



## 3. PHP变量

变量是用于存储信息的"容器":

```
<?php
$x=5;
$y=6;
$z=$x+$y;
echo $z;
?>
```

与代数类似:

$x=5$ ,  $y=6$ ,  $z=x+y$

在代数中, 我们使用字母 (如  $x$ ) , 并给它赋值 (如 5) 。

从上面的表达式  $z=x+y$  , 我们可以计算出  $z$  的值为 11。

在 PHP 中, 这些字母被称为**变量**。

### 3.1 php变量规则

- 变量以 \$ 符号开始, 后面跟着变量的名称
- 变量名必须以字母或者下划线字符开始
- 变量名只能包含字母数字字符以及下划线 (A-z、0-9 和 \_)
- 变量名不能包含空格

- 变量名是区分大小写的（\$y 和 \$Y 是两个不同的变量）



PHP 语句和 PHP 变量都是区分大小写的。

## 3.2 创建（声明）PHP 变量

PHP 没有声明变量的命令。

变量在第一次赋值给它的时候被创建：

```
<?php
$txt="Hello world!";
$x=5;
$y=10.5;
?>
```

在上面的语句执行中，变量 **txt** 将保存值 **Hello world!**，且变量 **x** 将保存值 **5**。



当赋一个文本值给变量时，需要在文本值两侧加上引号。

## 3.3 PHP 是一门弱类型语言

在上面的实例中，我们注意到，**不必向 PHP 声明该变量的数据类型**。

PHP 会根据变量的值，自动把变量转换为正确的数据类型。

在强类型的编程语言中，我们必须在使用变量前先声明（定义）变量的类型和名称。

## 3.4 PHP 变量作用域

变量的作用域是脚本中变量可被引用/使用的部分。

PHP 有四种不同的变量作用域：

- local
- global
- static
- parameter

### 3.4.1 局部和全局作用域

在所有函数外部定义的变量，拥有全局作用域。除了函数外，全局变量可以被脚本中的任何部分访问，要在一个函数中访问一个全局变量，需要使用 `global` 关键字。

在 PHP 函数内部声明的变量是局部变量，仅能在函数内部访问：

```
<?php
$x=5; // 全局变量

function myTest()
{
    $y=10; // 局部变量
    echo "<p>测试函数内变量:<p>";
    echo "变量 x 为: $x";
    echo "<br>";
    echo "变量 y 为: $y";
}

myTest();

echo "<p>测试函数外变量:<p>";
echo "变量 x 为: $x";
echo "<br>";
echo "变量 y 为: $y";
?>
```

在以上实例中 `myTest()` 函数定义了 `$x` 和 `$y` 变量。`$x` 变量在函数外声明，所以它是全局变量，`$y` 变量在函数内声明所以它是局部变量。

当我们调用 `myTest()` 函数并输出两个变量的值，函数将会输出局部变量 `$y` 的值，但是不能输出 `$x` 的值，因为 `$x` 变量在函数外定义，无法在函数内使用，如果要在一个函数中访问一个全局变量，需要使用 `global` 关键字。

然后我们在 `myTest()` 函数外输出两个变量的值，函数将会输出全局变量 `$x` 的值，但是不能输出 `$y` 的值，因为 `$y` 变量在函数中定义，属于局部变量。



你可以在不同函数中使用相同的变量名称，因为这些函数内定义的变量名是局部变量，只作用于该函数内。

### 3.4.2 PHP `global` 关键字

`global` 关键字用于函数内访问全局变量。

在函数内调用函数外定义的全局变量，我们需要在函数中的变量前加上 `global` 关键字：

```

<?php
$x=5;
$y=10;

function myTest()
{
    global $x,$y;
    $y=$x+$y;
}

myTest();
echo $y; // 输出 15
?>

```

PHP 将所有全局变量存储在一个名为 `$GLOBALS[index]` 的数组中。*index* 保存变量的名称。这个数组可以在函数内部访问，也可以直接用来更新全局变量。

上面的实例可以写成这样：

```

<?php
$x=5;
$y=10;

function myTest()
{
    $GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];
}

myTest();
echo $y;
?>

```

### 3.4.3 Static 作用域

当一个函数完成时，它的所有变量通常都会被删除。然而，有时候您希望某个局部变量不要被删除。

要做到这一点，请在您第一次声明变量时使用 **static** 关键字：

```

<?php
function myTest()
{
    static $x=0;
    echo $x;
    $x++;
    echo PHP_EOL;    // 换行符
}

myTest();

```

```
myTest();  
myTest();  
?>
```

然后，每次调用该函数时，该变量将会保留着函数前一次被调用时的值。



**注释：**该变量仍然是函数的局部变量。

### 3.4.4 参数作用域

参数是通过调用代码将值传递给函数的局部变量。

参数是在参数列表中声明的，作为函数声明的一部分：

```
<?php  
function myTest($x)  
{ echo $x;  
}  
myTest(5);  
?>
```