

## PHP

- 1 PHP简介
- 2 PHP基本语法
- 3 PHP流程控制
- 4 PHP函数
  - 4.1 PHP函数是什么
  - 4.2 PHP定义函数
  - 4.3 PHP函数的参数
  - 4.4 PHP声明参数类型
  - 4.5 PHP函数参数的传递
  - 4.6 PHP函数返回值
  - 4.7 PHP匿名函数
  - 4.8 PHP可变函数
  - 4.9 PHP回调函数
  - 4.10 PHP递归函数
- 5 PHP字符串操作
- 6 PHP数组
- 7 PHP时间和日期
- 8 PHP面向对象
- 9 正则表达式
- 10 PHP会话控制
- 11 PHP错误和异常处理
- 12 MySQL数据库的基础操作

[🏠 首页](#) > [PHP](#) > [PHP函数](#)

阅读：9,499

## PHP函数参数传递的四种方式



小白入职大厂完全攻略，很硬很肝  
学习路线 / 笔试面试 / 升职加薪 / 跳槽技巧

[猛击查看详情](#)

C语言中文网推出辅导班啦，包括「C语言辅导班、C++辅导班、算法/数据结构辅导班」，全部都是一对一教学：一对一辅导 + 一对一答疑 + 布置作业 + 项目实践 + 永久学习。QQ在线，随时响应！

在调用函数时，需要向函数传递参数，被传入函数的参数称为实参，而函数定义的参数称为形参。而向函数传递参数的方式有四种，分别是值传递、引用传递、默认参数和可变长度参数。

### 1. 值传递

值传递是 PHP 中函数的默认传值方式，也称为“拷贝传值”。顾名思义值传递的方式会将实参的值复制一份再传递给函数的形参，所以在函数中操作参数的值并不会对函数外的实参造成影响。因此如果不希望函数修改实参的值，就可以通过值传递的方式。

【示例】下面定义一个简单的函数，函数有两个参数，在函数中交换参数的值。

```
01. <?php
02.     function swap($a, $b){
03.         echo '函数内，交换前 $a = '.$a.', $b = '.$b.'<br>';
04.         $temp = $a;
05.         $a = $b;
06.         $b = $temp;
07.         echo '函数内，交换后 $a = '.$a.', $b = '.$b.'<br>';
```

[13 PHP文件目录操作](#)[14 PHP图像处理](#)

```
08.     }  
09.  
10.     $x = 5;  
11.     $y = 7;  
12.     echo '函数外, 交换前 $x = ' . $x . ', $y = ' . $y . '<br>';  
13.     swap($x, $y);  
14.     echo '函数外, 交换后 $x = ' . $x . ', $y = ' . $y . ';<br>';  
15.     ?>
```

运行结果如下:

```
函数外, 交换前 $x = 5, $y = 7  
函数内, 交换前 $a = 5, $b = 7  
函数内, 交换后 $a = 7, $b = 5  
函数外, 交换后 $x = 5, $y = 7
```

通过运行结果可以看到, 在函数内, 数值确实是交换了, 而在函数外部, 数值却没有变化。因此可以说, 函数的值传递只是传递变量的副本。那么要是想让函数能够操作函数外实参的话, 就需要使用引用传递的方式了。

## 2. 引用传递

参数的引用传递就是把实参的内存地址复制一份, 然后传递给函数的形参, 实参和形参都指向同一个内存地址, 因此函数对形参的操作, 会影响到函数外的实参。

按引用传递就是将实参的内存地址传递到函数的形参中。因此实参和形参指向的是同一个内存地址。这时在函数内部的所有操作都会影响到函数外实参的值。引用传递的方式就是在值传递的基础上加上一个 `&` 符号, 如下所示:

```
function name (&参数1, &参数2, ..., &参数3) {  
    ...  
}
```

