Windows
COMSPEC => C:\Windows\system32\cmd.exe
PATHEXT => .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
WINDIR => C:\Windows
SERVER_SIGNATURE =>
Apache/2.4.3 (Win32) OpenSSL/1.0.1c PHP/5.4.7 Server at localhost Port 80
SERVER_SOFTWARE => Apache/2.4.3 (Win32) OpenSSL/1.0.1c PHP/5.4.7
SERVER_NAME => localhost
SERVER_ADDR => 127.0.0.1
SERVER_PORT => 80
REMOTE_ADDR => 127.0.0.1
DOCUMENT_ROOT => C:/xampp/htdocs
REQUEST_SCHEME => http
CONTEXT_PREFIX =>
CONTEXT_DOCUMENT_ROOT => C:/xampp/htdocs
SERVER_ADMIN => postmaster@localhost
SCRIPT_FILENAME => C:/xampp/htdocs/CPS - PHP/aula01/variaveis_superblobais.php
REMOTE_PORT => 50549
GATEWAY_INTERFACE => CGI/1.1
SERVER_PROTOCOL => HTTP/1.1
REQUEST_METHOD => GET
QUERY_STRING =>
REQUEST_URI => /CPS%20-%20PHP/aula01/variaveis_superblobais.php
SCRIPT_NAME => /CPS - PHP/aula01/variaveis_superblobais.php
PHP_SELF => /CPS - PHP/aula01/variaveis_superblobais.php
REQUEST_TIME_FLOAT => 1444606773.755
```


REQUEST_TIME => 1444606773

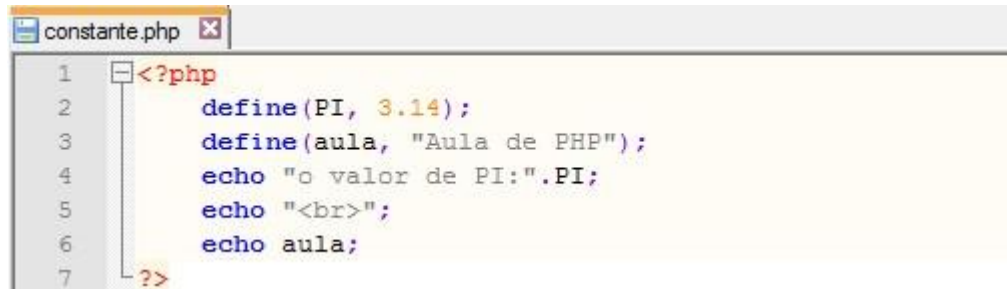
Veja algumas utilidades:



```
1 <?php
2     echo "Oi, internauta <br>";
3     echo "<b>IP: </b>".$_SERVER['REMOTE_ADDR']."<br>";
4     echo "<b>Navegador: </b>".$_SERVER['HTTP_USER_AGENT']."<br>";
5 ?>
```

9 Constantes

A função define() registra uma constante a partir de um nome.



```
1 <?php
2     define(PI, 3.14);
3     define(aula, "Aula de PHP");
4     echo "o valor de PI:".PI;
5     echo "<br>";
6     echo aula;
7 ?>
```

10 Expressões

Uma expressão é uma frase representando uma ação em particular, constituída de um operando e um ou mais operadores.

10.1 Operandos

São as entradas de uma expressão.

`$a++` // `$a` é um operando.

`$soma = $x e $y` // `$soma`, `$x` e `$y` são operandos.

10.2 Operadores

É um símbolo que especifica uma ação em particular em uma expressão, e sua precedência e associação devem ser respeitadas.

Associação	Operador	Objetivo
NA	new	instanciação do objeto
NA	()	subgrupos de expressão
Direita	[]	índice de posição
Direita	++ --	incremento e decremento
Direita	@	supressão de erro
Esquerda	/ * %'	divisão, multiplicação e módulo
Esquerda	+ - .	adição, subtração e concatenação
NA	< <= > >=	menor, menor igual, maior, maior igual
NA	== != === <>	igual, não é igual, idêntico, diferente
Direita	= += -= *= /= %=	atribuição de operadores
Esquerda	&&	AND e OR (booleano)
Esquerda	AND XOR OR	AND, XOR e OR (booleano)
Esquerda	, (virgule)	separação de expressões

Fonte: http://php.net/manual/pt_BR/language.operators.precedence.php