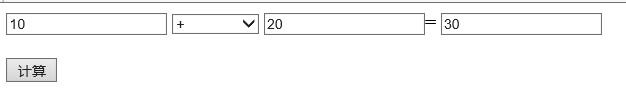


主讲教师：传智.神龙教主

传智播客 PHP学院 <http://php.itcast.cn> 发布

## 作业

### 计算器



<?php

$num1='';

$num2='';

$sign='';

$result='';

if(!empty($\_POST)) {

$num1=$\_POST['num1'];

$num2=$\_POST['num2'];

$sign=$\_POST['sign'];

switch($sign) {

case '+':

$result=$num1+$num2;

break;

case '-':

$result=$num1-$num2;

break;

case '\*':

$result=$num1\*$num2;

break;

case '/':

if($num2==0)

$result='除数不能为0';

else

$result=$num1/$num2;

break;

case '%':

$result=$num1%$num2;

break;

}

}

?>

<form method="post" action="">

<input type="text" name="num1" value="<?php echo $num1?>">

<select name="sign">

<option value="+" <?php echo $sign=='+'?'selected':''?>>+</option>

<option value="-" <?php echo $sign=='-'?'selected':''?>>-</option>

<option value="\*" <?php echo $sign=='\*'?'selected':''?>>\*</option>

<option value="/" <?php echo $sign=='/'?'selected':''?>>/</option>

<option value="%" <?php echo $sign=='%'?'selected':''?>>%</option>

</select>

<input type="text" name="num2" value='<?php echo $num2?>'>

=

<input type="text" name="result" value="<?php echo $result?>"> <br /><br />

<input type="submit" name="button" value="计算">

</form>

### 通过for循环求数组元素的和

<?php

$num=array(10,20,30,40,50);

$result=0;

for($i=0,$n=count($num); $i<$n; $i++) {

$result+=$num[$i];

}

echo '数组元素的和是：'.$result;



### 求阶乘

<?php

if(isset($\_POST['button'])){

$num=trim($\_POST['num']);

if($num=='' || !is\_numeric($num)) {

echo '请输入数字';

}else {

$num+=0; //将字符串数字转成数字型

if(is\_int($num)) {

if($num<0) {

echo '请输入非负整数';

}else {

if($num==0) {

echo '0!=1';

}else {

$result=1;

for($i=1; $i<=$num; $i++) {

$result\*=$i;

}

echo "{$num}!={$result}";

}

}

}else {

echo '您输入的不是整数';

}

}

}

?>

<form method="post" action="">

请输入一个数：<input type="text" name="num"> <br />

<input type="submit" name="button" value="计算阶乘">

</form>



### 循环输出1-100,3的倍数输出A,5的倍数输出B，15的倍数输出C

for($i=1; $i<=100; $i++) {

if($i%15==0)

echo 'C','<br>';

elseif($i%5==0)

echo 'B','<br>';

elseif($i%3==0)

echo 'A','<br>';

else

echo $i,'<br>';

}

### 打印水仙花数 a3+b3+c3=abc

方法一：

for($i=100; $i<999; $i++) {

$a=(int)($i/100); //获取百位的数

$b=(int)($i%100/10); //获取十位的数

$c=$i%10; //获取个位的数

if($i==pow($a,3)+pow($b,3)+pow($c,3))

echo $i,'<br>';

}



//方法二：

for($a=1; $a<=9; $a++) {

for($b=0; $b<=9; $b++) {

for($c=0; $c<=9; $c++) {

if(pow($a,3)+pow($b,3)+pow($c,3)==$a.$b.$c)

echo $a.$b.$c,'<br>';

}

}

}

echo '<hr>';

//方法三：

for($i=100; $i<999; $i++) {

$a=substr($i,0,1);

$b=substr($i,1,1);

$c=substr($i,2,1);

if($i==pow($a,3)+pow($b,3)+pow($c,3))

echo $i,'<br>';

}

### 打印100以内的斐波那契数列 1 1 2 3 5 8

<?php

$num1=1;

$num2=1;

echo $num1,'&nbsp;',$num2,'&nbsp;';

for(;;) {

$num3=$num1+$num2;

if($num3>100) {

break;

}

echo $num3,'&nbsp;';

$num1=$num2;

$num2=$num3;

