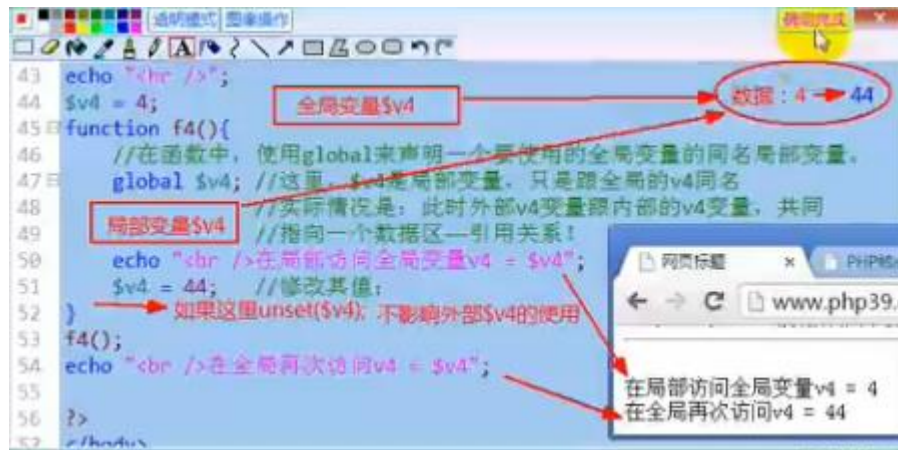
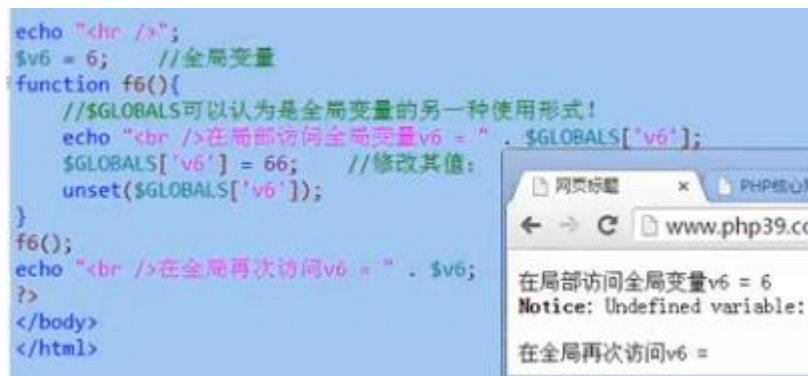


在局部作用域中引用全局变量

做法 1:



做法 2:



有关函数的系统函数:

- Function\_exists() : 判断一个函数是否被定义过,其中使用的参数为"函数名";
- Func\_get\_arg()
- Func\_get\_args()
- Func\_num\_args()

有关函数的编程思想

递归思想---递归函数

递归函数 : 就是在一个函数内部调用它自己的函数.

实用的递归函数是 : 能够控制这个调用的过程中,会在某个时刻(条件下)停下来.

```

14 //现在，假设，有一个函数，该函数“能够”计算n的阶乘。
15 function jiecheng( $n ){
16     echo "<br />开始：有人要求($n)的阶乘";
17     if( $n == 1){
18         echo "<br />结束：终于求到了($n)的阶乘：1";
19         return 1;
20     }
21     $jieguo = $n * jiecheng($n-1);
22     echo "<br />结束：终于求到了($n)的阶乘：$jieguo";
23     return $jieguo;
24 }
25 $v2= jiecheng(5); //结果应该是5的阶乘
26 /*
27 演示调用过程：
28 $v2 = jiecheng(5)相当于：
29 $v2 = 5 * jiecheng(4)==>>
30 $v2 = 5 * (4 * jiecheng(3) ) ==>>
31 $v2 = 5 * (4 * (3 * jiecheng(2) ) ) ==>>
32 $v2 = 5 * (4 * (3 * (2 * jiecheng(1) ) ) ) ==>>
33 $v2 = 5 * (4 * (3 * (2 * 1 ) ) ) ==>>
34 $v2 = 5 * (4 * (3 * 2 ) ) ==>>
35 $v2 = 5 * (4 * 6 ) ==>>
36 $v2 = 5 * 24 ==>>
37 $v2 = 120
38 */
39 echo "<br />$v2 = $v2";

```

网页标题 x PHP教程网 x

← → C www.php39.com/d

开始：有人要求5的阶乘  
 开始：有人要求4的阶乘  
 开始：有人要求3的阶乘  
 开始：有人要求2的阶乘  
 开始：有人要求1的阶乘  
 结束：终于求到了1的阶乘：1  
 结束：终于求到了2的阶乘：2  
 结束：终于求到了3的阶乘：6  
 结束：终于求到了4的阶乘：24  
 结束：终于求到了5的阶乘：120  
 v2 = 120

### 递归思想的总结

当面对一个“大问题”,该大问题可以经由该问题的同类问题的“小一级问题”而经过简单计算获得,而且,可以获知(已知)这类问题的“最小一级问题”的答案.则,此时就可以使用递归方法来解决.

