ESCOLA DE FORMAÇÃO DE
TRABALHADORES EM INFORMÁTICA

# Funções

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

# Sintaxe



```php
function foo ($arg_1, $arg_2, /* ..., */ $arg_n)
{
    echo "Exemplo de função.\n";
    return $valor_retornado;
}
```

```php
if ($makefoo) {
  function foo ()
  {
    echo "Eu não existo até que o programa passe por aqui.\n";
  }
}
```

# Sintaxe



```php
function foo()
{
  function bar()
  {
    echo "Eu não existo até foo() ser chamada.\n";
  }
}

function recursion($a)
{
    if ($a < 20) {
        echo "$a\n";
        recursion($a + 1);
    }
}
```

***Evite o uso de funções/métodos recursivos com mais de 100-200 níveis de recursão já que isso pode estourar a pilha e causar o encerramento do script atual.***

# Argumentos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Informações podem ser passadas para funções através da lista de argumentos, que é uma lista de expressões delimitados por vírgulas.
- O PHP suporta a passagem de argumentos por valor (o padrão), passagem por referência e valores padrão de argumento.
- Listas de argumentos de comprimento variável são suportadas.

# Argumentos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Passagem por referência

```php
function add_some_extra(&$string)
{
    $string .= ' e alguma coisa mais.';
}
$str = 'Isto é uma string,';
add_some_extra($str);
echo $str; // imprime 'Isto é uma string, e alguma coisa mais.'
```

***Referências, em PHP, significa acessar o mesmo conteúdo de variável através de vários nomes. Isto não é parecido como os ponteiros em C: aqui temos apelidos numa tabela simbólica. Note que no PHP nomes de variáveis e conteúdo de variável são tratados diferentemente, então um mesmo conteúdo pode ter nomes diferentes. A analogia mais próxima é a dos arquivos e seus nomes em sistemas UNIX: nomes de variáveis são o caminho completo dos arquivos, enquanto o conteúdo da variável são os dados desse arquivo. Assim, referências pode ser tomadas como os links físicos dentro do sistema de arquivos UNIX.***

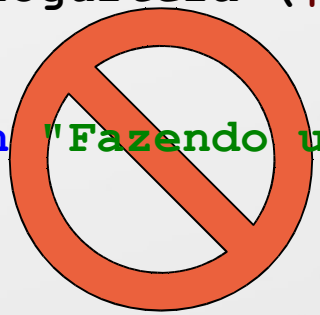
# Argumentos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Parâmetros default

```
function cafeteira ($tipo = "cappuccino")
{
    return "Fazendo uma xícara de café $tipo.\n";
}
echo cafeteira ();
echo cafeteira (null);
echo cafeteira ("expresso");

function iogurtera ($tipo = "azeda", $sabor)
{
    return "Fazendo uma taça de $sabor $tipo.\n";
}
```



# Argumentos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Parâmetros default não escalares

```
function makecoffee($types = array("cappuccino"), $coffeeMaker =
   NULL)
{
    $device = is_null($coffeeMaker) ? "hands" : $coffeeMaker;
    return "Making a cup of ".join(", ", $types)." with $device.\n";
}
echo makecoffee();
echo makecoffee(array("cappuccino", "lavazza"), "teapot");
```

***O valor padrão precisa ser uma expressão constante, não (por exemplo) uma variável, um membro de classe ou uma chamada de função.***

# Argumentos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- Número de argumentos variáveis

```
function foo() {
    $numargs = func_num_args();
    echo "Number of arguments: $numargs<br>\n";
    if ($numargs >= 2) {
        echo "Second argument is: " . func_get_arg (1) .
  "<br>\n";
    }
    $arg_list = func_get_args();
    for ($i = 0; $i < $numargs; $i++) {
        echo "Argument $i is: " . $arg_list[$i] . "<br>\n";
    }
}

foo (1, 2, 3);
```

# Retornos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

```php
function quadrado ($num)
{
    return $num * $num;
}
echo quadrado (4);


function numeros_pequenos()
{
    return array (0, 1, 2);
}
list ($zero, $um, $dois) = numeros_pequenos();
```

# Retornos

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

```php
$alguma_referencia = "aaa";
function &retorna_referencia()
{
global $alguma_referencia;
    return $alguma_referencia;
}

$nova_referencia =& retorna_referencia();
$nova_referencia2 =& retorna_referencia();
$nova_referencia2 = 'bbb';
print $nova_referencia;
```

***Quando não há um retorno declarado explicitamente o PHP retorna NULL. Não há a idéia de void como em outras linguagens.***

# Funções variáveis

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

- O PHP suporta o conceito de funções variáveis. Isto significa que se um nome de variável tem parênteses no final dela, o PHP procurará uma função com o mesmo nome, qualquer que seja a avaliação da variável, e tentará executá-la. Entre outras coisas, isto pode ser usado para implementar callbacks, tabelas de função e assim por diante.
- Funções variáveis não funcionam com construtores de linguagem como **echo()**, **print()**, **unset()**, **isset()**, **empty()**, **include()**, **require()** e outras assim. Você precisa antes construir uma função interceptadora (wrapper) para utilizar qualquer um desses construtores como funções convencionais.

# Funções variáveis

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

```php
function foo() {
    echo "Chamou foo()<br>\n";
}

// Essa eh uma funcao wrapper para echo()
function echoit($string)
{
    echo $string;
}

$func = 'foo';
$func();        // Chama foo()

$func = 'echoit';
$func('test');  // Chama echoit()
```

# Métodos variáveis

ESCOLA DE FORMAÇÃO DE TRABALHADORES EM INFORMÁTICA

```php
class Foo {
    function MetodoVariavel() {
        $name = 'Bar';
        $this->$name(); // Isto chama o método Bar()
    }

    function Bar() {
        echo "Bar foi chamada!";
    }
}

$foo = new Foo();
$funcname = "MetodoVariavel";
$foo->$funcname();  // Isto chama $foo->MetodoVariavel()
```