

第7章 自定义函数

学习要点：

- 1.标准函数
- 2.自定义函数
- 3.文件包含
- 4.魔法常量

主讲教师：李炎恢

合作网站：<http://www.ibeifeng.com>

讲师博客：<http://hi.baidu.com/李炎恢>

一般来讲，冗余的代码都是不好的。一而再，再而三地重写代码不仅浪费时间，从布局结构角度看也显得粗制滥造。与所有优秀的编程语言一样，PHP 采用了很多方法来缓解冗余代码的问题，其中最常见而且最容易实现的方法就是使用函数（function）。

一．标准函数

标准的PHP 发行包中有1000 多个标准函数，这些标准函数都是系统内置的，不需要用户自己创建而可以直接使用。

```
<?
```

```
echo md5('123456'); //MD5函数对字符串进行加密处理
```

```
?>
```

二．自定义函数

PHP 内置函数允许和文件进行交互、使用数据库、创建图形，还可以连接其他的服务器。但是，在实际工作中，有许多时候所需要的东西是语言的创建者无法预见到的。声明一个函数可以让我们想内置函数那样使用自己的代码。只要简单地调用这个函数并提供给它必须的参数。这就意味着，在整个脚本中，都可以调用和多次重复使用相同的函数。

创建函数

```
<?
```

```
function functionName() {  
echo '这是一个无参无返回自定义函数';  
}
```

```
?>
```

调用函数

```
<?
```

```
functionName();
```

?>

函数命名

- 1.函数名不能和已有的函数名重名。
- 2.函数名只能包含字母、数字和下划线。
- 3.函数名不能以数字开头。

包含参数无返回的函数调用

<?

```
function functionArea($radius) {  
$area = $radius * $radius * pi();  
echo $area;  
}  
functionArea(10);
```

?>

包含参数有返回的函数调用：使用return()语句可以向函数调用者返回任意确定值，将程序控制权返回到调用者的作用域。

<?

```
function functionArea($radius) {  
return $radius * $radius * pi();  
}  
echo functionArea(10);
```

?>

包含默认参数的函数调用：可以为输入参数指定默认值，在没有提供其他值的情况下，就会把这个默认值自动赋给该参数。

<?

```
function functionArea($radius=10) {  
return $radius * $radius * pi();  
}  
echo functionArea();
```

?>

返回多个值的函数调用：可以通过返回一个数组然后使用list()函数构造即可。

<?

```
function functionInfo($name,$age,$job) {  
$userInfo = array($name,$age,$job); //可以用追加的方式比较常用  
return $userInfo;  
}  
list($name,$age,$job) = functionInfo('吴祁',19,'学生');  
echo $name.'今年'.$age.'岁了，目前还是个'.$job;
```

```
?>
```

包含引用传参的函数调用：引用传递可以在函数内对参数的修改在函数范围外也能反应。

```
<?
```

```
$prices = 50;
$tax = 0.5;
function functionPrices(&$prices,$tax) {
    $prices = $prices + ($prices * $tax);
    $tax = $tax * 2;
}
functionPrices($prices,$tax);
echo $prices;
echo '<br />';
echo $tax;
```

```
?>
```

请注意，函数调用将不区分大小写，所以调用functionname()、FunctionName()或FUNCTIONNAME()都是有效的，而且都将返回相同的结果。为了方便，这里都用小写。注意到函数名称和变量名称是不同的，这一点很重要。变量名是区分大小写的，所以\$Name和\$name是两个不同的变量，但Name()和name()则是同一个函数。

理解作用域

变量的作用域可以控制变量在哪里是可见并且可用的。不同的编程语言有不同的变量作用域规则。PHP 具有相当简单的规则：

在函数内部声明的变量作用域是从声明它们的那条语句开始到函数末尾。这叫做函数作用域。这些变量成为局部变量。

在函数外部声明的变量作用域是从声明它们的那条语句开始到文件末尾，而不是函数内部。这叫做全局作用域。这些变量成为全局变量。

特殊的超级全局变量在函数内外部都是可见的。

使用require()和include()并不影响作用域。如果这两个语句用于函数内部，函数作用域适用。如果它不在函数内部，全局作用域适用。

关键字“global”可以用来手动指定一个在函数中定义或使用的变量具有全局作用域。

通过调用unset(\$variable_name)可以手动删除变量。如果变量被删除，它就不在参数所指定的作用域中了。

全局变量定义global

```
<?
```

```
$a = 5;
function fna() {
```

```

global $a;
$a = 20;
}
fna();
echo $a;
?>

```

可以通过使用超级全局变量\$GLOBAL，可以访问或改变全局作用域中的任何变量。

```

<?
$GLOBALS['a'] = 5;
function fna() {
$GLOBALS['a'] = 20;
}
fna();
echo $GLOBALS['a'];
?>

```

创建自己的函数库

通常将函数集文件存放在library 文件夹里，然后通过文件调用即可。文件名约定促成可以取名为tool.library.php，tool 可以根据情况来设定，后面两个照抄！

三. 文件包含

为了确保重用性和模块性，最普遍的方式是把功能组建隔离为单独的文件，然后在需要时重新组装。PHP 提供了四种在应用程序中包含文件的语句。

include()语句将在其被调用的位置处判断并包含一个文件。包含一个文件与在该语句所在位置复制该文件的数据具有相同的结果。

```

<?
include 'include.php';
?>

```

include_once()函数的作用与include()相同，不过它会首先验证是否已经包含了该文件。如果包含了该文件，则不再执行include_once()。

```

<?
include_once 'include.php';
?>

```

require()在很大程度与include()相同，都是将一个模板文件包含到require()调用所在的位置。

```

<?
require('require.php');
?>

```

require_once()函数的作用与require()相同，不过require_once()函数确保文件只包含一次。

在遇到require_once()后，后面再试图包含相同的文件时都将被忽略。

```
<?
```

```
require_once('require.php');
```

```
?>
```

require()语句和include()语句几乎是等价的。二者的差异在于，当这两个语句调用失败后，require()将给出一个致命错误，而include()只是给出一个警告

四. 魔法常量

PHP 实现了一些所谓的魔法常量，他们并不真的是常量，因为这些魔法常量会根据使用的场合改变值。

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城Web

俱乐部(yc60.com)

联合提供：

本次主讲老师：李炎恢

我的博客：hi.baidu.com/

李炎恢/

我的邮件：yc60.com@gmail.com

名称描述

__FILE__ 当前文件名

__LINE__ 当前行号

__FUNCTION__ 当前函数名

__CLASS__ 当前类名

__METHOD__ 当前方法名