PHP 语法 PHP 字符串变量

PHP 变量

变量是用于存储信息的"容器"：

实例

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

运行实例 »

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**。

变量是用于存储数据的容器。

PHP 变量

与代数类似，可以给 PHP 变量赋予某个值（x=5）或者表达式（z=x+y）。

变量可以是很短的名称（如 x 和 y）或者更具描述性的名称（如 age、carname、totalvolume）。

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。

创建（声明）PHP 变量

PHP 没有声明变量的命令。

变量在您第一次赋值给它的时候被创建：

实例

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

运行实例 »

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**。

注释：当您赋一个文本值给变量时，请在文本值两侧加上引号。

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。

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

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：

实例

```
<?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";
?>
```

运行实例 »

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。

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：

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 »

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：

实例

```
<?php
$x=5;
$y=10;
function myTest()
{
$GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];
}
myTest();
echo $y;
?>
```

运行实例 »

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：

实例

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

运行实例 »

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。

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：

实例

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

我们将在 PHP 函数 章节对它做更详细的讨论。

← PHP 语法 PHP 字符串变量 →

1 篇笔记

写笔记

- 1、定义在函数外部的就是全局变量，它的作用域从定义处一直到文件结尾。
- 2、函数内定义的变量就是局部变量，它的作用域为函数定义范围内。
- 3、函数之间存在作用域互不影响。
- 4、函数内访问全局变量需要 `global` 关键字或者使用 `$GLOBALS[index]` 数组

在 php 中函数是有独立的作用域，所以局部变量会覆盖全局变量，即使局部变量中没有全局变量相同的变量，也会被覆盖。如下：

```
<?php
$a=5;
$b=3;
function t()
{
    echo $a-$b; // 输出 0
}
t();
?>
```

要想在函数中直接使用全局变量可以通过 `global` 关键字声明或者通过 php 中自定义的 `$GLOBALS` 数组获取:

```
<?php
$a=5;
$b=3;
function t1()
{
    global $a,$b;
    echo $a-$b;  // 输出 2
}
t1();

echo PHP_EOL;
```

```
function t2()
{
    echo $GLOBALS['a']-$GLOBALS['b'];  // 输出 2
}
t2();
?>
```

佛系coder 11个月前 (04-26)