ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

PHP Básico

Profº Ricardo Aragão - raragao@raragao.eng.br
Zend Certified Engineer PHP 5

Sintaxe - PHP Tags

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

Standard Tags	<?php ... code ?>
Short Tags	<? ... code ?> <?= $variable ?>
Script Tags	<script language="php"> ... code </script>
ASP Tags	<% ... code %>

Sintaxe – PHP Tags

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Facilitar a inclusão de código PHP em simples arquivos texto
- No PHP 5 deve ser utilizado <?php para evitar conflitos com XML que usa <?, que era o padrão anterior do PHP.
- Configurável no php.ini (PHP_INI_PERDIR)
- Short tags, script tags e ASP tags são todas consideradas depreciadas e seu uso é desencorajado
- <?= só funciona com short tags habilitada

Sintaxe – Diretivas de configuração

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Locais

Constante	Valor	Significado
PHP_INI_USER	1	Valor pode ser modificado nos scripts do usuário ou no registro do Windows
PHP_INI_PERDIR	2	Valor pode ser modificado nos arquivos `php.ini`, `.htaccess` ou `httpd.conf`
PHP_INI_SYSTEM	4	Valor pode ser modificado no arquivo `php.ini` ou `httpd.conf`
PHP_INI_ALL	7	Valor pode ser modificado em qualquer lugar

Anatomia do PHP

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Todo script PHP é constituído de expressões, funções, variáveis, diretivas, etc.
- Cada instrução deve terminar com um ponto-e-vírgula (;) - exatamente como em C e Perl -, salvo em raras exceções

```
<?php
$var = “Olá Mundo”;
print $var
?>
```

Anatomia do PHP - Comentários

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

```
// Comentário de linha simples
# Comentário de linha simples
/*
Comentário de múltiplas
Linhas
*/
/**
 * Comentário de múltiplas linhas especial
 * @author Ricardo
 */
```

Anatomia do PHP – Espaço em branco

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Espaços em branco podem ser utilizados, ou não, em qualquer parte do código fonte, salvo em poucas exceções:
  - Não pode ter espaço entre <? e php
  - Não pode quebrar palavras reservadas:
    - `whi le`
    - `fo r`
    - `function`
  - Não pode quebrar nomes de variáveis, funções e classes utilizando espaço

Anatomia do PHP – Bloco de código

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- É uma série de instruções incluídas entre chaves ({ })
- Convenientemente utilizado para criar grupos de linhas de script que serão executadas em circunstâncias específicas: funções ou condicionais

```
{
//bloco de código
funcao(); //chamada de função
}
```

Anatomia do PHP – Construtores da linguagem

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- São elementos que são parte da linguagem
- Exemplos comuns:
  - echo
  - print()
  - die()

Tipos de dados

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Tipos ESCALARES:
  - boolean
  - int
  - float
  - string
- Tipos COMPOSTOS:
  - array
  - object

Tipos de dados - Numéricos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- int (Inteiros)

Decimal	10; 250; -102; 10023	Notação padrão de decimais
Octal	042; 0234;0731	Notação de octal é precedida de 0
Hexadecimal	0x2A; 0X834; 0xFF28A	Notação na base 16 é precedida de 0x ou 0X

O tamanho de um inteiro é dependente de plataforma, sendo um numero aproximado a 2 bilhões o valor mais comum (número de 32 bits com sinal). O PHP não suporta inteiros sem sinal.

Tipos de dados - Numéricos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Float (ponto flutuante)

Decimal	10.1; 1231.3; -1.02; 0.023; .923	Notação tradicional de decimais
Exponencial	2E7, 1.2e2	Notação exponencial

Tipos de dados - Numéricos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Cuidados com tipos de dados numéricos
  - Overflow: caso ocorra um *overflow* de inteiro, haverá, automaticamente, uma conversão para *float*
  - *PHP não faz alertas de overflow*
  - *Problemas de precisão:*
    - *$a = (int) ((0.1 + 0.7) * 10), qual resultado ?*
- *Conheça as limitações de seus tipos numéricos.*
- *Utilize a extensão BCMath quando precisar de precisão arbitrária*

Tipos de dados – String

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- String em PHP não são apenas seqüências de texto, são coleções ordenadas de dados binários que podem ser textos, arquivos de imagens, planilhas, músicas, etc.

