

## 函数的定义

- 定义时使用的形参,其实就是一个变量 –只能在该函数内部使用的变量
- 形参作为变量,其名字是“自己定义” – 自然应该遵循命名规范;

## 函数的调用

- 实参应该跟要调用的函数的形参一一对应
- 实参就是“数据值”,可能是直接值(比如 5,“abc”),也可能是变量值比如(\$v1);

形参: parameter

实参: argument

函数形参的默认值,可以只给部分形参设置默认值,但设置默认值性的形参,都要放在“右边”(后边);

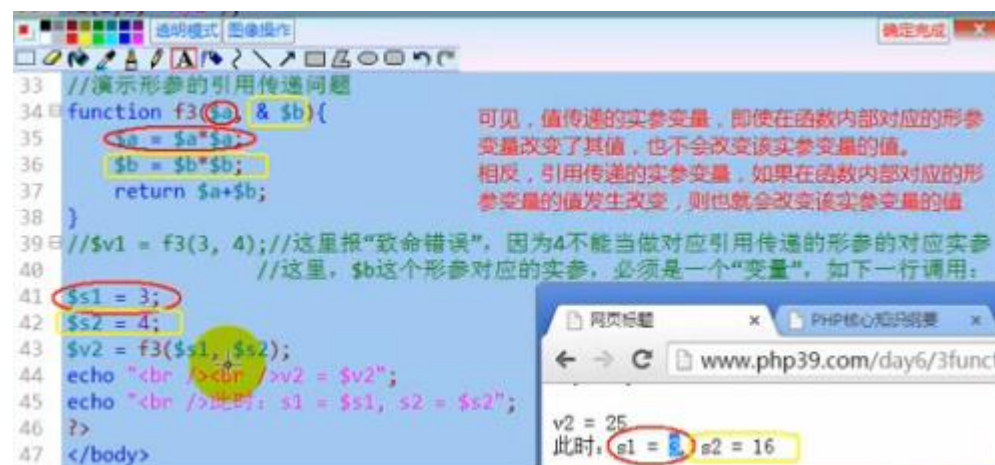
## 形参的传递问题

值传递:

这是默认值,如果没有特别设定,参数传递都是值传递;

引用传递:

需要在形参的前面加&符号;



```
33 //演示形参的引用传递问题
34 function f3($a, &$b){
35     $a = $a*$a;
36     $b = $b*$b;
37     return $a+$b;
38 }
39 // $v1 = f3(3, 4); //这里报“致命错误”,因为4不能当做对应引用传递的形参的对应实参
40 //这里, $b这个形参对应的实参,必须是一个“变量”,如下一行调用:
41 $s1 = 3;
42 $s2 = 4;
43 $v2 = f3($s1, $s2);
44 echo "<br /><u>v2 = $v2";
45 echo "<br />此时: s1 = $s1, s2 = $s2";
46 ?>
47 </body>
```

可见,值传递的实参变量,即使在函数内部对应的形参变量改变了其值,也不会改变该实参变量的值。  
相反,引用传递的实参变量,如果在函数内部对应的形参变量的值发生改变,则也就会改变该实参变量的值。

v2 = 25  
此时: s1 = 9, s2 = 16