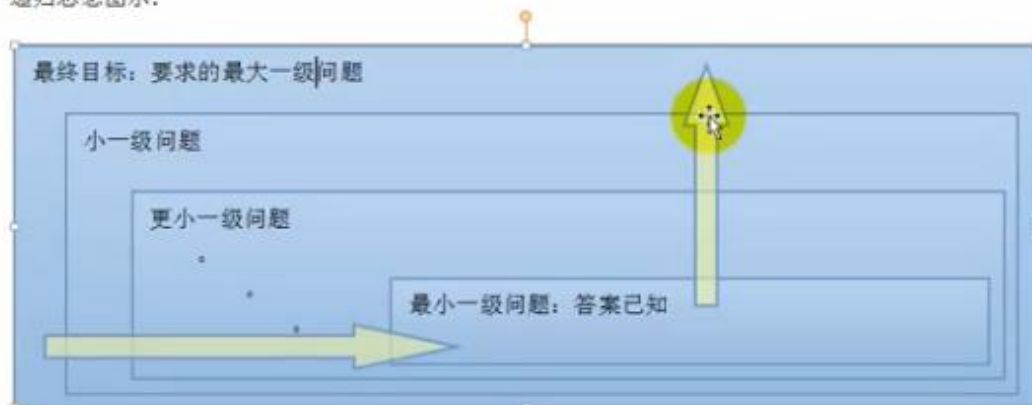
则此时该函数的基本模式是:

```

function digui( $n ){
    if(是最小一级){
        return 已知的答案;
    }
    $jieguo = 对 digui($n-1) 进行简单运算;
    return $jieguo;
}

```

递归思想图示:



## 递推总结

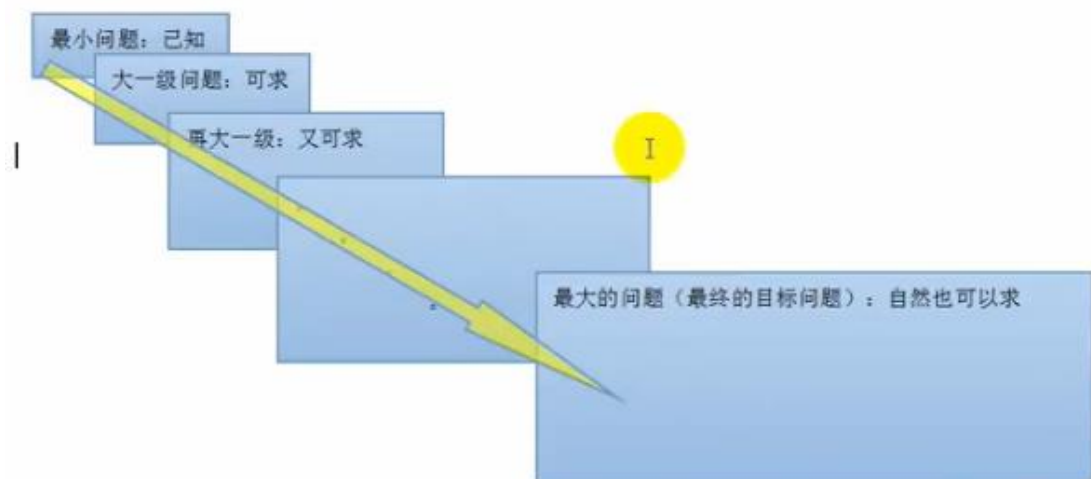
如果要求一个“大问题”,且该问题有如下 2 个特点:

- 1, 已经该问题的同类问题的最小问题的答案;
- 2, 如果知道这种问题的小一级问题的答案,就可轻松求得其“大一级”问题的答案,并且此问题的级次有一定的规律;

则此时就可以使用递推思想来解决该问题.代码模式为:

```
$qian = 已知的最小 一级问题的答案;  
For ($i = 最小一级的下一级; $i <= 最大一级的级次; ++$i){  
    $jieguo = 对 $qian 进行一定的计算,通常需要使用到 $i;  
    $qian = $jieguo;  
}  
Echo “结果为:” . $jieguo ;
```

递推思想图示:



- 通常,如果一个问题,既能使用递归计算解决,又能使用递推算法解决,则应该使用递推算法;