**原理**

自己调用自己 称之为“递归”，而不是回调  
  
你也知道回调的关键是这个回  
既然是回，那么就有一个谁是主体的问题，因为回调是往回调用的意思  
我调用了**函数A**，而**函数A在执行过程中调用了我提供的函数B**，这个**函数B就称为函数A的回调函数**  
显然主体是函数A  
  
我们知道，函数是完成某个特定功能的代码集合，在函数执行的过程中，一般是不能去干预他的行为的  
**当函数被设计成带有回调功能时，我们就有可能在函数的执行过程中，通过回调函数去干预他**。例如

function foo($n, $f='') {

  if($n < 1) return;

  for($i=0; $i<$n; $i++) {

    echo $f ? $f($i) : $i;

  }

}

//无回调时

foo(5); //01234

//有回调时

function f1($v) {

  return $v + $v;

}

foo(5, 'f1'); //02468

这里就是foo函数去调用（回调）了f1函数

<?php

function text1(){

echo '这个是text1';

}

function text2($username){

echo '这个是text2'.'<br/>'.'这里是username的值：'.$username;

}

//回调函数:函数体内的名称是我们传入的参数（）；

function callBack($call,$str){

$call($str);

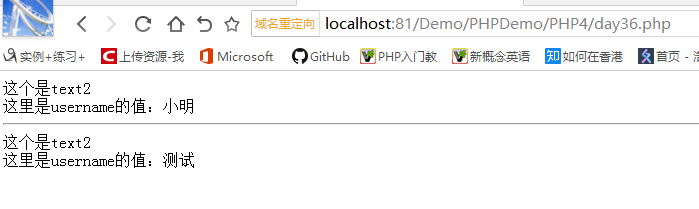
}

//回调函数的使用就是传入的参数是你想要回调的函数名称

callBack('text2','小明');

echo '<hr/>';

call\_user\_func('text2','测试');



回调函数的函数名，和回调函数需要的参数值

### 回调有自己写的回调函数和call\_user\_func两种

回调函数，通过别的函数来调用这个函数，所以要传给调用这个函数的那个函数这个函数的名字和需要的参数。