}



## 函数

函数是一个固定的代码块，或者称为一个子程序，一个函数可以多次调用。函数实现了计算机模块化编程。

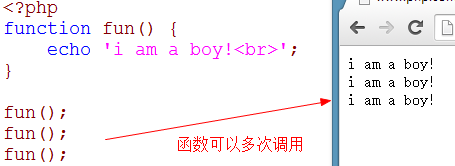
### 函数的定义

function 函数名(参数,参数,…){

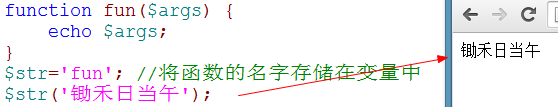
//函数体

}

1. 函数的命名规则：函数以字母、下划线开头，后面跟的是数字、字母、下划线。
2. 函数名不区分大小写。
3. 函数可以多次调用。

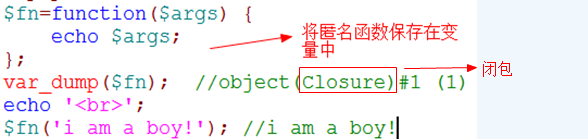


### 可变函数



### 匿名函数

PHP从5.3开始支持匿名函数，匿名函数就是没有名字的函数，匿名函数保存在变量中。



## create\_function(参数字符串，函数体字符串)

create\_function()用来创建函数，通过此函数只需要定义需要的函数即可。

$lang='eng';

if($lang=='ch')

$fn=create\_function('$name','echo \'你好\'.$name;');

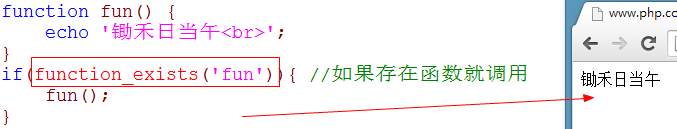
elseif($lang=='eng')

$fn=create\_function('$name','echo \'hello\'.$name;');

$fn('李白');



## function\_exists()：判断函数是否存在

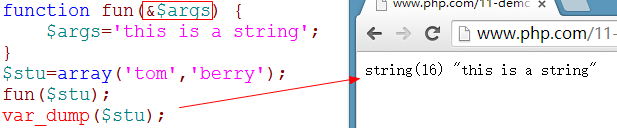
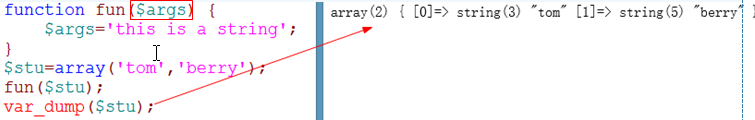


## 函数的参数

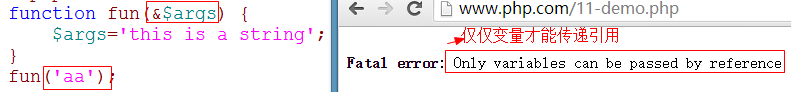
1. 形式参数 形参
2. 实际参数 实参

### 函数的传值与传址

默认情况下，函数的值都是值传递，在定义函数的时候前面的参数上加上取地址，表示传递的是变量的地址

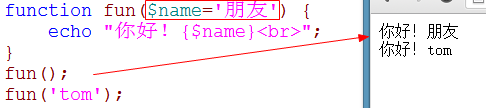


脚下留心：如果在传递地址，在调用函数的时候不能直接写值，只能写变量。

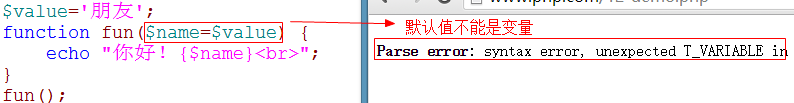


### 参数的默认值

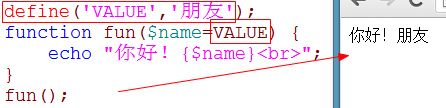
1. 形式参数在定义的时候可以给参数赋一个默认值。



1. 默认值必须是一个存在的值，不能通过变量来代替



1. 默认值可以是常量



1. 函数有多个参数的时候，没有默认值的参数要放在前面，有默认值的参数放在后面。

function fun($name,$sex='男',$age='未知',$add='地址不详') {

echo "你好：{$name}<br>";

echo "性别：{$sex}<br>";

echo "年龄：{$age}<br>";

echo "地址：{$add}<hr>";

}

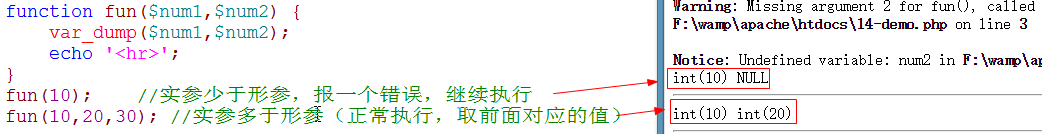
fun('李白');

fun('杜甫','女');

fun('白居易','男',22);

fun('李清照','女',23,'北京');

