

后盾网 人人做后盾

www.houdunwang.com

函数

后盾网 2011-2015 v5.0

函数的概念:

函数 (function) 是一段完成指定任务的已命名代码, 函数可以遵照给它的一组值或参数完成任务, 并且可能返回一个值。使用函数可以节省编译时间, 因为无论调用多少次, 函数只需被编译一次。

函数的特性:

- 控制程序设计的复杂性
- 提高软件的可靠性
- 提高软件的开发效率
- 提高软件的可维护性
- 提高程序的重用性

自定义函数

```
function 函数名称 (参数1, 参数2 ...) {  
    程序内容叙述;  
    return 返回值; //如需函数有返回值时使用;  
}
```

函数的使用:

- 函数名称(参数1,参数2 ...);
\$返回值变量 = 函数名称(参数1,参数2);

自定义函数语法格式

函数名称定义：

它是函数在程序代码中的识别名称，函数名可以是以字母或下划线开头后跟零个或多个字母、下划线和数字的任何字符串。函数名不区分大小写。命名函数时不可使用已声明的函数，或PHP内建的函数名称。

函数参数：

所谓的参数就是用来把数值由函数外部传入函数体中，并用来加以运算处理。参数之间用“，”号隔开。当函数不需要任何数值传入时，可以省略参数。

函数

返回值：

当调用函数时需要它返回一些数值，那么就要函数体中用return语句实现。格式如下：

return 返回值； //返回值也可以是一个表达式

如果函数中没有使用return返回值时则函数返回值为null

函数返回值

函数名称定义：

`function_exists()` 函数的作用是检查类的方法是否存在

检测hd这个函数是否定义：

`var_dump(function_exists('hd'));`

函数检测

变量的能见度

- 所谓变量的能见度，意指变量在程序中的可作用范围。当一个变量执行赋值动作后，会随着声明局部的差异，而有不同的作用范围。大致上来说变量会依据声明的局部分为下列两种：局部变量和全局变量

变量的范围

在函数之中声明的变量就是局部变量，并且该变量只有在函数范围之中才能加以使用。如果其它程序局部需要调用使用该变量值时，必须透过「return」指令，来将其传回至主程序区块以作后续处理。

```
<?php
$hd=0;
function p(){
    $hd = 3; //定义局部变量
    echo "在函数中显示局部变量$hd值: $hd </br>";
}
p();
echo "在函数外显示变量$hd值: $hd <br>";
```

局部变量

在函数范围之外声明的变量为全局变量，在函数无法直接使用全局变量，函数要使用全局变量，必须要利用global引入。

```
<?php
    $hd="后盾网";
    function p() {
        global $hd; //利用 global 引入全局变量
        echo $hd;
    }
    p();
?>
```

可以使用全局数组\$GLOBALS调用全局变量。

```
echo $GLOBALS["hd"];
```

全局变量

PHP支持声明函数变量为静态的(static)。一个静态变量在所有对该函数的调用之间共享，并且仅在脚本的执行期间函数第一次被调用时被初始化。要声明函数变量为静态的用关键字static。通常，静态变量的第一次使用时赋予一个初始值。

静态变量

按值传递参数

- 父程序直接传递指定的值或是变量给函数使用。由于所传递的值或变量，与函数里的数值分别储存于不同的内存区块，所以当函数对所导入的数值作了任何变动时，并不会对父程序造成直接影响。

按址传递参数(用“&”符号实现)

- 相对于按值传递模式，并不会将父程序中的指定数值或目标变量传递给函数，而是把该数值或变量的内存储存区块相对地址导入函数之中。因此当该数值在函数中有任何变动时，会连带对父程序造成影响。

参数的传递

默认参数

- 在php中，支持函数的默认方式调用。如果在调用函数时没有指定参数的值，在函数中会使用参数的默认值。默认参数必须列在所有没有默认值参数的后面。

参数默认值

变量函数

- PHP 支持变量函数的概念。这意味着如果一个变量名后有圆括号，PHP 将寻找与变量的值同名的函数，并且将尝试执行它。除了别的事情以外，这个可以被用于实现回调函数，函数表等等。
- 变量函数不能用于语言结构，例如 `echo()`，`print()`，`unset()`，`isset()`，`empty()`，`include()`，`require()` 以及类似的语句。需要使用自己的外壳函数来将这些结构用作变量函数。

变量函数

递归调用

- 所谓的函数递归调用，就是函数可以在其声明的执行叙述之中调用执行自己。
- 通常在此类型的函数之中会附加一个条件判断叙述，以判断是否需要执行递归调用，并且在特定条件下终止函数的递归调用动作，把目前流程的主控权交回上一层函数执行。因此当某个执行递归调用的函数，没有附加条件判断叙述时可能会造成无限循环的错误情形
- 函数递归调用最大的好处在于可以精简程序中繁杂重复调用程序，并且能以这种特性来执行一些较为复杂的运算动作。

函数调用

```
<?php
```

```
    echo "利用递归调用来运算 N 的阶乘<p>";
```

```
    function facto($n)
```

```
    {
```

```
        if ($n > 1){
```

```
            $r=$n*facto($n-1);}
```

```
        else{
```

```
            $r=$n;
```

```
        }
```

```
        return $r;
```

```
    }
```

```
    echo "当 N 值为 5 时,阶乘为: ".facto(5)."<br>";
```

```
    echo "当 N 值为 10 时,阶乘为: ".facto(10);
```

```
?>
```

用递归求n阶乘