Tipos de dados - boolean

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- TRUE ou FALSE
- Regras de conversão:
  - Um número convertido para *boolean* será *false* se for 0 ou *true* nos outros casos
  - Um *string* será convertida para *false* se ela for vazia ou contiver um simples caracter 0, em outros casos será convertida para *true*
  - Quando um *boolean* for convertido para um número ou string, será 1 se verdadeiro ou 0 se falso

Tipos de dados - Compostos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Arrays
  - São conjuntos de elementos de dados ordenados
  - Podem ser usados para armazenar ou retornar qualquer outro tipo de dados
- Objects
  - São conjuntos de dados e código
  - São a base da programação orientada a objetos

Tipos de dados – Outros Tipos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Tipos utilizados em situações especiais
  - NULL
    - A variável é considerada NULL se
      - ela foi assimilada com a constante NULL.
      - ela ainda não recebeu nenhum valor ainda.
      - ela foi apagada com unset().
  - RESOURCE
    - É uma variável especial, mantendo uma referência de recurso externo. Recursos são criados e utilizados por funções especiais.
    - Os tipos resource sustentam manipuladores especiais para arquivos abertos, conexões de bancos de dados, pinceis de desenho e coisas assim, você não pode converter nenhum valor para o tipo resource.

Tipos de dados - Conversão

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- float para inteiros:
  - $a = (int) 10.88;
  - $a = (integer) 10.88;
  - $a = intval(10.88);
  - $a = intval(“10.88”, 10);
  - $a = settype(10.88, “int”);

Tipos de dados - Conversão

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Quando convertendo para booleano, os seguintes valores são considerados FALSE:
  - o próprio 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 membros
  - o tipo especial NULL (incluindo variáveis não definidas)
- Qualquer outro valor é considerado TRUE (incluindo qualquer recurso).

Tipos de dados - Conversão

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- String para integer e float

```
$foo = 1 + "10.5";                // $foo é float (11.5)
$foo = 1 + "-1.3e3";              // $foo é float (-1299)
$foo = 1 + "bob-1.3e3";           // $foo é integer (1)
$foo = 1 + "bob3";                // $foo é integer (1)
$foo = 1 + "10 Small Pigs";       // $foo é integer (11)
$foo = 4 + "10.2 Little Piggies"; // $foo é float (14.2)
$foo = "10.0 pigs " + 1;          // $foo é float (11)
$foo = "10.0 pigs " + 1.0;        // $foo é float (11)
```

Tipos de dados - Conversão

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Para array

```php
$a = (array) 10;


class Pessoa {
    public $nome = "Fulano";
    public $peso = 70;
}
$c = (array) new Pessoa;
```

Tipos de dados - Casting

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- As moldagens(casting) permitidas são:
  - (int), (integer) - molde para inteiro
  - (bool), (boolean) - molde para booleano
  - (float), (double), (real) - molde para número de ponto flutuante
  - (string) - molde para string
  - (array) - molde para array
  - (object) - molde para objeto

Variáveis

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- São armazenamentos temporários.
- Regra de nomeclatura por ER
  - '[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*'

```
$var = "Bob";
$Var = "Joe";

$4site = 'not yet';      // inválido; começa com um número
$_4site = 'not yet';     // válido; começa com um sublinhado
$täyte = 'mansikka';     // válido; 'ä' é um caracter ASCII
   (extendido) 228
```

Variáveis pré-definidas

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- $GLOBALS
  - Contém um referência para todas as variáveis que são atualmente disponíveis dentro do escopo global do script. As chaves desse array são os nomes das variáveis globais. $GLOBALS existe desde o PHP 3.
- $_SERVER
  - Variáveis criadas pelo servidor web ou diretamente relacionadas ao ambiente de execução do script atual. Análogo ao antigo array $HTTP_SERVER_VARS (que ainda continua disponível, mas em decadência).
- $_GET
  - Variáveis postadas para o script via método HTTP GET. Análogo ao antigo array $HTTP_GET_VARS (que ainda continua disponível, mas em decadência).

Variáveis pré-definidas

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- $_POST
  - Variáveis postadas para o script via método HTTP POST. Análogo ao antigo array $HTTP_POST_VARS (que ainda continua disponível, mas em decadência).
- $_COOKIE
  - Variáveis postadas para o script via cookies HTTP. Análogo ao antigo array $HTTP_COOKIE_VARS (que ainda continua disponível, mas em decadência).
- $_FILES
  - Variáveis postadas para o script via transferência de arquivos HTTP. Análogo ao antigo array $HTTP_POST_FILES (que ainda continua disponível, mas em decadência). Veja uploads via método POST para maiores informações.