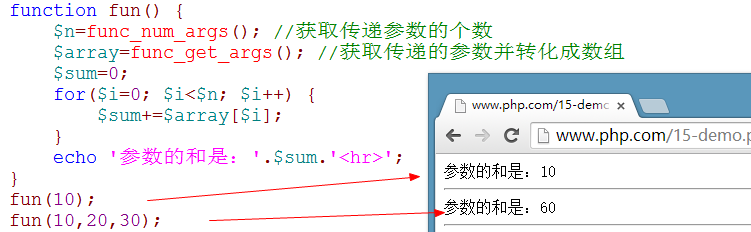
1. 形参和实参个数不一样



1. 传递不确定个数的参数

1、func\_num\_args()：获取参数的个数

2、func\_get\_args()：获取传递的所有参数并转成数组

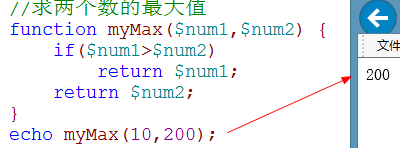


## 函数的返回（return语句）

return的作用：

1. 立即从函数中退出
2. Return返回值

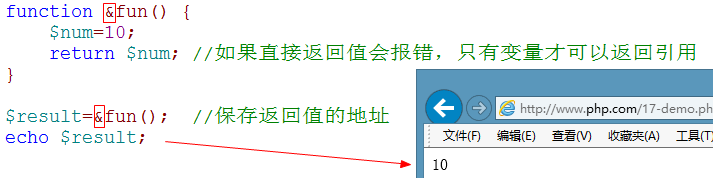
例题：求两个数的最大值



### 返回函数的引用

定义函数时，在函数名前面加上”&”表示返回函数的引用

调用函数时，在函数名前面加上“&”，如果没有“&”符，保存的是返回值的值。



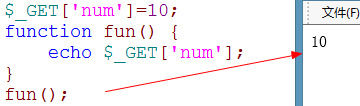
## 变量的作用域

1. 全局变量：在函数外部的变量
2. 局部变量：在函数内部的变量
3. 超全局变量：在函数的内部和外部使用。

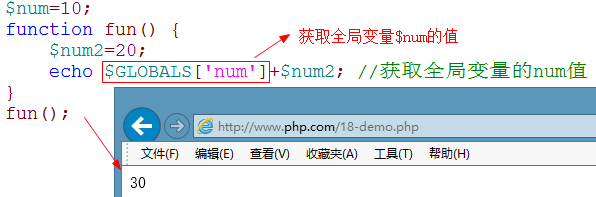
在javascript中，局部变量只能在函数内部使用，全局变量在函数内部和函数外部使用。

默认情况下在PHP中全局变量只能在函数外部使用，局部变量在函数内部使用。

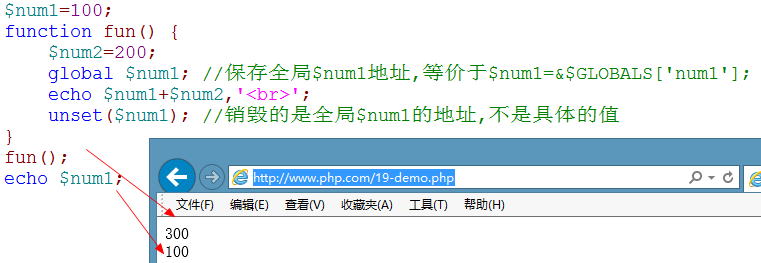
//通过超全局变量实现变量在函数内部和外部访问



//通过$GLOBALS[]来获取全局变量的值

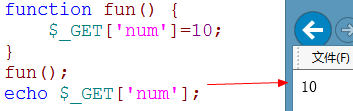


//global关键字用于创建一个对全局变量的引用

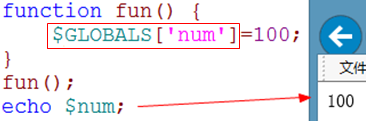


思考：如果在函数内部声明全局变量？

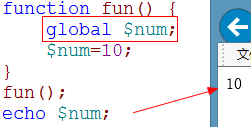
方法一：



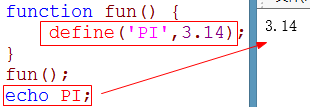
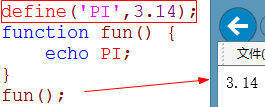
方法二：(推荐使用)



