PHP 在变量定义中不需要（或不支持）明确的类型定义；变量类型是根据使用该变量的上下文所决定的。也就是说，如果把一个 [string](https://www.php.net/manual/zh/language.types.string.php) 值赋给变量 *$var*，*$var* 就成了一个 [string](https://www.php.net/manual/zh/language.types.string.php)。如果又把一个[integer](https://www.php.net/manual/zh/language.types.integer.php) 赋给 *$var*，那它就成了一个[integer](https://www.php.net/manual/zh/language.types.integer.php)。

PHP 的自动类型转换的一个例子是乘法运算符“\*”。如果任何一个操作数是[float](https://www.php.net/manual/zh/language.types.float.php)，则所有的操作数都被当成[float](https://www.php.net/manual/zh/language.types.float.php)，结果也是[float](https://www.php.net/manual/zh/language.types.float.php)。否则操作数会被解释为[integer](https://www.php.net/manual/zh/language.types.integer.php)，结果也是[integer](https://www.php.net/manual/zh/language.types.integer.php)。注意这并*没有*改变这些操作数本身的类型；改变的仅是这些操作数如何被求值以及表达式本身的类型。

<?php  
$foo = "1";  // $foo 是字符串 (ASCII 49)  
$foo \*= 2;   // $foo 现在是一个整数 (2)  
$foo = $foo \* 1.3;  // $foo 现在是一个浮点数 (2.6)  
$foo = 5 \* "10 Little Piggies"; // $foo 是整数 (50)  
$foo = 5 \* "10 Small Pigs";     // $foo 是整数 (50)  
?>

如果上面两个例子看上去古怪的话，参见[字符串转换为数值](https://www.php.net/manual/zh/language.types.string.php" \l "language.types.string.conversion)。

如果要强制将一个变量当作某种类型来求值，参见[类型强制转换](https://www.php.net/manual/zh/language.types.type-juggling.php" \l "language.types.typecasting)一节。如果要改变一个变量的类型，参见 [settype()](https://www.php.net/manual/zh/function.settype.php)。

如果想要测试本节中任何例子的话，可以用 [var\_dump()](https://www.php.net/manual/zh/function.var-dump.php) 函数。

**Note**:

自动转换为 数组 的行为目前没有定义。

此外，由于 PHP 支持使用和数组下标同样的语法访问字符串下标，以下例子在所有 PHP 版本中都有效：

<?php  
$a    = 'car'; // $a is a string  
$a[0] = 'b';   // $a is still a string  
echo $a;       // bar  
?>

请参阅[存取和修改字符串中的字符](https://www.php.net/manual/zh/language.types.string.php" \l "language.types.string.substr)一节以获取更多信息。

**类型强制转换**

PHP 中的类型强制转换和 C 中的非常像：在要转换的变量之前加上用括号括起来的目标类型。

<?php  
$foo = 10;   // $foo is an integer  
$bar = (boolean) $foo;   // $bar is a boolean  
?>

允许的强制转换有：

* (int), (integer) - 转换为整形 [integer](https://www.php.net/manual/zh/language.types.integer.php)
* (bool), (boolean) - 转换为布尔类型 [boolean](https://www.php.net/manual/zh/language.types.boolean.php)
* (float), (double), (real) - 转换为浮点型 [float](https://www.php.net/manual/zh/language.types.float.php)
* (string) - 转换为字符串 [string](https://www.php.net/manual/zh/language.types.string.php)
* (array) - 转换为数组 [array](https://www.php.net/manual/zh/language.types.array.php)
* (object) - 转换为对象 [object](https://www.php.net/manual/zh/language.types.object.php)
* (unset) - 转换为 [NULL](https://www.php.net/manual/zh/language.types.null.php) (PHP 5)

(binary) 转换和 b 前缀转换支持为 PHP 5.2.1 新增。

注意在括号内允许有空格和制表符，所以下面两个例子功能相同：

<?php  
$foo = (int) $bar;  
$foo = ( int ) $bar;  
?>

将字符串文字和变量转换为二进制字符串：

<?php  
$binary = (binary)$string;  
$binary = b"binary string";  
?>

**Note**:

可以将变量放置在双引号中的方式来代替将变量转换成字符串：

<?php  
$foo = 10;            // $foo 是一个整数  
$str = "$foo";        // $str 是一个字符串  
$fst = (string) $foo; // $fst 也是一个字符串  
  
// 输出 "they are the same"  
if ($fst === $str) {  
    echo "they are the same";  
}  
?>

有时在类型之间强制转换时确切地会发生什么可能不是很明显。更多信息见如下小节：

* [转换为布尔型](https://www.php.net/manual/zh/language.types.boolean.php#language.types.boolean.casting)
* [转换为整型](https://www.php.net/manual/zh/language.types.integer.php#language.types.integer.casting)
* [转换为浮点型](https://www.php.net/manual/zh/language.types.float.php#language.types.float.casting)
* [转换为字符串](https://www.php.net/manual/zh/language.types.string.php#language.types.string.casting)
* [转换为数组](https://www.php.net/manual/zh/language.types.array.php#language.types.array.casting)
* [转换为对象](https://www.php.net/manual/zh/language.types.object.php#language.types.object.casting)
* [转换为资源](https://www.php.net/manual/zh/language.types.resource.php#language.types.resource.casting)
* [转换为 NULL](https://www.php.net/manual/zh/language.types.null.php#language.types.null.casting)
* [类型比较表](https://www.php.net/manual/zh/types.comparisons.php)