Variáveis pré-definidas

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- $_ENV
  - Variáveis disponíveis no script do ambiente de execução. Análogo ao antigo array $HTTP_ENV_VARS (que ainda continua disponível, mas em decadência).
- $_REQUEST
  - Variáveis postadas para o script por todas os mecanismos de input GET, POST, e COOKIE não podem ter seu conteúdo garantido de qualquer forma. A presença e a ordem de inclusão das variáveis nesse array é definida de acordo com a diretiva de configuração variables_order. Este array não tem um equivalente nas versões anteriores do PHP 4.1.0. Veja também import_request_variables().
- $_SESSION
  - Variáveis que estão atualmente registradas na sessão do script. Análogo ao antigo array $HTTP_SESSION_VARS (que ainda continua disponível, mas em decadência). Veja a sessão funções de manipulação de Sessões para maiores informações.

Variáveis - Escopo

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- O escopo de uma variável é o contexto onde ela foi definida.
- A maior parte das variáveis do PHP tem somente escopo local. Este escopo local inclui os arquivos incluídos.
- Com as funções definidas pelo usuário, um escopo local é introduzido. Quaisquer variáveis utilizadas dento da função é por default limitada dentro do escopo local da função.

Variáveis variáveis

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

```php
$a = "nome";
$nome = "Fulano";
echo $$a;

function teste() {
    print "Teste";
}

$funcao = "teste";
$funcao();
```

Variáveis variáveis

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- O código abaixo funciona ?

```
$a = '123';
$$a = '456';

print ${'123'};
```

Constantes

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- São valores imutáveis definidos pelo programador

```
define('EMAIL', 'davey@php.net');
echo EMAIL;
define('USE_XML', true);
if (USE_XML) { }
define('1CONSTANT', 'some value'); //nome inválido
```

Operadores

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Operadores Aritméticos
- Operadores de Atribuição
- Operador Bit-a-bit
- Operadores de Comparação
- Operadores de controle de erro
- Operadores de Execução
- Operadores de Incremento/Decremento
- Operadores Lógicos
- Operadores de String
- Operadores de Arrays

Operadores - precedência

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

Associação	Operador
não associativo	new
direita	[
direita	! ~ ++ -- (int) (float) (string) (array) (object) @
esquerda	* / %
esquerda	+ - .
esquerda	<< >>
não associativo	< <= > >=
não associativo	== != === !==
esquerda	&
esquerda	^
esquerda	|
esquerda	&&
esquerda	||
esquerda	? :
direita	= += -= *= /= .= %= &= |= ^= <<= >>=
direita	print
esquerda	and
esquerda	xor
esquerda	or
esquerda	,

Operadores aritméticos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

Exemplo	Nome	Resultado
$a + $b	Adição	Soma de $a e $b.
$a - $b	Subtração	Diferença entre $a e $b.
$a * $b	Multiplicação	Produto de $a e $b.
$a / $b	Divisão	Quociente de $a por $b.
$a % $b	Módulo	Resto de $a dividido por $b.

Operadores de atribuição

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

```
$a = ($b = 4) + 5; // $a é igual a 9 agora e $b foi configurado
   como 4.
```

```
$a = 3;

$a += 5; // configura $a para 8, como se disséssemos: $a = $a + 5;

$b = "Bom ";

$b .= "Dia!"; // configura $b para "Bom Dia!", como em $b = $b .
   "Dia!";
```

Operadores bit-a-bit

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

Exemplo	Nome	Resultado
$a & $b	E	Os bits que estão ativos tanto em $a quanto em $b são ativados.
$a | $b	OU	Os bits que estão ativos em $a ou em $b são ativados.
$a ^ $b	XOR	Os bits que estão ativos em $a ou em $b, mas não em ambos, são ativados.
~ $a	NÃO	Os bits que estão ativos em $a não são ativados, e vice-versa.
$a << $b	Deslocamento à esquerda	Desloca os bits de $a $b passos para a esquerda (cada passo significa "multiplica por dois")
$a >> $b	Deslocamento à direita	Desloca os bits de $a $b passos para a direita (cada passo significa "divide por dois")

Operadores de comparação

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

Exemplo	Nome	Resultado
$a == $b	Igual	Verdadeiro (TRUE) se $a é igual a $b.
$a === $b	Idêntico	Verdadeiro (TRUE) se $a é igual a $b, e eles são do mesmo tipo (introduzido no PHP4).
$a != $b	Diferente	Verdadeiro se $a não é igual a $b.
$a <> $b	Diferente	Verdadeiro se $a não é igual a $b.
$a !== $b	Não idêntico	Verdadeiro de $a não é igual a $b, ou eles não são do mesmo tipo (introduzido no PHP4).
$a < $b	Menor que	Verdadeiro se $a é estritamente menor que $b.
$a > $b	Maior que	Verdadeiro se $a é estritamente maior que $b.
$a <= $b	Menor ou igual	Verdadeiro se $a é menor ou igual a $b.
$a >= $b	Maior ou igual	Verdadeiro se $a é maior ou igual a $b.