方法三：



多学一招：变量有作用域的概念，常量没有作用域的概念



## 变量的生命周期

1. 全局变量：声明的时候产生，页面执行完毕后销毁
2. 局部变量：函数执行的时候产生，函数执行完毕销毁。

可以通过unset()手动销毁变量。

## 静态变量

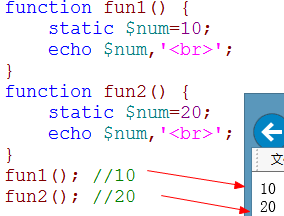
1、静态变量放在函数内部

2、作用域不变，还是在函数的内部访问，但是生命周期变了，页面执行完毕后才销毁。

3、静态变量只能被初始化一次。

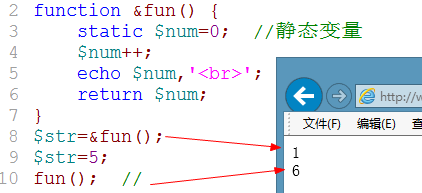
|  |  |
| --- | --- |
|  |  |

思考：如下代码结果是什么？



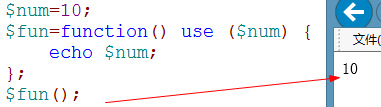
静态变量作用域没有变，只是生命周期发生了变化。

思考，如下代码结果是什么？



## 匿名函数的use()

匿名函数的作用域和普通函数是一样的，函数内部不能访问函数外部的变量，匿名函数中的use()可以将匿名函数外的变量引入到匿名函数内部。



思考：如下代码输出什么？

$num=10;

function myfun() {

$num=20;

$fun=function() use($num){

echo $num;

};

$fun(); //20

}

myfun();

结论：use()引入和匿名函数所在同一作用域的变量

### use()的传值和传址

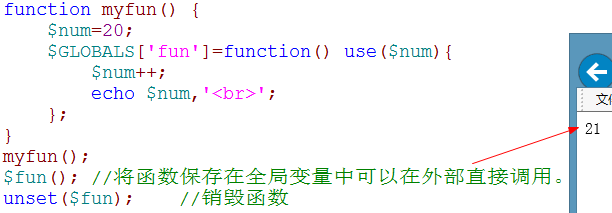
默认情况下，use()是传值的。

|  |  |
| --- | --- |
|  |  |

### 匿名函数的作用

适合建立一个临时函数，用完以后可以销毁，如果一个函数只需要执行一次就可以声明成匿名函数。

思考：如何将函数内部的匿名函数在函数外部调用



## 递归

语法：函数的内部自己调用自己

要素：

1. 递归点：递归调用的地方
2. 递归出口：递归调用结束。

### 递归：从1加到100

function fun($num) {

if($num==1)

return 1; //递归出口

return $num+fun($num-1); //递归点

}

echo fun(100);

/\*\*\*分析

fun(100)=100+fun(99);

=100+99+fun(98);

=100+99+98+fun(97);

=100+99+98+97++fun(1);

=100+99+98+97+1

\*/



### 递归：前10个斐波那契数列

//第n个斐波那契数

function cal($n) {

if($n==1 || $n==2)

return 1;

return cal($n-1)+cal($n-2);

}

/\*\*

\*分析：第5个斐波那契数

cal(5) =cal(4)+cal(3)

=2\*cal(3)+cal(2)

=2\*[cal(2)+cal(1)]+1

=2\*(1+1)+1

=5

\*/

//前10个斐波那契数列

for($i=1;$i<=10;$i++)

echo cal($i),'&nbsp;';

脚下留心：在递归调用的时候，有时候可能PHP执行的时间不够用，可以通过php.ini配置文件更改php最大执行时间。



## 函数的代理调用

不直接通过函数名去调用函数，而是通过PHP的内置函数call\_user\_func\_array()来调用函数

<?php

function concat() {

$array=func\_get\_args();

if(empty($array)){

echo '您没有传递参数<br>';

return;

}

$str='';

for($i=0,$n=count($array); $i<$n; $i++) {

$str.=$array[$i];

}

echo $str;

}

//concat(1,2,3);

if(isset($\_POST['button'])) {

$str=$\_POST['str'];

$array=explode(',',$str);

call\_user\_func\_array('concat',$array); //函数的代理调用

}

?>

<form method="post" action="">

请输入字符，用逗号分隔：<input type="text" name="str">

<input type="submit" name="button" value="连接字符串">

</form>



## 作业：

1. 自定义两个函数求数组中的最大值和最小值。（必须完成）
2. 递归法：因数分解 8=2\*2\*2
3. 递归法：求最大公约数和最小公倍数（8251和6105）
4. 递归法：一个猴子守着一堆桃子，第一天吃了一半又多吃了一个，第二天一样，到第十天就剩下一个桃子，请问原来有几个桃子。