**PHP 变量作用域**

在 PHP 中，可以在脚本的任意位置对变量进行声明。

变量的作用域指的是变量能够被引用/使用的那部分脚本。

PHP 有三种不同的变量作用域：

* local（局部）
* global（全局）
* static（静态）

**Local 和 Global 作用域**

函数之外声明的变量拥有 Global 作用域，只能在函数以外进行访问。

函数内部声明的变量拥有 LOCAL 作用域，只能在函数内部进行访问。

下面的例子测试了带有局部和全局作用域的变量：

**实例**

<?php

$x=5; // 全局作用域

function myTest() {

$y=10; // 局部作用域

echo "<p>测试函数内部的变量：</p>";

echo "变量 x 是：$x";

echo "<br>";

echo "变量 y 是：$y";

}

myTest();

echo "<p>测试函数之外的变量：</p>";

echo "变量 x 是：$x";

echo "<br>";

echo "变量 y 是：$y";

?>

[运行实例](http://www.w3school.com.cn/tiy/s.asp?f=demo_php_var_local)

在上例中，有两个变量 $x 和 $y，以及一个函数 myTest()。$x 是全局变量，因为它是在函数之外声明的，而 $y 是局部变量，因为它是在函数内声明的。

如果我们在 myTest() 函数内部输出两个变量的值，$y 会输出在本地声明的值，但是无法输出 $x 的值，因为它在函数之外创建。

然后，如果在 myTest() 函数之外输出两个变量的值，那么会输出 $x 的值，但是不会输出 $y 的值，因为它是局部变量，并且在 myTest() 内部创建。

注释：您可以在不同的函数中创建名称相同的局部变量，因为局部变量只能被在其中创建它的函数识别。

**PHP global 关键词**

global 关键词用于在函数内访问全局变量。

要做到这一点，请在（函数内部）变量前面使用 global 关键词：

**实例**

<?php

$x=5;

$y=10;

function myTest() {

global $x,$y;

$y=$x+$y;

}

myTest();

echo $y; // 输出 15

?>

[运行实例](http://www.w3school.com.cn/tiy/s.asp?f=demo_php_var_global)

PHP 同时在名为 $GLOBALS[index] 的数组中存储了所有的全局变量。下标存有变量名。这个数组在函数内也可以访问，并能够用于直接更新全局变量。

上面的例子可以这样重写：

**实例**

<?php

$x=5;

$y=10;

function myTest() {

$GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];

}

myTest();

echo $y; // 输出 15

?>

[运行实例](http://www.w3school.com.cn/tiy/s.asp?f=demo_php_var_global2)

**PHP static 关键词**

通常，当函数完成/执行后，会删除所有变量。不过，有时我需要不删除某个局部变量。实现这一点需要更进一步的工作。

要完成这一点，请在您首次声明变量时使用 *static* 关键词：

**实例**

<?php

function myTest() {

static $x=0;

echo $x;

$x++;

}

myTest();

myTest();

myTest();

?>

[运行实例](http://www.w3school.com.cn/tiy/s.asp?f=demo_php_var_static)

然后，每当函数被调用时，这个变量所存储的信息都是函数最后一次被调用时所包含的信息。

注释：该变量仍然是函数的局部变量。