

函数基础

1. 函数的定义：

形式为：function 函数名(【\$行参1】 【， \$行参2】 【， ...】){
 函数体
};

说明：

(1). 定义时使用的形参，其实就是一个变量——只能在该函数内使用

(3). 形参作为变量，其名字可以是“自己定义”——自然应该遵循命名规范

2. 函数的调用：

形式为：函数名(\$实参1, \$实参2, ...);

说明：

(1). 实参应该跟要调用的函数的形参一一对应；

(2). 实参就是“数据值”，可能是直接值(如5, "abc"), 也可能是变量值；

```
<?php  
  
function f1($x, $y){  
    $s = $x * $x + $y * $y; //平方和  
    $result = sqrt($s); //开根号  
    return $result;  
}  
  
echo f1(3, 4);  
  
?>
```

3. 函数的调用详细过程

(1) 首先，将函数调用时的实参数据，传递(赋值)给函数的形参(变量)

(2) 程序的执行流程，进入到函数内部去——此时可以认为是一个跟外界“隔离”的“独立运行空间”

(3) 函数内部，按正常的流程顺序，执行其中的代码

(4) 直到函数结束，则退出该“运行空间”，而返回到函数调用的位置，继续执行后续的代码

(5) 如果在函数的内部执行的过程中有return语句，则会立即终止函数，并回到函数调用的位置，继续执行后续的代码

4. 函数的参数问题

(1) 函数形参的默认值问题

定义：我们可以给一个函数定义时的形参，赋值一个“默认值”，则这个函数调用的时候，该形参对应的实参，可以不用传入

```
function f2($x, $y = 4){  
    $s = $x * $x + $y * $y; //平方和  
    $result = sqrt($s); //开根号  
    return $result;  
}  
  
echo f2(2). "<br>";
```

一个函数形参可以部分设置默认值，但是有一个要求，那就是，被设置默认值的形参必需放在右边，没有默认值的放左边。

(2) 形参的传值问题

一句话，其实就是“变量之间的传值问题”：无非就是实参变量传值给形参变量的问题

即：

此时，也同样有两种传值方式：

值传递、引用传递

```
function f3(&$x, $y = 4){  
    $s = $x * $x + $y * $y; //平方和  
    $x = $s;  
    $result = sqrt($s); //开根号  
    return $result;  
}  
  
$x = 3;  
echo f3(3). "<br>";  
echo "x=" . $x. "<br>";
```

上述程序，运行后结果报致命错误，原因是：3不能作为引用传递的形参对应的实参

正确的做法应该是一个正确的变量：

```
function f3(&$x, $y = 4){  
    $s = $x * $x + $y * $y; //平方和  
    $x = $s;  
    $result = sqrt($s); //开根号  
    return $result;  
}  
  
$x = 3;  
echo f3($x). "<br>";  
echo "x=" . $x. "<br>";
```

最常访问 火狐官方站点 新手上路

5
5
5
x=25

传递(值传递)的实参的值，在函数内部发生了改变，并不会改变该实参变量的值

引用传递的实参的值，在函数内部发生了改变，那么，该实参的值也会被改变

(3) 函数参数的数量问题


- 1) 通常函数的实参数量，应该和函数定义时候的形参数量保持一致
- 2) 如果定义时，形参有默认值，则对应的实参就可以进行一定程度的省略
注意：省略只能从右往左进行
- 3) 有一种定义和使用函数的特别形式(并不常见)：它不定义形参，而实参任意给出，其实，系统中也有类似的函数，比如：

```
var_dump($v1);
```

```
var_dump($v1, $v2);
```

可见，这种函数就可以接收任意个数的实参数

示例如下：



```
<?php

function f4(){
    $arr = func_get_args();
    for ($i=0; $i <= count($arr) - 1; $i++) {
        echo $arr[$i]."<br>";
    }
}

f4(3, 5);
?>
```

for循环遍历，中获取数组大小的方式，除了count函数，还可以使用其他方式比如：func_num_args()

5. 函数的返回值问题

- (1) 函数返回值，不是必需的，根据具体需求而定
- (2) 如果需要返回值，则使用return语句后面跟一个数据(直接数据或变量或常量)，就可以返回该数据
- (3) 如果不需要返回值，则函数中可以没有return语句，或有return语句，但没有数据
- (4) return语句的作用，不管后面跟不跟数据，都会立即终止函数的执行，返回到函数调用的位置，并且继续其他后续代码执行