- Operador ternário:
  - (expr1) ? (expr2) : (expr3)

Operadores de controle de erros

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- O PHP suporta um operador de controle de erro: o sinal 'arroba' (@). Quando ele precede uma expressão em PHP, qualquer mensagem de erro que possa ser gerada por aquela expressão será ignorada.

```
/* Erro de arquivo intencional */
$my_file = @file ('arquivo_nao_existente') ou
    die ("Falha abrindo arquivo: '$php_errormsg'");

// Isto funciona para qualquer expressão, não apenas para
   funções:
$value = @$cache[$key];
// você não receberá nenhum aviso se a chave $key não existir.
```

Operadores de execução

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- O PHP suporta um operador de execução: acentos graves (``).

```
$output = `ls -al`;
echo "<pre>$output</pre>";
```

Operadores de incremento/decremento

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

Exemplo	Nome	Efeito
++$a	Pré-incremento	Incrementa $a em um, e então retorna $a.
$a++	Pós-incremento	Retorna $a, e então incrementa $a em um.
--$a	Pré-decremento	Decrementa $a em um, e então retorna $a.
$a--	Pós-decremento	Retorna $a, e então decrementa $a em um.

Operadores de comparação

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

Exemplo	Nome	Resultado
$a and $b	E	Verdadeiro (`TRUE`) se tanto $a quanto $b são verdadeiros.
$a or $b	OU	Verdadeiro se $a ou $b são verdadeiros.
$a xor $b	XOR	Verdadeiro se $a ou $b são verdadeiros, mas não ambos.
! $a	NÃO	Verdadeiro se $a não é verdadeiro.
$a && $b	E	Verdadeiro se tanto $a quanto $b são verdadeiros.
$a || $b	OU	Verdadeiro se $a ou $b são verdadeiros.

Operadores de string

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Há dois operadores de string. O primeiro é o operador de concatenação ('.'), que retorna a concatenação dos seus argumentos direito e esquerdo. O segundo é o operador de atribuição de concatenação ('.='), que acrescenta o argumento do lado direito no argumento do lado esquerdo.

```
$a = "Olá ";
$b = $a . "mundo!"; // agora $b contém "Olá mundo!"

$a = "Olá ";
$a .= "mundo!";     // agora $a contém "Olá mundo!"
```

Operadores de array

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

Exemplo	Nome	Resultado
$a + $b	União	União de $a e $b.
$a == $b	Igualdade	TRUE se $a e $b tem os mesmos elementos.
$a === $b	Identidade	TRUE se $a e $b tem os mesmos elementos na mesma ordem.
$a != $b	Desigualdade	TRUE se $a não é igual a $b.
$a <> $b	Desigualdade	TRUE se $a não é igual a $b.
$a !== $b	Não identidade	TRUE se $a não é identico a $b.

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- IF

```
if ($a > $b)
    echo "a é maior que b";

if ($a > $b) {
    echo "a é maior que b";
    $b = $a;
}

if ($a > $b) {
    echo "a é maior que b";
} else {
    echo "a NÃO é maior que b";
}
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- IF

```
if ($a > $b) {
    echo "a é maior que b";
} elseif ($a == $b) {
    echo "a é igual a b";
} else {
    echo "a é menor que b b";
}
?>

<?php if ($a == 5): ?>
A é igual a 5
<?php endif; ?>
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- IF

```
if ($a == 5):
    echo "a igual a 5";
    echo "...";
elseif ($a == 6):
    echo "a igual a 6";
    echo "!!!";
else:
    echo "a não é nem 5 nem 6";
endif;
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- while

```php
$i = 1;
while ($i <= 10) {
    echo $i++;  /* o valor impresso será
                   $i depois do acréscimo
                   (post-increment) */
}

$i = 1;
while ($i <= 10):
    echo $i;
    $i++;
endwhile;
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- do while

```php
$i = 0;
do {
    echo $i;
} while ($i > 0);
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- for

