

[< VOLTAR](#)

Criação de funções

Neste tópico apresentaremos como trabalhar com funções em PHP, entendendo sua declaração, passagem de parâmetros, retorno e acesso.

NESTE TÓPICO

- > Introdução
- > Passagem de parâmetros para funções em PHP
- > Referências

Marcar
tópico



Introdução

Função é um trecho de código que pode ser invocado de qualquer parte do código para executar uma tarefa qualquer e retornar algum valor (CONVERSE, T. & PARK, J., 2005). Ou seja, você pode chamar quantas vezes forem necessárias uma determinada função. Você pode enviar ou não parâmetros para uma função, e esta poderá retornar apenas um valor.

Vamos ao exemplo de uma função simples:

```
1. function nome_da_funcao(){  
2.     código a ser executado;  
3. }
```

O nome de uma função deverá seguir as mesmas regras de criação de variáveis, visto no **TÓPICO IV– Sintaxe, constantes, variáveis**.

Algumas considerações sobre o código acima:

- A palavra reservada *function* é imutável e obrigatória, é ela que define o *código a ser executado* como uma função.

- O *nome_da_funcao* deve ser um valor significativo, pelo qual a função seja facilmente identificada.
- Os valores entre parênteses são chamados de parâmetros ou argumentos e são valores dos quais a função depende para ser executada corretamente. Nem sempre os argumentos são necessários, então pode-se omiti-los na definição da função, mantendo apenas os parênteses.
- As funções podem ser classificadas quanto ao seu retorno como vazias (void) ou com retorno. As funções *void* ou sem retorno apenas executam uma série de comandos sem a obrigação de devolver um valor específico como resultado. Já as funções com retorno, ao serem executadas, resultam diretamente em um dado valor que, no PHP, não tem tipo definido.

Vamos a um exemplo prático:

```
1. <?php
2.     /* -----
3.     |
4.     |function boas_vindas()
5.     |
6.     |Objetivo: Enviar mensagem de boas vindas
7.     |ao visitante do site.
8.     |Data de criação: 26/10/2015
9.     |Autor: Prof. Adriano Milanez
10.    |
11.    ----- */
12.    function boas_vindas(){
13.        echo "Seja bem vindo!";
14.    }
15.    boas_vindas();
16. ?>
```

O código acima, produzirá o resultado abaixo:



Figura 1: Trabalhando com funções no PHP.

Como puderam perceber no código acima, as boas práticas de programação, sugerem que tenhamos o hábito de criar comentários nas funções, contendo o objetivo daquela função, data de criação, assim como seu autor, informações adicionais, como por exemplo, de quais páginas nossa função depende para ser executada, as páginas que serão executadas quando do término da execução da mesma, isto facilita uma manutenção futura.

Passagem de parâmetros para funções em PHP

Muitas vezes desejaremos ir além da simples impressão de um texto, preferiremos criar uma função que receba uma informação, realize uma operação sobre a informação recebida e produza outra informação como saída, este é o real valor de uma função em PHP.

É aí que entram os parâmetros, também chamados de *argumentos de funções*. Eles são valores passados do código externo para o trecho de código dentro da função.

Para utilizar os parâmetros, basta especificar as informações que a função espera receber no código de criação, digitando a variável, ou lista de variáveis, dentro dos parênteses, no código de criação da função (CONVERSE, T. & PARK, J., 2005). E na execução, você passa as informações nas variáveis listadas no código de criação das funções em PHP.

Vejamos o exemplo abaixo:

```
1.  <?php
2.      /* -----
3.      |
4.      |function total()
5.      |
6.      |Objetivo: Criar uma função para somar três valores numéricos
7.      |e retornar o resultado para o código externo.
8.      |Data de criação: 26/10/2015
9.      |Autor: Prof. Adriano Milanez
10.     |
11.     ----- */
12.     function total($valor1, $valor2, $valor3){
13.         $total = $valor1 + $valor2 + $valor3;
14.         return $total;
15.     }
16.     echo total(365,221, 105);
17.  ?>
```

Perceba que a palavra reservada *return*, na linha 14, é utilizada para definir o resultado da função e sinaliza também o fim da execução desta. Qualquer código que venha a ser posto após o *return* (o que não deve ocorrer) será desconsiderado.