【示例】将上面示例的代码略作调整, 使用引用传递的方式向 `swap` 函数中传递参数, 代码如下所示:

```
01. <?php
02.     function swap(&$a, &$b) {
03.         echo '函数内, 交换前 $a = '.$a.', $b = '.$b.'<br>';
04.         $temp = $a;
05.         $a = $b;
06.         $b = $temp;
07.         echo '函数内, 交换后 $a = '.$a.', $b = '.$b.'<br>';
08.     }
09.
10.     $x = 5;
11.     $y = 7;
12.     echo '函数外, 交换前 $x = '.$x.', $y = '.$y.'<br>';
13.     swap($x, $y);
14.     echo '函数外, 交换后 $x = '.$x.', $y = '.$y;
15. ?>
```

运行结果如下:

```
函数外, 交换前 $x = 5, $y = 7
函数内, 交换前 $a = 5, $b = 7
函数内, 交换后 $a = 7, $b = 5
函数外, 交换后 $x = 7, $y = 5
```

### 3. 默认参数

默认参数就是给函数的某个或多个形式参数指定一个默认的值, 如果调用函数时不传入对应的值, 那么函数就会使用这个默认值, 这样可以避免调用时出现没有参数的错误, 也可以使一些程序显得更加合理。如果传入对应的参数, 就会替换这个默认值。

函数的默认参数如下所示:



```
function name ($str = 'C语言中文网', $url) {  
    echo $str;  
}
```

其中，形式参数 \$str 后面的“C语言中文网”就是它的默认值，形参与默认值之间需要使用 `=` 连接。

【示例】下面来定义一个具有默认参数的函数，如下所示：

```
01. <?php  
02.     function add($a, $b=56) {  
03.         echo $a.' + '.$b.' = '.($a+$b).'  
<br>';  
04.     }  
05.  
06.     add(11);  
07.     add(37, 29);  
08. ?>
```

运行结果如下：

```
11 + 56 = 67  
37 + 29 = 66
```

默认参数也可以是多个，而且默认参数必须放在非默认参数右边，并且指定默认参数的值必须是一个具体的值，如数字、字符串，而不能是一个变量。

【示例】下面来定义一个具有多个默认参数的函数，如下所示：

```
01. <?php  
02.     function add($a, $b=33, $c=57) {  
03.         echo $a.' + '.$b.' + '.$c.' = '.($a+$b+$c).'  
<br>';  
04.     }  
05.  
06.     add(11);
```

```
07.      add(31, 22);  
08.      add(64, 9, 7);  
09.      ?>
```

运行结果如下:

```
11 + 33 + 57 = 101  
31 + 22 + 57 = 110  
64 + 9 + 7 = 80
```

## 4. 可变长度参数

在 PHP 5.6 及以后的版本中, 函数的形式参数可使用 `...` 来表示函数可接受一个可变数量的参数, 可变参数将会被当作一个数组传递给函数。示例如下:

```
01.  <?php  
02.      function test(...$arr) {  
03.          print_r($arr);  
04.      }  
05.      echo '<pre>';  
06.      test(1, 2, 3, 4);  
07.      test(5, 6, 7, 8, 9, 10);  
08.      ?>
```

运行结果如下:

```
Array  
(  
    [0] => 1  
    [1] => 2  
    [2] => 3  
    [3] => 4
```

```
)  
Array  
(  
    [0] => 5  
    [1] => 6  
    [2] => 7  
    [3] => 8  
    [4] => 9  
    [5] => 10  
)
```

关注微信公众号「站长严长生」，在手机上阅读所有教程，随时随地都能学习。本公众号由[C语言中文网站](#)长运营，每日更新，坚持原创，敢说真话，凡事有态度。



微信扫描二维码关注公众号

### 优秀文章

setbuf与setvbuf函数，C语言setbuf与setvbuf函

迁移学习及实操（使用预训练的VGG16网络）详解

Docker HTTP路由网格（HRM）

JS实现无刷新图片预览

linux端口及查询方法详解

C语言模块化编程中的头文件

编写Python开发文档的重要性

PHP array\_flip(): 交换数组中的键和值



精美而实用的网站，分享优质编程教程，帮助有志青年。千锤百炼，只为大作；精益求精，处处斟酌；这种教程，看一眼就倾心。

[关于网站](#) | [关于站长](#) | [如何完成一部教程](#) | [联系我们](#) | [网站地图](#)

Copyright ©2012-2022 biancheng.net, 陕ICP备15000209号

*biancheng.net*