```
for ($i = 1; $i <= 10; $i++) {
    echo $i;
}

for ($i = 1; ; $i++) {
    if ($i > 10) {
        break;
    }
    echo $i;
}
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- for

```
$i = 1;
for (; ; ) {
    if ($i > 10) {
        break;
    }
    echo $i;
    $i++;
}

for ($i = 1, $j = 0; $i <= 10; $j += $i, print $i, $i++);
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- foreach

```php
$a = array(1, 2, 3, 17);

foreach ($a as $v) {
    echo "Valor atual de \$a: $v.\n";
}

foreach ($a as $k => $v) {
    echo "\$a[$k] => $v.\n";
}

foreach ($a as &$value) {
    $value = $value * 2;
}
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- break / switch

```
$i = 0;
while (++$i) {
    switch ($i) {
    case 5:
        echo "No 5<br />\n";
        break 1;  /* Sai somente do switch. */
    case 10:
        echo "No 10; saindo<br />\n";
        break 2;  /* Sai do switch e while. */
    default:
        break;
    }
}
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- continue

```
$i = 0;
while ($i++ < 5) {
    echo "Fora<br />\n";
    while (1) {
        echo "  Meio<br />\n";
        while (1) {
            echo "  Dentro<br />\n";
            continue 3;
        }
        echo "Isto nunca será exibido.<br />\n";
    }
    echo "Nem isso.<br />\n";
}
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- switch

```
switch ($i) {
    case 0:
    case 1:
        echo "i é menor que 3 mas não negativo";
        break;
    case 2:
        echo "i é 3";
}
switch ($i):
    case 0:
        echo "i igual a 0";
        break;
    default:
        echo "i não é igual a 0, 1 ou 2";
endswitch;
```

Estruturas de controle

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- return
- include
- require
- include_once
- require_once

Erros

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

Valor	Constante	Descrição	Nota
1	`E_ERROR` (integer)	Erros em tempo de execução fatais. Estes indicam erros que não podem ser recuperados, como problemas de alocação de memória. A execução do script é interrompida.	
2	`E_WARNING` (integer)	Avisos em tempo de execução (erros não fatais). A execução do script não é interrompida.	
4	`E_PARSE` (integer)	Erro em tempo de compilação. Erros gerados pelo interpretador.	
8	`E_NOTICE` (integer)	Notícia em tempo de execução. Indica que o script encontrou alguma coisa que pode indicar um erro, mas que também possa acontecer durante a execução normal do script.	
16	`E_CORE_ERROR` (integer)	Erro fatal que acontece durante a inicialização do PHP. Este é parecido com `E_ERROR`, exceto que é gerado pelo núcleo do PHP.	Desde PHP 4
32	`E_CORE_WARNING` (integer)	Avisos (erros não fatais) que aconteçam durante a inicialização do PHP. Este é parecido com `E_WARNING`, exceto que é gerado pelo núcleo do PHP.	Desde PHP 4
64	`E_COMPILE_ERROR` (integer)	Erro fatal em tempo de compilação. Este é parecido com `E_ERROR`, exceto que é gerado pelo Zend Scripting Engine.	Desde PHP 4

Erros

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

128	`E_COMPILE_WARNING` (integer)	Aviso em tempo de compilação. Este é parecido com `E_WARNING`, exceto que é geredo pelo Zend Scripting Engine.	Desde PHP 4
256	`E_USER_ERROR` (integer)	Erro gerado pelo usuário. Este é parecido com `E_ERROR`, exceto que é gerado pelo código PHP usando a função trigger_error().	Desde PHP 4
512	`E_USER_WARNING` (integer)	Aviso gerado pelo usuário. Este é parecido com `E_WARNING`, exceto que é gerado pelo código PHP usando a função trigger_error().	Desde PHP 4
1024	`E_USER_NOTICE` (integer)	Notícia gerada pelo usuário. Este é parecido com `E_NOTICE`, exceto que é gerado pelo código PHP usando a função trigger_error().	Desde PHP 4
2047	`E_ALL` (integer)	Todos os erros e avisos, como suportado, exceto do nível `E_STRICT`.	
2048	`E_STRICT` (integer)	Nóticias em tempo de execução. Permite ao PHP sugerir modificações em seu código para segurar melhor interoperabilidade e compatibilidade futura do seu código.	Desde PHP 5

Funções de manipulação de erros

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- error_reporting
- set_error_handler
- trigger_error
- set_exception_handler
- error_log
- debug_backtrace