Quantidade de parâmetros variável

A linguagem PHP possui uma funcionalidade bastante útil que permite definir funções que recebam um número indefinido de argumentos (NIEDERAUER, J., 2004), com isso, pode-se acessar vários parâmetros dinamicamente, mesmo sem que seja previamente definida a quantidade e nome dos mesmos. Dentro da função, utiliza-se as funções `func_get_args()` e `func_num_args()` que retornam um array contendo os argumentos e a quantidade destes, respectivamente.

A seguir é apresentado um exemplo, retomando a função “total”, que define uma função para a qual se pode passar vários números como parâmetro e ela retornará a soma deles.

```

1.  <?php
2.      /* -----
3.      |
4.      |function total()
5.      |
6.      |Objetivo: Criar uma função para somar N valores numéricos
7.      |e retornar o resultado para o código externo.
8.      |Data de criação: 26/10/2015
9.      |Autor: Prof. Adriano Milanez
10.     |
11.     ----- */
12.     function total(){
13.         $numeros = func_get_args();
14.         $qtd_nums = func_num_args();
15.         $soma = 0;
16.         for($i = 0; $i < $qtd_nums;$i++){
17.             $soma += $numeros[$i];
18.         }
19.         return $soma;
20.     }
21.     echo total(365,221,105,112,27,33,208);
22.  ?>

```

Antes de encerrar este tópico é interessante lembrar que o PHP é *case-sensitive*, faz diferença entre letras maiúsculas e minúsculas, porém esta regra não vale para as funções, ou seja, ele não diferencia maiúsculas de minúsculas. Sendo assim, um código, como o mostrado abaixo, funciona perfeitamente:

```

1.  <?php
2.      /* -----
3.      |
4.      |function boas_vindas()
5.      |
6.      |Objetivo: Enviar mensagem de boas vindas
7.      |ao visitante do site.
8.      |Data de criação: 26/10/2015
9.      |Autor: Prof. Adriano Milanez
10.     |
11.     ----- */
12.     function boas_vindas(){
13.         echo "Seja bem vindo!";
14.     }
15.     BOAS_VINDAS();
16.  ?>

```

Trabalhar com funções em PHP é bastante simples e eficiente, devido aos recursos e sintaxe da linguagem. Assim, torna-se fácil e prático promover a reusabilidade de código nas aplicações.

Espero que o conteúdo tenha sido proveitoso. Estudem e até o próximo tópico!

Quiz

Exercício Final

Criação de funções

INICIAR ➤

Referências

CONVERSE, T. & PARK, J. **PHP 5 - A Bíblia**. 2. Ed. Rio de Janeiro: Campus, 2005.

DAL'OGGIO, P. **Programando com orientação a objetos (Inclui *Design Patterns*)**. São Paulo: Novatec, 2009.

MARTIN, J. **Princípios de análise e projeto baseado em objetos**. Rio de Janeiro: Editora Campus, 1994.

MATHEUS, D. **Variáveis e constantes no PHP**. Disponível em <<http://www.diogomatheus.com.br/blog/php/variaveis-e-constantes-no-php/>>, acessado em 19/09/2015, às 12h15min. Publicado em 26/09/2012.

MUTO, C. A. **PHP e MySQL: Guia Introdutório**. 3. Ed. Rio de Janeiro: Brasport, 2006.

NIEDERAUER, J. **Desenvolvendo Websites com PHP**. 2. Ed. São Paulo: Novatec, 2004.

PHP. **PHP: Funções definidas pelo usuário**. Disponível em <http://php.net/manual/pt_BR/functions.user-defined.php>. Acessado em 26/10/2015, às 14h.

SOARES, L.; AUGUSTO, B. **Aprendendo a Linguagem PHP**. 1. Ed. Rio de Janeiro: Ciência Moderna, 2007.

SOARES, W. **PHP 5 - Conceitos, Programação e Integração com Banco de Dados**. 4. Ed. São Paulo: Érica, 2004.

TANSLEY, D. **Como criar Web Pages rápidas e eficientes usando PHP e MySQL**. 1. Ed. Rio de Janeiro: Ciência Moderna, 2002.

W3Schols. PHP Functions. Disponível em: <http://www.w3schools.com/php/php_functions.asp>, acessado em 26/10/2015, as 21h25min.



Avalie este tópico



ANTERIOR

Resgatando informações de Formulários

HTML



Índice

Biblioteca

(<https://www.uninove.br/conhec>

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site

Ajuda?

PRÓXIMO: (<https://ava.un>

Diferenças entre Cookies e Sessões)

© Todos os direitos reservados

