

# 常用系统函数及递归思想

## 1. 常用系统函数

### (1) 有关函数的函数

```
if(!function_exists("f5")){  
    function f5(){  
        $arr = func_get_args();  
        var_dump($arr). "<br>";  
  
        echo "<pre>";  
        for($i = 0; $i < func_num_args(); $i++){  
            echo func_get_arg($i). "\t";  
        }  
    }  
}  
  
f5(5, 6, 7);
```

array(3) { [0]=> int(5) [1]=> int(6) [2]=> int(7) }

5          6          7

### (2) 字符串函数

输出与格式化: echo, print, printf, print\_r, var\_dump

字符串去除与填充: trim, ltrim, rtrim, str\_pad

字符串截取: substr, strchr, strrchr

字符串替换: str\_replace, substr\_replace

字符串长度与位置: strlen, strpos, strrpos

字符串转换: strtolower, strtoupper, lcfirst, ucfirst, ucwords

特殊字符串处理:

nl2br, addslashes, htmlspecialchars, htmlspecialchars\_decode

### (3) 时间函数

time, microtime, mktime, date, idate, strtotime, date\_add, date\_default\_timezone\_set, date\_default\_timezone\_get

### (4) 数学相关函数

max, min, round, ceil, floor, abs, sqrt, pow, rand

## 2. 有关函数的编程思想

### (1) 递归思想——递归函数

```
//递归思想
//计算n的阶层
function jiecheng($n){
    if($n == 1)
        return 1;

    $s = $n * jiecheng($n - 1);
    return $s;
}

$result = jiecheng(5);
echo "5的阶层为$result". "<br><br>";

//一个队伍n个人，按年龄排，第n个年龄比n-1个年龄多2岁，第一个年龄是12，求队伍中所有人的年龄
function get_ages($n, $completion){
    if($n == 1){
        $completion("第($n)个人年龄为"."12");
        return 12;
    }

    $age = get_ages($n - 1, $completion) + 2;
    $completion("第($n)个人年龄为"."$age");
    return $age;
}

get_ages(5, function($completion){
    echo "$completion"."<br>"; //过程回调，得到每个人年龄得出的过程
});
```

方式2:1234

array(3) { [0]=> int(5) [1]=> int(6) [2]=> int(7) }

5          6          7

5的阶层为120

第(1)个人年龄为12  
第(2)个人年龄为14  
第(3)个人年龄为16  
第(4)个人年龄为18  
第(5)个人年龄为20

1. 只要明白递归调用每次都是在一个独立的函数代码区执行代码即可，走到满足临界条件的函数代码中开始往回return结果，一直return到第一次调用的函数代码区，返回最终结果，那就好理解了。
2. 当面对一个“大问题”，该问题可以经由该问题的同类问题的“小一级问题”经过简单计算获得答案，而且，可以获知(已知)这类问题的“最小一级”问题的答案，则此时可以采用递归思想来解决此大问题。

## (2) 递推(迭代)思想

如果要求一个“大问题”，且该问题有如下2个特点：

1. 已知该问题的同类问题的最小问题的答案
2. 如果知道这种小一级的答案，就可以轻松求的大一级问题的答案，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

```
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
//求以下数列
//1 1 2 3 5 8 13 ...
//第7项的值
$qian1 = 1;
$qian2 = 1;
for($i = 3; $i <= 7; ++$i){
    $jieguo = $qian1 + $qian2;
    $qian1 = $qian2;
    $qian2 = $jieguo;
}
echo "结果为: ".$jieguo."<br>";
```

结果为: 13