浮点型（也叫浮点数 float，双精度数 double 或实数 real）可以用以下任一语法定义：

<?php  
$a = 1.234;   
$b = 1.2e3;   
$c = 7E-10;  
?>

浮点数的形式表示：

LNUM [0-9]+

DNUM ([0-9]\*[\.]{LNUM}) | ({LNUM}[\.][0-9]\*)

EXPONENT\_DNUM [+-]?(({LNUM} | {DNUM}) [eE][+-]? {LNUM})

浮点数的字长和平台相关，尽管通常最大值是 1.8e308 并具有 14 位十进制数字的精度（64 位 IEEE 格式）。

**Warning**

**浮点数的精度**

浮点数的精度有限。尽管取决于系统，PHP 通常使用 IEEE 754 双精度格式，则由于取整而导致的最大相对误差为 1.11e-16。非基本数学运算可能会给出更大误差，并且要考虑到进行复合运算时的误差传递。

此外，以十进制能够精确表示的有理数如 *0.1* 或 *0.7*，无论有多少尾数都不能被内部所使用的二进制精确表示，因此不能在不丢失一点点精度的情况下转换为二进制的格式。这就会造成混乱的结果：例如，*floor((0.1+0.7)\*10)* 通常会返回 *7* 而不是预期中的 *8*，因为该结果内部的表示其实是类似 *7.9999999999999991118...*。

所以永远不要相信浮点数结果精确到了最后一位，也永远不要比较两个浮点数是否相等。如果确实需要更高的精度，应该使用[任意精度数学函数](https://www.php.net/manual/zh/ref.bc.php)或者 [gmp 函数](https://www.php.net/manual/zh/ref.gmp.php)。

参见[» 浮点数指南](http://floating-point-gui.de/)网页的简单解释。

**转换为浮点数**

如果希望了解有关何时和如何将字符串转换成浮点数的信息，请参阅“[字符串转换为数值](https://www.php.net/manual/zh/language.types.string.php#language.types.string.conversion)”一节。对于其它类型的值，其情况类似于先将值转换成整型，然后再转换成浮点。请参阅“[转换为整型](https://www.php.net/manual/zh/language.types.integer.php#language.types.integer.casting)”一节以获取更多信息。自 PHP 5 起，如果试图将对象转换为浮点数，会发出一条 E\_NOTICE 错误消息。

**比较浮点数**

如上述警告信息所言，由于内部表达方式的原因，比较两个浮点数是否相等是有问题的。不过还是有迂回的方法来比较浮点数值的。

要测试浮点数是否相等，要使用一个仅比该数值大一丁点的最小误差值。该值也被称为机器极小值（epsilon）或最小单元取整数，是计算中所能接受的最小的差别值。

*$a* 和 *$b* 在小数点后五位精度内都是相等的。

<?php  
$a = 1.23456789;  
$b = 1.23456780;  
$epsilon = 0.00001;  
  
if(abs($a-$b) < $epsilon) {  
    echo "true";  
}  
?>

**NaN**

某些数学运算会产生一个由常量 **NAN** 所代表的结果。此结果代表着一个在浮点数运算中未定义或不可表述的值。任何拿此值与其它任何值（除了 **TRUE**）进行的松散或严格比较的结果都是 **FALSE**。

由于 **NAN** 代表着任何不同值，不应拿 **NAN** 去和其它值进行比较，包括其自身，应该用 [is\_nan()](https://www.php.net/manual/zh/function.is-nan.php) 来检查。